Analyzing Inference Privacy Risks Through Gradients In Machine Learning

Zhuohang Li
Vanderbilt University
Nashville, TN, USA
zhuohang.li@vanderbilt.edu

Andrew Lowy
University of Wisconsin–Madison
Madison, WI, USA
alowy@wisc.edu

Jing Liu
Mitsubishi Electric Research Laboratories
Cambridge, MA, USA
jiliu@merl.com

Toshiaki Koike-Akino
Mitsubishi Electric Research Laboratories
Cambridge, MA, USA
koike@merl.com

Kieran Parsons
Mitsubishi Electric Research Laboratories
Cambridge, MA, USA
parsons@merl.com

Bradley Malin
Vanderbilt University
Nashville, TN, USA
b.malin@vanderbilt.edu

Ye Wang
Mitsubishi Electric Research Laboratories
Cambridge, MA, USA
yewang@merl.com

ABSTRACT

In distributed learning settings, models are iteratively updated with shared gradients computed from potentially sensitive user data. While previous work has studied various privacy risks of sharing gradients, our paper aims to provide a systematic approach to analyze private information leakage from gradients. We present a unified game-based framework that encompasses a broad range of attacks including attribute, property, distributional, and user disclosures. We investigate how different uncertainties of the adversary affect their inferential power via extensive experiments on five datasets across various data modalities. Our results demonstrate the inefficacy of solely relying on data aggregation to achieve privacy against inference attacks in distributed learning. We further evaluate five types of defenses, namely, gradient pruning, signed gradient descent, adversarial perturbations, variational information bottleneck, and differential privacy, under both static and adaptive adversary settings. We provide an information-theoretic view for analyzing the effectiveness of these defenses against inference from gradients. Finally, we introduce a method for auditing attribute inference privacy, improving the empirical estimation of worst-case privacy through crafting adversarial canary records.

1 INTRODUCTION

Ensuring privacy is an important prerequisite for adopting machine learning (ML) algorithms in critical domains that require training on sensitive user data, such as medical records, personal financial information, private images, and speech. Prominent ML models, ranging from compact neural networks tailored for mobile platforms [40] to large foundation models [10, 72], are often trained on user data via gradient-based iterative optimization. In many cases, such as decentralized learning [19, 41] or federated learning (FL) [33, 38, 66], model gradients are directly exchanged in place of raw training data to facilitate joint learning, which opens up an additional channel for potential privacy leakage [61].

Recent works have explored information leakage through this gradient channel in various forms, albeit in isolation. For instance, Nasr *et al.* [69] showed that it is feasible to infer membership (i.e., single-bit information indicating the existence of a target record in the training data pool) from model updates in federated learning. Beyond membership, Melis *et al.* [68] demonstrated inference over sensitive properties of the training data in collaborative learning. Other independent lines of work additionally explored attribute inference [20, 62] and data reconstruction [31, 35, 103] through shared model gradients. However, some emerging privacy concerns that have so far only been considered under the centralized learning setting, such as the distributional inference [16, 85] and user-level inference [49, 54], have not been well investigated in the gradient leakage setting.

Existing studies on information leakage from gradients have several limitations. First, the majority of the current literature focuses on investigating each individual type of inference attack under their specific threat models while lacking a comprehensive examination of inference attack performance under various adversarial assumptions, which is essential for providing a holistic view of the adversary's capabilities. For instance, from the attack's perspective, assuming the adversary to have access to a reasonably-sized shadow dataset and limited rounds of access to the model's gradients helps to capture the realistic inference privacy risk under a practical threat model. Conversely, from the defense's perspective, assuming a powerful adversary with access to record-level gradients and auxiliary information about the private record helps to estimate the worst-case privacy risk, which may facilitate the design of more robust defenses. Second, while several types of heuristic defenses have been explored by prior work, their supposed effectiveness has not been fully verified under more challenging adaptive adversary settings. Moreover, existing studies do not adequately explain why some defenses succeed in reducing the inference risk over gradients, while others fail, which could provide important guidance on the design of more effective defenses.

In this paper, we conduct a systematic analysis of private information leakage from gradients. We start by defining a unified inference game that broadly encompasses four types of inference attacks that aims at inferring common private information of the data from gradients, namely, *attribute inference attack* (AIA), *property inference attack* (PIA), *distributional inference attack* (DIA), and *user inference attack* (UIA), as illustrated in Figure 1. Under this framework, we show that information leakage from gradients can be treated as performing statistical inference over a sensitive variable upon observing samples of the gradients, with different definitions of the information encapsulated by the variable being inferred, leading to a generic template for constructing different types of inference attacks. We additionally explore different tiers of adversarial assumptions, with varying numbers of available data samples, numbers of observable rounds of gradients, and varying batch sizes, to investigate how different priors and uncertainties in the adversary's knowledge about the gradient and data distribution affect the adversary's inferential power.

We perform a systematic evaluation of these attacks on five datasets (Adult [8], Health [47], CREMA-D [12], CelebA [59], UTKFace [102]) with three different data modalities (tabular, speech, and image). A common setting in distributed learning is that the data distribution is heterogeneous across different nodes but homogeneous within each node. Under this assumption, where the sensitive variable is common across a batch, we show that a larger batch size leads to higher inference privacy risk from gradients across all considered attacks, highlighting that *solely relying on data aggregation is insufficient for achieving meaningful privacy in distributed learning.* With a moderate batch size (e.g., 16), we show that an adversary can launch successful inference attacks with very few shadow data samples ($\leq$ 1,000). For instance, in the case of property inference on the Adult dataset, the adversary can achieve 0.92 AUROC with only 100 shadow data samples. Moreover, we demonstrate that an adversary with access to multiple rounds of gradient updates can perform Bayesian inference to aggregate adversarial knowledge, eventually leading to higher confidence and better attack performance.

We apply the developed inference attacks to evaluate the effectiveness of five common types of defenses from the privacy literature [45, 46, 77–79, 82, 84, 94, 103], including Gradient Pruning [103], Signed Stochastic Gradient Descent (SignSGD) [9], Adversarial Perturbations [64], Variational Information Bottleneck (VIB) [4], and Differential Privacy (DP-SGD) [1], against both *static* adversaries that are unaware of the defense and *adaptive* adversaries that can adapt to the defense mechanism. We find that most heuristic defense methods only offer a weak notion of "security through obscurity", in the sense that they defend against static adversaries empirically but can be easily bypassed by adaptive adversaries. Although DP-SGD shows consistent performance against both static and adaptive adversaries, to fully prevent inference attacks, it often requires injecting too much noise which diminishes the utility of the learning model. We provide an information-theoretic perspective for explaining and analyzing the (in)effectiveness of these considered defenses and show that *the key ingredient of a successful defense is to effectively reduce the mutual information between the released gradients and the sensitive variable*, which could serve as a guideline for designing future defenses. Finally, to provide practical guidance in selecting privacy parameters, we introduce an auditing approach for empirically estimating the privacy loss of attribute inference attacks through crafting adversarial canary records to approximate the privacy risk in the worst case.

In summary, our main contributions are as follows:

- We provide a holistic analysis of inference privacy from gradients through a unified inference game that broadly encompasses a range of attacks concerning attribute, property, distributional, and user inference.
- We demonstrate the weakness of solely relying on data aggregation to achieve privacy against inference attacks in distributed learning. We do this through a systematic evaluation of the four types of attacks on datasets with different modalities under various adversarial assumptions.
- Our analyses reveal that reducing the mutual information between the released gradients and the sensitive variable is the key ingredient of a successful defense. This is shown by investigating five common types of defense strategies against inference over gradients from an information-theoretic perspective.
- Our auditing results provide an empirical justification for tolerating large DP parameters when defending against attribute inference attacks (c.f. [60]). This is achieved by implementing an auditing method for empirically estimating the privacy loss against attribute inference attacks from gradients.

2 BACKGROUND AND RELATED WORK

2.1 Machine Learning Notation

A machine learning (ML) model can be denoted as a function $f_{\boldsymbol{\theta}}$: $\mathbf{x} \rightarrow \mathbf{y}$ parameterized by $\boldsymbol{\theta}$ that maps from the input (feature) space to the output (label) space. The training of an ML model involves a set of training data and an optimization procedure, such as stochastic gradient descent (SGD). At each step of SGD, a loss function $\mathcal{L}(\boldsymbol{\theta}, \mathcal{D}_b)$ is first computed based on the current model and a batch of k training samples $\mathcal{D}_b = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{k}$ and then a set of gradients is computed as $\boldsymbol{g} = \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \mathcal{D}_b)$. Finally, the model is updated by taking a gradient step towards minimizing the loss.

2.2 Related Work

Developing ML models in many applications involves training on the users' private data, which introduces privacy leakage risks from different components of the ML model across several stages of the development and deployment pipeline.

Leakage From Model Parameters ($\boldsymbol{\theta}$). The first way of exposing privacy information is through analyzing the model parameters. This is connected to the most prominent centralized ML setting, where the model is first developed on a local dataset and then released to the users for deployment. Various forms of privacy leakage have been studied in this setting. White-box membership inference [53, 69, 73] aims at identifying the presence of individual records in the training dataset given access to the full model. Data

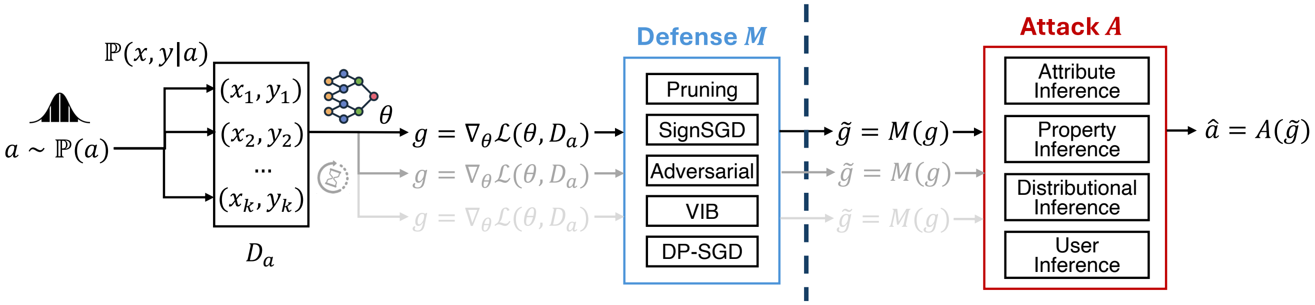


Figure 1: Overview of the unified inference game from gradients: the adversary infers the sensitive variable a from observations of the gradients $\tilde{g}$ computed on the private data batch $\mathcal{D}_a$.

extraction attacks exploit the memorization of the ML model to extract training samples [14, 37], whereas model inversion attacks generate synthetic data samples from the training distribution [93, 99]. In contrast, for distributional inference attacks [6, 30, 85], the attacker's goal is to make inferences about the entire training data distribution rather than individuals.

Leakage From Model Outputs ($f_\theta(x)$). Another source of privacy leakage is the model output, which is related to more restrictive settings such as machine learning as a service (MLaaS) in cloud APIs where only black-box access to the ML model is granted. Under this setting, researchers have studied several privacy attacks that can be launched by querying the model and observing the outputs. For instance, query-based model inversion attacks [28, 29] exploit the predicted confidence or labels from the model to make inferences about the input data instance [101] or attribute [67]. Model stealing attacks attempt to recover the confidential model weights [88] or hyper-parameters [92] given query access to the model. Black-box membership inference attacks [73, 76, 83, 90] and black-box distributional inference attacks [16, 65] allow an adversary to decide whether a data point was included in training or reveal information about the training data distribution by analyzing its output prediction or confidence.

Leakage From Model Gradients (g). The final source of privacy leakage is the gradient of the loss function with respect to the model parameters, which is essential for updating the model with stochastic gradient descent. This is relevant to ML settings that release intermediate model updates during model development, such as distributed training, federated learning, peer-to-peer learning, and online learning. Compared to model parameters, model gradients carry more nuanced information about a small batch of data used for computing the update and thus may reveal more information about the underlying data instances. Current literature studies different types of gradient-based privacy leakage in isolation. One line of work focused on data reconstruction from model gradients [31, 103] or updates [37, 74] with various data types, such as image [31, 57, 98, 103], text [35, 37], tabular [91], and speech data [56]. However, these attacks rely on strong adversarial assumptions and do not generalize to large batch sizes [42]. Another line of work investigated the extraction of private attributes or properties [26, 68] of the private data from model gradients. Specifically, Melis *et al.* [68] first revealed that gradients shared in collaborative learning can be used to infer properties of the training data that are uncorrelated with the task label. Lyu *et al.* [62] explored attribute reconstruction from epoch-averaged gradients on tabular and genomics data. Feng *et al.* [26] discovered that gradients of Speech Emotion Recognition models leak information about user demographics such as gender and age. Dang *et al.* [18] showed that speaker identities can be revealed from the gradients of Automatic Speech Recognition models. Kerkouche *et al.* [51] demonstrated the weakness of secure aggregation without differential privacy in Federated learning by designing a disaggregation attack that exploits the linearity of model aggregation and client participation across multiple rounds to capture client-specific properties. In contrast to existing studies that design separate treatments for each type of attack, in this work, we take a holistic view of information leakage from gradients.

3 PROBLEM FORMALIZATION

This section introduces four types of inference attacks from gradients, namely, *attribute inference*, *property inference*, *distributional inference*, and *user inference*. We formally define information leakage from gradients using a unified security game, following standard practices in machine learning privacy studies [75], and discuss variants of threat models that affect the adversary's inferential power. In Section 4, we describe methods to construct these attacks.

3.1 Attack Definitions

We consider four types of information leakage from model gradients that generally involve two parties, namely, a private learner who releases model gradients computed on a private data batch, and an adversary who tries to make inferences about the private data given access to the gradients. This generic setting captures multiple ML application scenarios such as distributed training, federated learning, and online learning.

Attribute Inference. Attribute inference attacks (AIA) seek to infer a data record's unknown attribute (feature) from its gradient. Prior works in both centralized [95, 97] and federated settings [20, 62] usually assume the record to be *partially known*. For instance, infer a missing entry (e.g., genotype) of a person's medical record [29]. It is worth noting that, in practice, when the attributes are not completely independent, an adversary with partial knowledge about the record may be able to infer the unknown attribute just from the known ones, as in data imputation [44].

Property Inference. Property inference attacks (PIA) aim to infer a global property of the private data batch that is not directly present in the data feature space but is correlated with some of the features (and consequently the gradients). For tabular data, these properties could be sensitive features that have been intentionally excluded from training (e.g., pseudo-identifiers in health records that are required to be removed for HIPAA compliance); for high-dimensional data like image and speech, they could be some high-level statistical features capturing the semantics of the data instance (e.g., race of a face image [68] or gender of a speech recording [26]).

