Secure Confidential Business Information When Sharing Machine Learning Models

Yunfan Yang
School of Management, Fudan University, yunfanyang24@m.fudan.edu.cn

Jiarong Xu
School of Management, Fudan University, jiarongxu@fudan.edu.cn

Hongzhe Zhang
School of Management and Economics, The Chinese University of Hong Kong, Shenzhen, zhanghongzhe@cuhk.edu.cn

Xiao Fang
Lerner College of Business and Economics, University of Delaware, xfang@udel.edu

Abstract. Model-sharing offers significant business value by enabling firms with well-established Machine Learning (ML) models to monetize and share their models with others who lack the resources to develop ML models from scratch. However, concerns over data confidentiality remain a significant barrier to model-sharing adoption, as Confidential Property Inference (CPI) attacks can exploit shared ML models to uncover confidential properties of the model provider's private model training data. Existing defenses often assume that CPI attacks are non-adaptive to the specific ML model they are targeting. This assumption overlooks a key characteristic of real-world adversaries: their responsiveness, i.e., adversaries' ability to dynamically adjust their attack models based on the information of the target and its defenses. To overcome this limitation, we propose a novel defense method that explicitly accounts for the responsive nature of real-world adversaries via two methodological innovations: a novel Responsive CPI attack and an attack–defense arms race framework. The former emulates the responsive behaviors of adversaries in the real world, and the latter iteratively enhances both the target and attack models, ultimately producing a secure ML model that is robust against responsive CPI attacks. Furthermore, we propose and integrate a novel approximate strategy into our defense, which addresses a critical computational bottleneck of defense methods and improves defense efficiency. Through extensive empirical evaluations across various realistic model-sharing scenarios, we demonstrate that our method outperforms existing defenses by more effectively defending against CPI attacks, preserving ML model utility, and reducing computational overhead.

Key words: Model-sharing; information privacy; machine learning; information systems-enabling technology

1. Introduction

Driven by rapid technological advancement and increasing complexity of business environments, the application of Machine Learning (ML) in business is no longer a distant aspiration but has become a strategic imperative. Firms across industries are increasingly turning to ML to streamline operations, personalize customer experiences, and unlock data-driven insights (Huang and Rust 2018, Pant and Sheng 2015, Fang et al. 2021, Zhao et al. 2023, Zhang et al. 2024). However, the path to ML implementation remains challenging, particularly for firms lacking in-depth technical skills or substantial computational resources (Ransbotham

et al. 2017). In reality, developing an effective ML model is far from trivial; it requires vast quantities of high-quality training data, specialized expertise, and significant computational infrastructure (Priestley et al. 2023, Thompson et al. 2023). Even constructing a high-quality training dataset, often the very first step of ML model training, can typically incur costs ranging from $10,500 to $85,000.[1] The total cost required to develop an in-house ML model can easily exceed one million US dollars.[2] As a result, not all firms have the capability to develop their own ML models. This challenge is particularly acute for small and medium-sized enterprises (SMEs) with limited funds and resources. Worse still, the inability of these SMEs to train ML models makes it difficult for them to compete with industry giants that are fully empowered by ML, placing SMEs at risk of bankruptcy and intensifying market monopolization (Bammens and Hünermund 2021).

To address this significant issue, a new business mode known as *model-sharing* has emerged as a viable solution. Specifically, model-sharing facilitates collaboration between two key stakeholders: the model provider, who develops high-quality ML models and shares them for profit, and the model user, who pays for the usage rights of these models. A real-world example of this model-sharing business can be found in Databricks[3], a well-known platform for building, deploying, and sharing enterprise-grade ML solutions. Databricks allows model providers to upload and monetize their well-trained models through the Databricks Marketplace, where other firms can pay for access to these models, either by downloading the models directly or querying them via APIs, for their own use. Through the model-sharing business, both stakeholders stand to benefit. On the one hand, model users, typically firms lacking capability to train ML models, can access advanced ML capabilities via partnering with model providers at relatively low cost, without bearing the substantial expenses associated with developing their own ML models from scratch. On the other hand, model providers, which are firms that have already trained effective ML models, can earn revenue by sharing these models, offset model training costs, and accumulate funds for developing next-generation ML models as ML technology rapidly evolves (Zhao et al. 2018, Gan et al. 2023).

Although model-sharing could yield significant benefits, numerous firms remain hesitant to share their models for profit. Particularly, the concern over data confidentiality continues to pose a significant barrier,

[1] See https://hackernoon.com/machine-learning-costs-price-factors-and-real-world-estimates (last accessed on Aug 18, 2025).

[2] See https://itrexgroup.com/blog/machine-learning-costs-price-factors-and-estimates/ (last accessed on Aug 18, 2025).

[3] See https://www.databricks.com (last accessed on Aug 18, 2025).

discouraging firms from monetizing their ML models (Carlini et al. 2019, Gan et al. 2023). Such concerns are not unfounded. Recent studies on *Confidential Property Inference* (CPI) attacks have shown that even when model providers share only their ML models but not the training data, certain confidential properties of the training data are still at risk of disclosure (Ganju et al. 2018, Zhang et al. 2021, Mahloujifar et al. 2022). These confidential properties are often descriptive statistics of attribute distributions across the entire dataset, such as the loan default rate for a bank, which is the proportion of users who default on their loans. These properties are highly sensitive as they often reveal critical insights into a company's operational strategies and competitive advantages. If these properties are leaked, the model provider could lose its competitive edge and face significant operational disruptions. Using a bank's loan default rate as an example, disclosure of a high loan default rate could undermine public trust in the bank, potentially leading to a bank run and ultimately causing the bank to fail (Diamond and Dybvig 1983).

Therefore, protecting model providers from CPI attacks has become a critical prerequisite for the broader adoption of model-sharing. Recent research efforts have been devoted to designing defense methods to address this issue. Existing defenses can be categorized as noise-based methods and model-based methods. In general, noise-based methods aim to obscure confidential properties by injecting noise at various stages of the model training process, thereby preventing adversaries from extracting sensitive attributes from the trained model through CPI attacks (e.g., Mahloujifar et al. 2022). Exemplar studies such as Ganju et al. (2018) and Suri et al. (2023) propose adding noise to the training data, while other works explore injecting noise into model gradients (Mahloujifar et al. 2022), intermediate embeddings, or final outputs (Wang and Wang 2022). However, the introduction of noise inevitably degrades the utility of the resulting ML models (Jayaraman and Evans 2019). Such reduced model utility directly undermines the practical and commercial value of ML models, thereby rendering noise-based methods inappropriate for real-world model-sharing business. To overcome this issue, recent research has shifted toward model-based methods. Typically, these defense methods first simulate a CPI adversary and allow it to attack the ML model intended for sharing (Stock et al. 2022, 2024). Next, the ML model parameters are adjusted to minimize the attack's effectiveness while preserving as much ML model utility as possible. Concretely, these defense methods extend the original

training objective of the ML model, which typically focuses solely on maximizing ML model performance, to additionally account for minimizing the effectiveness of the simulated CPI attack (Stock et al. 2024, Noorbakhsh et al. 2024). In doing so, the resulting ML model becomes robust to CPI attacks, capable of effectively protecting providers from CPI attacks while maintaining overall utility.

However, existing model-based defense methods still face two fundamental limitations in protecting model providers against CPI attacks in real-world scenarios. *First*, these methods commonly assume that adversaries are non-adaptive to the specific model they are targeting. This assumption overlooks a key characteristic of real-world adversaries: their responsiveness, that is, adversaries' ability to dynamically adjust their attack models based on the specific target and its defenses (Song and Mittal 2021). Such responsiveness of adversaries would inevitably invalidate defense methods that assume adversaries are non-adaptive. Therefore, an effective defense method must account for the responsive nature of real-world adversaries. Yet, none of the existing model-based defenses are capable of doing so. *Second*, current model-based defense methods generally involve high computational complexity in producing robust models, primarily because the first step of the defense process, i.e., simulating CPI adversaries, incurs substantial computational cost (Suri et al. 2023). This high computational complexity often makes these defense methods time-consuming to apply.

In response to the above challenges, we propose a novel model-based defense method. This method explicitly accounts for the responsive nature of real-world adversaries and significantly improves the computational efficiency in defense, thereby addressing two critical limitations of existing model-based defenses: the assumption of non-adaptive adversaries and the high computational overhead. To achieve this, we first devise a Responsive CPI attack model, which allows the adversary to adapt its attack model based on the target ML model. This Responsive CPI attack model thus emulates the responsive behaviors of adversaries that shared ML models may encounter in real-world scenarios. Next, to defend against this responsive attack, we introduce an attack–defense arms race framework, where both the attack model and the defense are iteratively improved. This iterative process ultimately yields an ML model that is robust to responsive CPI attacks, thereby effectively protecting the confidentiality of the model provider's training data during model-sharing. Moreover, we propose an approximation technique that significantly reduces

the computational costs associated with attack simulation, ensuring the practicality of our defense method. Taken together, our study makes three methodological contributions: a Responsive CPI attack model for emulating real-world responsive adversaries, an arms race framework for developing effective defenses against responsive adversaries, and an approximation strategy for efficiently simulating CPI adversaries in the defense process. Extensive empirical evaluations demonstrate that our proposed defense method significantly outperforms existing benchmarks in defending against CPI attacks, but also maintains the utility of the ML model while reducing computational time.

2. Literature Review

In this section, we first review three streams of research related to our study: privacy attacks on ML models, CPI attacks threatening model-sharing business, and existing defense methods against CPI attacks. We then identify the research gaps and highlight the key novelty of our study.

2.1. Privacy Attacks on ML Models

Our study falls into the domain of Information Privacy, a research area that has received substantial attention from Information Systems (IS) scholars (Smith et al. 2011, Li and Sarkar 2011, Li and Qin 2017, Xu and Dinev 2022). Within this domain, one line of IS research empirically investigates individuals' or organizations' information disclosure behaviors, aiming to explain how they perceive, evaluate, and respond to risks related to confidential information disclosure and associated government regulations (Xu et al. 2012, Buckman et al. 2019, Xu and Zhang 2022). For instance, Xu et al. (2012) examine how different privacy assurance methods affect context-specific concerns for information privacy and find that perceived control over personal information is a key determinant. Another line of IS research focuses on developing technical methods to protect information privacy when sharing data for new business opportunities (Li and Sarkar 2011, Li and Qin 2017, Han et al. 2025). For example, Li and Qin (2017) propose a clustering-based anonymization approach to enable privacy-preserving sharing of medical text records in health information systems. Our study contributes to the latter line of work by addressing an information privacy challenge that arises when ML model owners share their models.

Privacy concerns surrounding ML models have recently received widespread attention. Indeed, privacy attacks on ML models have become increasingly prevalent and can be broadly classified into four types:

model extraction attacks, membership inference attacks, reconstruction attacks, and confidential property inference (CPI) attacks (Rigaki and Garcia 2023). Specifically, model extraction attacks seek to duplicate the ML model by recovering its parameters or architecture through query access to the target model (Tramèr et al. 2016, Oh et al. 2018). For example, Tramèr et al. (2016) demonstrate that by carefully designing queries to a linear or logistic regression model and then analyzing its returned confidence scores, an adversary can accurately recover parameters of the model. In contrast, membership inference attacks aim to determine whether a specific individual's data is included in the training set (Shokri et al. 2017, Salem et al. 2018). Such attacks are particularly critical in the healthcare domain, as inferring whether an individual's data appears in a private medical dataset could expose the individual's medical conditions, constituting a significant privacy violation (Hu et al. 2022). Shokri et al. (2017) exemplify this attack by training a classifier on the posterior probabilities produced by the target ML model to differentiate members from non-members of the training dataset. Unlike membership inference attacks focusing solely on confirming an individual's presence in the training set, reconstruction attacks take the threat a step further by recovering sensitive attributes of that individual (Fredrikson et al. 2015, Jia and Gong 2018). For instance, Fredrikson et al. (2014) show that given a patient's non-sensitive attributes, such as height, age, and weight, it is possible to reconstruct the patient's sensitive genetic traits from the outputs of an ML model trained for pharmacogenetic prediction. Lastly, CPI attacks target the extraction of confidential properties from the training datasets of ML models (Ateniese et al. 2015, Melis et al. 2019). These properties are typically descriptive statistics of attribute distributions across the entire dataset. For firms that train ML models using proprietary data, such properties are highly sensitive, as they potentially reveal trade secrets and strategic information (Ganju et al. 2018). Consequently, CPI attacks constitute a significant obstacle for ML model-sharing businesses, which we review next.

2.2. Confidential Property Inference (CPI) Attacks

Existing research has demonstrated that adversaries can infer confidential properties of training data even when their access is limited solely to trained ML models. Such attacks are referred to as Confidential Property Inference (CPI) attacks. A seminal study by Ateniese et al. (2015) introduces the first CPI attack, operating under the assumption that adversaries have full knowledge of the target ML model's architecture

and parameters, a scenario later defined as the "white-box setting." To execute this attack, Ateniese et al. (2015) propose training numerous surrogate models (commonly known as shadow models) using datasets with varying confidential properties. The parameters of these shadow models, along with their associated confidential properties, form the training data for the white-box CPI attack model. By learning the correlation between model parameters and confidential properties, this attack model can then use the parameters of a targeted ML model to infer confidential properties present in the target model's training data. Although this shadow-model-based CPI attack framework has been widely adopted in subsequent studies, the significant computational overhead involved in training numerous shadow models has been consistently highlighted (Suri et al. 2023, Bertran et al. 2023, Chaudhari et al. 2023).

Built on Ateniese et al. (2015), Ganju et al. (2018) further enhance CPI attacks specifically focusing on deep learning models, characterized by extensive parameters (e.g., neuron weights). Concretely, they propose a systematic organization of parameters whereby neurons are first sorted according to the sum of their associated weights, followed by sorting the weights within each neuron by their magnitude. Such reorganization helps the CPI attack model better learn the correlation between model parameters and confidential properties, significantly improving the attack's success rate on deep learning models. However, rising concerns over intellectual property protection have made firms more cautious, limiting adversaries' ability to obtain the full model architectures and parameters needed for white-box attacks. In response, Zhang et al. (2021) explore CPI attacks under the more practical "black-box setting," where adversaries can only query the target ML model and observe its outputs. Unlike the white-box scenario, Zhang et al. (2021) do not utilize model parameters directly; instead, they query shadow models with a predefined query set and combine the resultant outputs with their corresponding confidential properties to create training data for the black-box CPI attack model. This enables the CPI attack model to learn correlations between ML model outputs and confidential properties of the ML models' training data, thereby facilitating CPI attacks in the black-box setting. Extending this line of research, Mahloujifar et al. (2022) develop a black-box CPI attack for collaborative learning environments. In this scenario, adversaries act as malicious collaborators and inject poisoning data during joint ML model training. Besides, Wang and Wang (2022) focus on inferring

confidential properties in social network data, such as node group properties (e.g., whether the network contains more female nodes than male nodes) and link group properties (e.g., whether there are more links between male users compared to female users). All these studies follow a general paradigm of CPI attacks. We provide details of this paradigm in Section 3.2.

2.3. Defense Methods Against CPI Attacks

Given the significant risk posed by CPI attacks, recent research has increasingly emphasized developing effective defenses. Extant defense methods can be broadly classified into two categories: noise-based and model-based methods.

Noise-based defense methods obscure confidential properties by introducing noise at various stages during the ML model training. A straightforward approach is to perturb the training dataset itself. For instance, Ganju et al. (2018) flip the confidential labels of a subset of training samples, whereas Zhou et al. (2022) and Suri et al. (2023) modify the dataset by removing or adding samples. Another line of work injects noise into the gradients during training. Exemplar studies by Mahloujifar et al. (2022) and Chaudhari et al. (2023) employ differential-privacy mechanisms to perturb gradients during training, thereby reducing an adversary's ability to infer confidential properties from parameter updates. Additionally, Wang and Wang (2022) propose injecting differentially private noise directly into model parameters in the white-box setting or into model outputs in the black-box setting. However, noise-based defense methods inevitably degrade the performance of the ML model due to the noise injected during training. Particularly, such a reduction in model utility directly undermines the practical and commercial value of ML models, making noise-based defenses unsuitable for model-sharing business.

Unlike noise-based defenses that indiscriminately introduce noise into the training process, model-based defense methods strategically modify ML model parameters to diminish the effectiveness of a pretrained attack model. Concretely, these defense methods typically begin by simulating a CPI adversary and training an attack model that aims to infer confidential properties either from the target model's parameters (in the white-box setting) or from its query outputs (in the black-box setting). The target model's parameters are then deliberately retrained to minimize the objective function of such attack model (e.g., the inference accuracy).

In this vein, Stock et al. (2022) propose a property unlearning algorithm, which calculates gradients with respect to ML model parameters, nudging to decrease the accuracy of a well-trained CPI attack proposed by Ganju et al. (2018). The algorithm then carefully selects a suitable learning rate to apply these gradients, ensuring a reduction in the adversary's ability to infer confidential properties. However, while this approach can effectively diminish the success of the simulated CPI adversary, it still does not guarantee preservation of the ML model's utility. To this end, Stock et al. (2024) further propose retraining the ML model with a combined objective that minimizes the inference accuracy of the simulated CPI attack while preserving as much ML model's utility as possible. By optimizing this joint objective, the retrained ML model can achieve robustness against that CPI attack while maintaining high predictive performance.

Our literature review suggests the following research gaps. *First*, existing model-based defense methods typically assume a non-adaptive attack model, meaning that the attack model is trained independently of the target model and remains unchanged throughout the defense optimization process. In reality, however, adversaries frequently adjust their attack models dynamically in response to the specific information of the targeted ML model and its defense. As a result, current model-based defenses overlook the responsive nature of real-world adversaries, inevitably rendering such defenses ineffective in practice. To address this gap, we propose a novel model-based defense method which explicitly accounts for the responsive nature of real-world adversaries. In doing so, we first develop a Responsive CPI attack, which effectively emulates the responsive behavior of real-world adversaries—a capability not achieved by any existing CPI attack methods. Next, we introduce an attack-defense arms race framework that iteratively modifies the ML model, thereby enhancing its robustness against responsive CPI adversaries. *Second*, recall that simulating an attack model is a necessary first step in model-based defense methods. However, this step is computationally intensive due to the requirement of training numerous shadow models, making existing model-based defenses highly time-consuming to implement. Such computational overhead significantly limits the practical applicability of current defenses in real-world ML model-sharing scenarios. To this end, we propose an approximation technique that substantially reduces the computational costs associated with attack model simulation, ensuring the practicality of our defense method. In summary, the proposed Responsive CPI attack, the arms race-based defense framework, and the approximation strategy for computational efficiency together constitute the key methodological contributions of this study.

3. Problem Formulation and Preliminaries

We formally define the secure model-sharing problem in Section 3.1, and present preliminaries on the general paradigm of existing CPI attacks in Section 3.2, which serves as the technical foundation of our study.

3.1. Problem Formulation

Consider a model provider intending to share a well-trained ML model $f_\theta : \mathcal{X} \to \mathcal{Y}$, parameterized by θ and trained on the provider's proprietary dataset $D = (X, Y)$. Here, X is a matrix in which each row represents the attribute values of an instance, with each instance belonging to the input space $\mathcal{X}$. Correspondingly, Y is a vector of labels for these instances, associated with the target prediction task, where each label in Y belongs to the output space $\mathcal{Y}$. For instance, a bank may wish to share its financial product telemarketing model f_θ. In this case, X may include demographic characteristics and credit histories of its clients, and Y indicates whether a client purchases financial products or not. Notably, even though the model provider shares only the trained ML model (or query access to the model) without disclosing the proprietary dataset D, confidential properties of the dataset (as defined in Definition 1) can still be disclosed through Confidential Property Inference (CPI) attacks (Definition 2).

DEFINITION 1. (**Confidential Property**) A confidential property of the training dataset D is defined as the result of applying an aggregate function to the vector X_p of a specific confidential property-related attribute p, which corresponds to one column of attribute matrix X. Formally, let P denote a confidential property and AGG$(\cdot)$ denote an aggregate function (e.g. mean). Then, $P :=$ AGG(X_p).[4]

For example, consider a bank's dataset containing client attributes such as demographic characteristics (e.g., age and job), along with financial information indicating whether clients hold loans and their corresponding default statuses. The attribute matrix can be thus represented as $X = [X_{age}, X_{job}, X_{loan}, X_{default}]$. Since the default rate reflects the bank's effectiveness in risk management and overall asset quality, competitors gaining access to this information could strategically undermine the bank's market position and competitive advantage (Agoraki et al. 2011). Hence, the default rate of clients is one confidential property of this dataset, i.e., $P =$ MEAN$(X_{default})$, with the corresponding confidential property-related attribute being *default*.

[4] In this study, we focus exclusively on categorical attributes, which is a common setting in existing works (Ganju et al. 2018, Zhang et al. 2021).

DEFINITION 2. (**Confidential Property Inference (CPI) Attack**) Consider an ML model $f_\theta : \mathcal{X} \to \mathcal{Y}$ as the target model, parameterized by θ and trained on a dataset $D = (X, Y)$. A Confidential Property Inference (CPI) attack aims to infer a confidential property P of training data D, using information accessible from the target model. This information may consist of the target ML model's parameters in the white-box setting or the model's outputs on a predefined query set in the black-box setting. Precisely, let $\mathcal{F}_\theta$ denote the accessible information from the target model f_θ. A CPI attack trains an attack model h_ϕ parameterized by ϕ, with the objective of estimating the confidential property P from $\mathcal{F}_\theta$, i.e., $\hat{P} = h_\phi(\mathcal{F}_\theta)$.

As a result, given the threat posed by CPI attacks, when a model provider intends to share its ML model for profit, it becomes imperative to adjust the model such that it is CPI-attack robust, that is, the confidential properties of the models' training data cannot be inferred through CPI attacks. Moreover, it is noted that a shared model holds commercial value only if it maintains high utility (i.e., target predictive performance). Therefore, the adjusted ML model, while being robust to CPI attacks, should also preserve as much utility as possible. Now, we are ready to formally define our secure model-sharing problem.

DEFINITION 3. (**Secure Model-Sharing (SMS) Problem**) A model provider intends to share its ML model f_θ for profit, where f_θ is trained on the model provider's proprietary dataset D that contains a confidential property P. Given the threat posed by CPI attacks, the model provider needs to adjust the model's parameters θ prior to sharing, with the following two objectives:

- **CPI-attack robustness.** The confidential property P of the training data cannot be inferred by adversaries through CPI attacks when they have access to the model information $\mathcal{F}_\theta$ after the model is shared.
- **Utility retention.** The shared ML model should retain as much utility as possible for the target prediction task.

3.2. Preliminary: General Paradigm of CPI Attacks

In this section, we introduce the general paradigm of existing CPI attacks. The realization of CPI attack is grounded in the fact that ML models trained on analogous datasets tend to exhibit similar characteristics, such as similar parameters or outputs (Ateniese et al. 2015, Ganju et al. 2018). Such fact implies the existence

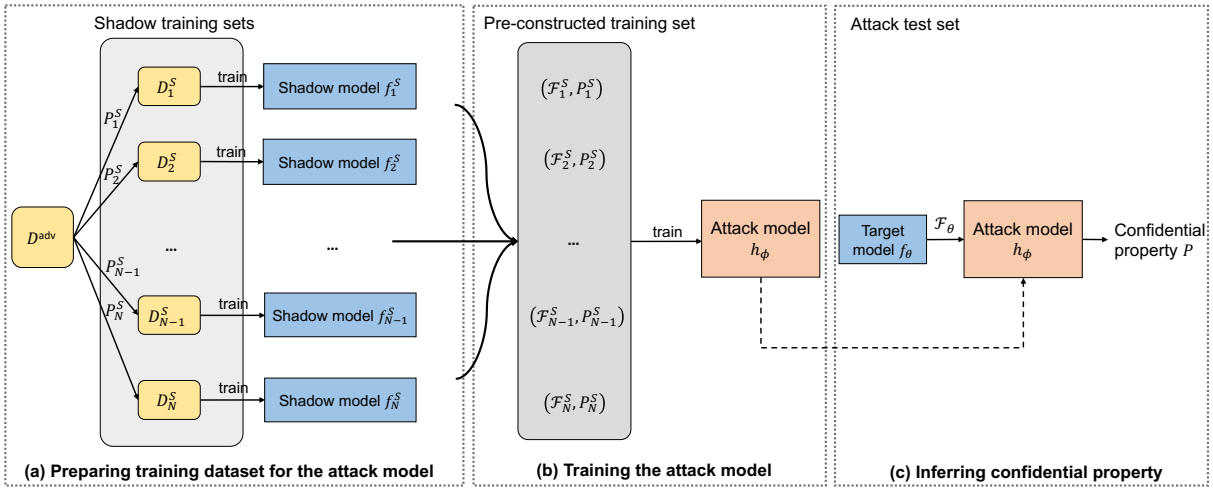


Figure 1 **A common CPI attack paradigm consists of three phases: (a) preparing the training dataset for the attack model, (b) training the attack model, and (c) inferring confidential properties from the target model's training dataset.**

of a learnable mapping from a model's information $\mathcal{F}_\theta$ (e.g., model parameters) to the confidential properties of its training dataset. CPI attacks aim to learn this mapping by training an attack model h_ϕ.

To facilitate this, an appropriate training dataset must be constructed, consisting of training samples that include model parameters or outputs along with their corresponding confidential properties. Once the attack model is well-trained, it can be utilized to infer the confidential property in the target model's training dataset. Therefore, the commonly used paradigm for conducting such a CPI attack consists of three phases: (1) preparing the training dataset for the attack model h_ϕ, (2) training the attack model h_ϕ, and (3) inferring the confidential property using the trained h_ϕ. The pipeline of a CPI attack is illustrated in Figure 1.

PHASE 1: Preparing the Training Dataset for the Attack Model h_ϕ. To effectively train the attack model, the adversary must first prepare the attack model's training dataset, where the i-th training sample $(\mathcal{F}_i^S, P_i^S)$ includes model information $\mathcal{F}_i^S$ (i.e., the model parameters or outputs on a predefined query set) of a trained ML model, along with the corresponding confidential property P_i^S of that ML model's training data. Existing studies have shown that an adversary can obtain an auxiliary dataset $D^{\text{adv}} = (X^{\text{adv}}, Y^{\text{adv}})$ that shares the same attribute space and the target label space as the target ML model's training data (Shokri et al. 2017, Nasr et al.

2018, Song and Mittal 2021).[5] This auxiliary dataset D^{adv} allows the adversary to systematically construct the training data for the attack model.

Specifically, to construct a single training sample $(\mathcal{F}_i^S, P_i^S)$, the adversary first randomly samples a subset of data $D_i^S \subset D^{\text{adv}}$. This sampled dataset D_i^S is then used to train an ML model f_i^S, commonly referred to as a *shadow model* (Shokri et al. 2017, Ganju et al. 2018, Zhang et al. 2021).[6] Subsequently, the adversary derives the shadow model's information $\mathcal{F}_i^S$ and computes the confidential property P_i^S of the shadow model's training dataset D_i^S.[7] The resulting pair $(\mathcal{F}_i^S, P_i^S)$ constitutes one training sample for the attack model h_ϕ. By repeating this procedure for N shadow models, the adversary ultimately compiles the training dataset for the attack model, represented as $\{(\mathcal{F}_i^S, P_i^S)\}_{i=1}^N$.

PHASE 2: Training the Attack Model h_ϕ. Given the training dataset $\{(\mathcal{F}_i^S, P_i^S)\}_{i=1}^N$, the adversary aims to train the attack model h_ϕ with learnable parameters ϕ by optimizing

$$\min_{\phi} \sum_{i=1}^{N} l(h_\phi(\mathcal{F}_i^S), P_i^S),$$

where $l(\cdot,\cdot)$ is a loss function that quantifies the error of the attack model h_ϕ in predicting the confidential property P_i^S from the corresponding model information $\mathcal{F}_i^S$.

PHASE 3: Inferring Confidential Property. With the well-trained attack model h_ϕ, the adversary can now infer the confidential property of the target ML model's training data, treating the target model's information as a test sample for the attack model. Precisely, given the target ML model's information $\mathcal{F}_\theta$, the predicted confidential property $\hat{P}$ is computed as

$$\hat{P} = h_\phi(\mathcal{F}_\theta).$$

[5] This is because a model user would only purchase and use the provider's ML model if they have input data with attributes and target label compatible with the model. Therefore, the attribute space of the provider's training data is known to the model user and can be easily obtained by the adversary. The actual attribute values for D^{adv} can be further gathered from publicly available sources or the adversary's private data (Ganju et al. 2018, Zhang et al. 2021, Zhou et al. 2022, Mahloujifar et al. 2022).

[6] Existing literature on CPI attacks commonly assumes that the adversary has knowledge of the target model's architecture (Ateniese et al. 2015, Ganju et al. 2018, Wang and Wang 2022). Accordingly, the shadow model f_i^S is constructed using the same architecture as the target model. In white-box settings, this assumption is straightforward, as the adversary has direct access to the model parameters and architecture. Even in black-box settings, the model architecture can often be inferred through model architecture extraction techniques prior to conducting CPI attacks (Oh et al. 2018, Wang and Gong 2018).

[7] In the white-box setting, the accessible model information is model parameters. When the parameters are not naturally in vector form (e.g., in deep neural networks where weights span multiple layers), transformations proposed by Ganju et al. (2018) are applied to convert them into a vector format.

4. Solution Method: D-S^{2}HARE

In this section, we propose a novel defense method to address the SMS problem. Recall that the model provider must account for two key objectives when adjusting the ML model parameters: (1) ensuring CPI-attack robustness, and (2) retaining the ML model's utility for its target prediction task. Notably, our literature review has shown that model-based defense methods provide a viable approach to simultaneously achieving both objectives (Stock et al. 2022). Our proposed method thus follows the model-based defense paradigm.

More precisely, model-based defense methods begin by simulating a CPI adversary and training an attack model h_ϕ following the paradigm introduced in Section 3.2, which is capable of inferring the confidential property P from the ML model's information $\mathcal{F}_\theta$. Once such an attack model is trained, the defense proceeds by adjusting the target ML model's parameters θ to maximize a composite objective function that combines the error of the attack model in predicting the confidential property, denoted by $\mathcal{L}_\mathcal{P}$, and the utility of the ML model, denoted by $-\mathcal{L}_\mathcal{T}$. The resulting model parameters are thus obtained by solving the following optimization problem:

$$\arg\max_\theta \left[\mathcal{L}_\mathcal{P}(P, h_\phi(\mathcal{F}_\theta)) - \lambda \mathcal{L}_\mathcal{T}(\theta; D)\right], \tag{1}$$

where hyperparameter λ controls the relative importance of these two loss terms.

However, because these existing model-based defenses simulate the attack model in advance, they implicitly assume that the attack is static and trained independently of the target model. This neglects a critical characteristic of real-world adversaries: their responsiveness. In practice, adversaries often adapt their attack model based on the information of the target ML model $\mathcal{F}_\theta$, in order to improve attack effectiveness. Overlooking such responsiveness can thus result in ineffective defenses. To address this limitation, we first develop a Responsive CPI attack that effectively reflects the responsive behavior of real-world adversaries. Building on this, we propose an attack-defense arms race framework that iteratively refines the ML model to be robust against responsive adversaries. Figure 2 illustrates this key difference between our proposed defense method and existing model-based defenses. Additionally, considering the significant computational overhead in simulating attack models, which is primarily due to the need to train numerous shadow models, we propose an approximation technique to reduce the computational cost associated with simulating the

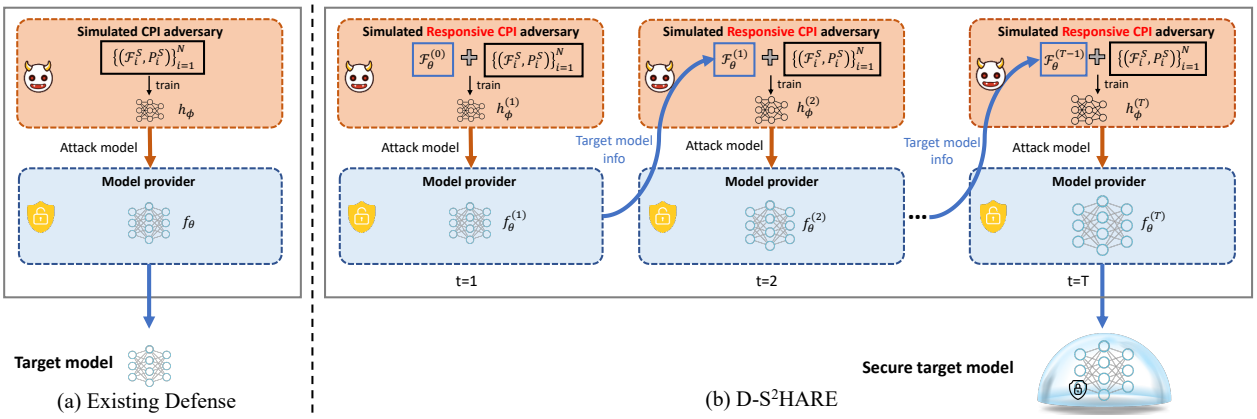


Figure 2 Comparison of Existing Defenses with Our Proposed D-S^{2}HARE

attack model, thus enhancing the practicality and efficiency of our defense method in realistic model-sharing scenarios.