Distributional Inference. Distributional inference attacks (DIA) aim to infer the ratio of the training samples (α) that satisfy some target property[1]. The majority of current literature on DIA [16, 30, 65, 85] is in the space of centralized learning, which captures leakage from model parameters. These studies usually define DIA as a distinguishing test between two worlds where the model is trained on two datasets with different ratios (α_0 and α_1) [85]. This can be further categorized into *property existence* tests that decide if there exists any data point with the target property in the training set and *property size estimation* tests that infer the exact ratio of the property in the training data [16]. In this work, we extend DIA to the gradient space and consider a general case that combines property existence and property size estimation by formulating DIA as performing ordinal classification between a set of m ratio bins ($m \geq 3$), i.e., $\{0\}, (0, \frac{1}{m-1}], (\frac{1}{m-1}, \frac{2}{m-1}], ..., (\frac{m-2}{m-1}, 1]$.

User Inference. User inference attacks (UIA) or re-identification attacks aim to identify which user's data was used to compute the observed gradients. Here, the adversary does not know the user's exact data used for computing the gradients. Instead, the adversary is provided a set of candidate users and their corresponding underlying user-level data distributions. This setting shares similarities with the *subject-level membership inference* [86] in the sense that both attacks measure the privacy risk at the granularity of each individual. However, the user inference attack aims to infer richer information that directly exposes the user's identity compared to the membership inference attack, which only discloses a single bit of information (i.e., whether a given user's data sample is involved in training). Thus user inference can be considered as a generalization of subject-level membership inference attack.

We note that except for attribute inference which directly exposes (part of) the user's private data, property inference, distributional inference, and user inference attacks are *inferential disclosures* (also known as *deductive disclosures*) that exploit the statistical correlation exists in data to infer sensitive information from the released gradients with high confidence. We exclude record-level privacy attacks such as membership inference and data reconstruction as our analysis here focuses on distributed learning scenarios where private information can be shared across different data samples within a batch.

3.2 Unified Inference Game

Our framework aims to capture an abstraction of privacy problems in distributed learning settings, where an attacker aims to recover some sensitive information of a particular client from their shared gradients (or model updates). In practical distributed learning settings, the data may be heterogeneously split across the clients, and an attacker may take advantage of side information about a particular client's local data distribution. Generally, the objective of the attacker is to recover the sensitive information, represented by the variable $\mathbf{a}$, which is related to the local data distribution of the client through a joint distribution $\mathbb{P}(\mathbf{x}, \mathbf{y}, \mathbf{a}) = \mathbb{P}(\mathbf{a})\,\mathbb{P}(\mathbf{x}, \mathbf{y}|\mathbf{a})$. As we will detail later, specific choices in what $\mathbf{a}$ represents and the corresponding specialized structure of $\mathbb{P}(\mathbf{x}, \mathbf{y}, \mathbf{a})$ enable the framework to capture attribute, property, distributional, and user inference privacy problems. This joint distribution may capture both the side information available to the attacker and the inherent heterogeneity of the data. To focus on evaluating the effectiveness of gradient-based attacks and defenses, we simplify the modeling of the overall training procedure, by updating the model in a centralized fashion on the entire training data set $\mathcal{D}$, but generating gradients for the attacker on batches drawn according to $\mathbb{P}(\mathbf{x}, \mathbf{y}, \mathbf{a})$.

Definition 3.1. ***Unified Inference Game.*** Let $\mathbb{P}(\mathbf{x}, \mathbf{y}, \mathbf{a})$ be the joint distribution, $\mathcal{L}$ the loss function, $\mathcal{T}$ the training algorithm, r the total number of training rounds, and $\mathcal{R} \subset [r]$ a set of rounds that are observable to the adversary[2]. The unified inference game from gradients between a challenger (private learner) and an adversary is as follows:

(1) Challenger initializes the model parameters as $\boldsymbol{\theta}_0$.
(2) Challenger samples a training dataset $\mathcal{D} = \{(\boldsymbol{x}_j, \boldsymbol{y}_j)\}_{j=1}^{n}$, where $(\boldsymbol{x}_j, \boldsymbol{y}_j) \overset{\text{i.i.d.}}{\sim} \mathbb{P}(\mathbf{x}, \mathbf{y})$.
(3) Challenger draws the sensitive variable $\boldsymbol{a} \sim \mathbb{P}(\mathbf{a})$.
(4) Challenger draws a batch of k data samples $\mathcal{D}_{\boldsymbol{a}} = \{(\boldsymbol{x}_p, \boldsymbol{y}_p)\}_{p=1}^{k}$, where $(\boldsymbol{x}_p, \boldsymbol{y}_p) \overset{\text{i.i.d.}}{\sim} \mathbb{P}(\mathbf{x}, \mathbf{y}|\boldsymbol{a})$, for the given $\boldsymbol{a}$.
(5) Challenger computes the gradient of the loss on the data batch, $\boldsymbol{g}_i = \nabla_{\boldsymbol{\theta}_{i-1}} \mathcal{L}(\boldsymbol{\theta}_{i-1}, \mathcal{D}_{\boldsymbol{a}})$.
(6) Challenger applies the defense mechanism $\mathcal{M}$ to produce a privatized version of the gradient $\tilde{\boldsymbol{g}}_i = \mathcal{M}(\boldsymbol{g}_i)$. When no defense is applied, $\mathcal{M}$ is simply the identity function, i.e., $\tilde{\boldsymbol{g}}_i = \boldsymbol{g}_i$.
(7) The model is updated by applying the training algorithm on the training dataset for one epoch $\boldsymbol{\theta}_i \leftarrow \mathcal{T}(\boldsymbol{\theta}_{i-1}, \mathcal{D}, \mathcal{L}, \mathcal{M})$.
(8) Steps (5)-(7) are repeated for r rounds.
(9) A *static* adversary $\mathcal{A}_s$ gets access to $\mathcal{L}, \mathcal{T}, \mathbb{P}(\mathbf{x}, \mathbf{y}, \mathbf{a})$, and the set of (intermediate) model parameters $\Theta = \{\boldsymbol{\theta}_{i-1} | i \in \mathcal{R}\}$ and released gradients $\mathcal{G} = \{\tilde{\boldsymbol{g}}_i | i \in \mathcal{R}\}$. An *adaptive* adversary $\mathcal{A}_a$ also gets the defense mechanism $\mathcal{M}$.
(10) The adversary outputs its inference $\hat{\boldsymbol{a}}$ of the sensitive variable, i.e., $\hat{\boldsymbol{a}} \leftarrow \mathcal{A}_s(\mathcal{L}, \mathcal{T}, \mathbb{P}(\mathbf{x}, \mathbf{y}, \mathbf{a}), \Theta, \mathcal{G})$ for the static adversary, or $\hat{\boldsymbol{a}} \leftarrow \mathcal{A}_a(\mathcal{L}, \mathcal{T}, \mathbb{P}(\mathbf{x}, \mathbf{y}, \mathbf{a}), \Theta, \mathcal{G}, \mathcal{M})$ for the adaptive adversary. The adversary wins if $\hat{\boldsymbol{a}} = \boldsymbol{a}$ and loses otherwise.

[1] Some prior work also refers to distributional inference as property inference.

[2] We use $[r]$ to denote the discrete set $\{1, 2, ..., r\}$.

In the above general game, the flexibility of the joint distribution $\mathbb{P}(\mathrm{x}, \mathrm{y}, \mathrm{a})$ allows capturing various scenarios. Rather than explicitly defining this joint distribution, which anyways depends on the unknown data distribution, we implicitly define it through transformations/filtering of a given data set. Further, providing the adversary with knowledge of the distribution $\mathbb{P}(\mathrm{x}, \mathrm{y}, \mathrm{a})$ is realized by providing the adversary with suitable shadow datasets drawn according to such transformations and filtering operations.

Attribute Inference Game. The variable $\boldsymbol{a} \in [m]$ is a discrete attribute within the features $\boldsymbol{x}$. Sampling $\boldsymbol{a} \sim \mathbb{P}(\mathrm{a})$ is accomplished by drawing uniformly or according to its marginal empirical distribution within the given training data set $\mathcal{D}$. Drawing the data batch $\mathcal{D}_{\boldsymbol{a}}$ according to the distribution $\mathbb{P}(\mathrm{x}, \mathrm{y}|\boldsymbol{a})$, is accomplished by uniformly selecting data samples $(\boldsymbol{x}, \boldsymbol{y})$ from the entire training data set $\mathcal{D}$ with features $\boldsymbol{x}$ that possess the attribute $\boldsymbol{a}$.

Property Inference Game. This scenario is similar to attribute inference, except that $\boldsymbol{a} \in [m]$ is a property associated with, but external to the features of, each data sample (i.e., $\boldsymbol{a}$ may be some meta-data property of each sample, but excluded from the features of $\boldsymbol{x}$). Drawing the data batch $\mathcal{D}_{\boldsymbol{a}}$ is handled similarly to the attribute inference case.

Distributional Inference Game. In this class of scenarios, we have a general set of m transformations $\{\Phi_{\boldsymbol{a}}|\boldsymbol{a} \in [m]\}$, which are selected by the sensitive variable $\boldsymbol{a}$. Each transformation $\Phi_{\boldsymbol{a}}$ corresponds to implicitly realizing the corresponding $\mathbb{P}(\mathrm{x}, \mathrm{y}|\boldsymbol{a})$, by applying a general transformation that involves selective sampling from the overall training set $\mathcal{D}$. For example, the selection of $\boldsymbol{a}$ may indicate a particular proportion for the prevalence of a certain attribute or property, and thus the corresponding transformation would select batches of data according to that proportion.

User Inference Game. This is a special case of property inference, where $\boldsymbol{a}$ specifically corresponds to the identity of an individual that provided the corresponding data samples. Unlike other inference attacks, the sensitive variable, as it represents identity, does not take on a fixed set of values. To make the attack more operational, similar to prior work on data reconstruction [39], we assume the inference is over a fixed set of m candidate users randomly sampled from the population at the beginning of each game.

3.3 Threat Model

In this work, we assume the adversary has no control over the training protocol and only passively observes gradients as the model is being updated. In practice, the adversary could be an honest-but-curious parameter server [55] in a distributed learning or federated learning setting, a node in decentralized learning [19], or an attacker who eavesdrops on the communication channel. The game as defined in Definition 3.1 is similar to games defined in many prior works [13, 97] which captures the average-case privacy as the performance of the attack is measured by its expected value over the random draw of data samples. In Section 7, we consider an alternative game where the data samples are adversarially chosen to provide a measure of worst-case privacy for privacy auditing.

We consider the following aspects that reflect different levels of the adversary's knowledge:

- **Knowledge of Data Distribution.** Similar to many prior works on inference attacks [13, 16, 58, 68, 80, 85, 96], we model the adversarial knowledge of the data distribution through access to data samples drawn from this distribution, which are referred to as shadow datasets. A larger shadow dataset implies a more powerful adversary that has more knowledge about the underlying data distribution. For discrete attributes, we additionally consider a more informed adversary who knows the prior distribution of the attribute, which can be estimated by drawing a large amount of data from the population.
- **Continuous Observation.** We use the observable set $\mathcal{R}$ to capture the adversary's ability to observe the gradients continuously. Intuitively, an adversary observing multiple rounds should perform better than a single-round adversary. Assuming a powerful adversary is beneficial for analyzing and auditing defenses. For instance, the privacy analysis in DP-SGD [1] assumes that the adversary has access to all rounds of gradients.
- **Adaptive Adversary.** When evaluating defenses, in addition to the static adversary, we consider a stronger adaptive adversary who is aware of the underlying defense mechanism. This has been demonstrated as pivotal for thoroughly assessing the effectiveness of security defenses [15, 87].

4 ATTACK CONSTRUCTION

4.1 Inference Attacks

The objective of the inference adversary is to infer the sensitive variable from the observed gradient, i.e., modeling the posterior distribution $\mathbb{P}(\mathrm{a}|\mathrm{g})$. The general strategy of implementing inference attacks from gradients is to exploit the following two adversarial assumptions as defined in the unified inference game in Section 3.2. First, the adversary possesses knowledge about the underlying population data distribution. Operationally, this implies that the adversary is able to draw data samples $(\boldsymbol{x}, \boldsymbol{y})$ with corresponding sensitive variable $\boldsymbol{a}$ from $\mathbb{P}(\mathrm{x}, \mathrm{y}, \mathrm{a})$ to construct a shadow dataset. Second, the adversary has access to the training algorithm and the current model parameters, which allows the adversary to compute the gradients $\boldsymbol{g}$ for each batch of samples within the shadow dataset. With this information, the adversary can train a predictive model $P_{\boldsymbol{\omega}}(\mathrm{a}|\mathrm{g})$ to approximate the posterior.

Attribute & Property Inference. The attribute and property inference attacks follow a similar attack procedure, with the difference being whether the sensitive variable $\mathbf{a}$ is internal or external to the data record. Specifically, the adversary first constructs a shadow dataset $\mathcal{D}_{\boldsymbol{s}}$ by sampling from the population distribution, i.e., $\mathcal{D}_{\boldsymbol{s}} = \{(\boldsymbol{x}_j, \boldsymbol{y}_j, \boldsymbol{a}_j)\}_{j=1}^{s}$ where $(\boldsymbol{x}_j, \boldsymbol{y}_j, \boldsymbol{a}_j) \overset{\text{i.i.d.}}{\sim} \mathbb{P}(\mathrm{x}, \mathrm{y}, \boldsymbol{a})$. Then the adversary draws data batches $\mathcal{D}_{\boldsymbol{a}} = \{(\boldsymbol{x}_j, \boldsymbol{y}_j)\}_{j=1}^{k}$ from the shadow dataset through bootstrapping. This is achieved by repeatedly sampling the sensitive attribute $\boldsymbol{a}$ and then drawing k records that have the sensitive attribute from $\mathcal{D}_{\boldsymbol{s}}$. Next, for each data batch $\mathcal{D}_{\boldsymbol{a}}$, the adversary computes the gradient $\boldsymbol{g}_{\boldsymbol{a}} = \nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}, \mathcal{D}_{\boldsymbol{a}})$ using the current model parameters $\boldsymbol{\theta}$. This results in a set of labeled data pairs $(\boldsymbol{g}_{\boldsymbol{a}}, \boldsymbol{a})$, which can then be used for training an ML model $P_{\boldsymbol{\omega}}(\mathrm{a}|\mathrm{g})$ that predicts the sensitive variable from gradient observations. In practice, we find that it is beneficial to train the predictive model using a balanced dataset, which can be seen as modeling $\frac{\mathbb{P}(\mathrm{a}|\mathrm{g})}{\mathbb{P}(\mathrm{a})}$, and capture the prior knowledge in a separate term. This provides more stable performance for small shadow dataset sizes and skewed sensitive variable distributions.

It is worth noting that here we are considering a more restrictive setting for attribute inference where the adversary holds no additional knowledge about the private data besides the gradients compared to prior works that assume the private record to be partially known (e.g., [20, 62] assume that everything is known except for the sensitive attribute). Our framework can be easily extended to the general case where the adversary holds arbitrary additional knowledge $\varphi(\boldsymbol{x})$ about the private record $\boldsymbol{x}$ by training a predictive model $P_{\boldsymbol{\omega}}(\mathrm{a}|\mathrm{g}, \varphi(\boldsymbol{x}))$ using shadow data drawn from $\mathbb{P}(\mathbf{x}, \mathrm{y}, \mathrm{a}|\varphi(\boldsymbol{x}))$.