These three innovations, the Responsive CPI attack, the arms race-based defense framework, and the approximation technique, constitute our novel model-based defense method for addressing the SMS problem. Together, they enable our proposed defense method to effectively counter responsive adversaries while ensuring high defense efficiency. We detail them in Sections 4.1, 4.2, and 4.3, respectively. Accordingly, we designate our defense method as D-S^{2}HARE, which stands for a Defense method for the Secure model-SHaring problem that considers Adversarial Responsiveness and defense Efficiency. The complete defense procedure for D-S^{2}HARE is summarized in Section 4.4.

4.1. Responsive CPI Attack

Recall that real-world adversaries are highly responsive, constantly adapting their attack models to changes in the target models. Therefore, our proposed CPI attack must be capable of capturing this responsiveness. Moreover, since adversaries seek to effectively extract confidential properties, our simulated attack also requires prioritizing inference effectiveness. These two fundamental requirements: responsiveness and effective inference, establish the objectives of our attack design, ensuring that it precisely reflects real-world adversaries. In what follows, we begin by discussing why current CPI attacks fail to satisfy these two requirements, then introduce our proposed Responsive CPI attack, which is grounded in a rigorous theoretical foundation.

Limitations of existing CPI attacks. As discussed in Section 2.2, existing CPI attacks train attack models by minimizing the empirical error on training data, with the expectation that these attacks will effectively infer the confidential properties of unseen target ML models. More precisely, an attack model is trained on a pre-constructed dataset $\{(\mathcal{F}_i^S, P_i^S)\}_{i=1}^N$, consisting of N training samples. For a given training instance $(\mathcal{F}_i^S, P_i^S)$, $\mathcal{F}_i^S$ represents the i-th shadow model f_i^S's model information (i.e., parameters or outputs), which follows the distribution $\mathcal{D}_S$. P_i^S represents the confidential property associated with the training dataset of shadow model f_i^S. The training objective of the attack model h_ϕ is to minimize the empirical error $\widehat{R}(h_\phi)$, which is given by:

$$\widehat{R}(h_\phi) = \frac{1}{N}\sum_{i=1}^{N} l(h_\phi(\mathcal{F}_i^S), P_i^S),$$

where $l(h_\phi(\mathcal{F}_i^S), P_i^S)$ denotes the loss measuring the discrepancy between the predicted confidential property $h_\phi(\mathcal{F}_i^S)$ and the true confidential property P_i^S in the i-th training sample.

Once trained, the attack model is expected to effectively infer the confidential property of a target ML model, given the target model's information $\mathcal{F}_\theta$. The target model thus serves as a test instance for the well-trained attack model h_ϕ. Accordingly, the attack model's inference effectiveness corresponds to its test error $R(h_\phi)$, computed as:

$$R(h_\phi) = E_{\mathcal{D}_T}[l(h_\phi(\mathcal{F}_\theta), P)]. \tag{2}$$

The test error $R(h_\phi)$ quantifies the expected performance of h_ϕ in inferring a target model's confidential property, with the expectation taken over the distribution $\mathcal{D}_T$ of the target model's information $\mathcal{F}_\theta$.

The above formulation illustrates that existing CPI attacks fundamentally cannot meet either of the two critical requirements: responsiveness and effective inference. First, with respect to responsiveness, the attack model in existing CPI attacks is trained to minimize the empirical error $\widehat{R}(h_\phi)$ on a pre-constructed dataset, without incorporating any information from the target ML model. This renders the attack model's training process entirely independent of the target, resulting in an attack model that remains static and unable to adapt to the specific information of the target ML model. Second, existing CPI attacks fail to achieve effective inference due to their neglect of distribution shift[8] between the pre-constructed training dataset and the

[8] Here, distribution shift refers to a discrepancy between the distribution of the ML model's training data and that of its test data.

target ML model (i.e., the test instance). Concretely, when the distribution of model information in the pre-constructed training datasets (i.e., $\mathcal{D}_S$) differs significantly from that of the target ML model (i.e., $\mathcal{D}_T$), the attack effectiveness inevitably deteriorates, as the attack model struggles to generalize across different data distributions. For instance, consider an adversary from one financial institution (Bank A) who trains an attack model using customer data reflecting Bank A's specific demographic and behavioral patterns. When this adversary attempts to apply the trained model to infer the confidential property of a target ML model shared by a different financial institution (Bank B), discrepancies between the customer populations would cause substantial distribution shifts. Consequently, the attack model, trained on Bank A's data, fails to generalize effectively to Bank B's target model, resulting in a significant increase in test error $R(h_\phi)$ when inferring Bank B's confidential property.

To this end, we propose a novel Responsive CPI attack that explicitly incorporates information of the target ML model into the attack model's training process, thereby capturing the responsive nature of real-world adversaries. Meanwhile, it mitigates the distribution shift between the pre-constructed training dataset and the target model, which enhances the inference effectiveness of the attack.

Theoretical foundation of Responsive CPI attack. Before we delve into the detailed method design of Responsive CPI attack, we build on the work of Shimodaira (2000) to establish the following theorem, which provides the theoretical guideline for designing our attack model's training objective.

THEOREM 1. *Let h_ϕ denote an attack model that maps an ML model's information $\mathcal{F}_i^S$ to its training data's confidential property P_i^S. Define $\mathcal{D}_S$ as the distribution of model information from the attack model's pre-constructed training data, and $\mathcal{D}_T$ as the corresponding distribution for the test data (i.e., the target model's information in our problem). Suppose that the support of the distribution $\mathcal{D}_T$ is contained in the support of distribution $\mathcal{D}_S$. Then, the test error of the attack model can be expressed as:*

$$R(h_\phi) = E_{\mathcal{D}_S}[\frac{1}{N}\sum_{i=1}^{N} r_i l(h_\phi(\mathcal{F}_i^S), P_i^S)], \tag{3}$$

where $r_i = \frac{\Pr_{\mathcal{D}_T}(\mathcal{F}_i^S)}{\Pr_{\mathcal{D}_S}(\mathcal{F}_i^S)}$ is the training sample weight, $\Pr_{\mathcal{D}_T}(\mathcal{F}_i^S)$ and $\Pr_{\mathcal{D}_S}(\mathcal{F}_i^S)$ are the probability densities of $\mathcal{F}_i^S$ under the distributions $\mathcal{D}_S$ and $\mathcal{D}_T$.

Proof: See Appendix A.1.

The theorem demonstrates that the test error of the attack model, $R(h_\phi)$, can be represented as a weighted sum of the losses over the training samples. In particular, the training sample weight r_i adjusts the contribution of each training sample based on the probability density of $\mathcal{F}_i^S$ under the training sample distribution $\mathcal{D}_S$ relative to that under the test distribution $\mathcal{D}_T$. This theoretical insight suggests that to minimize the test error under distribution shift, the training objective for the attack model should be formulated as minimizing the following weighted empirical loss $\mathcal{L}_\mathcal{A}$:

$$\mathcal{L}_\mathcal{A} := \frac{1}{N}\sum_{i=1}^{N} r_i \times l(h_\phi(\mathcal{F}_i^S), P_i^S), \tag{4}$$

where $\mathcal{L}_\mathcal{A}$ corresponds to the integrand of the expected test error in Eq. (3).

By adopting this training objective, Responsive CPI attack achieves dual capabilities: responsiveness to the target model and effective inference of confidential properties.

- *Responsiveness*: The weights of the training samples in training objective (4) can be adjusted based on the distribution of the target model's information. Specifically, as the target model changes, the training sample weights are reassigned to better match the new target model. This reweighting ensures that the attack model remains responsive to the specific information of the target model.
- *Effective inference*: The weighted training loss $\mathcal{L}_\mathcal{A}$ (i.e., Eq. (4)) is precisely the integrand of the expected test error defined in Eq. (2). Therefore, as guaranteed by Theorem 1, minimizing $\mathcal{L}_\mathcal{A}$ directly contributes to reducing the test error in the presence of distribution shift, thereby improving the effectiveness of inferring confidential properties.

Training sample weight estimation. Although the key to ensuring the responsiveness of the proposed attack model is adjusting the weights r_i based on the distribution of the target model's information, calculating r_i remains a significant challenge for the adversary, since the distribution of the target model's information is typically unknown. Inspired by Sugiyama et al. (2007), we derive the following theorem to estimate r_i for each training sample $\mathcal{F}_i^S$ in the context of the SMS problem.

THEOREM 2. *Let* $K_\sigma(a,b) = exp\{-\frac{\|a-b\|^2}{2\sigma^2}\}$ *denote the Gaussian kernel with kernel width* σ. *For the SMS problem, the weight* $r_i = \frac{Pr_{\mathcal{D}_T}(\mathcal{F}_i^S)}{Pr_{\mathcal{D}_S}(\mathcal{F}_i^S)}$ *for the training sample* $\mathcal{F}_i^S$ *can be estimated by:*

$$\hat{r}_i = \frac{N \cdot K_\sigma(\mathcal{F}_i^S, \mathcal{F}_\theta)}{\sum_{j=1}^N K_\sigma(\mathcal{F}_j^S, \mathcal{F}_\theta)},$$

where $\mathcal{F}_\theta$ *denotes the accessible model information of the target ML model.*

Proof: See Appendix A.2.

With the estimated training sample weight $\hat{r}_i$, the loss for Responsive CPI attack model is formulated as:

$$\begin{aligned} \mathcal{L}_\mathcal{A} &= \frac{1}{N}\sum_{i=1}^N \hat{r}_i \times l(h_\phi(\mathcal{F}_i^S), P_i^S) \\ &= \sum_{i=1}^N \frac{K_\sigma(\mathcal{F}_i^S, \mathcal{F}_\theta)}{\sum_{j=1}^N K_\sigma(\mathcal{F}_j^S, \mathcal{F}_\theta)} \times l(h_\phi(\mathcal{F}_i^S), P_i^S). \end{aligned} \tag{5}$$

Notably, this training objective explicitly incorporates the target ML model's information $\mathcal{F}_\theta$, enabling the attack model to update its parameters based on $\mathcal{F}_\theta$ and capture the responsive nature of the real-world adversary. Moreover, as established in Theorem 1, minimizing $\mathcal{L}_\mathcal{A}$ reduces the expected test error under distribution shift, enhancing the attack model's effectiveness in inferring target ML model's confidential property.

4.2. Attack-Defense Arms Race

Having introduced the Responsive CPI attack, which more accurately emulates real-world adversaries, we now propose a defense method against such attacks, thereby enhancing the protection of the target model's confidential properties in practical model-sharing scenarios.

However, defending against the Responsive CPI attack is inherently challenging. Unlike existing model-based defense methods that assume a static attack model and adjust the target ML model's parameters accordingly, the responsive nature of real-world adversaries renders such one-shot defense strategies ineffective. Specifically, when the model provider adjusts the target ML model to defend against a particular attack model, a responsive adversary may adapt its attack in response, thereby invalidating the initial defense. As a result, static defenses fail to keep pace with adversaries that continuously adapt their attack strategies.

To this end, we propose an arms race-based framework that mimics such iterative competition between the model provider and the adversary. This ongoing arms race enables both parties to improve continuously:

the model provider iteratively refines the target model f_θ to enhance its robustness, while a simulated adversary updates its attack model h_ϕ according to the information $\mathcal{F}_\theta$ of the updated target model. Upon convergence of this iteration, the two sides reach an equilibrium, resulting in a target model that is robust against responsive CPI attacks.

More precisely, in the t-th round of iteration, given the current attack model $h_{\phi^{(t)}}$, the model provider updates the target model's learnable parameters θ by solving the following optimization problem:

$$\theta^{(t)} = \arg\max_{\theta} \left[\mathcal{L}_{\mathcal{P}}(P, h_{\phi^{(t)}}(\mathcal{F}_\theta)) - \lambda \mathcal{L}_{\mathcal{T}}(\theta; D) \right], \tag{6}$$

where $\mathcal{L}_{\mathcal{P}}$ represents the test error of the attack model $h_{\phi^{(t)}}$ in inferring the confidential property P of the target model given its model information $\mathcal{F}_\theta$. Maximizing $\mathcal{L}_{\mathcal{P}}$ reduces the attack's effectiveness, thereby strengthening the target model's protection of its confidential property. Moreover, $\mathcal{L}_{\mathcal{T}}$ denotes the standard training loss of the target model on its training dataset $D = (X, Y)$.[9] Maximizing the negative of $\mathcal{L}_{\mathcal{T}}$ ensures that the target model retains high utility for its target task. The hyperparameter λ controls the trade-off between CPI-attack robustness and maintaining the utility of the target model.

Once the target model has been updated, the simulated adversary subsequently refines its Responsive CPI attack model (as illustrated in Section 4.1) based on the information of the adjusted target model, initiating the next iteration. Specifically, the adversary updates the learnable parameters ϕ of the attack model by solving the following optimization problem:

$$\phi^{(t+1)} = \arg\min_{\phi} \mathcal{L}_{\mathcal{A}}(\phi, \mathcal{F}_{\theta^{(t)}}),$$

where $\mathcal{L}_{\mathcal{A}}$ is the training loss of the attack model, as defined in Eq. (5). Next, we show the convergence of this iteration of attack-defense arms race.

THEOREM 3. *The proposed attack-defense arms race framework converges.*

Proof: See Appendix A.3.

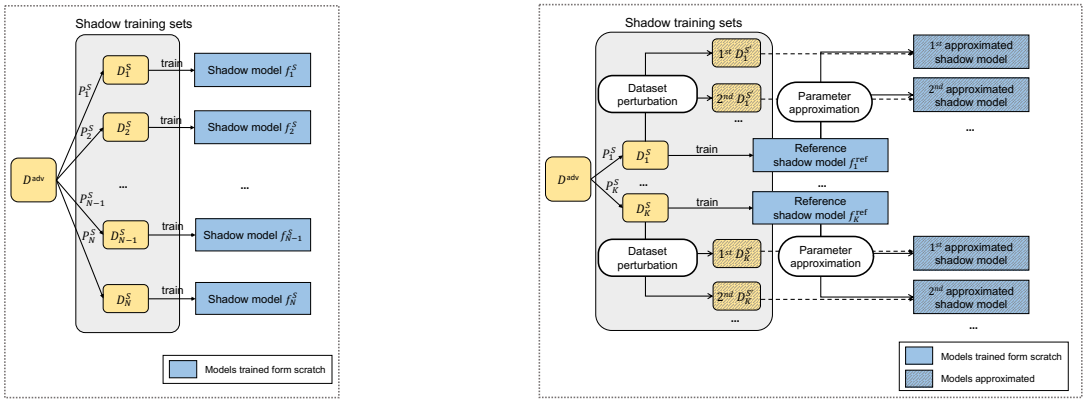
[9] For example, let the i-th instance in the training dataset $D = (X, Y)$ be denoted as $(\mathbf{x}_i, y_i)$. Assuming that y_i is a numerical label, the training loss $\mathcal{L}_{\mathcal{T}}$ can be defined as $\mathcal{L}_{\mathcal{T}} = \sum_i (f_\theta(\mathbf{x}_i) - y_i)^2$.

Taken together, this iterative arms race drives continuous improvements in both the model provider's defenses and the simulated adversary's attacks. Once convergence is reached, as guaranteed by Theorem 3, the resulting secure target model f_{θ^*} is robust against real-world responsive adversaries while preserving its model utility, making it suitable for secure model-sharing.

4.3. Approximation Strategy

An essential step in the proposed defense method is simulating a responsive adversary, which must first construct a training dataset $\{(\mathcal{F}_i^S, P_i^S)\}_{i=1}^N$ to train its attack model. Here, P_i^S represents the confidential property of the dataset D_i^S, and $\mathcal{F}_i^S$ denotes the model information of the shadow model f_i^S, which is trained on D_i^S. This dataset construction process implies that in order to obtain N training samples, the adversary must train N shadow models. As noted in prior work (Ganju et al. 2018, Mahloujifar et al. 2022, Stock et al. 2024), N can often reach hundreds or even thousands. Consequently, training such a large number of shadow models has emerged as a significant computational bottleneck in simulating the adversary, thereby affecting the overall efficiency of the defense method. To this end, we propose a novel approximation strategy that substantially reduces the computational cost associated with shadow model training.

Figure 3 compares our approximation strategy with the vanilla method for training shadow models. As illustrated, the vanilla approach samples N distinct datasets and independently trains N corresponding shadow models. In contrast, our proposed approximation strategy samples only K datasets and trains K shadow models using these datasets, where $K \ll N$. These K shadow models are referred to as *reference shadow models*, denoted by $f_1^{\text{ref}}, f_2^{\text{ref}}, \ldots, f_K^{\text{ref}}$. For the remaining $N - K$ models, instead of sampling new datasets and training each shadow model from scratch, we perturb the original K datasets and estimate how the parameters of the corresponding reference shadow models would change due to these training data perturbations. This enables us to approximate the parameters of the shadow models that would have otherwise been trained on the perturbed datasets. We refer to these models, whose parameters are not obtained through actual retraining but are instead estimated based on perturbations of the training data, as *approximated shadow models*. As summarized in the overview, our approximation strategy comprises three key steps: sampling datasets and training reference shadow models, perturbing the datasets, and approximating model parameter changes due to training data perturbations. We describe each of these steps in detail below.



(a) Existing CPI Attacks (b) Our Approximation Strategy

Figure 3 Comparison of Shadow Model Training: Existing CPI Attacks vs. Our Approximation Strategy

Data sampling and reference shadow model training. Instead of training all N shadow models from scratch, we train only K reference shadow models $f_1^{\text{ref}}, f_2^{\text{ref}}, \ldots, f_K^{\text{ref}}$. For the k-th reference shadow model, we begin by randomly sampling a subset of data $D_k^S = (X^S, Y^S)$ (where X^S and Y^S represent the attributes and the labels for the target task respectively) from the adversary's auxiliary dataset D^{adv}. The sampled dataset D_k^S is then used to train the reference shadow model f_k^{ref}, which mimics the target model by learning to map attribute values X^S to the corresponding labels Y^S for the target task. Specifically, given the training dataset D_k^S of the k-th reference shadow model f_k^{ref}, its parameters θ_k are obtained by solving the following optimization problem:

$$\theta_k = \arg\min_{\theta} L_{\mathcal{T}}(\theta; D_k^S),$$

where $L_{\mathcal{T}}(\theta; D_k^S)$ denotes the training loss of f_k^{ref} evaluated on D_k^S.

Data perturbation. Next, we perturb the datasets $D_1^S, D_2^S, \ldots, D_K^S$, each of which serves as the training dataset for the corresponding reference shadow model $f_1^{\text{ref}}, f_2^{\text{ref}}, \ldots, f_K^{\text{ref}}$. Specifically, given an original dataset D_k^S, a perturbation operation $g(\cdot)$ is applied to generate a perturbed dataset $D_k^{S'}$:

$$D_k^{S'} = g(D_k^S), g \sim G, \tag{7}$$

where G denotes the set of candidate perturbation operations. Each operation $g(\cdot)$ is sampled uniformly from G and maps D_k^S to a perturbed dataset $D_k^{S'}$. In particular, we consider the following perturbation operations:

- *Shuffling values of confidential property-related attribute*: This operation randomly shuffles a subset of values within the confidential property-related attribute while keeping all other attributes unchanged.

Thus, it introduces greater variability into the attribute joint distribution while still preserving the confidential property.

- *Shuffling values of multiple attributes*: This operation extends the above by also shuffling subsets of values from m additional attributes, randomly selected from the dataset. All selected attributes, including the confidential property-related one, are shuffled independently. This introduces variation in attribute joint distribution while preserving the confidential property.
- *Mutating values of the confidential property-related attribute*: This operation randomly replaces a subset of values in the confidential property-related attribute with alternative valid values from its domain. Unlike shuffling, this operation directly changes the dataset's confidential property.
- *Mutating values of multiple attributes*: This operation extends the previous mutation by additionally replacing subsets of values in m other randomly selected attributes. Consequently, this introduces variation into both the joint distribution of attributes and the confidential property itself.

Moreover, to ensure the perturbed dataset $D_k^{S'}$ remains a realistic variant of the original dataset D_k^S, we constrain the extent of perturbation using a perturbation budget $\delta \in \mathbb{Z}^+$. Specifically, we require:

$$|D_k^S - D_k^{S'}|_0 \leq \delta,$$

where $|D_k^S - D_k^{S'}|_0$ denotes the number of elements that differ between D_k^S and $D_k^{S'}$. This constraint ensures that each perturbed dataset remains close to its original counterpart, preserving the realism and distributional fidelity of perturbed datasets.

Parameter approximations. Given a dataset $D_k^{S'}$ obtained by perturbing the original dataset D_k^S, we are now in a position to approximate the parameters θ'_k of the shadow model that would have been trained on $D_k^{S'}$. We denote such an approximated shadow model as f'_k. Concretely, instead of retraining f'_k on $D_k^{S'}$, we estimate how the parameters θ_k of the reference shadow model f_k^{ref}, trained on D_k^S, would change due to the perturbation from D_k^S to $D_k^{S'}$. This allows us to obtain an approximation of θ'_k efficiently.

Let $Z_k \subset D_k^S$ be the subset of samples that have at least one attribute value modified by the perturbation, and $Z'_k \subset D_k^{S'}$ represent the corresponding perturbed samples. The model parameter change is thus denoted as $\Delta(Z_k, Z'_k)$, such that:

$$\theta'_k = \theta_k + \Delta(Z_k, Z'_k).$$

Following Koh and Liang (2017) and Warnecke et al. (2023), we approximate the parameter change $\Delta(Z_k, Z_k')$ according to the theorem below.

THEOREM 4. *Suppose the loss function $l(\cdot;\cdot)$ employed in shadow model training is twice differentiable and strictly convex. Then, the shadow model parameter change due to dataset perturbation can be approximated as:*

$$\Delta(Z_k, Z_k') \approx -\frac{1}{|D_k^S|} H_{\theta_k}^{-1} \Big(\sum_{z' \in Z_k'} \nabla_{\theta_k} l(\theta_k; z') - \sum_{z \in Z_k} \nabla_{\theta_k} l(\theta_k; z) \Big), \tag{8}$$

where $|D_k^S|$ is the size of training dataset, $H_{\theta_k}^{-1}$ denotes the inverse Hessian of the empirical loss $\frac{1}{|D_k^S|} \sum_{i=1}^{|D_k^S|} l(f_k^{ref}(X_i^S), Y_i^S)$ evaluated at θ_k, and $\nabla_{\theta_k} l(\theta_k; z)$ is the gradient of the loss with respect to model parameters θ_k for sample z.

Proof: See Appendix A.4.

Algorithm 1 summarizes our approximation strategy. By doing so, we eliminate the need to train all N shadow models from scratch. Instead, we train only K reference shadow models (lines 1-5) and apply the proposed parameter approximation strategy to the remaining $N - K$ models (lines 6-12). This approach significantly reduces the overall computational overhead compared to conventional methods that require independent training of all N shadow models. Consequently, our strategy substantially improves the computational efficiency of the proposed defense method and ensures its practical viability for real-world secure model-sharing scenarios.

4.4. Overall Algorithm

Given a vanilla target ML model $f_{\theta^{(0)}}$ trained on the model provider's proprietary dataset D with a confidential property P, the overall defense procedure of D-S^{2}HARE for producing a secure target model is summarized in Algorithm 2.

Specifically, line 1 of Algorithm 2 invokes the proposed approximation strategy (Algorithm 1) to efficiently acquire N shadow models, thereby significantly reducing the computational overhead associated with shadow model training. Using these shadow models, lines 2-6 derive each model's information $\mathcal{F}_i^S$ and compute the corresponding training dataset's confidential property P_i^S; together, these form the dataset $\{(\mathcal{F}_i^S, P_i^S)\}_{i=1}^N$ for training Responsive CPI attack models that simulate real-world adversaries. Line 8 initializes a Responsive

Algorithm 1 Approximation Strategy for Shadow Model Training

Input: Auxiliary dataset D^{adv}; total number of shadow models N; number of reference shadow models K.

Output: Shadow models f_i^S with parameters θ_i and corresponding training datasets D_i^S, for $i = 1, \ldots, N$.

1: **for** $i \in [1, K]$ **do**
2: Sample training dataset D_i^S from D^{adv}.
3: Train reference shadow model f_i^{ref} with parameters $\theta_i = \arg\min_\theta \mathcal{L}_\mathcal{T}(\theta; D_i^S)$.
4: Set $f_i^S \leftarrow f_i^{\text{ref}}$.
5: **end for**
6: **for** $i \in [K+1, N]$ **do**
7: Uniformly sample $k \sim [1, K]$.
8: Generate perturbed dataset $D_k^{S'}$ from D_k^S using the procedure in Eq (7).
9: Apply Theorem 4 to approximate parameters θ_k' of shadow model f_k' if trained on $D_k^{S'}$.
10: Set $f_i^S \leftarrow f_k'$ and $\theta_i \leftarrow \theta_k'$.
11: Set $D_i^S \leftarrow D_k^{S'}$.
12: **end for**
13: **return** Shadow models f_i^S with parameters θ_i and their training datasets D_i^S, for $i = 1, \ldots, N$.

CPI attack model that is specifically tailored to model information $\mathcal{F}_{\theta^{(0)}}$ of the vanilla target ML model. Starting from line 9, the attack-defense arms race proceeds iteratively: the model provider first updates the target model parameters to reduce the attack's effectiveness while preserving utility (line 11), after which the simulated adversary refines its attack model in response (line 12). The process repeats until convergence or until the maximum iteration count T is reached, ultimately yielding a secure target model f_{θ^*} that is robust to Responsive CPI attacks yet retains high utility for its original task. In addition, a computational complexity analysis of D-S^{2}HARE can be found in Appendix B.

5. Empirical Evaluation

This section provides a comprehensive evaluation of our proposed defense method. We describe the experimental setup, including the model-sharing scenario, dataset, evaluation metrics, and benchmark methods in Section 5.1. Next, we report the evaluation results on CPI-attack robustness and utility retention for our proposed defense and benchmarks in Section 5.2. We then present the robustness checks in Section 5.3 and the defense efficiency assessment in Section 5.4. Lastly, we analyze the performance contribution of key

Algorithm 2 Overall Defense Procedure of D-S^{2}HARE

Input: Vanilla target ML model $f_{\theta^{(0)}}$ and accessible model information $\mathcal{F}_{\theta^{(0)}}$;
proprietary dataset D and its confidential property P;
convergence threshold ϵ; maximum iterations T; trade-off hyperparameter λ.

Output: Secure target ML model f_{θ^*}.

Acquire shadow model f_i^S with parameters θ_i and training dataset D_i^S by Algorithm 1, for $i = 1, \ldots, N$.
for $i \in [1, N]$ **do**
  Derive model information $\mathcal{F}_i^S$ of shadow model f_i^S.
  Compute confidential property P_i^S of dataset D_i^S.
end for
Construct training dataset $\{(\mathcal{F}_i^S, P_i^S)\}_{i=1}^N$ for the attack model.
Set iteration count $t = 0$.
Initialize Responsive CPI attack $h_{\phi^{(1)}}$ with $\phi^{(1)} = \arg\min_\phi \mathcal{L}_\mathcal{A}(\phi, \mathcal{F}_{\theta^{(0)}})$. ▷ Eq. (5)
repeat
  $t \leftarrow t + 1$.
  Update target ML model to $f_{\theta^{(t)}}$ with $\theta^{(t)} = \arg\max_\theta \left[\mathcal{L}_\mathcal{P}(P, h_{\phi^{(t)}}(\mathcal{F}_\theta)) - \lambda \mathcal{L}_\mathcal{T}(\theta; D)\right]$. ▷ Eq. (6)
  Refine Responsive CPI attack model to $h_{\phi^{(t+1)}}$ with $\phi^{(t+1)} = \arg\min_\phi \mathcal{L}_\mathcal{A}(\phi, \mathcal{F}_{\theta^{(t)}})$. ▷ Eq. (5)
until $|\theta^{(t)} - \theta^{(t-1)}| < \epsilon \bigvee t \geq T$.
$f_{\theta^*} \leftarrow f_{\theta^{(t)}}$
return Secure target ML model f_{θ^*}.

components in our proposed method in Section 5.5. Additional experiments, such as parameter sensitivity analysis, are provided in Appendix D.

5.1. Experimental Setup

Scenario and dataset. We evaluated our method in a model-sharing scenario where a bank (the model provider) intends to share an ML telemarketing model (i.e., the target model). The target task of this telemarketing model is to predict whether a customer will subscribe to a long-term deposit, to aid banks in targeting their telemarketing efforts toward customers most likely to respond. The model provider must protect the confidential property of the telemarketing model's training dataset from being inferred by potential adversaries, while ensuring that the telemarketing model effectively performs the target task. The model provider can share the target model in two ways: white-box and black-box settings. In the white-box setting, the model provider directly shares the parameters and architectures of the target model. In the black-box

setting, the model provider shares only query access to the target model without exposing the model details. Our evaluation was conducted under both settings.

We utilized the Bank Marketing dataset (Moro et al. 2012) to simulate this model-sharing scenario. The dataset contains records related to telemarketing campaigns of a Portuguese banking institution. Each record describes a customer's information with three main categories: (1) demographic attributes, including age, job, marital status and education level; (2) bank marketing actions, including contact channels via cellular or telephone, last contact time, number of contacts during and before the campaign, and the outcome of the previous marketing campaign; (3) banking profile, including whether the customer has a personal loan, a housing loan, the customer's average yearly balance, and whether the customer has defaulted. Among these, attributes in the banking profile are particularly confidential, as their properties reflect key indicators of a bank's risk management and overall asset quality. Here, we designated the default rate, i.e., the percentage of customers who have defaulted, as the confidential property of the dataset. We also defined an additional confidential property for robustness checks in Section 5.3.

The Bank Marketing dataset contains 45,211 records. We randomly sampled 25,000 records from the dataset for the model provider to constitute the provider's confidential data. The remaining records were the adversary's auxiliary dataset. To comprehensively evaluate performance under varying cases where the model provider and adversary possess distinct datasets, we repeated this sampling process 100 times. This resulted in 100 unique cases, each with a different partitioning of the model provider's private data and the adversary's auxiliary dataset.

Evaluation metrics. The performance of each method was evaluated using two metrics, *success rate* of CPI attacks and *target accuracy*, to reflect the CPI-attack robustness and utility retention capacity of the shared target model, respectively.

Specifically, to evaluate the robustness of defense methods against CPI attacks, we framed the CPI attack task as a binary classification task that predicts whether the dataset's default rate is below or above 5%. Additional CPI attack tasks were examined as part of the robustness checks in Section 5.3. Accordingly, the

success rate of CPI attacks is defined as the proportion of shared target models for which the attack model accurately infers the confidential property. We have

$$\text{Attack success rate} = \frac{N_p}{N_t},$$

where N_p is the number of target models whose confidential properties are correctly inferred by the attack model, and N_t is the total number of target models. A lower success rate indicates a stronger ability of defense methods to defend against CPI attacks.

The target accuracy was introduced to evaluate the performance of the target ML models in the target telemarketing task. It is calculated as the ratio of the correctly predicted customer outcomes (N_m) to the total number of customers (N_c):

$$\text{Target accuracy} = \frac{N_m}{N_c}.$$

Higher target accuracy suggests that the target model maintains its utility effectively.

Benchmark methods. We compared our defense method D-S^{2}HARE with the following benchmarks.

- W/O defense, which is the simplest baseline to train the target model as usual, without any defenses. Compared with this benchmark, we can assess the robustness of defense methods against CPI attacks.