Distributional Inference. In distributional inference, the sensitive variable is the index of the ratio bin to which the property ratio belongs. The adversary first samples a random bin index $\boldsymbol{a}$ and then samples a property ratio α within that bin. Next, the adversary draws a data batch $\mathcal{D}_{\boldsymbol{a}}$ with $\lfloor \alpha k \rfloor$ records with the property and the rest without the property and derives the gradient $\boldsymbol{g_a}$. This process is repeated by the adversary to collect a set of labeled gradients and attribute pairs $(\boldsymbol{g_a}, \boldsymbol{a})$ to train a predictive model. We note that in the setting of distributional inference, the sensitive variable is a series of ordinal numbers indicative of the continuous property ratio α and thus should not be treated as regular multi-class classification. To utilize the ordering information, we adopt a simple strategy to ordinal classification [27], which transforms the m-class ordinal classification problem into $m-1$ binary classifications. Specifically, the adversary trains a series of $m-1$ binary classifiers, with the i-th classifier $P_{\omega_i}(\mathrm{a} > i|\mathrm{g})$ trained to decide whether or not $\boldsymbol{a}$ is larger than i. The final posterior probability can be obtained as

$$P_{\boldsymbol{\omega}}(\mathrm{a} = \boldsymbol{a}|\mathrm{g}) = \begin{cases} 1 - P_{\omega_1}(\mathrm{a} > 1|\mathrm{g}), & \text{if } \boldsymbol{a} = 1 \\ P_{\omega_{\boldsymbol{a}-1}}(\mathrm{a} > \boldsymbol{a} - 1|\mathrm{g}) - P_{\omega_{\boldsymbol{a}}}(\mathrm{a} > \boldsymbol{a}|\mathrm{g}), & \text{if } 1 < \boldsymbol{a} < m \\ P_{\omega_{m-1}}(\mathrm{a} > m - 1|\mathrm{g}), & \text{if } \boldsymbol{a} = m \end{cases}.$$

User Inference. In contrast to other inference attacks where the sensitive variable is sampled from a well-defined set of values, in user inference, the sensitive variable is the user's identity, which does not take on a fixed set of values. Moreover, the identities that occur during test time are likely not seen during the development of the attack model. As a result, the posterior $\mathbb{P}(\mathbf{a}|\mathbf{g})$ cannot be directly modeled. To resolve this, we employ a training strategy analogous to the prototypical network [81] for few-shot learning. Specifically, we first train a neural network $f_{\boldsymbol{\omega}} \circ u$ that is composed of an encoder $f_{\boldsymbol{\omega}} : \mathbf{g} \to \mathbf{h}$ that maps the gradient vector to a continuous embedding space and a classifier $u : \mathbf{h} \to \mathbf{a}$ that takes the embedding as input and outputs the predicted user identity. Given gradient and sensitive variable pairs $(\boldsymbol{g}, \boldsymbol{a})$ created from the shadow dataset, as the number of available users in the shadow dataset is finite, the neural network can be trained in an end-to-end manner using standard multi-class classification loss such as cross-entropy. After training, the classifier u is discarded. At the time of inference, the adversary is provided with an observed gradient $\tilde{\boldsymbol{g}}$ and a set of m candidate data batches $\{\mathcal{D}_i | i \in [m]\}$, where $\mathcal{D}_i = \{(\boldsymbol{x}_j, \boldsymbol{y}_j)\}_{j=1}^{k}$. Then, the adversary can derive the corresponding set of candidate gradients $\{\boldsymbol{g}_i | i \in [m]\}$ based on the current model parameters $\boldsymbol{\theta}$. Finally, the adversary computes the probability of each candidate identity after observing the gradient as

$$P_{\boldsymbol{\omega}}(\mathrm{a} = \boldsymbol{a}|\mathrm{g} = \tilde{\boldsymbol{g}}) = \frac{\exp\left(-||f_{\boldsymbol{\omega}}(\boldsymbol{g_a}) - f_{\boldsymbol{\omega}}(\tilde{\boldsymbol{g}})||_2\right)}{\sum_{i \in [m]} \exp\left(-||f_{\boldsymbol{\omega}}(\boldsymbol{g}_i) - f_{\boldsymbol{\omega}}(\tilde{\boldsymbol{g}})||_2\right)}.$$

4.2 Continual Attack and Adaptive Attack

The inference attack can be further improved if the adversary has access to multiple rounds of gradients or the defense mechanism.

Inference under Continual Observation. In cases where continual observation of the gradients is allowed, the adversary can use the set of observed gradients $\mathcal{G} = \{\tilde{\boldsymbol{g}}_i | i \in \mathcal{R}\}$ from multiple rounds to improve the attack. A naive solution would be to train a model to directly approximate $\mathbb{P}(\mathrm{a}|\mathcal{G})$. However, this would be generally infeasible in practice because of the high dimensionality of $\mathcal{G}$. Instead, the adversary can use Bayesian updating to accumulate adversarial knowledge. Specifically, given a set of observed gradients, the log-posterior can be formulated as

$$\begin{aligned} &\log \mathbb{P}(\mathrm{a} = \boldsymbol{a}|\mathcal{G}) && (1) \\ &= \log \mathbb{P}(\mathcal{G}|\mathrm{a} = \boldsymbol{a}) + \log \mathbb{P}(\mathrm{a} = \boldsymbol{a}) - \log \mathbb{P}(\mathcal{G}) && (2) \\ &\approx \sum_{i \in \mathcal{R}} \log \mathbb{P}(\tilde{\boldsymbol{g}}_i|\mathrm{a} = \boldsymbol{a}) + \log \mathbb{P}(\mathrm{a} = \boldsymbol{a}) - \log \mathbb{P}(\mathcal{G}) && (3) \\ &= \sum_{i \in \mathcal{R}} \Big(\log \mathbb{P}(\mathrm{a} = \boldsymbol{a}|\tilde{\boldsymbol{g}}_i) + \log \mathbb{P}(\tilde{\boldsymbol{g}}_i) - \log \mathbb{P}(\mathrm{a} = \boldsymbol{a}) \Big) \\ &\quad + \log \mathbb{P}(\mathrm{a} = \boldsymbol{a}) - \log \mathbb{P}(\mathcal{G}) && (4) \\ &= \sum_{i \in \mathcal{R}} \log \mathbb{P}(\mathrm{a} = \boldsymbol{a}|\tilde{\boldsymbol{g}}_i) - (|\mathcal{R}| - 1) \log \mathbb{P}(\mathrm{a} = \boldsymbol{a}) + C, && (5) \end{aligned}$$

where Eq. (3) makes the approximating assumption that the gradients are conditionally independent given $\boldsymbol{a}$. Since $C = -\log \mathbb{P}(\mathcal{G}) + \sum_{i \in \mathcal{R}} \log \mathbb{P}(\tilde{\boldsymbol{g}}_i)$ is independent of $\boldsymbol{a}$, and therefore it can be treated as a constant. $C = 0$ if the gradients $\tilde{\boldsymbol{g}}_i$ are additionally mutually independent. In Eq. (5), the prior term is known and $\mathbb{P}(\mathrm{a} = \boldsymbol{a}|\tilde{\boldsymbol{g}}_i)$ can be approximated by training a fresh model for each round of observation. The sensitive variable can thus be estimated as $\hat{\boldsymbol{a}} = \arg\max_{\boldsymbol{a}} \log \mathbb{P}(\mathrm{a} = \boldsymbol{a}|\mathcal{G})$.

Adaptive Attack. The adversary can design adaptive attacks if the defense mechanism $\mathcal{M}$ is known. Instead of training the predictive model $P_{\boldsymbol{\omega}}(\mathrm{a}|\mathrm{g})$ using clean gradient pairs $(\boldsymbol{g_a}, \boldsymbol{a})$, a simple strategy for adaptive attack is to apply the same defense mechanism to the shadow data's gradients and use the transformed gradient pairs $(\mathcal{M}(\boldsymbol{g_a}), \boldsymbol{a})$ to train the predictive model $P_{\boldsymbol{\omega}}(\mathrm{a}|\mathcal{M}(\mathrm{g}))$. As we will show in Section 6, this simple strategy is sufficient to bypass several heuristic-based defenses.

5 ATTACK EVALUATION

In this section, we evaluate the four inference attacks on datasets with different modalities to investigate the impact of various adversarial assumptions. The findings we present below indicate the key factors that affect the attack performance are: (1) *Continual Observation*: an adversary can improve the inference by accumulating information from multiple rounds of updates, (2) *Batch Size*: when the private information is shared across the batch, using a large batch averages out the effect of the other variables, making it easier to infer the sensitive variable, and (3) *Adversarial Knowledge*: the attack improves with the amount of knowledge of the data

Table 1: Summary of datasets used in experiments.

Dataset	Type	Task Label	Sensitive Variable	Correlation
Adult	Tabular	Income	Gender	-0.1985
Health	Tabular	Mortality	Gender	-0.1123
CREMA-D	Speech	Emotion	Gender	-0.0133
CelebA	Image	Smiling	High Cheekbones	0.6904
UTKFace	Image	Age	Ethnicity	-0.1788

distribution (as captured by the number of available shadow data points).

5.1 Experimental Setup

5.1.1 Datasets and Model Architecture. We consider the following five datasets with different data modalities (tabular, speech, and image) in our experiments.

(1) **Adult** [8] is a tabular dataset containing 48,842 records from the 1994 Census database. We train a fully-connected neural network to predict the person's annual income (whether or not more than 50K a year) and use gender (male or female) as the private attribute. For property and distributional inference attacks, the sex feature is removed.
(2) **Health** [47] (Heritage Health Prize) is a tabular dataset from Kaggle that contains de-identified medical records of over 55,000 patients' inpatient or emergency room visits. We train a fully-connected neural network to predict whether the Charlson Index (an estimate of patient mortality) is greater than zero. We use the patient's gender (male, female, or unknown) as the private attribute, which is removed for property and distributional inference attacks.
(3) **CREMA-D** [12] is a multi-modal dataset that contains 7,442 emotional speech recordings collected from 91 actors (48 male and 43 female). Speech signals are pre-processed using OpenSMILE [25] to extract a total number of 23,990 utterance-level audio features for automatic emotion recognition. Following prior work [26], we use EmoBase which is a standard feature set that contains the MFCC, voice quality, fundamental frequency, and other statistical features, resulting in a feature dimension of 988 for each utterance [36]. We train a fully connected neural network to classify four emotions, including happy, sad, anger, and neutral. We use the speaker's gender (male or female) as the target property for inference attacks.
(4) **CelebA** [59] contains 202,599 face images, each of which is labeled with 40 binary attributes. We resize the images to 32×32 pixels and train a convolutional neural network to classify whether the person is smiling and use whether or not the person has high cheekbones as the target property.
(5) **UTKFace** [102] consists of over 20,000 face images annotated with age, gender, and ethnicity. We resize the images to 32×32 pixels and select 22,012 images from the four largest ethnicity groups (White, Black, Asian, or Indian) to train a convolutional neural network to classify three age groups ($0 - 30$, $31 - 60$, and ≥ 61 years old). Ethnicity is used as the target property.

We split each dataset three-fold into a training set, a testing set, and a public set. The training set is considered to be private and is only used for model training and inference attack evaluation. The testing set is reserved for evaluating the utility of the ML model. The public set is accessible to both the adversary and the private learner, which can be used as the shadow dataset for training the adversary's predictive model or developing defenses as described in Section 6. We provide a summary of the datasets in Table 1, including the task label $\mathbf{y}$, the sensitive variable $\mathbf{a}$ for AIA and PIA, and the Pearson correlation between $\mathbf{y}$ and $\mathbf{a}$.

5.1.2 Metrics. We define the following metrics for measuring inference attack performance:

(1) **Attack Success Rate (ASR)**: We measure the attack performance by the number of times the adversary successfully guesses the sensitive variable, i.e., $p = \sum_{t \in [T]} \mathbb{1}_{\hat{\mathbf{a}}=\boldsymbol{a}}/T$, where T is the total number of trials (i.e., repetitions of the inference game).
(2) **AUROC**: We additionally report the area under the receiver operating characteristic curve (AUROC). For sensitive variables that have more than two classes, we report the macro-averaged AUROC.
(3) **Advantage**: We follow prior work [34, 97] and use the advantage metric to measure the gain in the adversary's inferential power upon observing the gradients. Specifically, the advantage of an adversary is defined by comparing its success rate p to a baseline adversary who doesn't observe the gradients, i.e., $\texttt{Adv}(p) := \max(p - p^*, 0)/(1 - p^*) \in [0, 1]$, where p^* is the success rate of the baseline adversary. The Bayes optimal strategy for the baseline adversary without observing gradients is to guess the majority class, i.e., $p^* = \arg\max_{\boldsymbol{a}} \mathbb{P}(\mathbf{a} = \boldsymbol{a})$.
(4) **TPR@1%FPR**: Besides average performance metrics, recent work on membership inference [13, 96] argue the importance of understanding the privacy risk on worst-case training data by examining the low false positive rate (FPR) region. Inspired by this, we additionally report the true positive rate (TPR) when the FPR is 1%.

5.1.3 Adversary's Model. We conducted preliminary experiments with various types and configurations of ML models and found that random forest with 50 estimators performs the best (especially in the low FPR region) for estimating the posterior in AIA, PIA, and DIA with small shadow dataset sizes. For UIA, we use a fully-connected network with one hidden layer as the encoder. The embedding dimension is set to be 50 for the CREMA-D dataset of 100 for CelebA dataset. As the gradient vector is extremely high dimensional (e.g., the gradient dimensions for CREMA-D and CelebA datasets are 67,716 and 45,922, respectively), we apply a 1-dimensional max-pooling layer before the adversary's predictive model with a kernel size of 3 for tabular datasets and 10 for other datasets for dimensionality reduction.

5.1.4 Other Attack Settings. We assume the model parameters $\boldsymbol{\theta}$ are randomly initialized at the beginning of the inference game. During the game, the model parameters are updated at each epoch using SGD with a learning rate of 0.01. We evaluate AIA on the tabular datasets and UIA on datasets that contain user labels (CREMA-D and CelebA), while PIA and DIA are evaluated on all datasets. For AIA, PIA, and DIA, we use a training set of 5,000 samples and a balanced public set that contains a default number of 1,000 samples equally divided for each sensitive attribute/property class. For UIA, we first filter out user identities that contain less than 2×

batch size number of samples and then split the dataset according to user identities. We select 15 and 30 users on the CREMA-D dataset, and 150 and 300 users on the CelebA dataset as the training and public sets, respectively. We select more users on the CelebA dataset because the majority of users only have very few samples (≤ 16). We set $m = 6$ for DIA, i.e., inferring over 6 ratio bins ($\{0\}, (0, 0.2], (0.2, 0.4], ..., (0.8, 1]$), and $m = 5$ for UIA, i.e., choosing from 5 candidate users. For AIA and PIA, we assume the adversary has access to a prior of the sensitive variable that is estimated from the population. For DIA and UIA, we assume the adversary holds an uninformed prior, and thus the baseline is simply random guessing. The default batch sizes are 16 for AIA and PIA, 128 for DIA, and 8 for UIA. For AIA, PIA, and DIA, the total number of trials T of each experiment is equal to the number of random draws of training batches (i.e., 5,000); for UIA, T is the number of random draws of candidate sets, which we set to be 1,000. We repeat each experiment with 5 different random seeds and report the mean and standard deviation of the results.