In addition, we compared our method with existing noise-based defense methods, which introduce noise into different phases during the target model training. The comparison with these methods demonstrates the advantage of model-based defense methods in maintaining utility.

- Noisy Label (Ganju et al. 2018), which introduces noise to the labels of the target task by flipping a certain proportion of labels.
- DP-SGD (Mahloujifar et al. 2022), which adds Laplace noise to the target model gradients to obscure sensitive information during training.
- Resampling (Suri et al. 2023), which under-samples the training data for the target model so that the confidential property is altered.

Moreover, we benchmarked our method against state-of-the-art model-based defense methods, which aim to defend against a presumed adversary. This comparison highlights the superiority of our method over existing model-based defenses.

- Property Unlearning (Stock et al. 2022), which modifies the parameters of the target model to reduce the efficacy of the CPI attack model proposed by Ganju et al. (2018). This method is specifically designed for the white-box setting only.
- Adversarial Defense (Stock et al. 2024), which adjusts the parameters of the target model to minimize the effectiveness of the CPI attack proposed by Ganju et al. (2018), while simultaneously preserving the utility of the target ML model.

Implementation details. In our model-sharing scenario, we assumed the target model is a Multilayer Perceptron (MLP) with 2 hidden layers of sizes 32 and 16. For the model provider, we sampled 10,000 records for training the target ML model and the remaining 15,000 records for simulating CPI attacks. When implementing our defense method, we set the batch size to 32 and the learning rate to 1e-3. Both $\mathcal{L}_{\mathcal{P}}$ and $\mathcal{L}_{\mathcal{T}}$ in the objective function (1) were instantiated as cross-entropy losses. The default value of the trade-off parameter λ was set as 0.3. When simulating the adversary using the Responsive CPI attack, we trained $K = 100$ reference shadow models with the same architecture as the target model. From each reference shadow model's training dataset, we generated 4 perturbed datasets for approximated shadow models, obtaining 400 approximated shadow models. When implementing the Responsive CPI attack in the black-box setting, we randomly sampled 1000 samples from the adversary's auxiliary data as the query set. This query set was then used to obtain outputs from both the shadow models and the target model, enabling the attack model to learn and infer the confidential properties based on these outputs. In the approximation strategy, we set the perturbation budget δ at 1,000. The attack model was an MLP network with 3 hidden layers of sizes 32, 16 and 8. The $l(\cdot,\cdot)$ term used in the attack loss $\mathcal{L}_{\mathcal{A}}$ was instantiated as the cross-entropy loss. When estimating the training sample weight r_i, we set the kernel width σ^2 to 0.75.

For the benchmark methods, we followed Ganju et al. (2018) to implement Noisy Label by flipping labels for the target task. DP-SGD was implemented following Yousefpour et al. (2021). We adapted the publicly released codes[10] from Suri et al. (2023) for implementing Resampling and Property Unlearning. Additionally, we implemented Adversarial Defense based on the algorithm described by Stock et al. (2024).

[10] https://github.com/iamgroot42/propertyinference

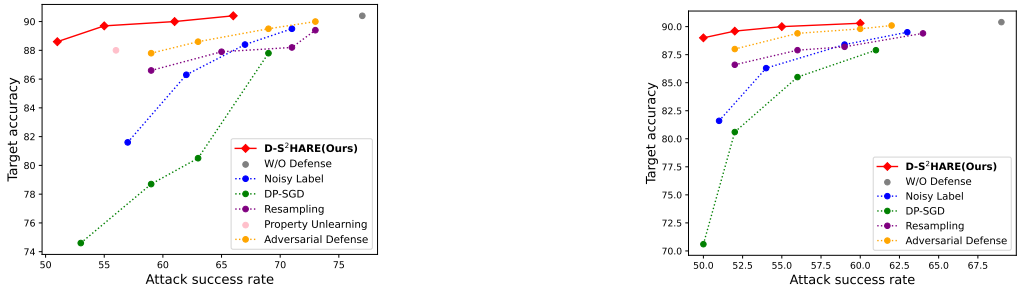
All parameters were set to their default values, except for the confidentiality-utility trade-off parameters, if such parameters exist for a given benchmark. Specifically, the confidentiality-utility trade-off parameter is typically used to control the amount of allowed confidential information leakage. Generally speaking, as the allowed confidential information leakage increases, the model's robustness to CPI attacks decreases, while its utility (i.e., target accuracy) improves. Except for Property Unlearning, all benchmarks include this trade-off parameter. To ensure comparability across benchmarks, we conducted a comprehensive analysis of performance under different trade-off parameter settings. Details on the settings of trade-off parameters for each compared method are provided in Appendix C.

5.2. Performance Under Responsive CPI Attack

In this section, we evaluated the performance of our proposed defense method and other benchmarks against the Responsive CPI attack, the most realistic and strongest attack that captures the adaptive nature of real-world adversaries. Performance of the defenses under other non-responsive CPI attacks is reported in Section 5.3. Specifically, the performance of each defense method was evaluated using two metrics: the success rate of the CPI attack, which evaluates the robustness of the target ML model, and the target accuracy, which reflects the model's utility retention capability.

Figure 4 presents a comparison of our defense method and benchmark methods under both white-box (i.e., adversaries have access to model parameters) and black-box settings (i.e., adversaries have query access only). The horizontal axis represents the success rate of the Responsive CPI attack, while the vertical axis indicates the target accuracy. A lower value on the horizontal axis reflects higher robustness to Responsive CPI attacks, whereas a higher value on the vertical axis corresponds to better utility retention. Except for Property Unlearning, which does not include a confidentiality-utility trade-off parameter, we evaluated each defense method under four different trade-off parameter settings. For each method, each point represents the result under one configuration. Notably, points closer to the upper-left corner indicate better overall performance, combining both higher robustness and better utility retention.

As shown in the figure, compared to all the benchmark methods, our defense method positions in the upper-left corner under all settings, indicating the best performance in both CPI-attack robustness (lower



(a) P as binary categories of default rate, white-box setting

(b) P as binary categories of default rate, black-box setting

Figure 4 Performance Comparison between Our Method and Benchmarks under Varying Trade-off Parameters

attack success rate) and utility retention (higher target accuracy). By comparison, noise-based defense methods (i.e., Noisy Label, DP-SGD and Resampling) often result in a significant reduction in model utility, which degrades the target task's performance and makes them less suitable for secure model-sharing. Existing model-based defense methods (i.e., Property Unlearning and Adversarial Defense) typically achieve better utility but fall short in defending against Responsive CPI attack compared to our method. The superior performance of our method is attributed to the consideration of responsive nature of CPI attacks during the training process of the target model. By progressively defending against responsive attacks, our method enhances the robustness against CPI attacks.

5.3. Robustness Check

Moreover, we conducted a series of robustness checks by varying the experimental settings to assess whether our method consistently outperforms others. Specifically, we examined performance of the compared methods under four types of robustness settings.

First, we altered the CPI attack task. Instead of the binary classification of the default rate, we further constructed a 3-class classification task that categorizes the default rate into one of three intervals: below 5%, between 5% and 10%, or between 10% and 15%. The results are presented in Figure 5(a) and 5(b).

Second, we evaluated performance of the methods under a different confidential property: the housing loan rate, defined as the mean value of the attribute indicating whether a customer has a housing loan. The corresponding CPI task here is a binary classification that determines whether it is below or above 50%. The results are reported in Figure 5(c) and 5(d).

Third, we examined the performance of our method and benchmark methods against non-responsive CPI attacks. For white-box CPI attacks, we evaluated the performance of the methods against the attack proposed

by Ganju et al. (2018), which assumes that adversary can obtain the parameters of the target model and thus inferring the confidential property by sorted network neurons. For black-box CPI attacks, we evaluated the performance of the methods against the attack proposed by Zhang et al. (2021), which uses probability vectors from a specific set of queries to the target model to infer confidential properties. The results are shown in Figure 5(e) and 5(f).

Lastly, we replaced the target ML model with a logistic regression model to evaluate the robustness and superiority of our proposed defense across different model architectures. The results are presented in Figure 5(g) and 5(h). Across all robustness checks, we can observe that our defense method consistently remains positioned at the upper-left corner of each figure, indicating superior performance in both CPI-attack robustness and utility retention under a wide range of experimental settings. We provide the implementation details of these robustness checks in Appendix C.

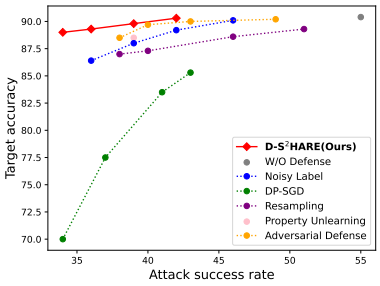
5.4. Defense Efficiency

Table 1 reports the training time of our method and the most competitive benchmarks, specifically existing model-based defense methods. Additionally, since no current methods account for the responsive nature of adversaries, a straightforward solution is to combine the best-performing benchmark, Adversarial Defense, with our proposed Responsive CPI attack. This integration initiates the arms race attack–defense framework to produce a secure target ML model. We refer to this modified version as Adversarial Defense+.

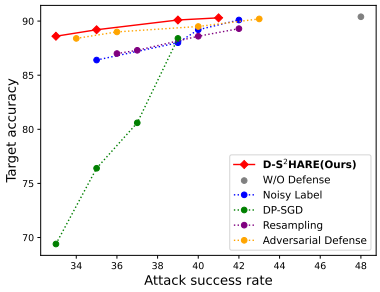
Table 1 **Training Time Comparison between D-S^{2}HARE and Existing Model-based Defenses (in Second)**

Method	White-box		Black-box	
	Training time	Time reduced	Training time	Time reduced
D-S^{2}HARE (Ours)	141.86s	-	258.81s	-
Property Unlearning	581.73s	75.61%	-	-
Adversarial Defense	566.31s	74.95%	606.84s	57.35%
Adversarial Defense+	2641.57s	94.63%	3147.39s	91.78%

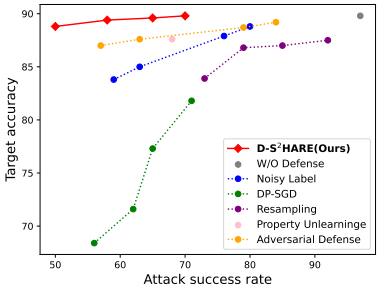
The results show that our method is the most time-efficient, achieving a substantial reduction in training time—specifically, by 74.95% to 94.63% in the white-box setting and by 57.35% to 91.78% in the black-box setting. The significant improvement in efficiency is attributed to our proposed approximation strategy, which eliminates the need for extensive training shadow models from scratch to simulate CPI attacks. In comparison,



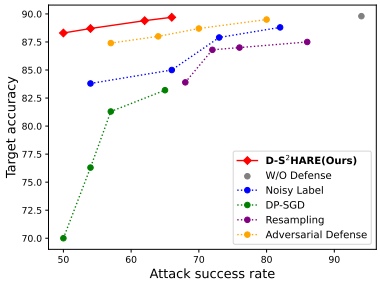
(a) P as 3-class categories of default rate, white-box



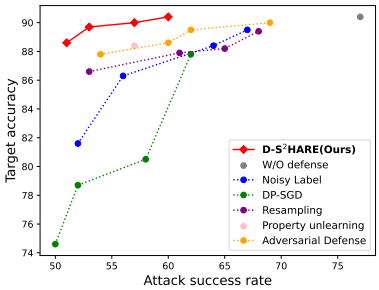
(b) P as 3-class categories of default rate, black-box



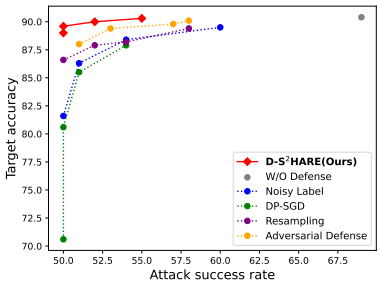
(c) P as binary categories of housing loan rate, white-box



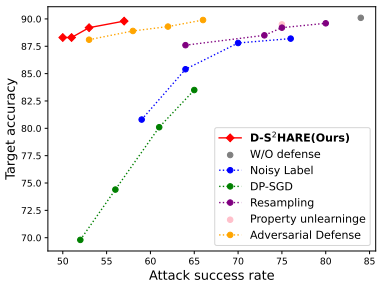
(d) P as binary categories of housing loan rate, black-box



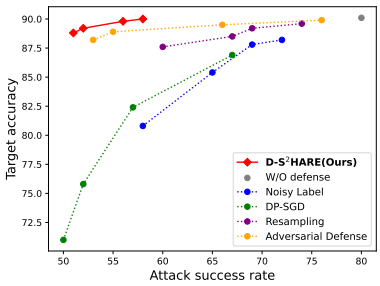
(e) P as binary categories of default rate, white-box, under non-responsive CPI attack proposed by Ganju et al. (2018)



(f) P as binary categories of default rate, black-box, under non-responsive CPI attack proposed by Zhang et al. (2021)



(g) P as binary categories of default rate, white-box, logistic regression model as the target model



(h) P as binary categories of default rate, black-box, logistic regression model as the target model

Figure 5 Robustness Checks

the substantial computational cost of Adversarial Defense+ underscores that existing model-based defense methods, when accounting for responsive adversaries, exhibit very high training time, thereby failing to meet practical efficiency requirements. Other model-based defense benchmarks exhibit longer runtime than ours, as they necessitate training all the shadow models from scratch when simulating CPI attacks. In summary,

our method not only achieves superior efficiency but also better performance in defending against CPI attacks and maintaining utility.

5.5. Ablation Analysis

We further conducted an ablation analysis to investigate the contribution of each component of our proposed defense method D-S^{2}HARE to its performance. Our focus is on two key methodological components: the consideration of Responsive CPI attacks and the approximation strategy. More concretely, we first replaced the simulated Responsive CPI attack in our D-S^{2}HARE with existing CPI attacks, designating the resulting method as Ours-R. In the white-box setting, we replaced the Responsive CPI attack with the attack proposed by Ganju et al. (2018), while in the black-box setting, we replaced it with the attack proposed by Zhang et al. (2021). The performance difference between D-S^{2}HARE and Ours-R reveals the contribution of considering the responsive nature of CPI attacks. Furthermore, to evaluate the contribution of the approximation strategy, we removed this component from our method and trained all shadow models from scratch. We designate this resulting method as Ours-A. The performance difference between D-S^{2}HARE and Ours-A uncovers the contribution of the approximation strategy.

Table 2 Ablation Analysis of D-S^{2}HARE

	White-box			Black-box		
	Success rate	Target acc	Run time	Success rate	Target acc	Run time
D-S^{2}HARE (Ours)	**0.55**	**0.897**	**141.86s**	**0.52**	**0.896**	**258.81s**
Ours-R	0.60	0.900	135.71s	0.61	0.890	232.03s
Ours-A	0.50	0.890	544.12s	0.50	0.893	649.85s

Table 2 shows the performance of D-S^{2}HARE, Ours-R and Ours-A under both white-box and black-box settings. As shown, the attack success rate of Ours-R is notably higher than that of D-S^{2}HARE, highlighting the importance of considering responsive nature of CPI attacks during the defense process. Without this component, the shared target ML model fails to effectively defend against Responsive CPI attacks, resulting in reduced CPI-attack robustness. The variant Ours-A performs comparable to D-S^{2}HARE in terms of CPI-attack robustness and utility retention, but the training time is significantly higher. This underscores the necessity to improve efficiency through the proposed approximation strategy. Overall, the ablation analysis highlights the substantial contributions of both the consideration of the responsive nature of adversaries and the approximation strategy to the effectiveness and efficiency of our defense method.

6. Conclusion and Discussion

6.1. Summary and Contributions

Model sharing offers considerable business value, but concerns over data confidentiality remain a critical barrier. Such concerns arise from well-known CPI attacks, which can infer training dataset's confidential properties solely from access to the shared model. Although prior studies have proposed defenses against CPI attacks, existing methods often assume that attacks are static and overlook the responsive nature of real-world adversaries, making these defenses ineffective in practical model-sharing scenarios. To address this limitation, we propose a novel defense method that explicitly accounts for the responsive nature of real-world adversaries. Specifically, we first devise a Responsive CPI attack that simulates such adversaries. Building on this, we introduce an attack–defense arms race framework, where the target ML model and the attack model are iteratively enhanced. This iterative process results in a secure ML model that is robust against responsive CPI attacks. Furthermore, we propose and integrate a novel approximation strategy into our defense, which addresses the critical computational bottleneck of existing methods, thereby improving defense efficiency. Through extensive empirical evaluations across various realistic model-sharing scenarios, we demonstrate that our method outperforms existing defenses by more effectively defending against CPI attacks, preserving ML model utility, and reducing computational overhead.

Our study contributes to the extant literature in two ways. First, our study belongs to the computational genre of design science research in the IS field, which emphasizes the development of computational methods to address important business or societal problems and aims to make significant methodological contributions (Abbasi et al. 2024, Rai 2017, Padmanabhan et al. 2022). In this regard, we focus on the valuable model-sharing business and propose a novel defense method to address a critical information privacy issue arising in its applications. Specifically, the key innovations of our novel defense method—Responsive CPI attack, arms race-based defense framework, and approximation strategy for computational efficiency—constitute the core methodological contributions of this work. Second, our study contributes to the research field of information privacy, which has increasingly captured the attention of IS scholars due to its substantial social and business implications (e.g., Li and Qin 2017, Xu and Dinev 2022). By proposing a novel method that enables model providers to share ML models securely, our work addresses a critical information privacy

challenge that discourages model providers from engaging in the model-sharing. In doing so, our study promotes technological equity and helps mitigate the risks associated with technological monopolies.

6.2. Implications and Future Work

Our study offers several managerial implications for stakeholders in the model-sharing business. First, by adopting our secure model-sharing method, model providers can monetize their AI intellectual property (i.e., well-established ML models) without compromising data confidentiality. This approach addresses information privacy concerns and provides a reliable source of secure revenue streams. Second, our proposed Responsive CPI attack equips model providers with advanced tools for proactive risk management and risk assessment. This enables them to identify and address potential vulnerabilities early so as to better prepare both for sharing their ML models externally and reinforcing their internal privacy frameworks. This proactive stance is crucial for effective AI governance, helping ensure compliance with regulatory standards and promoting secure, trustworthy collaborations across various industries (Berente et al. 2021). Third, our method significantly empowers model users, especially SMEs, by providing them access to well-established ML models without the heavy investments typically required for data collection and computing infrastructure. This access supports the growth and development of SMEs, allowing them to utilize advanced ML models to enhance their decision-making capabilities and drive business expansion. Consequently, SMEs can accelerate their innovation, optimize their operations, and significantly strengthen their market position in the industry.

This study also carries implications for policymakers and regulatory authorities who aim to make policy to foster secure and responsible AI ecosystems. First, as ML models increasingly become strategic assets for firms, our study demonstrates that data confidentiality risks can arise solely from sharing trained ML models. This suggests that data privacy regulations may need to extend their scope to explicitly regulate model-sharing practices, ensuring that derived or inferred confidential information from ML models is also covered under information privacy compliance standards. Second, our study highlights that traditional defense strategies, which are typically designed based on static adversary assumptions, fail to safeguard ML models against responsive attacks. These responsive attacks more accurately reflect real-world threats. Therefore, regulatory guidelines should also evolve beyond static risk assumptions and incorporate evaluation criteria that assess model robustness against responsive privacy attacks.

Our study opens up multiple directions for future research. First, adapting our secure model-sharing paradigm to dynamic, streaming data environments presents a valuable future direction. In real-world applications, the model provider's training data and the confidential properties contained within it often change over time. Therefore, future studies could extend our approach to allow for the continuous updating of the shared model in response to incoming data, while ensuring confidentiality, by incorporating continual learning and online learning approaches (Bottou and Cun 2003, Lesort et al. 2020). Second, our secure model-sharing paradigm introduces an attack–defense arms race framework where both the attack model and the target model are iteratively updated to achieve robustness against Responsive CPI attacks. Future research could explore methods to accelerate convergence in this iterative process to further improve the efficiency of secure model sharing.

References

Abadi M, Chu A, Goodfellow I, McMahan HB, Mironov I, Talwar K, Zhang L (2016) Deep learning with differential privacy. *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*.

Abbasi A, Parsons J, Pant G, Sheng ORL, Sarker S (2024) Pathways for design research on artificial intelligence. *Information Systems Research* 35(2):441–459.

Agoraki MEK, Delis MD, Pasiouras F (2011) Regulations, competition and bank risk-taking in transition countries. *Journal of Financial Stability* 7(1):38–48.

Ateniese G, Mancini LV, Spognardi A, Villani A, Vitali D, Felici G (2015) Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *International Journal of Security and Networks* 10(3):137–150.

Bammens Y, Hünermund P (2021) How midsize companies can compete in ai. *Harvard Business Review* 6.

Berente N, Gu B, Recker J, Santhanam R (2021) Managing artificial intelligence. *MIS quarterly* 45(3).

Bertran M, Tang S, Roth A, Kearns M, Morgenstern JH, Wu SZ (2023) Scalable membership inference attacks via quantile regression. *Advances in Neural Information Processing Systems* 36:314–330.

Bottou L, Cun Y (2003) Large scale online learning. *Advances in neural information processing systems* 16.

Buckman JR, Bockstedt JC, Hashim MJ (2019) Relative privacy valuations under varying disclosure characteristics. *Information Systems Research* 30(2):375–388.

Carlini N, Liu C, Erlingsson Ú, Kos J, Song D (2019) The secret sharer: Evaluating and testing unintended memorization in neural networks. *28th USENIX security symposium (USENIX security 19)*.

Chaudhari H, Abascal J, Oprea A, Jagielski M, Tramer F, Ullman J (2023) Snap: Efficient extraction of private properties with poisoning. *2023 IEEE Symposium on Security and Privacy (SP)* (IEEE).

Diamond DW, Dybvig PH (1983) Bank runs, deposit insurance, and liquidity. *Journal of political economy* 91(3):401–419.

Fang X, Gao Y, Hu PJ (2021) A prescriptive analytics method for cost reduction in clinical decision making. *MIS Quarterly* 245(1):83–115.

Fredrikson M, Jha S, Ristenpart T (2015) Model inversion attacks that exploit confidence information and basic countermeasures. *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*.

Fredrikson M, Lantz E, Jha S, Lin S, Page D, Ristenpart T (2014) Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. *23rd USENIX security symposium (USENIX Security 14)*.

Gan W, Wan S, Philip SY (2023) Model-as-a-service (maas): A survey. *2023 IEEE International Conference on Big Data (BigData)* (IEEE).

Ganju K, Wang Q, Yang W, Gunter CA, Borisov N (2018) Property inference attacks on fully connected neural networks using permutation invariant representations. *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*.

Han X, Wang L, Wu J, Fang X (2025) Data valuation for vertical federated learning: A model-free and privacy-preserving method. *MIS Quarterly* forthcoming.

Hu H, Salcic Z, Sun L, Dobbie G, Yu PS, Zhang X (2022) Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)* 54(11s):1–37.

Huang MH, Rust RT (2018) Artificial intelligence in service. *Journal of service research* 21(2):155–172.

Jayaraman B, Evans D (2019) Evaluating differentially private machine learning in practice. *28th USENIX Security Symposium (USENIX Security 19)*.

Jia J, Gong NZ (2018) Attriguard: A practical defense against attribute inference attacks via adversarial machine learning. *27th USENIX Security Symposium (USENIX Security 18)*.

Koh PW, Liang P (2017) Understanding black-box predictions via influence functions. *International conference on machine learning* (PMLR).

Lesort T, Lomonaco V, Stoian A, Maltoni D, Filliat D, Díaz-Rodríguez N (2020) Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Information fusion* 58:52–68.

Li XB, Qin J (2017) Anonymizing and sharing medical text records. *Information Systems Research* 28(2):332–352.

Li XB, Sarkar S (2011) Protecting privacy against record linkage disclosure: A bounded swapping approach for numeric data. *Information Systems Research* 22(4):774–789.

Mahloujifar S, Ghosh E, Chase M (2022) Property inference from poisoning. *2022 IEEE Symposium on Security and Privacy (SP)* (IEEE).

Melis L, Song C, De Cristofaro E, Shmatikov V (2019) Exploiting unintended feature leakage in collaborative learning. *2019 IEEE symposium on security and privacy (SP)* (IEEE).

Moro S, Rita P, Cortez P (2012) Bank Marketing. UCI Machine Learning Repository, DOI: https://doi.org/10.24432/C5K306.

Nasr M, Shokri R, Houmansadr A (2018) Comprehensive privacy analysis of deep learning. *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP)*.

Noorbakhsh SL, Zhang B, Hong Y, Wang B (2024) Inf2guard: An information-theoretic framework for learning privacy-preserving representations against inference attacks. *33rd USENIX Security Symposium (USENIX Security 24)*.

Oh SJ, Augustin M, Fritz M, Schiele B (2018) Towards reverse-engineering black-box neural networks. *International Conference on Learning Representations*.

Padmanabhan B, Fang X, Sahoo N, Burton-Jones A (2022) Machine learning in information systems research. *MIS Quarterly* 46(1).

Pant G, Sheng OR (2015) Web footprints of firms: Using online isomorphism for competitor identification. *Information Systems Research* 26(1):188–209.

Priestley M, O'donnell F, Simperl E (2023) A survey of data quality requirements that matter in ml development pipelines. *ACM Journal of Data and Information Quality* 15(2):1–39.

Rai A (2017) Editor's comments: Diversity of design science research. *MIS Quarterly* 41(1):iii–xviii.

Ransbotham S, Kiron D, Gerbert P, Reeves M (2017) Reshaping business with artificial intelligence: Closing the gap between ambition and action. *MIT sloan management review* 59(1).

Rigaki M, Garcia S (2023) A survey of privacy attacks in machine learning. *ACM Computing Surveys* 56(4):1–34.

Salem A, Zhang Y, Humbert M, Berrang P, Fritz M, Backes M (2018) Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246* .

Shimodaira H (2000) Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference* 90(2):227–244.

Shokri R, Stronati M, Song C, Shmatikov V (2017) Membership inference attacks against machine learning models. *2017 IEEE symposium on security and privacy (SP)* (IEEE).

Smith HJ, Dinev T, Xu H (2011) Information privacy research: an interdisciplinary review. *MIS quarterly* 989–1015.

Song L, Mittal P (2021) Systematic evaluation of privacy risks of machine learning models. *30th USENIX Security Symposium (USENIX Security 21)*.

Stock J, Lange L, Rahm E, Federrath H (2024) Property inference as a regression problem: Attacks and defense. *Proceedings of the International Conference on Security and Cryptography*.

Stock J, Wettlaufer J, Demmler D, Federrath H (2022) Lessons learned: How (not) to defend against property inference attacks. *arXiv preprint arXiv:2205.08821* .

Sugiyama M, Nakajima S, Kashima H, Buenau P, Kawanabe M (2007) Direct importance estimation with model selection and its application to covariate shift adaptation. *Advances in neural information processing systems* 20.

Suri A, Lu Y, Chen Y, Evans D (2023) Dissecting distribution inference. *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)* (IEEE).

Thompson N, Manso G, Ge S (2023) The importance of (exponentially more) computing power. *Academy of Management Proceedings* (Academy of Management Briarcliff Manor, NY 10510).

Tramèr F, Zhang F, Juels A, Reiter MK, Ristenpart T (2016) Stealing machine learning models via prediction {APIs}. *25th USENIX security symposium (USENIX Security 16)*.

Wang B, Gong NZ (2018) Stealing hyperparameters in machine learning. *2018 IEEE symposium on security and privacy (SP)* (IEEE).

Wang X, Wang WH (2022) Group property inference attacks against graph neural networks. *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*.

Warnecke A, Pirch L, Wressnegger C, Rieck K (2023) Machine unlearning of features and labels. *30th Annual Network and Distributed System Security Symposium, (NDSS 2023)* (The Internet Society).

Xu H, Dinev T (2022) Guest editorial: Reflections on the 2021 impact award: Why privacy still matters. *MIS Quarterly* 46(4):xx–xxxii.

Xu H, Teo HH, Tan BC, Agarwal R (2012) Research note—effects of individual self-protection, industry self-regulation, and government regulation on privacy concerns: a study of location-based services. *Information systems research* 23(4):1342–1363.

Xu H, Zhang N (2022) From contextualizing to context theorizing: Assessing context effects in privacy research. *Management Science* 68(10):7383–7401.

Yousefpour A, Shilov I, Sablayrolles A, Testuggine D, Prasad K, Malek M, Nguyen J, Ghosh S, Bharadwaj A, Zhao J, Cormode G, Mironov I (2021) Opacus: User-friendly differential privacy library in PyTorch. *arXiv preprint arXiv:2109.12298* .

Zhang H, Zhao X, Fang X, Chen B (2024) Proactive resource request for disaster response: A deep learning-based optimization model. *Information Systems Research* 35(2):528–550.

Zhang W, Tople S, Ohrimenko O (2021) Leakage of dataset properties in multi-party machine learning. *30th USENIX security symposium (USENIX Security 21)*.

Zhao S, Talasila M, Jacobson G, Borcea C, Aftab SA, Murray JF (2018) Packaging and sharing machine learning models via the acumos ai open platform. *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)* (IEEE).

Zhao X, Fang X, He J, Huang L (2023) Exploiting expert knowledge for assigning firms to industries: A novel deep learning method. *MIS quarterly* 47(3):1147–1176.

Zhou J, Chen Y, Shen C, Zhang Y (2022) Property inference attacks against gans. *30th Network and Distributed System Security Symposium (NDSS 2022)*.

Appendix A: Proof of Theorems

A.1. Proof of Theorem 1

Recalling the definition of $R(h_\phi)$ in Eq. (2), we have

$$\begin{aligned}
R(h_\phi) &= E_{\mathcal{D}_T}[l(h_\phi(\mathcal{F}_\theta), P)] \\
&= \int_{Pr_{\mathcal{D}_T}(\mathcal{F}_\theta)\neq 0} Pr_{\mathcal{D}_T}(\mathcal{F}_\theta) l(h_\phi(\mathcal{F}_\theta), P)\, d\mathcal{F}_\theta \\
&= \int_{Pr_{\mathcal{D}_S}(\mathcal{F}_i^S)\neq 0} \frac{Pr_{\mathcal{D}_T}(\mathcal{F}_i^S)}{Pr_{\mathcal{D}_S}(\mathcal{F}_i^S)} Pr_{\mathcal{D}_S}(\mathcal{F}_i^S) l(h_\phi(\mathcal{F}_i^S), P_i^S)\, d\mathcal{F}_i^S \\
&= \int_{Pr_{\mathcal{D}_S}(\mathcal{F}_i^S)\neq 0} r_i Pr_{\mathcal{D}_S}(\mathcal{F}_i^S) l(h_\phi(\mathcal{F}_i^S), P_i^S)\, d\mathcal{F}_i^S \\
&= E_{\mathcal{D}_S}[r_i l(h_\phi(\mathcal{F}_i^S), P_i^S)] \\
&= \frac{1}{N}\sum_{i=1}^{N} E_{\mathcal{D}_S}[r_i l(h_\phi(\mathcal{F}_i^S), P_i^S)] \\
&= E_{\mathcal{D}_S}[\frac{1}{N}\sum_{i=1}^{N} r_i l(h_\phi(\mathcal{F}_i^S), P_i^S)],
\end{aligned}$$

where the third equation holds under the assumption that the support of $\mathcal{D}_T$ is contained in the support of $\mathcal{D}_S$, the fourth equation uses the definition $r_i = \frac{Pr_{\mathcal{D}_T}(\mathcal{F}_i^S)}{Pr_{\mathcal{D}_S}(\mathcal{F}_i^S)}$, and the last two equations hold under the assumption that the training samples $(\mathcal{F}_i^S, P_i^S)$ are drawn i.i.d..