5.2 Evaluation of Inference Attacks

We evaluate each type of inference attack with a small shadow dataset (1,000 samples) and compare the results of single-round attacks (where the adversary only observes a single round of gradients) to multi-round attacks (where the adversary gets continual observation of the gradients). Due to space limits, we only include a snapshot of the results (one dataset per attack) in Figure 2 and provide the full results in Appendix Figure 11.

Attribute Inference. We present the results of AIA in Figure 11a. We observe that the adversary is able to infer the sensitive attribute with high confidence using only 1,000 shadow data samples. For instance, on the Adult dataset, the multi-round adversary is able to achieve a high average AUROC of 0.9991 and a TPR@1%FPR of 0.9823. On the Health dataset, however, the AUROC of the multi-round adversary reduces slightly to 0.8122 while the TPR@1%FPR drops drastically to 0.1611. This is likely because the sensitive attribute on the Health dataset contains an "unknown" class (18.9%) that is uncorrelated with other features, making it hard to estimate statistically.

Property Inference. Figure 11c depicts the results of PIA, where we observe that the adversary is able to achieve high performance across all five datasets. Namely, the average AUROCs of the multi-round adversary on the Adult, Health, CREMA-D, CelebA, and UTKFace datasets are 0.9919, 0.8294, 0.8970, 0.9993, and 0.9167, respectively. This consistent high attack performance is in contrast to the general low correlation between the sensitive properties and the task labels across all datasets as indicated in Table 1 (except for CelebA, where a spurious relationship exists), which suggests that the information leakage observed is intrinsic to the computed gradients [68], regardless of the specific data type and learning task.

Distributional Inference. Figure 11d summarizes the results of DIA. Although distributional inference is a more challenging task (6-class ordinal classification), we observe that the multi-round adversary still performs fairly well with a batch size of 128, achieving an average AUROC of 0.8848, 0.7806, 0.7572, 0.9522, and 0.7664 on the Adult, Health, CREMA-D, CelebA, and UTKFace datasets, respectively.

User Inference. We report the results of UIA in Figure 11b. We observe that the adversary is able to identify the user with relatively high confidence on the CelebA dataset, with an average AUROC and TPR@1%FPR of 0.8935 and 0.2828 for the multi-round adversary. On the CREMA-D dataset, the average AUROC of the multi-round adversary is only 0.6808, which may be due to the low identifiability of the features extracted for emotion recognition.

General Observations. Additionally, we have the following general observations across different type of attacks and datasets. First, the performance of single-round attacks decreases as the training progresses. This is because the gradients of the training data will become smaller in magnitude as the training loss decreases and thus the variation within these gradients will become harder to capture. Second, on most datasets, the multi-round attack performs better than any single-round attack, proving the effectiveness of the Bayesian attack framework. Third, we observe very similar performance for AIA and PIA on the tabular datasets. This indicates that whether the sensitive variable is internal or external to the data features does not affect the inference performance.

5.3 Attack Analyses

We investigate the following factors that may affect the performance of inference attacks.

Impact of Batch Sizes. In Figure 3, we study the impact of varying batch sizes on the performance of the inference attacks. We report the results on the Adult dataset for AIA, PIA, and DIA, and results on the CREMA-D dataset for UIA. We observe that the performance of all four considered inference attacks improves as the batch size increases. This is because the records within the batch are sampled from the same conditional distribution $\mathbb{P}(\mathbf{x}, \mathbf{y}|\mathbf{a})$. As the private information $\boldsymbol{a}$ is shared across the batch, a larger batch size would amplify the private information and suppress other varying signals, thereby improving inference performance on $\boldsymbol{a}$. For distributional inference, the difference in the number of samples with the property between each ratio bin $\lfloor \alpha k \rfloor$ also increases as the batch size increases and thus becomes easier to distinguish. For AIA and PIA, we observe that the gap between the single-round adversary (solid lines) and multi-round adversary (dashed lines) is the largest when the batch size is 4, and then gradually reduces as the batch size increases further due to performance saturation. This result suggests that simply aggregating more data does not protect gradients from inference. In fact, it may even increase the privacy risk in distributed learning where data are sampled from the same conditional distribution. This indicates that data aggregation alone is insufficient to achieve meaningful privacy in these settings.

Impact of Adversary's Knowledge. To investigate the impact of the adversary's knowledge on the performance of the attack, we use PIA as an example and plot the attack performance with varying shadow data size and number of observations on the Adult dataset in Figure 5. We observe the general trend that the attack performance increases with the number of observations and available shadow data samples. Interestingly, the attack performance does not always increase monotonically along each axis. For instance, given a small shadow dataset of only 100 samples, the AUROC of an adversary that observes 10 rounds does not outperform an adversary that only observes 5 rounds of gradients. This is likely because when

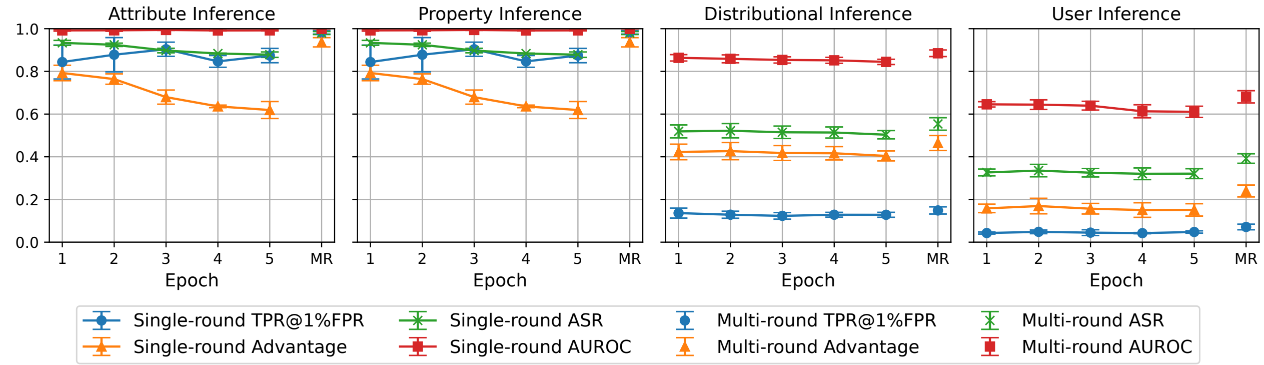


Figure 2: Comparison of single-round and multi-round inference attacks on the Adult (AIA, PIA, DIA) and CREMA-D (UIA) datasets. A complete result on all datasets is provided in Appendix Figure 11.

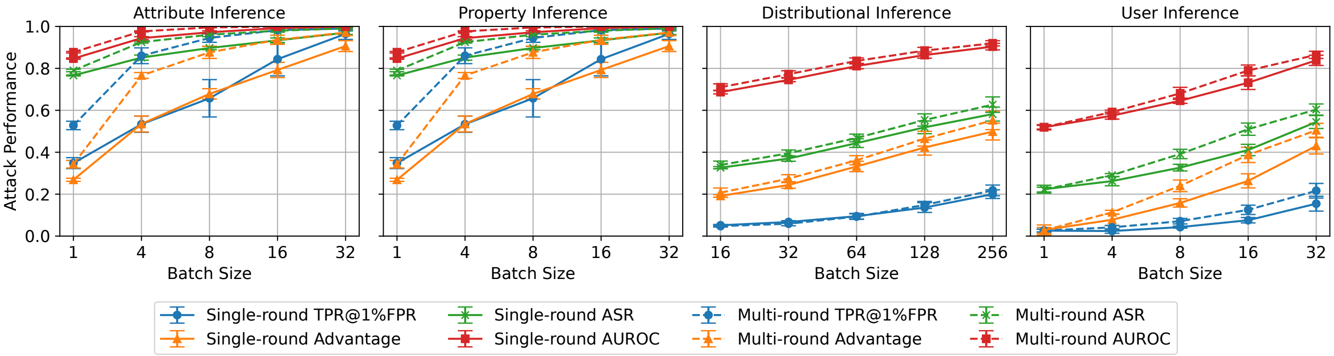


Figure 3: Sensitivity analysis of the impact of varying batch sizes on the performance of inference attacks.

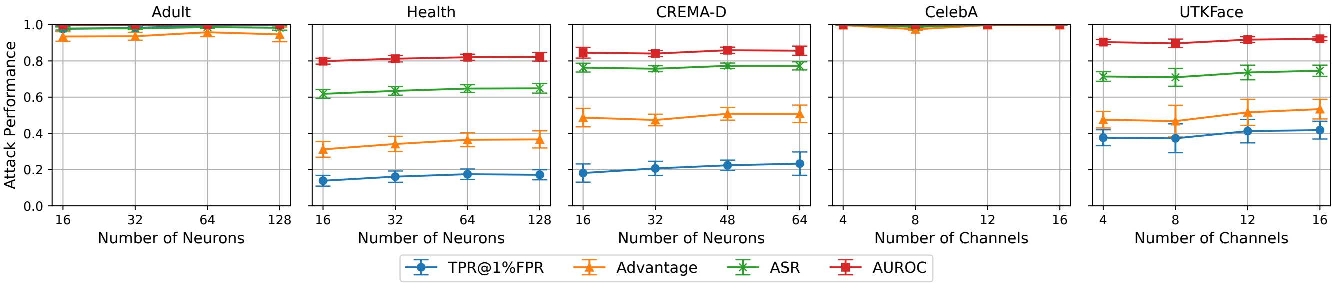


Figure 4: Sensitivity analysis of the impact of varying model sizes on the performance of *Property Inference Attack*.

the model is near convergence, the gradients are small and thus have low variance, which requires more shadow data to accurately estimate the posterior. Such errors in the predictive model will accumulate when using the summation of the log-likelihoods of all single rounds to approximate the joint distribution (Eq. (3)), eventually leading to suboptimal performance.

Impact of Model Size. In Figure 4, we use PIA as an example to study the impact of the machine learning model size. We control the size of the models by varying the model width. Specifically, for fully connected neural networks, we control the number of neurons for the hidden layer. For convolutional neural networks, we control the number of output channels for the first convolutional layer, with the remaining convolutional layers being scaled accordingly. We observe that the attack performance tends to improve slightly with increasing model size, except for the Adult and UTKFace datasets, where performance is saturated. However, most of these improvements are not statistically significant (falling within the margin of error) and thus do not allow for a conclusive statement. We include additional results of other types of inference attacks in Appendix Figure 13, where we make similar observations. These

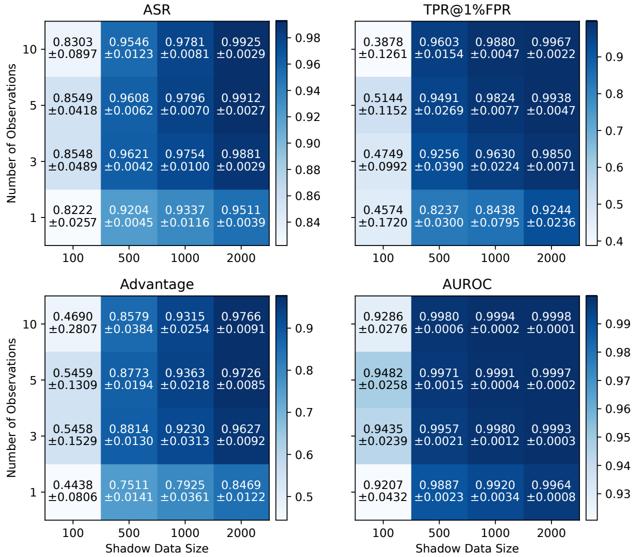


Figure 5: Sensitivity analysis of the impact of adversary's knowledge on the performance of *Property Inference Attack* on the Adult dataset with a batch size of 16.

results demonstrate that all four types of inference attacks can be generalized to larger model sizes.

6 DEFENSES

In this section, we investigate five types of strategies for defending inference from gradients against both static and adaptive adversaries and analyze their performance from an information-theoretic view. The main takeaways from our analyses are: (1) heuristic defenses can defend static adversaries but are ineffective against adaptive adversaries, (2) DP-SGD [1] is the only considered defense that remains effective against adaptive attacks, at the cost of sacrificing model utility, and (3) reducing the mutual information between the released gradients and the sensitive variable is a key ingredient for a successful defense.

6.1 Privacy Defenses Against Inference

Privacy-enhancing strategies in machine learning generally follow two principles: *data minimization* and *data anonymization.* Data minimization strategies, such as the application of cryptographic techniques (e.g., Secure Multi-party Computation and Homomorphic Encryption) and Federated Learning, aim to reveal only the minimal amount of information that is necessary for achieving a specific computational task - and only to the necessary parties. As shown by prior work [24, 51, 52, 89], data minimization alone may not provide sufficient privacy protection and, thus, should be applied in combination with data anonymization defenses to further reduce privacy risks. However, for heuristic-based privacy defenses, it is important to conduct a careful evaluation of their effectiveness against adaptive adversaries. We consider the following five types of representative defenses from the current literature in our experiments:

(1) **Gradient Pruning.** Gradient pruning creates a sparse gradient vector by pruning gradient elements with small magnitudes. This strategy has been used as a baseline for privacy defense in federated learning [84, 94, 103]. By default, we set the pruning rate to be 99%.
(2) **SignSGD.** SignSGD [9] binarizes the gradients by applying an element-wise sign function to the gradients, thereby compressing the gradients to 1-bit per dimension. Similar to gradient pruning, it has been explored in prior work [94, 100] as a defense against data reconstruction attacks in federated learning. Along similar lines, Kerkouche *et al.* [50] evaluated SignFed, a variant of the SignSGD protocol adapted for federated settings, and found it to be more resilient to privacy and security attacks than the standard federated learning scheme.
(3) **Adversarial Perturbation.** Inspired by prior research on protecting privacy through adopting evasion attacks in adversarial machine learning [45, 46, 71, 79], we explore a heuristic defense strategy against inference attacks that inject adversarial perturbation to the gradients. Specifically, at each round of observation, the adversary first trains a neural network $f_{\boldsymbol{\phi}} : \mathbf{g} \rightarrow \mathbf{a}$ to classify the sensitive variable $\boldsymbol{a}$ from the gradient $\boldsymbol{g}$ using a public dataset (same as the shadow dataset). Then, the defense generates a protective adversarial perturbation to cause $f_{\boldsymbol{\phi}}$ to misclassify the perturbed gradients. We adopt l_∞-bounded projected gradient descent (PGD) [64], which generates the adversarial example $\boldsymbol{g}'$ (perturbed gradient) by iteratively taking gradient steps. For AIA, PIA, and DIA, this defense generates an untargeted adversarial perturbation through gradient ascent, i.e., $\tilde{\boldsymbol{g}} \leftarrow \prod_{\mathcal{B}_\infty(\boldsymbol{g},\gamma)} \left(\tilde{\boldsymbol{g}} + \alpha \cdot \text{sign}(\nabla_{\boldsymbol{g}} \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{g}, \boldsymbol{a}))\right)$, where $\mathcal{B}_\infty(\boldsymbol{g}, \epsilon)$ is the l_∞ norm ball centered around $\boldsymbol{g}$ with radius ϵ. For UIA, the defense generates a targeted adversarial perturbation through gradient descent, i.e., $\tilde{\boldsymbol{g}} \leftarrow \prod_{\mathcal{B}_\infty(\boldsymbol{g},\gamma)} \left(\tilde{\boldsymbol{g}} - \alpha \cdot \text{sign}(\nabla_{\boldsymbol{g}} \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{g}, \boldsymbol{a}_t))\right)$, to make the gradients misrecognized as the target user $\boldsymbol{a}_t$. By default, we set the total number of steps to be 5, $\gamma = 0.005$, and $\alpha = 0.002$.
(4) **Variational Information Bottleneck (VIB).** This defense inserts an additional VIB layer [4] that splits the neural network $f_{\boldsymbol{\theta}}$ into a probabilistic encoder $p(\mathbf{h}|\mathbf{x})$ and a decoder $q(\mathbf{y}|\mathbf{h})$, where $\mathbf{h}$ is a latent representation that follows a Gaussian distribution. An additional Kullback-Leibler (KL) divergence term is introduced to the training loss: $\mathcal{L}_{VIB} = \mathcal{L}(\boldsymbol{\theta}, \mathcal{D}) + \beta \cdot KL(p(\mathbf{h}|\mathbf{x})||q(\mathbf{z}))$, where $q(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ is the standard Gaussian. Optimizing this VIB objective reduces the mutual information $I(\mathbf{x}; \mathbf{h})$ between the representation and the input by minimizing a variational upper bound. Prior work suggests that this helps to reduce the model's dependence on input's sensitive attributes and improve privacy [77, 78, 82]. We set $\beta = 0.01$ as the default for our experiments.
(5) **Differential Privacy (DP-SGD).** Differential privacy (DP) [23] provides a rigorous notion of algorithmic privacy.

Definition 6.1. ***(ε, δ)-Differential Privacy.*** An algorithm $\mathcal{M}$ is said to satisfy (ε, δ)-DP if for all sets of events S defined on the output of $\mathcal{M}$ and all neighboring datasets $\mathcal{D}, \mathcal{D}'$ that differ in one sample, the following inequality holds:

$$\mathbb{P}(\mathcal{M}(\mathcal{D}) \in S) \leq e^{\varepsilon}\, \mathbb{P}(\mathcal{M}(\mathcal{D}') \in S) + \delta.$$

The most widely adopted DP algorithm for training ML model is DP-SGD [1]. At each step of training, the DP-SGD algorithm first clips the l_2 norm of per-sample gradients $\tilde{\boldsymbol{g}}_i \leftarrow \boldsymbol{g}_i / \max(1, \frac{||\boldsymbol{g}_i||_2}{\Delta})$ and then injects calibrated Gaussian noise to get the aggregated gradients $\tilde{\boldsymbol{g}} \leftarrow \frac{1}{k} \sum_{i=0}^{k} \left(\tilde{\boldsymbol{g}}_i + \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})\right)$. DP-SGD achieves (ε, δ)-differential privacy for any $\delta > 0$ with $\varepsilon = \Delta \sqrt{2 \log \frac{1.25}{\delta}} / \sigma$ for each step, while the total privacy loss can be obtained through composition. By default, we set $\Delta = 2$ and $\sigma = 0.1$ (corresponds to per-step $\varepsilon = 96.90$ when $\delta = 10^{-5}$).

6.2 Defense Evaluation

In Figure 6, we compare the performance of defenses against static and adaptive adversaries. Due to space limits, here we focus on PIA on the adult dataset. The full results including all four types of inference attacks are available in Appendix Figure 12. We observe that heuristic defenses such as Gradient Pruning, SignSGD, and Adversarial Perturbation can successfully defend against static adversaries in terms of reducing the advantage of the adversary to zero. However, these defenses are ineffective against adaptive adversaries aware of the defense. For instance, in the case of gradient pruning, the adaptive adversary can achieve a high advantage (0.7841) that is only slightly decreased compared to no defense (0.9363). Interestingly, in the case of Adversarial Perturbation, we found that the adaptive adversary's performance is increased, rather than decreased, compared to no defense, reaching a perfect advantage and AUROC of 1.00. For the rest of the defenses, namely, VIB and DP-SGD, the attack performance is consistent across static and adaptive adversaries. However, only DP-SGD manages to effectively reduce the advantage of the adaptive adversary to near zero.

To understand the privacy-utility trade-off of these defenses, we plot the PIA adversary's advantage evaluated on the training data versus the measured AUROC of the network on predicting the task label on the test dataset on the Adult dataset in Figure 7. We consider three different sets of parameters for each type of defense (details in Appendix). We observe that in the case of static adversaries, SignSGD achieves the best trade-off that approximates the ideal defense (upper left corner) by reducing the advantage to zero without affecting model utility. However, in the case of adaptive adversary, only DP-SGD provides a meaningful notion of privacy, at the cost of diminishing model utility. Moreover, there may exist stronger adversaries that are more resilient against these defenses. For instance, in Table 2, we show that an adversary using principal component analysis (PCA) with 50 principal dimensions as dimensionality reduction can bypass the DP-SGD defense with $\varepsilon = 96.90$ and $\delta = 10^{-5}$ that defends an adversary using max-pooling, and requires 15× larger noise to thwart.

In the next section, we analyze the underlying principles of these defenses and the necessary ingredients for a successful defense.

6.3 Defense Analyses

In this section, we provide an information-theoretic perspective for understanding and analyzing defenses against inference attacks from gradients.

Information-theoretic View on Inference Privacy. The inference attacks captured in the unified game can be viewed as performing statistical inference [21] on properties of the underlying data distributions upon observing samples of the gradients. A well-known information-theoretic result for analyzing inference is Fano's inequality, which guarantees a lower bound on the estimation error of any inference adversary. Formally, consider any arbitrary data release mechanism that provides $\mathbf{Y}$ computed from the private discrete random variable $\mathbf{X}$ supported on $\mathcal{X}$. Any inference from the observation $\mathbf{Y}$ must produce an estimate $\hat{\mathbf{X}}$ that satisfies the Markov chain $\mathbf{X} \rightarrow \mathbf{Y} \rightarrow \hat{\mathbf{X}}$. Let $\mathbf{e}$ be a binary random variable that indicates an error, i.e., $\mathbf{e} = 1$ if $\hat{\mathbf{X}} \neq \mathbf{X}$. Then we have

$$H(\mathbf{X}|\mathbf{Y}) \leq H(\mathbf{X}|\hat{\mathbf{X}}) \leq H_2(\mathbf{e}) + \mathbb{P}(\mathbf{e} = 1) \log(|\mathcal{X}| - 1), \tag{6}$$

where $H_2(\mathbf{e}) = -\mathbb{P}(e = 1) \log \mathbb{P}(e = 1) - \big(1 - \mathbb{P}(e = 1)\big) \log \big(1 - \mathbb{P}(e = 1)\big)$ is the binary entropy. For $|\mathcal{X}| > 2$, a standard treatment is to consider the mutual information $I(\mathbf{X}; \mathbf{Y}) = H(\mathbf{X}) - H(\mathbf{X}|\mathbf{Y})$ and $H_2(\mathbf{e}) \leq \log 2$, and thereby we can obtain a lower bound on the error probability:

$$\mathbb{P}(\hat{\mathbf{X}} \neq \mathbf{X}) \geq \frac{H(\mathbf{X}) - I(\mathbf{X}; \mathbf{Y}) - \log 2}{\log(|\mathcal{X}| - 1)}. \tag{7}$$

Note that this bound is vacuous when $|\mathcal{X}| = 2$, and a slightly tighter bound can be obtained by considering $H_2(\mathbf{e})$ exactly (rather than using the approximating bound of log 2) and numerically computing the lowest error probability that satisfies the inequality in (6), as noted by prior work [34]. The bound in inequality (7) captures both the prior (via $H(\mathbf{X})$) and the cardinality of the sensitive variable alphabet, indicating that data with a large degree of uncertainty is hard to infer or reconstruct, which aligns with intuition from Balle *et al.* [7]. Inequality (7) generically holds for any data release mechanism. In the context of inference from gradients, the adversary's goal is to obtain an estimate of $\mathbf{a}$ upon observing $\tilde{\mathbf{g}}$, which can be described as a Markov chain of $\mathbf{a} \rightarrow \mathbf{x} \rightarrow \mathbf{g} \rightarrow \tilde{\mathbf{g}} \rightarrow \hat{\mathbf{a}}$. Since the adversary's success rate is $p = 1 - \mathbb{P}(\mathbf{e} = 1)$, one can get an immediate upper bound on the adversary's advantage:

$$\mathsf{Adv}(p) \leq 1 - \frac{H(\mathbf{a}) - I(\mathbf{a}; \tilde{\mathbf{g}}) - \log 2}{(1 - p^*) \log(m - 1)}. \tag{8}$$

As $H(\mathbf{a})$ is a constant, this indicates that reducing $I(\mathbf{a}; \tilde{\mathbf{g}})$ results in increasing the lower bound of the error probability and consequently diminishing the adversary's advantage. This analysis can be generalized to continuous sensitive variables by applying continuum Fano's inequality [22].

Understanding Defenses. Next, we provide an explanation of the failures of heuristic defenses using the above framework and argue that a successful defense should effectively minimize the mutual information $I(\mathbf{a}; \tilde{\mathbf{g}})$ between the gradients and the sensitive variable. The Gradient Pruning and SignSGD defenses can be viewed as trying to reduce the number of transmitted bits in the gradients. However, this does not necessarily reduce the mutual information. The neural network classifier $f_{\boldsymbol{\phi}} : \mathbf{g} \rightarrow \mathbf{a}$ used in the Adversarial Perturbation defense is trained to minimize cross-entropy loss, which provides an approximate upper bound on the conditional entropy $H(\mathbf{a}|\mathbf{g})$, and serves as a proxy for estimating the mutual information $I(\mathbf{a}; \tilde{\mathbf{g}}) = H(\mathbf{a}) - H(\mathbf{a}|\mathbf{g})$. However, generating adversarial perturbations to produce $\tilde{\mathbf{g}}$ against this fixed classifier does not necessarily result in a reduction of the mutual information $I(\mathbf{a}; \tilde{\mathbf{g}})$, and likely increases it. This is because the gradient steps

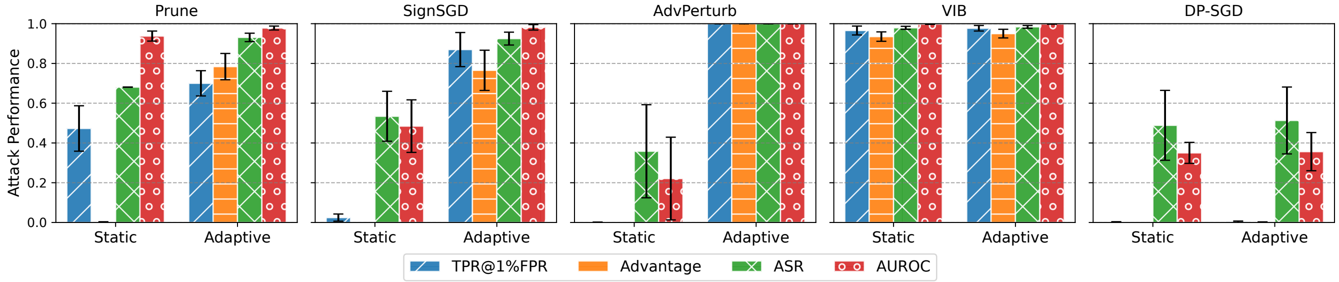


Figure 6: Comparison of various defenses against static and adaptive *Property Inference Attack* on the Adult dataset with a batch size of 16. A complete result of all inference attacks is provided in Appendix Figure 12.

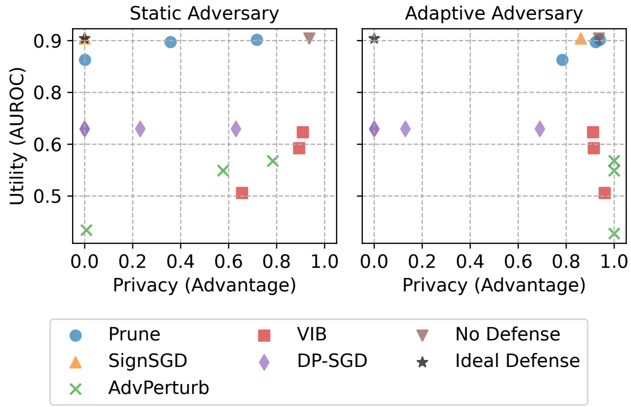


Figure 7: Privacy-Utility trade-off of various defenses against *Property Inference Attack* on the Adult dataset.

$\nabla_{\boldsymbol{g}}\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{g}, \boldsymbol{a})$ used to generate the protective perturbation also contain information about $\boldsymbol{a}$. As the perturbation generation process is deterministic, an adaptive adversary can learn to pick up these patterns and gain additional advantage. In the case of VIB, the mechanism is stochastic but optimizing the VIB objective only gradually reduces the mutual information $I(\mathbf{x}; \mathbf{h})$ between the latent representation $\mathbf{h}$ and the input $\mathbf{x}$, which still does not guarantee a reduction in $I(\mathbf{a}; \tilde{\mathbf{g}})$ during the optimization process. By design, differential privacy is not intended to protect against statistical inference as its goal is to preserve the statistical properties of the dataset while protecting the privacy of individual samples. However, an alternative information-theoretical interpretation of differential privacy is that it places a constraint on mutual information [11, 17]. An easy way to see this is that by adding Gaussian noises to the gradients, the DP-SGD algorithm essentially creates a Gaussian channel between the true and released gradients, thereby placing a constraint on $I(\mathbf{g}; \tilde{\mathbf{g}})$, which further bounds $I(\mathbf{a}; \tilde{\mathbf{g}})$ as $I(\mathbf{a}; \tilde{\mathbf{g}}) \leq I(\mathbf{g}; \tilde{\mathbf{g}})$ according to the data processing inequality. More concretely, due to the Gaussian channel $\tilde{\mathbf{g}} = \mathbf{g} + \mathcal{N}(\mathbf{0}, \sigma^2\boldsymbol{I})$, we have the upper bound given by the channel capacity $I(\mathbf{g}; \tilde{\mathbf{g}}) \leq \frac{1}{2}\log(1 + \frac{P}{\sigma})$, if the gradients $\boldsymbol{g}$ satisfy an average power constraint $\mathbb{E}[\|\boldsymbol{g}\|_2^2] \leq nP$, where n is the dimensionality of $\boldsymbol{g}$. One can obtain a stronger result in cases where the l_2 sensitivity is bounded (e.g., Theorem 2 in [34]).