A.2. Proof of Theorem 2

Recall that we have $r_i = \frac{\Pr_{\mathcal{D}_T}(\mathcal{F}_i^S)}{\Pr_{\mathcal{D}_S}(\mathcal{F}_i^S)}$ that adjusts the contribution of each training sample based on the probability density of $\mathcal{F}_i^S$ under the training sample distribution $\mathcal{D}_S$ relative to that under the test distribution $\mathcal{D}_T$. A good estimate of r_i, denoted by $\hat{r}_i$, is thus supposed to ensure that

$$\hat{r}_i \Pr_{\mathcal{D}_S}(\mathcal{F}_i^S) \approx \Pr_{\mathcal{D}_T}(\mathcal{F}_i^S), \forall i = 1, \ldots, N.$$

This can be achieved by minimizing the Kullback-Leibler (KL) divergence of the probability density under the test distribution $\mathcal{D}_T$ from the probability density under the adjusted training sample distribution with respect to r_i:

$$\min_{r_i} \mathrm{KL}[\Pr_{\mathcal{D}_T}(\mathcal{F}_i^S) \| r_i \Pr_{\mathcal{D}_S}(\mathcal{F}_i^S)], \forall i = 1, \ldots, N,$$

where the KL divergence is given by

$$\begin{aligned}
\mathrm{KL}[\Pr_{\mathcal{D}_T}(\mathcal{F}_i^S) \| r_i \Pr_{\mathcal{D}_S}(\mathcal{F}_i^S)] &= \int_{\mathcal{D}_T} \Pr_{\mathcal{D}_T}(\mathcal{F}_i^S) \log \frac{\Pr_{\mathcal{D}_T}(\mathcal{F}_i^S)}{r_i \Pr_{\mathcal{D}_S}(\mathcal{F}_i^S)} d\mathcal{F}_i^S \\
&= \int_{\mathcal{D}_T} \Pr_{\mathcal{D}_T}(\mathcal{F}_i^S) \log \frac{\Pr_{\mathcal{D}_T}(\mathcal{F}_i^S)}{\Pr_{\mathcal{D}_S}(\mathcal{F}_i^S)} d\mathcal{F}_i^S - \int_{\mathcal{D}_T} \Pr_{\mathcal{D}_T}(\mathcal{F}_i^S) \log r_i d\mathcal{F}_i^S.
\end{aligned}$$

Since the first term in the last equation is independent of r_i, the optimization problem focusing on minimizing the KL divergence with respect to r_i is equivalent to maximizing the second term:

$$\int_{\mathcal{D}_T} \Pr_{\mathcal{D}_T}(\mathcal{F}_i^S) \log r_i d\mathcal{F}_i^S \approx \frac{1}{N_T} \sum_i^{N_T} \log r_i,$$

where N_T is the number of samples from the test distribution (i.e., target models in our problem).

Note that r_i is non-negative by definition. Furthermore, to ensure that the probability density under the adjusted training sample distribution $r_i \Pr_{\mathcal{D}_S}(\mathcal{F}_i^S)$ forms a valid probability density function, r_i must satisfy the normalization constraint:

$$1 = \int_{\mathcal{D}_S} r_i \Pr_{\mathcal{D}_S}(\mathcal{F}_i^S) d\mathcal{F}_i^S \approx \frac{1}{N} \sum_i^N r_i.$$

Consider the training sample weight r_i modeled as a linear model as suggested by (Sugiyama et al. 2007):

$$r_i = \sum_{j=1}^{N_T} \alpha_j K_\sigma(\mathcal{F}_i^S, \mathcal{F}_j),$$

where $K_\sigma(\mathcal{F}_i^S, \mathcal{F}_j) = exp\{-\frac{\|\mathcal{F}_i^S - \mathcal{F}_j\|^2}{2\sigma^2}\}$ is the Gaussian kernel with kernel width σ, $\mathcal{F}_i^S$ denotes the model information of the i-th training sample, $\mathcal{F}_j$ denotes the model information of the j-th target ML model, and $\{\alpha_j\}_{j=1}^{N_T}$ are the kernel weights to be determined. Given the non-negativity and normalization constraints, the training sample weight r_i can be estimated by solving the following optimization problem:

$$\min_{\{\alpha_j\}_{j=1}^{N_T}} \frac{1}{N_T} \sum_i^{N_T} \log\left(\sum_{j=1}^{N_T} \alpha_j K_\sigma(\mathcal{F}_i^S, \mathcal{F}_j)\right)$$
$$\text{subject to } \frac{1}{N} \sum_{i=1}^N r_i = 1 \text{ and } r_i \geq 0, \forall i = 1, \ldots, N.$$

In our model-sharing scenario, the model provider typically shares only one target model, implying that $N_T = 1$. In this case, r_i simplifies to $r_i = \alpha K_\sigma(\mathcal{F}_i^S, \mathcal{F}_\theta)$, where $\mathcal{F}_\theta$ is the model information of the target model to be shared. The optimization problem then reduces to

$$\max_\alpha \log(\alpha K_\sigma(\mathcal{F}_i^S, \mathcal{F}_\theta))$$
$$\text{subject to } \frac{1}{N} \sum_i^N \alpha K_\sigma(\mathcal{F}_i^S, \mathcal{F}_\theta) = 1 \text{ and } \alpha \geq 0.$$

The closed-form solution of this problem is obtained by solving for α:

$$\alpha = \frac{N}{\sum_{j=1}^N K_\sigma(\mathcal{F}_j^S, \mathcal{F}_\theta)}.$$

Substituting this into $r_i = \alpha K_\sigma(\mathcal{F}_i^S, \mathcal{F}_\theta)$, we obtain the estimate of training sample weight r_i as

$$\hat{r}_i = \frac{N \cdot K_\sigma(\mathcal{F}_i^S, \mathcal{F}_\theta)}{\sum_{j=1}^N K_\sigma(\mathcal{F}_j^S, \mathcal{F}_\theta)}.$$

A.3. Theorem 3 And Its Proof

We first define the necessary notations to ensure clarity and simplify the convergence analysis. Subsequently, we outline the assumptions required for the convergence analysis. Following this, we detail Theorem 3 and finally provide its proof.

Notations. Let $\mathcal{L}_{\text{defend}}(\theta) := \lambda\mathcal{L}_{\mathcal{T}}(\theta) - \mathcal{L}_{\mathcal{P}}(\phi^*(\theta), \mathcal{F}_\theta)$ be the defender's objective function, where $\mathcal{L}_{\mathcal{T}}(\theta)$ is the average loss $l(f_\theta(X_i), Y_i))$ over the target model's training dataset, and $\mathcal{L}_{\mathcal{P}}(\phi^*(\theta), \mathcal{F}_\theta) = l(h_{\phi^*}(\mathcal{F}_\theta), P))$. Thus, we define the defender's per-sample loss as $l_{\text{defend}}(\theta, \phi^*(\theta)) = \lambda l(f_\theta(X_i), Y_i)) - l(h_{\phi^*}(\mathcal{F}_\theta), P))$, with $\mathcal{L}_{\text{defend}}(\theta) = E[l_{\text{defend}}(\theta, \phi^*(\theta))]$. Similarly, the adversary's per-sample loss is given by $l_{\text{adv}}(\theta, \phi) = r_i l(h_\phi(\mathcal{F}_i^S), P_i^S)$ for each training sample $(\mathcal{F}_i^S, P_i^S)$. Let h_{a}^t denote the unbiased stochastic gradient of $l_{\text{adv}}(\theta^{(t)}, \phi^{(t)})$, and let h_{d}^t denote the approximate stochastic gradient of $\mathcal{L}_{\text{defend}}(\theta^{(t)})$ at the t-th iteration. We define $\overline{\nabla} l_{\text{defend}}(\theta, \phi) := \nabla_\theta l_{\text{defend}}(\theta, \phi) - \nabla^2_{\theta\phi} l_{\text{adv}}(\theta, \phi)[\nabla^2_{\phi\phi} l_{\text{adv}}(\theta, \phi)]^{-1} \nabla_\phi l_{\text{defend}}(\theta, \phi)$ and $\bar{h}_{\text{d}}^t := E[h_{\text{d}}^t | \theta^{(t)}, \phi^{(t+1)}]$.

Assumptions for convergence analysis. We make the following assumptions for the convergence analysis.

Assumption 1 (Lipschitz continuity). $\nabla l_{\text{adv}}(\theta, \phi)$, $\overline{\nabla} l_{\text{defend}}(\theta, \phi)$, $\nabla \mathcal{L}_{\text{defend}}(\theta)$, $\phi^*(\theta)$ and $\nabla\phi^*(\theta)$ are Lipschitz continuous with constant L_1, L_2, L_3, L_4 and L_5.

$$\|\nabla l_{\text{adv}}(\theta_1, \phi_1) - \nabla l_{\text{adv}}(\theta_2, \phi_2)\| \le L_1 \|(\theta_1, \phi_1) - (\theta_2, \phi_2)\|,$$

$$\|\overline{\nabla} l_{\text{defend}}(\theta, \phi^*(\theta)) - \overline{\nabla} l_{\text{defend}}(\theta, \phi)\| \le L_2 \|\phi^*(\theta) - \phi\|,$$

$$\|\nabla \mathcal{L}_{\text{defend}}(\theta_1) - \nabla \mathcal{L}_{\text{defend}}(\theta_2)\| \le L_3 \|\theta_1 - \theta_2\|,$$

$$\|\phi^*(\theta_1) - \phi^*(\theta_2)\| \le L_4 \|\theta_1 - \theta_2\|,$$

$$\|\nabla\phi^*(\theta_1) - \nabla\phi^*(\theta_2)\| \le L_5 \|\theta_1 - \theta_2\|.$$

Assumption 2 (Strong convexity). For any fixed θ, $l_{\text{adv}}(\theta, \phi)$ is μ-strongly convex in ϕ.

Assumption 3 (Bounded variance of stochastic gradients). The variances of stochastic derivatives of $\nabla l_{\text{defend}}(\theta, \phi)$, $\nabla l_{\text{adv}}(\theta, \phi)$, $\nabla^2 l_{\text{adv}}(\theta, \phi)$ are bounded by constant σ_1^2, σ_2^2, σ_3^2, respectively.

Assumption 4 (Bounded gradient estimates). The stochastic gradient estimates satisfy: $E[\|h_{\text{d}}^t\|^2 | \theta^{(t)}, \phi^{(t+1)}] \le \tilde{C}^2$, $\|\bar{h}_{\text{d}}^t - \overline{\nabla} l_{\text{defend}}(\theta^{(t)}, \phi^{(t+1)})\| \le b_t$, $E[\|h_{\text{d}}^t - \bar{h}_{\text{d}}^t\|^2] \le \tilde{\sigma}^2$.

Restatement of Theorem 3 .

THEOREM 3. *Suppose Assumptions 1-4 hold. Let α_t and β_t be the stepsizes when training the target model and the attack model, respectively. Choose α_t and β_t as*

$$\alpha_t = min\{\bar{\alpha}_1, \bar{\alpha}_2, \frac{\alpha}{\sqrt{T}}\},\ \beta_t = \frac{8L_2L_4 + 2\eta L_5\tilde{C}^2\bar{\alpha}_1}{4T\mu}\alpha_t,$$

where $\alpha > 0$ is a control constant. Constants $\bar{\alpha}_1$ and $\bar{\alpha}_2$ are defined as

$$\bar{\alpha}_1 = \frac{1}{2L_3 + 4L_3L_4 + \frac{2L_2L_5}{L_4\eta}},\ \bar{\alpha}_2 = \frac{8\frac{2\mu L_1}{\mu+L_1}}{(\mu+L_1)(8L_2L_4 + 2\eta L_5\tilde{C}^2\bar{\alpha}_1)},$$

where $\eta > 0$ is a control constant. Then the outputs $\{\theta^{(t)}, \phi^{(t)}\}$ generated by the attack-defense arms race framework satisfy

$$\frac{1}{T}\sum_{t=1}^{T} E[\|\nabla\mathcal{L}_{defend}(\theta^{(t)})\|^2] = \mathcal{O}(\frac{1}{\sqrt{T}}) \text{ and } E[\|\phi^{(T)} - \phi^*(\theta^{(T)})\|^2] = \mathcal{O}(\frac{1}{\sqrt{T}}).$$

Proof. The attack-defense arms race framework can be formulated as the following bilevel optimization problem:

$$\text{minimize } \mathcal{L}_{\text{defend}}(\theta) := E_\xi[l_{\text{defend}}(\theta, \phi^*(\theta); \xi)]$$

$$\text{subject to } \phi^*(\theta) = \arg\min_{\phi} E_\zeta[l_{\text{adv}}(\theta, \phi; \zeta)],$$

where ξ and ζ are random variables associated stochastic gradient descent. Then in the $(t+1)$-th iteration of the arms race framework, the model parameters θ and ϕ are updated as follows:

$$\phi^{(t+1)} = \phi^{(t)} - \beta_t h_{\text{a}}^t,\ \theta^{(t+1)} = \theta^{(t)} - \alpha_t h_{\text{d}}^t.$$

Let $\mathcal{T}_t'$ denote the all the history information available up to iteration t, defined as $\mathcal{T}_t' = \sigma\{\phi^1, \theta^1, \ldots, \phi^{(t)}\}$, where $\sigma\{\cdot\}$ is the σ-algebra generated by the random variables. Using the Lipschitz continuity of $\nabla\mathcal{L}_{\text{defend}}(\theta)$, we have

$$\mathcal{L}_{\text{defend}}(\theta^{(t+1)}) \le \mathcal{L}_{\text{defend}}(\theta^{(t)}) + \langle\mathcal{L}_{\text{defend}}(\theta^{(t)}), \theta^{(t+1)} - \theta^{(t)}\rangle + \frac{L_3}{2}\|\theta^{(t+1)} - \theta^{(t)}\|^2.$$

Taking expectations, we have

$$\begin{aligned}
&E[\mathcal{L}_{\text{defend}}(\theta^{(t+1)})|\mathcal{T}_t'] \\
&\le \mathcal{L}_{\text{defend}}(\theta^{(t)}) + E[\langle\nabla\mathcal{L}_{\text{defend}}(\theta^{(t)}), \theta^{(t+1)} - \theta^{(t)}\rangle|\mathcal{T}_t'] + \frac{L_3}{2}E[\|\theta^{(t+1)} - \theta^{(t)}\|^2|\mathcal{T}_t'] \\
&= \mathcal{L}_{\text{defend}}(\theta^{(t)}) - \alpha_t\langle\nabla\mathcal{L}_{\text{defend}}(\theta^{(t)}), \bar{h}_{\text{d}}^t\rangle + \frac{L_3\alpha_t^2}{2}E[\|h_{\text{d}}^t\|^2|\mathcal{T}_t'] \\
&= \mathcal{L}_{\text{defend}}(\theta^{(t)}) - \frac{\alpha_t}{2}\|\nabla\mathcal{L}_{\text{defend}}(\theta^{(t)})\|^2 - \frac{\alpha_t}{2}\|\bar{h}_{\text{d}}^t\|^2 + \frac{\alpha_t}{2}\|\nabla\mathcal{L}_{\text{defend}}(\theta^{(t)}) - \bar{h}_{\text{d}}^t\|^2 + \frac{L_3\alpha_t^2}{2}\|\bar{h}_{\text{d}}^t\|^2 + \frac{L_3\alpha_t^2}{2}E[\|h_{\text{d}}^t - \bar{h}_{\text{d}}^t\|^2|\mathcal{T}_t'] \\
&\le \mathcal{L}_{\text{defend}}(\theta^{(t)}) - \frac{\alpha_t}{2}\|\nabla\mathcal{L}_{\text{defend}}(\theta^{(t)})\|^2 - (\frac{\alpha_t}{2} - \frac{L_3\alpha_t^2}{2})\|\bar{h}_{\text{d}}^t\|^2 + \frac{\alpha_t}{2}\|\nabla\mathcal{L}_{\text{defend}}(\theta^{(t)}) - \bar{h}_{\text{d}}^t\|^2 + \frac{L_3\alpha_t^2}{2}\tilde{\sigma}^2,
\end{aligned}$$

where the first equation uses $\theta^{(t+1)} - \theta^{(t)} = -\alpha_t h_{\mathrm{d}}^t$ and $\bar{h}_{\mathrm{d}}^t = E[h_{\mathrm{d}}^t|\mathcal{T}_t']$, the second equation uses $\langle \nabla \mathcal{L}_{\text{defend}}(\theta^{(t)}), \bar{h}_{\mathrm{d}}^t \rangle = \frac{1}{2}(\|\nabla \mathcal{L}_{\text{defend}}(\theta^{(t)})\|^2 + \|\bar{h}_{\mathrm{d}}^t\|^2 - \|\nabla \mathcal{L}_{\text{defend}}(\theta^{(t)}) - \bar{h}_{\mathrm{d}}^t\|^2)$ and $\|h_{\mathrm{d}}^t\|^2 = \|\bar{h}_{\mathrm{d}}^t + h_{\mathrm{d}}^t - \bar{h}_{\mathrm{d}}^t\|^2 = \|\bar{h}_{\mathrm{d}}^t\|^2 + \|h_{\mathrm{d}}^t - \bar{h}_{\mathrm{d}}^t\|^2$, the last inequality uses $E[\|h_{\mathrm{d}}^t - \bar{h}_{\mathrm{d}}^t\|^2] \le \tilde{\sigma}^2$. Further, we have

$$\begin{aligned}
\|\nabla \mathcal{L}_{\text{defend}}(\theta^{(t)}) - \bar{h}_{\mathrm{d}}^t\|^2 &= \|\overline{\nabla} l_{\text{defend}}(\theta^{(t)}, \phi^*(\theta^{(t)})) - \overline{\nabla} l_{\text{defend}}(\theta^{(t)}, \phi^{(t+1)}) + \overline{\nabla} l_{\text{defend}}(\theta^{(t)}, \phi^{(t+1)}) - \bar{h}_{\mathrm{d}}^t\|^2 \\
&\le 2\|\overline{\nabla} l_{\text{defend}}(\theta^{(t)}, \phi^*(\theta^{(t)})) - \overline{\nabla} l_{\text{defend}}(\theta^{(t)}, \phi^{(t+1)})\|^2 + 2\|\overline{\nabla} l_{\text{defend}}(\theta^{(t)}, \phi^{(t+1)}) - \bar{h}_{\mathrm{d}}^t\|^2 \\
&\le 2L_2^2\|\phi^{(t+1)} - \phi^{(t)}\|^2 + 2b_t^2,
\end{aligned}$$

where the last inequality uses Assumption 1 and Assumption 4. Taking expectations over $\mathcal{T}_t'$ on both sides of the inequality, we have

$$\begin{aligned}
E[\mathcal{L}_{\text{defend}}(\theta^{(t+1)})] - E[\mathcal{L}_{\text{defend}}(\theta^{(t)})] \le &-\frac{\alpha_t}{2} E[\|\nabla \mathcal{L}_{\text{defend}}(\theta^{(t)})\|^2] - (\frac{\alpha_t}{2} - \frac{L_3\alpha_t^2}{2}) E[\|h_{\mathrm{d}}^t\|^2] \\
&+ L_2^2\alpha_t E[\|\phi^{(t+1)} - \phi^*(\theta^{(t)})\|^2] + \alpha_t b_t^2 + \frac{L_3\alpha_t^2}{2}\tilde{\sigma}^2.
\end{aligned}$$