Table 2: Comparison of different dimensionality reduction strategies in PIA on Adult against DP-SGD defense ($\delta = 10^{-5}$).

ε	Adversary Type	AUROC	TPR@1%FPR	ASR	Advantage
96.90	MaxPooling	0.3004 ±0.0773	0.0017 ±0.0010	0.5732 ±0.1124	0.0001 ±0.0002
96.90	PCA	0.9825 ±0.0112	0.7284 ±0.1679	0.9437 ±0.0222	0.8239 ±0.0694
6.46	PCA	0.7010 ±0.0278	0.0471 ±0.0120	0.6995 ±0.0091	0.0598 ±0.0286

It is worth noting that the goal of our analyses here is to provide a perspective for understanding the effectiveness of a class of defense strategies, rather than deriving tight bounds. Additionally, as mutual information is a statistical quantity, the mutual information interpretation of inference privacy inherently only captures the average-case privacy risk. In the next section, we provide a privacy auditing framework for empirically estimating the privacy risk by approximating the worst-case scenario.

7 EMPIRICAL ESTIMATION OF PRIVACY RISK

In the privacy game defined in Definition 3.1, the data is randomly sampled from the distribution, which only captures the average-case privacy risk and therefore cannot be used for reasoning about the minimal level of noise required for ensuring a certain level of privacy, as it may underestimate the privacy risk in the worst case. To better understand the privacy risk in the worst-case scenario, we provide a privacy auditing framework for empirically estimating the privacy leakage of a specific type of inference attack, namely, attribute inference, by allowing the data to be chosen adversarially. We start with a formal definition of per-attribute privacy following prior work [3, 32]:

Definition 7.1. ***Per-attribute DP.*** A randomized mechanism $\mathcal{M}$ is (ε, δ)-per-attribute DP if for all pairs of inputs x, x' differing only on a single attribute and for all events S defined on the output of $\mathcal{M}$, the following inequality holds:

$$\mathbb{P}[\mathcal{M}(x) \in S] \leq e^{\varepsilon} \cdot \mathbb{P}[\mathcal{M}(x') \in S] + \delta.$$

One can show that DP-SGD satisfies (ε, δ)-per-attribute DP. However, it is hard to derive the privacy parameter analytically, as the per-attribute sensitivity of the gradient is not readily tractable and the common technique of gradient clipping only provides a very loose bound on sensitivity. Instead, we seek to obtain an empirical estimate of the per-attribute DP for *each step* through the following audit game.

Definition 7.2. ***Per-Attribute Privacy Audit Game.*** Suppose $\mathrm{a} \in [m]$ is a discrete attribute that takes on m values. The per-attribute privacy audit game between a challenger (private learner) and an adversary (auditor) is as follows:

(1) Adversary chooses a record $\boldsymbol{z}$ with attribute value $\boldsymbol{a}$.
(2) Challenger samples a uniformly random private bit $\mathrm{b} \in \{0, 1\}$. If $\mathrm{b} = 1$, assign the attribute in $\boldsymbol{z}$ with a new value uniformly sampled from $[m]\backslash\{\boldsymbol{a}\}$.
(3) Challenger obtains the latest model parameters as $\boldsymbol{\theta}$ through the training algorithm $\mathcal{T}$.
(4) Challenger computes the gradient of the record, $\boldsymbol{g} = \nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{z})$.
(5) Challenger applies the DP-SGD algorithm $\mathcal{M}$ to produce a privatized version of the gradient $\tilde{\boldsymbol{g}} = \mathcal{M}(\boldsymbol{g})$.
(6) The adversary $\mathcal{A}$ gets access to $\mathcal{L}$, $\mathcal{T}$, and the model parameters $\boldsymbol{\theta}$, released gradients $\boldsymbol{g}$, the auxiliary information about the record $\varphi(\boldsymbol{z})$, and the defense mechanism $\mathcal{M}$.
(7) The adversary outputs the inferred information $\hat{\mathrm{b}}$, i.e., $\hat{\mathrm{b}} \leftarrow \mathcal{A}(\mathcal{L}, \mathcal{T}, \boldsymbol{\theta}, \boldsymbol{g}, \varphi(\mathrm{z}), \mathcal{M})$. The adversary wins if $\hat{\mathrm{b}} = \mathrm{b}$ and loses otherwise.

There are two major differences between the audit game and the inference game as defined in Definition 3.1. First, the record is chosen by the adversary, instead of being randomly drawn from the distribution, which aims to simulate the worst-case scenario over all adjacent input pairs as captured by the per-attribute DP definition. Second, instead of having access to distributional information $\mathbb{P}(\mathrm{x}, \mathrm{y}, \mathrm{a})$, the adversary gets access to some auxiliary information about the record $\varphi(\boldsymbol{z})$, which is assumed to be all the remaining features except for $\boldsymbol{a}$. This is to approximate the strong adversarial assumption in per-attribute DP where the adversary has access to everything except for one attribute.

Empirical Privacy Estimate. Analogous to the operational interpretation of canonical DP [48], we can interpret attribute DP as a hypothesis test with $\mathrm{b} = 0$ as the null hypothesis ($\mathcal{H}_0$) and $\mathrm{b} = 1$ as the alternative hypothesis ($\mathcal{H}_1$). We compute the test statistics $\mathrm{t}(\tilde{\boldsymbol{g}}) = ||\tilde{\boldsymbol{g}} - \boldsymbol{g}_{\mathcal{H}_0}||_2$ using the l_2 norm between the observed gradient $\tilde{\boldsymbol{g}}$ and the hypothetical gradient $\boldsymbol{g}_{\mathcal{H}_0}$ under $\mathcal{H}_0$ (i.e., $\boldsymbol{g}_{\mathcal{H}_0} = \boldsymbol{g}/\max(1, \frac{||\boldsymbol{g}||_2}{\Delta})$ when $\varphi(\boldsymbol{z}) = \boldsymbol{a}$). This is connected to the likelihood of observing $\tilde{\boldsymbol{g}}$ under $\mathcal{H}_0$ since $\tilde{\boldsymbol{g}}_{\mathcal{H}_0} \sim \mathcal{N}(\boldsymbol{g}_{\mathcal{H}_0}, \sigma^2\boldsymbol{I})$. We then execute the game several times to get an empirical distribution of the test statistics. Building on prior works on auditing canonical DP with membership inference attacks [5, 43, 63, 70], finally, we derive the empirical privacy loss parameter $\hat{\varepsilon}$ given the false positive rate (FPR) and false negative rate (FNR) at critical value c as

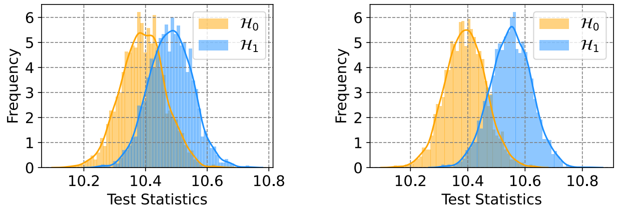
$$\hat{\varepsilon} = \max\left(\log\frac{1-\delta-\text{FPR}}{\text{FNR}}, \log\frac{1-\delta-\text{FNR}}{\text{FPR}}\right),$$

where the critical value c is chosen over all possible values to maximize the empirical estimate $\hat{\varepsilon}$ to obtain a worst-case measure. Similar to previous work [70], we additionally compute and report the 95% confidence intervals for $\hat{\varepsilon}$ using the Clopper-Pearson method.

Crafting the Worst-case Sample. To further improve the estimate, we approximate the worst-case scenario by crafting a canary record $\boldsymbol{z}^*$ to maximize the expected difference in the test statistics between $\mathcal{H}_0$ and $\mathcal{H}_1$, via the optimization

$$\boldsymbol{z}^* = \arg\max_{\boldsymbol{z}} \mathrm{Dist}(\boldsymbol{g}_{\mathcal{H}_0}, \boldsymbol{g}_{\mathcal{H}_1}),$$

where $\mathrm{Dist}(\cdot, \cdot)$ is a distance measure. We experimented with cosine similarity and mean squared error (MSE) and found that MSE performs better empirically.



(a) Randomly sampled record. **(b) Adversarially crafted record.**

Figure 8: Comparison of the test statistics distribution from auditing games with different choices of test record $\boldsymbol{z}$.

Empirical Results. We first conduct experiments on the Adult dataset using a fully-connected neural network with one hidden layer of 100 neurons to verify the effectiveness of the adversarially crafted sample. We compute the test statistics for 5,000 trials with $\Delta = 2$ and $\sigma = 0.1$ and plot the histogram of the test statistics in Figure 8. We observe that the distributions of test statistics under $\mathcal{H}_0$ and $\mathcal{H}_1$ are more separable using the adversarially crafted canary record, compared to a randomly drawn record from the data distribution, thereby providing a better estimate on the worst-case privacy risk. We then compare the empirical estimated $\hat{\varepsilon}$ to the theoretical ε computed with the gradient clipping bound Δ as the per-attribute sensitivity, at $\delta = 10^{-5}$, using adversarially crafted records. Figure 9 plots the empirically estimated $\hat{\varepsilon}$ and the theoretical ε normalized by the total number of attributes ($N = 14$) with varying clipping bound Δ and noise level σ. We observe that using the clipping bound as the per-attribute sensitivity indeed leads to a very conservative estimate of the privacy loss, with a large gap ($\varepsilon/\hat{\varepsilon} = 1.86N$) when $\sigma = 0.1$ and $\Delta = 4$. As the clipping bound reduces, the ratio gradually approximates to N ($\varepsilon/\hat{\varepsilon} = 1.14N$ when $\Delta = 1.5$). When the clipping bound is fixed to $\Delta = 2$, the gap is relatively consistent across different noise levels (e.g., $\varepsilon/\hat{\varepsilon} = 1.20N$ when $\sigma = 0.08$ and $\varepsilon/\hat{\varepsilon} = 1.33N$ when $\sigma = 0.13$).

8 CONCLUSION AND DISCUSSION

We conduct a systematic analysis of private information leakage from gradients under different levels of adversarial uncertainties within a unified inference framework. We provide an information-theoretic perspective for explaining and analyzing the efficacy of defenses for preventing inference through the gradient channel. Finally, we introduce an auditing approach for estimating realistic privacy risks against attribute inference. There are three primary takeaways from this study: (1) data aggregation alone does not provide sufficient privacy in distributed learning, (2) reducing the mutual information is a key ingredient of successful defenses against inference from gradients, and (3) it is important to specify the privacy context (e.g., average vs. worst-case vulnerability of a dataset) when estimating privacy risks.

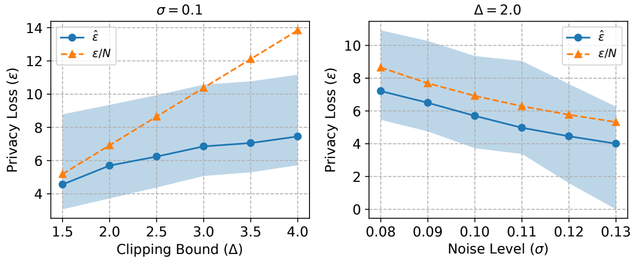


Figure 9: Comparison of the empirical per-attribute privacy loss ($\hat{\varepsilon}$) with 95% confidence interval and the theoretical privacy loss (ε) normalized by the total number of attributes N.

Our findings open up several interesting discussions. Firstly, information leakage from the gradient exhibits distinct characteristics compared to leakage from the model parameters, suggesting a competing relationship. In the extreme case, a perfectly memorized sample could pose high privacy risks via model parameters, yet disclose no information through gradients if the loss is zero. Secondly, while the quantification of inference attack risks with mutual information provides theoretical guarantees and a guiding principle for privacy mechanism design, there are practical disadvantages in tractability and composition. As mutual information is a statistical quantity that depends on the unknown data distribution, practical application requires estimation from data, which may be challenging in distributed learning scenarios. Further, addressing information leakage across multiple rounds requires dealing with the mutual information between the private variable and all gradient observations handled jointly, i.e., $I(\mathrm{a}; \mathrm{g}_1, \ldots, \mathrm{g}_r)$, where g_i denotes the gradients observed in each round. However, a complication of this combined mutual information is that it cannot simply be exactly decomposed as a summation of single-round mutual information terms $I(\mathrm{a}; \mathrm{g}_i)$, i.e., mutual information lacks a convenient composition property. Finally, we demonstrated that among the defenses considered, only DP-SGD provides a meaningful notion of privacy against adaptive adversaries, despite affecting model utility. However, the loss in utility can softened by improving the algorithms. For instance, recent research [2] showed in the centralized setting that state-of-the-art DP-SGD solutions provide better privacy than most empirical defenses at similar utility. Improving multi-round privacy analysis and tuning defenses (e.g., by minimizing the mutual information objective) towards better privacy-utility trade-offs using public data are interesting avenues for future research.

ACKNOWLEDGMENTS

This research was sponsored in part by the National Institute of Health (NIH) under grant number U54HG012510.

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 308–318.

[2] Michael Aerni, Jie Zhang, and Florian Tramèr. 2024. Evaluations of Machine Learning Privacy Defenses are Misleading. *arXiv preprint arXiv:2404.17399* (2024).

[3] Faraz Ahmed, Alex X Liu, and Rong Jin. 2016. Social graph publishing with privacy guarantees. In *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 447–456.

[4] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. 2016. Deep Variational Information Bottleneck. In *International Conference on Learning Representations*.

[5] Galen Andrew, Peter Kairouz, Sewoong Oh, Alina Oprea, Hugh Brendan McMahan, and Vinith Menon Suriyakumar. 2023. One-shot Empirical Privacy Estimation for Federated Learning. In *The Twelfth International Conference on Learning Representations*.

[6] Giuseppe Ateniese, Luigi V Mancini, Angelo Spognardi, Antonio Villani, Domenico Vitali, and Giovanni Felici. 2015. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *International Journal of Security and Networks* 10, 3 (2015), 137–150.

[7] Borja Balle, Giovanni Cherubin, and Jamie Hayes. 2022. Reconstructing training data with informed adversaries. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1138–1156.

[8] Barry Becker and Ronny Kohavi. 1996. Adult. UCI Machine Learning Repository. (1996). DOI: https://doi.org/10.24432/C5XW20.

[9] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. 2018. signSGD: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*. PMLR, 560–569.

[10] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[11] Mark Bun and Thomas Steinke. 2016. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*. Springer, 635–658.

[12] Houwei Cao, David G Cooper, Michael K Keutmann, Ruben C Gur, Ani Nenkova, and Ragini Verma. 2014. Crema-d: Crowd-sourced emotional multimodal actors dataset. *IEEE transactions on affective computing* 5, 4 (2014), 377–390.

[13] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. 2022. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1897–1914.

[14] Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. 2023. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*. 5253–5270.

[15] Nicholas Carlini and David Wagner. 2017. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*. 3–14.

[16] Harsh Chaudhari, John Abascal, Alina Oprea, Matthew Jagielski, Florian Tramer, and Jonathan Ullman. 2023. SNAP: Efficient extraction of private properties with poisoning. In *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 400–417.

[17] Paul Cuff and Lanqing Yu. 2016. Differential privacy as a mutual information constraint. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 43–54.

[18] Trung Dang, Om Thakkar, Swaroop Ramaswamy, Rajiv Mathews, Peter Chin, and Françoise Beaufays. 2022. A method to reveal speaker identity in distributed asr training, and how to counter it. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 4338–4342.

[19] Akash Dhasade, Anne-Marie Kermarrec, Rafael Pires, Rishi Sharma, and Milos Vujasinovic. 2023. Decentralized learning made easy with DecentralizePy. In *Proceedings of the 3rd Workshop on Machine Learning and Systems*. 34–41.

[20] Ilias Driouich, Chuan Xu, Giovanni Neglia, Frederic Giroire, and Eoin Thomas. 2022. A Novel Model-Based Attribute Inference Attack in Federated Learning. In *FL-NeurIPS'22-Federated Learning: Recent Advances and New Challenges workshop in Conjunction with NeurIPS 2022*.

[21] Flávio du Pin Calmon and Nadia Fawaz. 2012. Privacy against statistical inference. In *2012 50th annual Allerton conference on communication, control, and computing (Allerton)*. IEEE, 1401–1408.

[22] John C Duchi and Martin J Wainwright. 2013. Distance-based and continuum Fano inequalities with applications to statistical estimation. *arXiv preprint arXiv:1311.2669* (2013).

[23] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*. Springer, 265–284.

[24] Ahmed Roushdy Elkordy, Jiang Zhang, Yahya H Ezzeldin, Konstantinos Psounis, and Salman Avestimehr. 2023. How Much Privacy Does Federated Learning with Secure Aggregation Guarantee? *Proceedings on Privacy Enhancing Technologies* (2023).

[25] Florian Eyben, Martin Wöllmer, and Björn Schuller. 2010. Opensmile: the munich versatile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM international conference on Multimedia*. 1459–1462.

[26] Tiantian Feng, Hanieh Hashemi, Rajat Hebbar, Murali Annavaram, and Shrikanth S Narayanan. 2021. Attribute inference attack of speech emotion recognition in federated learning settings. *arXiv preprint arXiv:2112.13416* (2021).

[27] Eibe Frank and Mark Hall. 2001. A simple approach to ordinal classification. In *Machine Learning: ECML 2001: 12th European Conference on Machine Learning Freiburg, Germany, September 5–7, 2001 Proceedings 12*. Springer, 145–156.

[28] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*. 1322–1333.

[29] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. 2014. Privacy in pharmacogenetics: An {End-to-End} case study of personalized warfarin dosing. In *23rd USENIX security symposium (USENIX Security 14)*. 17–32.

[30] Karan Ganju, Qi Wang, Wei Yang, Carl A Gunter, and Nikita Borisov. 2018. Property inference attacks on fully connected neural networks using permutation invariant representations. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*. 619–633.

[31] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. 2020. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems* 33 (2020), 16937–16947.

[32] Badih Ghazi, Ravi Kumar, Pasin Manurangsi, and Thomas Steinke. 2022. Algorithms with More Granular Differential Privacy Guarantees. *arXiv preprint arXiv:2209.04053* (2022).

[33] Dhruv Guliani, Françoise Beaufays, and Giovanni Motta. 2021. Training speech recognition models with federated learning: A quality/cost framework. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 3080–3084.

[34] Chuan Guo, Alexandre Sablayrolles, and Maziar Sanjabi. 2023. Analyzing privacy leakage in machine learning via multiple hypothesis testing: A lesson from fano. In *International Conference on Machine Learning*. PMLR, 11998–12011.

[35] Samyak Gupta, Yangsibo Huang, Zexuan Zhong, Tianyu Gao, Kai Li, and Danqi Chen. 2022. Recovering private text in federated learning of language models. *Advances in neural information processing systems* 35 (2022), 8130–8143.

[36] Fasih Haider, Senja Pollak, Pierre Albert, and Saturnino Luz. 2021. Emotion recognition in low-resource settings: An evaluation of automatic feature selection methods. *Computer Speech & Language* 65 (2021), 101119.

[37] Niv Haim, Gal Vardi, Gilad Yehudai, Ohad Shamir, and Michal Irani. 2022. Reconstructing training data from trained neural networks. *Advances in Neural Information Processing Systems* 35 (2022), 22911–22924.

[38] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. 2018. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604* (2018).

[39] Jamie Hayes, Borja Balle, and Saeed Mahloujifar. 2024. Bounding training data reconstruction in dp-sgd. *Advances in Neural Information Processing Systems* 36 (2024).

[40] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).

[41] Kevin Hsieh, Aaron Harlap, Nandita Vijaykumar, Dimitris Konomis, Gregory R Ganger, Phillip B Gibbons, and Onur Mutlu. 2017. Gaia:{Geo-Distributed} machine learning approaching {LAN} speeds. In *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*. 629–647.

[42] Yangsibo Huang, Samyak Gupta, Zhao Song, Kai Li, and Sanjeev Arora. 2021. Evaluating gradient inversion attacks and defenses in federated learning. *Advances in Neural Information Processing Systems* 34 (2021), 7232–7241.

[43] Matthew Jagielski, Jonathan Ullman, and Alina Oprea. 2020. Auditing differentially private machine learning: How private is private sgd? *Advances in Neural Information Processing Systems* 33 (2020), 22205–22216.

[44] Bargav Jayaraman and David Evans. 2022. Are attribute inference attacks just imputation?. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. 1569–1582.

[45] Jinyuan Jia and Neil Zhenqiang Gong. 2018. {AttriGuard}: A practical defense against attribute inference attacks via adversarial machine learning. In *27th USENIX Security Symposium (USENIX Security 18)*. 513–529.

[46] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. 2019. Memguard: Defending against black-box membership inference attacks via adversarial examples. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*. 259–274.

[47] Kaggle. 2012. Heritage Health Prize. https://www.kaggle.com/c/hhp/data. (2012).

[48] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. 2015. The composition theorem for differential privacy. In *International conference on machine learning*. PMLR, 1376–1385.

[49] Nikhil Kandpal, Krishna Pillutla, Alina Oprea, Peter Kairouz, Christopher A Choquette-Choo, and Zheng Xu. 2023. User Inference Attacks on Large Language Models. In *International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023*.

[50] Raouf Kerkouche, Gergely Ács, and Claude Castelluccia. 2020. Federated learning in adversarial settings. *arXiv preprint arXiv:2010.07808* (2020).

[51] Raouf Kerkouche, Gergely Ács, and Mario Fritz. 2023. Client-specific property inference against secure aggregation in federated learning. In *Proceedings of the 22nd Workshop on Privacy in the Electronic Society*. 45–60.

[52] Maximilian Lam, Gu-Yeon Wei, David Brooks, Vijay Janapa Reddi, and Michael Mitzenmacher. 2021. Gradient disaggregation: Breaking privacy in federated learning by reconstructing the user participant matrix. In *International Conference on Machine Learning*. PMLR, 5959–5968.

[53] Klas Leino and Matt Fredrikson. 2020. Stolen memories: Leveraging model memorization for calibrated {White-Box} membership inference. In *29th USENIX security symposium (USENIX Security 20)*. 1605–1622.

[54] Guoyao Li, Shahbaz Rezaei, and Xin Liu. 2022. User-Level Membership Inference Attack against Metric Embedding Learning. In *ICLR 2022 Workshop on PAIR: Privacy, Accountability, Interpretability, Robustness, Reasoning on Structured Data*.

[55] Mu Li, David G Andersen, Jun Woo Park, Alexander J Smola, Amr Ahmed, Vanja Josifovski, James Long, Eugene J Shekita, and Bor-Yiing Su. 2014. Scaling distributed machine learning with the parameter server. In *11th USENIX Symposium on operating systems design and implementation (OSDI 14)*. 583–598.

[56] Zhuohang Li, Jiaxin Zhang, and Jian Liu. 2023. Speech Privacy Leakage from Shared Gradients in Distributed Learning. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.

[57] Zhuohang Li, Jiaxin Zhang, Luyang Liu, and Jian Liu. 2022. Auditing privacy defenses in federated learning via generative gradient leakage. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10132–10142.

[58] Yugeng Liu, Rui Wen, Xinlei He, Ahmed Salem, Zhikun Zhang, Michael Backes, Emiliano De Cristofaro, Mario Fritz, and Yang Zhang. 2022. {ML-Doctor}: Holistic risk assessment of inference attacks against machine learning models. In *31st USENIX Security Symposium (USENIX Security 22)*. 4525–4542.

[59] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*. 3730–3738.

[60] Andrew Lowy, Zhuohang Li, Jing Liu, Toshiaki Koike-Akino, Kieran Parsons, and Ye Wang. 2024. Why Does Differential Privacy with Large Epsilon Defend Against Practical Membership Inference Attacks? *arXiv preprint arXiv:2402.09540* (2024).

[61] Andrew Lowy and Meisam Razaviyayn. 2022. Private Federated Learning Without a Trusted Server: Optimal Algorithms for Convex Losses. In *The Eleventh International Conference on Learning Representations*.

[62] Lingjuan Lyu and Chen Chen. 2021. A novel attribute reconstruction attack in federated learning. *arXiv preprint arXiv:2108.06910* (2021).

[63] Samuel Maddock, Alexandre Sablayrolles, and Pierre Stock. 2022. CANIFE: Crafting Canaries for Empirical Privacy Measurement in Federated Learning. In *The Eleventh International Conference on Learning Representations*.

[64] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*.

[65] Saeed Mahloujifar, Esha Ghosh, and Melissa Chase. 2022. Property inference from poisoning. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1120–1137.

[66] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 1273–1282.

[67] Shagufta Mehnaz, Sayanton V Dibbo, Roberta De Viti, Ehsanul Kabir, Björn B Brandenburg, Stefan Mangard, Ninghui Li, Elisa Bertino, Michael Backes, Emiliano De Cristofaro, et al. 2022. Are your sensitive attributes private? Novel model inversion attribute inference attacks on classification models. In *31st USENIX Security Symposium (USENIX Security 22)*. 4579–4596.

[68] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. 2019. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE symposium on security and privacy (SP)*. IEEE, 691–706.

[69] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE symposium on security and privacy (SP)*. IEEE, 739–753.

[70] Milad Nasr, Shuang Songi, Abhradeep Thakurta, Nicolas Papernot, and Nicholas Carlin. 2021. Adversary instantiation: Lower bounds for differentially private machine learning. In *2021 IEEE Symposium on security and privacy (SP)*. IEEE, 866–882.

[71] Patrick O'Reilly, Andreas Bugler, Keshav Bhandari, Max Morrison, and Bryan Pardo. 2022. Voiceblock: Privacy through real-time adversarial attacks with audio-to-audio models. *Advances in Neural Information Processing Systems* 35 (2022), 30058–30070.

[72] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In

Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 10684–10695.

[73] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. 2019. White-box vs black-box: Bayes optimal strategies for membership inference. In *International Conference on Machine Learning*. PMLR, 5558–5567.

[74] Ahmed Salem, Apratim Bhattacharya, Michael Backes, Mario Fritz, and Yang Zhang. 2020. {Updates-Leak}: Data set inference and reconstruction attacks in online learning. In *29th USENIX security symposium (USENIX Security 20)*. 1291–1308.

[75] Ahmed Salem, Giovanni Cherubin, David Evans, Boris Köpf, Andrew Paverd, Anshuman Suri, Shruti Tople, and Santiago Zanella-Béguelin. 2023. SoK: Let the privacy games begin! A unified treatment of data inference privacy in machine learning. In *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 327–345.

[76] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. 2018. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246* (2018).

[77] Daniel Scheliga, Patrick Mäder, and Marco Seeland. 2022. Precode-a generic model extension to prevent deep gradient leakage. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 1849–1858.

[78] Daniel Scheliga, Patrick Mäder, and Marco Seeland. 2023. Privacy Preserving Federated Learning with Convolutional Variational Bottlenecks. *arXiv preprint arXiv:2309.04515* (2023).

[79] Shawn Shan, Emily Wenger, Jiayun Zhang, Huiying Li, Haitao Zheng, and Ben Y Zhao. 2020. Fawkes: Protecting privacy against unauthorized deep learning models. In *29th USENIX security symposium (USENIX Security 20)*. 1589–1604.

[80] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*. IEEE, 3–18.

[81] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems* 30 (2017).

[82] Congzheng Song and Vitaly Shmatikov. 2019. Overlearning Reveals Sensitive Attributes. In *International Conference on Learning Representations*.

[83] Liwei Song and Prateek Mittal. 2021. Systematic evaluation of privacy risks of machine learning models. In *30th USENIX Security Symposium (USENIX Security 21)*. 2615–2632.

[84] Jingwei Sun, Ang Li, Binghui Wang, Huanrui Yang, Hai Li, and Yiran Chen. 2021. Soteria: Provable defense against privacy leakage in federated learning from representation perspective. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9311–9319.

[85] Anshuman Suri and David Evans. 2022. Formalizing and Estimating Distribution Inference Risks. *Proceedings on Privacy Enhancing Technologies* (2022).

[86] Anshuman Suri, Pallika Kanani, Virendra J Marathe, and Daniel W Peterson. 2022. Subject membership inference attacks in federated learning. *arXiv preprint arXiv:2206.03317* (2022).

[87] Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. 2020. On adaptive attacks to adversarial example defenses. *Advances in neural information processing systems* 33 (2020), 1633–1645.

[88] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. 2016. Stealing machine learning models via prediction {APIs}. In *25th USENIX security symposium (USENIX Security 16)*. 601–618.

[89] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. 2019. A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM workshop on artificial intelligence and security*. 1–11.

[90] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei. 2019. Demystifying membership inference attacks in machine learning as a service. *IEEE Transactions on Services Computing* 14, 6 (2019), 2073–2089.

[91] Mark Vero, Mislav Balunovic, Dimitar Iliev Dimitrov, and Martin Vechev. 2023. TabLeak: Tabular Data Leakage in Federated Learning. In *International Conference on Machine Learning*. PMLR.

[92] Binghui Wang and Neil Zhenqiang Gong. 2018. Stealing hyperparameters in machine learning. In *2018 IEEE symposium on security and privacy (SP)*. IEEE, 36–52.

[93] Kuan-Chieh Wang, Yan Fu, Ke Li, Ashish Khisti, Richard Zemel, and Alireza Makhzani. 2021. Variational model inversion attacks. *Advances in Neural Information Processing Systems* 34 (2021), 9706–9719.

[94] Ruihan Wu, Xiangyu Chen, Chuan Guo, and Kilian Q Weinberger. 2023. Learning To Invert: Simple Adaptive Attacks for Gradient Inversion in Federated Learning. In *Uncertainty in Artificial Intelligence*. PMLR, 2293–2303.

[95] Xi Wu, Matthew Fredrikson, Somesh Jha, and Jeffrey F Naughton. 2016. A methodology for formalizing model-inversion attacks. In *2016 IEEE 29th Computer Security Foundations Symposium (CSF)*. IEEE, 355–370.

[96] Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. 2022. Enhanced membership inference attacks against machine learning models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. 3093–3106.

[97] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*. IEEE, 268–282.

[98] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. 2021. See through gradients: Image batch recovery via gradinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16337–16346.

[99] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. 2020. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8715–8724.

[100] Kai Yue, Richeng Jin, Chau-Wai Wong, Dror Baron, and Huaiyu Dai. 2023. Gradient obfuscation gives a false sense of security in federated learning. In *32nd USENIX Security Symposium (USENIX Security 23)*. 6381–6398.

[101] Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. 2020. The secret revealer: Generative model-inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 253–261.

[102] Zhifei Zhang, Yang Song, and Hairong Qi. 2017. Age progression/regression by conditional adversarial autoencoder. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5810–5818.

[103] Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep leakage from gradients. *Advances in neural information processing systems* 32 (2019).

A DETAILED EXPERIMENTAL SETUP

A.1 Details About ML Models

For the Adult and Health datasets, we train a fully-connected neural network with two hidden layers of sizes 32 and 16. For the CREMA-D, we train a fully-connected neural network with two hidden layers of 64 neurons. For the CelebA and UTKFace datasets, we train a convolutional neural network with 9 convolutional layers and 2 MaxPooling layers (details in Table 3). ReLU is used as the default activation function for all models.

Table 3: Model architecture for image datasets.

Layer	Kernel	Stride	Output
Conv2D	3×3	1×1	8
Conv2D	3×3	1×1	16
Conv2D	3×3	1×1	16
Conv2D	3×3	1×1	32
Conv2D	3×3	1×1	32
Conv2D	3×3	1×1	32
MaxPool2D	3×3	3×3	32
Conv2D	3×3	1×1	32
Conv2D	3×3	1×1	32
Conv2D	3×3	1×1	32
MaxPool2D	3×3	3×3	32
Flatten	–	–	–
FC	–	–	# of classes

A.2 Parameter Choices for Privacy-Utility Analysis of Defenses

For each type of defense, we consider three different sets of parameters in Figure 7: Gradient Pruning with 90%, 95%, and 99% rate; Adversarial Perturbation with ($\gamma = 5 \times 10^{-4}, \alpha = 2 \times 10^{-4}$), ($\gamma = 1 \times 10^{-3}, \alpha = 3 \times 10^{-4}$), and ($\gamma = 5 \times 10^{-3}, \alpha = 2 \times 10^{-3}$); VIB with $\beta = 10^{-1}, 10^{-2}, 10^{-3}$; and DP-SGD with ($\sigma = 1 \times 10^{-1}, \Delta = 2$), ($\sigma = 2 \times 10^{-2}, \Delta = 2$), and ($\sigma = 1 \times 10^{-2}, \Delta = 2$).

A.3 Details About Crafting The Canary Record

We generate synthetic canary records by initializing z using a vector sampled from the standard normal distribution and then solve the optimization using the Adam optimizer with a learning rate of 5×10^{-2} for 2,000 iterations.

B ADDITIONAL RESULTS

B.1 Comparison of Adversary Models

We conduct a preliminary experiment of attribute inference attack using five types of adversarial models, including Gaussian Naive Bayes (NB), Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Multi-layer Perceptron (MLP) with one hidden layer of 100 neurons, and Random Forest (RF) with 50 estimators on the Adult dataset with 1,000 shadow data samples and a batch size of 1. Figure 10 plots the ROC curves on both linear and log scales. We observe that the ensemble learning method (Random Forest) performs the best, especially in the low FPR region.

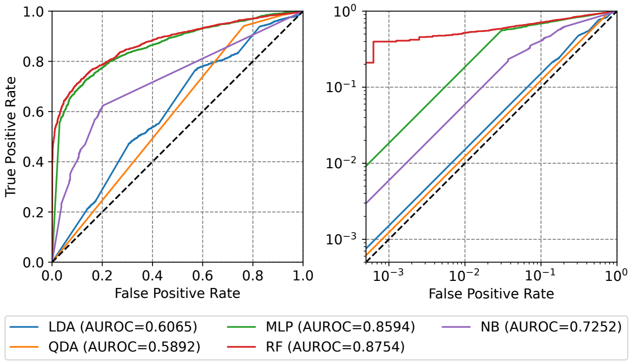


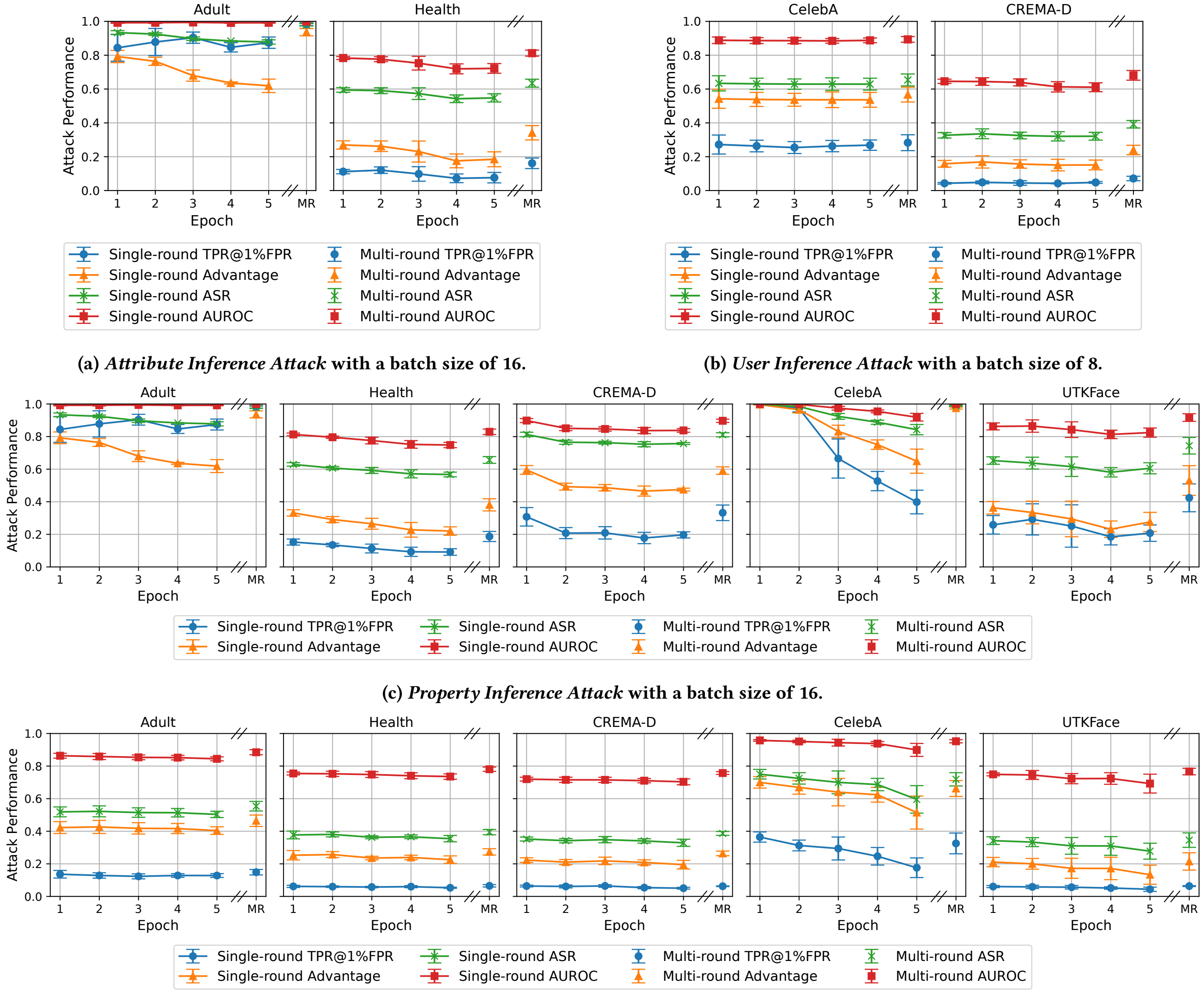
Figure 10: Comparison of AIA with different adversarial models and a batch size of 1 on the Adult dataset.

B.2 Full Results From Main Paper

Figure 11 provides a complete comparison between single-round and multi-round inference attacks on all datasets. Figure 12 evaluates defense against all four types of inference attacks under both the static and adaptive adversary settings.

B.3 Impact of Model Size

We conduct additional experiments to study the impact of model size for all four types of inference attacks in Figure 13.



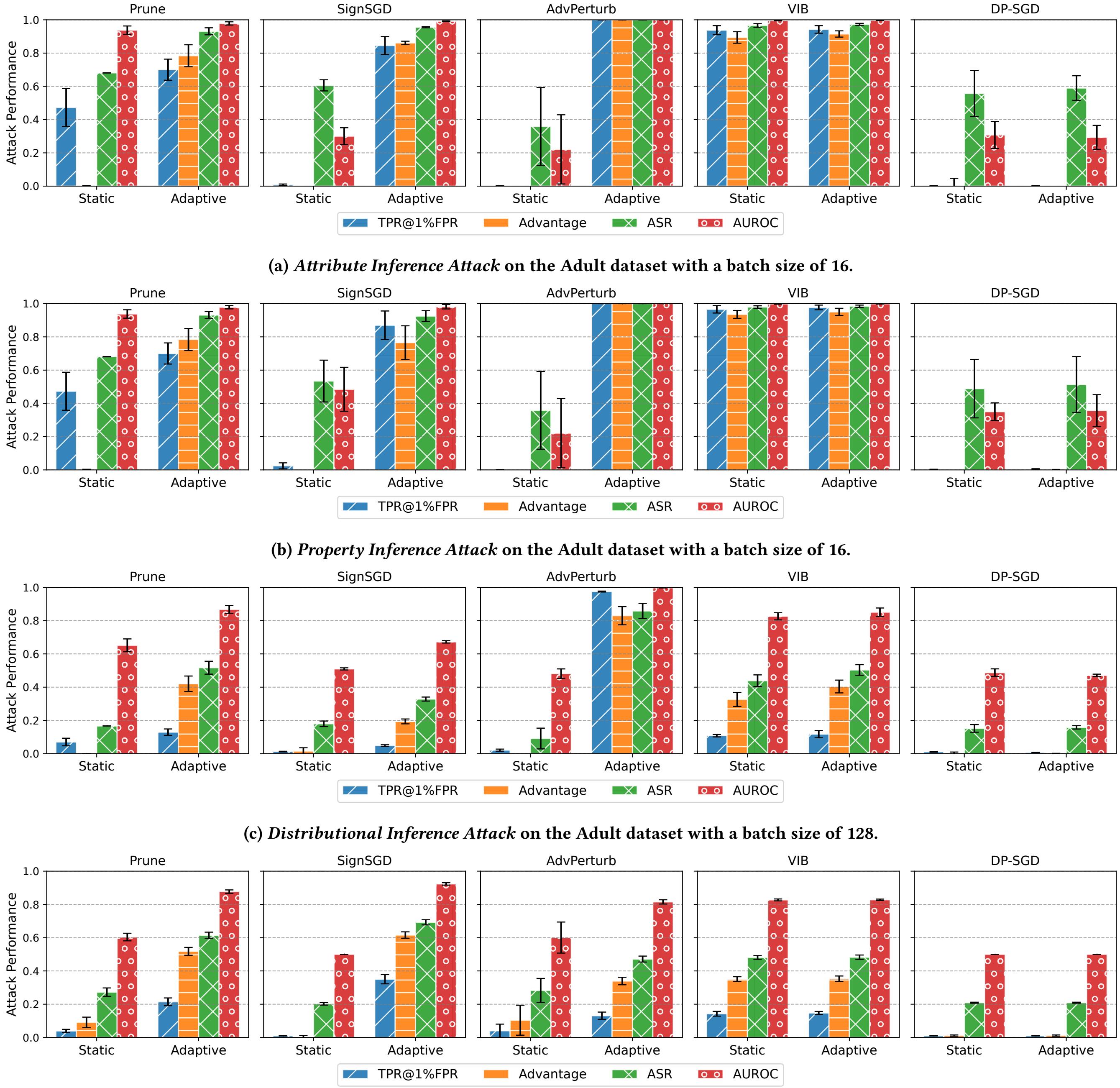
(a) *Attribute Inference Attack* with a batch size of 16.

(b) *User Inference Attack* with a batch size of 8.

(c) *Property Inference Attack* with a batch size of 16.

(d) *Distributional Inference Attack* with a batch size of 128.

Figure 11: Comparison of single-round and multi-round inference attacks.



(a) ***Attribute Inference Attack*** **on the Adult dataset with a batch size of 16.**

(b) ***Property Inference Attack*** **on the Adult dataset with a batch size of 16.**

(c) ***Distributional Inference Attack*** **on the Adult dataset with a batch size of 128.**

(d) ***User Inference Attack*** **on the CelebA dataset with a batch size of 8.**

Figure 12: Comparison of various defenses against static and adaptive adversaries.

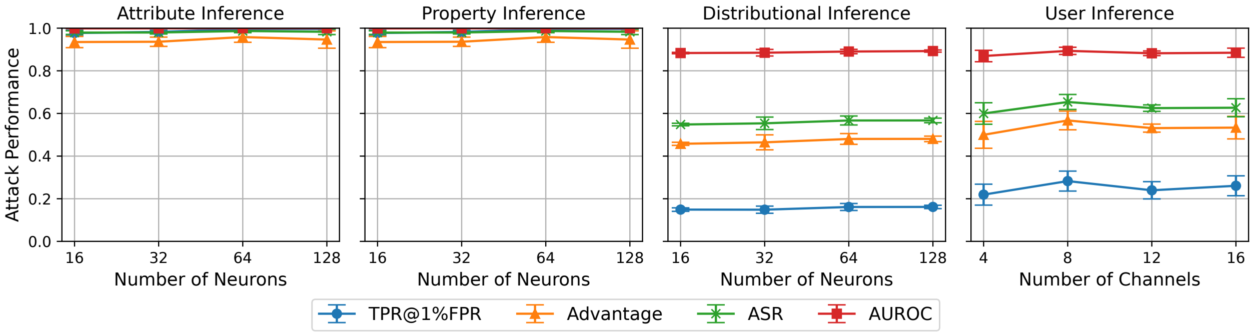


Figure 13: Sensitivity analysis of the impact of varying model sizes on the performance of inference attacks on the Adult (AIA, PIA, DIA) and CelebA (UIA) datasets.