We then analyze the attack model optimization, decomposing the error of parameters of attack model as

$$\|\phi^{(t+1)} - \phi^*(\theta^{(t+1)})\|^2 = \|\phi^{(t+1)} - \phi^*(\theta^{(t)})\|^2 + \|\phi^*(\theta^{(t+1)}) - \phi^*(\theta^{(t)})\|^2 + 2\langle \phi^{(t+1)} - \phi^*(\theta^{(t)}), \phi^*(\theta^{(t)}) - \phi^*(\theta^{(t+1)}) \rangle.$$

The upper bound of $\|\phi^{(t+1)} - \phi^*(\theta^{(t)})\|^2$ can be derived as

$$\begin{aligned}
E[\|\phi^{(t+1)} - \phi^*(\theta^{(t)})\|^2|\mathcal{T}_t'] &= E[\|\phi^{(t)} - \beta_t h_{\mathrm{a}}^t - \phi^*(\theta^{(t)})\|^2|\mathcal{T}_t'] \\
&= \|\phi^{(t)} - \phi^*(\theta^{(t)})\|^2 - 2\beta_t\langle \phi^{(t)} - \phi^*(\theta^{(t)}), E[h_{\mathrm{a}}^t|\mathcal{T}_t'] \rangle + \beta_t^2 E[\|h_{\mathrm{a}}^t\|^2|\mathcal{T}_t'] \\
&\le \|\phi^{(t)} - \phi^*(\theta^{(t)})\|^2 - 2\beta_t\langle \phi^{(t)} - \phi^*(\theta^{(t)}), \nabla l_{\text{adv}}(\theta^{(t)}, \phi^{(t)}) \rangle + \beta_t^2\|\nabla l_{\text{adv}}(\theta^{(t)}, \phi^{(t)})\|^2 + \beta_t^2\sigma_2^2 \\
&\le (1 - \frac{2\mu L_1}{\mu + L_1}\beta_t)\|\phi^{(t)} - \phi^*(\theta^{(t)})\|^2 + \beta_t(\beta_t - \frac{2}{\mu + L_1})\|\nabla l_{\text{adv}}(\theta^{(t)}, \phi^{(t)})\|^2 + \beta_t^2\sigma_2^2 \\
&\le (1 - \frac{2\mu L_1}{\mu + L_1}\beta_t)\|\phi^{(t)} - \phi^*(\theta^{(t)})\|^2 + \beta_t^2\sigma_2^2,
\end{aligned}$$

where the first inequality uses $E[\|h_{\mathrm{a}}^t\||\mathcal{T}_t'] = \nabla l_{\text{adv}}(\theta^{(t)}, \phi^{(t)})$ and $\mathrm{Var}[\|h_{\mathrm{a}}^t\|] = E[\|h_{\mathrm{a}}^t\|^2] - E[\|h_{\mathrm{a}}^t\|]^2$ the second inequality uses the μ-strong convexity and L_1 smoothness property of $l_{\text{adv}}(\theta, \phi)$, and the last inequality follows from choosing step size $\beta_t \le \frac{2}{\mu + L_1}$. Taking expectations over $\mathcal{T}_t'$ on both sides of the inequality, we have

$$E[\|\phi^{(t+1)} - \phi^*(\theta^{(t)})\|^2] \le (1 - \mu\beta_t)E[\|\phi^{(t)} - \phi^*(\theta^{(t)})\|^2] + \beta_t^2\sigma_2^2,$$

Using the smoothness of $\phi^*(\theta)$ and Assumption 4, we derive the upper bound of $\|\phi^*(\theta^{(t+1)}) - \phi^*(\theta^{(t)})\|^2$:

$$\begin{aligned}
E[\|\phi^*(\theta^{(t+1)}) - \phi^*(\theta^{(t)})\|^2] &\le L_4^2 E[\|\theta^{(t+1)} - \theta^{(t)}\|^2] \\
&= L_4^2\alpha_t^2 E[E[\|h_{\mathrm{d}}^t - \bar{h}_{\mathrm{d}}^t + \bar{h}_{\mathrm{d}}^t\|^2|\mathcal{T}_t']] \\
&\le L_4^2\alpha_t^2(E[\|\bar{h}_{\mathrm{d}}^t\|^2] + \tilde{\sigma}^2).
\end{aligned}$$

The term $E[\langle \phi^{(t+1)} - \phi^*(\theta^{(t)}), \phi^*(\theta^{(t)}) - \phi^*(\theta^{(t+1)})\rangle]$ can be decomposed into two terms:

$$\begin{aligned}E[\langle \phi^{(t+1)} - \phi^*(\theta^{(t)}), \phi^*(\theta^{(t)}) - \phi^*(\theta^{(t+1)})\rangle] = &- E[\langle \phi^{(t+1)} - \phi^*(\theta^{(t)}), \nabla\phi^*(\theta^{(t)})(\theta^{(t+1)} - \theta^{(t)})\rangle] \\ &- E[\langle \phi^{(t+1)} - \phi^*(\theta^{(t)}), \phi^*(\theta^{(t+1)}) - \phi^*(\theta^{(t)}) - \nabla\phi^*(\theta^{(t)})(\theta^{(t+1)} - \theta^{(t)})\rangle].\end{aligned}$$

The first term is bounded by

$$\begin{aligned}-E[\langle \phi^{(t+1)} - \phi^*(\theta^{(t)}), \nabla\phi^*(\theta^{(t)})(\theta^{(t+1)} - \theta^{(t)})\rangle] &= -\alpha_t E[\langle \phi^{(t+1)} - \phi^*(\theta^{(t)}), \nabla\phi^*(\theta^{(t)})\bar{h}_{\mathrm{d}}^t\rangle] \\ &\le \alpha_t E[\|\phi^{(t+1)} - \phi^*(\theta^{(t)})\|\|\nabla\phi^*(\theta^{(t)})\bar{h}_{\mathrm{d}}^t\|] \\ &\le \alpha_t L_4 E[\|\phi^{(t+1)} - \phi^*(\theta^{(t)})\|\|\bar{h}_{\mathrm{d}}^t\|] \\ &\le 2\gamma_t E[\|\phi^{(t+1)} - \phi^*(\theta^{(t)})\|^2] + \frac{L_4^2\alpha_t^2}{8\gamma_t}E[\|\bar{h}_{\mathrm{d}}^t\|^2],\end{aligned}$$

where the second inequality uses the Lipschitz continuity of $\phi^*(\theta)$, and the last inequality uses the Young's inequality that $ab \le 2\gamma_t a^2 + \frac{b^2}{8\gamma_t}$. Using the smoothness of $\phi^*(\theta)$, the second term is bounded by

$$\begin{aligned}&- E[\langle \phi^{(t+1)} - \phi^*(\theta^{(t)}), \phi^*(\theta^{(t+1)}) - \phi^*(\theta^{(t)}) - \nabla\phi^*(\theta^{(t)})(\theta^{(t+1)} - \theta^{(t)})\rangle] \\ \le\ & E[\|\phi^{(t+1)} - \phi^*(\theta^{(t)})\|\|\phi^*(\theta^{(t+1)}) - \phi^*(\theta^{(t)}) - \nabla\phi^*(\theta^{(t)})(\theta^{(t+1)} - \theta^{(t)})\|] \\ \le\ & \frac{L_5}{2}E[\|\phi^{(t+1)} - \phi^*(\theta^{(t)})\|\|\theta^{(t+1)} - \theta^{(t)})\|^2] \\ \le\ & \frac{\eta L_5}{4}E[\|\phi^{(t+1)} - \phi^*(\theta^{(t)})\|^2 E[\|\theta^{(t+1)} - \theta^{(t)})\|^2|\mathcal{T}_t']] + \frac{L_5}{4\eta}E[E[\|\theta^{(t+1)} - \theta^{(t)})\|^2|\mathcal{T}_t']] \\ \le\ & \frac{\eta L_5\tilde{C}^2\alpha_t^2}{4}E[\|\phi^{(t+1)} - \phi^*(\theta^{(t)})\|^2] + \frac{L_5\alpha_t^2}{4\eta}(E[\|\bar{h}_{\mathrm{d}}^t\|^2] + \tilde{\sigma}^2),\end{aligned}$$

where the third inequality uses the Young's inequality that $1 \le \frac{\eta}{2} + \frac{1}{2\eta}$, and the last inequality uses Assumption 4. Then we obtain the upper bound of $E[\langle \phi^{(t+1)} - \phi^*(\theta^{(t)}), \phi^*(\theta^{(t)}) - \phi^*(\theta^{(t+1)})\rangle]$:

$$\begin{aligned}&E[\langle \phi^{(t+1)} - \phi^*(\theta^{(t)}), \phi^*(\theta^{(t)}) - \phi^*(\theta^{(t+1)})\rangle] \\ &\le (2\gamma_t + \frac{\eta L_5\tilde{C}^2}{4}\alpha_t^2)E[\|\phi^{(t+1)} - \phi^*(\theta^{(t)})\|^2] + (\frac{L_4^2\alpha_t^2}{8\gamma_t} + \frac{L_5\alpha_t^2}{4\eta})E[\|\bar{h}_{\mathrm{d}}^t\|^2] + \frac{L_5\alpha_t^2}{4\eta}\tilde{\sigma}^2.\end{aligned}$$

Therefore, choosing $\gamma_t = L_2L_1\alpha_t$, the upper bound of the error of ϕ is

$$\begin{aligned}E[\|\phi^{(t+1)} - \phi^*(\theta^{(t+1)})\|^2] \le\ & (1 + 4L_2L_1\alpha_t + \frac{\eta L_5\tilde{C}^2}{2}\alpha_t^2)E[\|\phi^{(t)} - \phi^*(\theta^{(t)})\|^2] \\ &+ (L_4^2 + \frac{L_4}{4L_2\alpha_t} + \frac{L_4\theta}{2\eta})\alpha_t^2 E[\|h_{\mathrm{d}}^t\|^2] + (L_4^2 + \frac{L_5}{2\eta})\alpha_t^2\tilde{\sigma}^2.\end{aligned}$$

Define the Lyapunov function as $\mathbb{V}^{(t)} := \mathcal{L}_{\text{defend}}(\theta^{(t)}) + \frac{L_2}{L_4}\|\phi^{(t)} - \phi^*(\theta^{(t)})\|$, then $E[\mathbb{V}^{(t+1)}] - E[\mathbb{V}^{(t)}] = E[\mathcal{L}_{\text{defend}}(\theta^{(t+1)})] - E[\mathcal{L}_{\text{defend}}(\theta^{(t)})] + \frac{L_2}{L_4}(E[\|\phi^{(t+1)} - \phi^*(\theta^{(t+1)})\|^2] - E[\|\phi^{(t)} - \phi^*(\theta^{(t)})\|^2])$. Using the upper bounds

of $E[\mathcal{L}_{\text{defend}}(\theta^{(t+1)})] - E[\mathcal{L}_{\text{defend}}(\theta^{(t)})]$ and $E[\|\phi^{(t+1)} - \phi^*(\theta^{(t+1)})\|^2]$, the difference between two Lyapunov functions is bounded as follows

$$\begin{aligned}
E[\mathbb{V}^{(t+1)}] - E[\mathbb{V}^{(t)}] \leq & -\frac{\alpha_t}{2} E[\|\nabla \mathcal{L}_{\text{defend}}(\theta^{(t)})\|^2] - (\frac{\alpha_t}{2} - \frac{L_3\alpha_t^2}{2} - L_2L_1\alpha_t^2 - \frac{\alpha_t}{4} - \frac{L_2}{L_4}\frac{L_5}{2\eta}) E[\|h_{\text{d}}^t\|^2] \\
& + \frac{L_2}{L_4}((1 + 4L_2L_1\alpha_t + \frac{\eta L_5 \tilde{C}^2}{2}\alpha_t^2)(1 - \frac{2\mu L_1}{\mu + L_1}\beta_t) - 1) E[\|\phi^{(t)} - \phi^*(\theta^{(t)})\|^2] \\
& + \frac{L_2}{L_4}(1 + 5L_2L_1\alpha_t + \frac{\eta L_5 \tilde{C}^2}{4}\alpha_t^2)\beta_t^2\sigma_2^2 + \alpha_t b_t^2 + (\frac{L_3}{2} + L_2L_1 + \frac{L_5L_2}{2\eta L_1})\alpha_t^2\tilde{\sigma}^2.
\end{aligned}$$

To ensure the decrease of $\mathbb{V}^{(t)}$, the following conditions must be met:

$$\begin{aligned}
\alpha_t &\leq \frac{1}{2L_3 + 4L_2L_4 + \frac{2L_2L_5}{L_4\eta}}, \\
\frac{2\mu L_1}{\mu + L_1}\beta_t &\geq 2L_2L_4\alpha_t + \frac{\eta L_5\tilde{C}^2}{2}\alpha_t^2, \\
\beta_t &\leq \frac{2}{\mu + L_1}.
\end{aligned}$$

To satisfy these conditions, we choose the stepsizes as defined in Theorem 3, then the upper bound of the difference between the two Lyapunov functions is simplified as

$$E[\mathbb{V}^{(t+1)}] - E[\mathbb{V}^{(t)}] \leq -\frac{\alpha_t}{2} E[\|\nabla \mathcal{L}_{\text{defend}}(\theta^{(t)})\|^2] + c_1\alpha_t^2\sigma_2^2 + \alpha_t b_t^2 + c_2\alpha_t^2\tilde{\sigma}^2,$$

where constants c_1 and c_2 are defined as

$$\begin{aligned}
c_1 &= \frac{L_2}{L_1}(1 + 5L_2L_1\bar{\alpha}_1 + \frac{\eta L_5\tilde{C}^2}{4}\bar{\alpha}_1^2)(\frac{8L_2L_1 + 2\eta L_5\tilde{C}^2\bar{\alpha}_1}{4\frac{2\mu L_1}{\mu + L_1}})^2, \\
c_2 &= (\frac{L_3}{2} + L_2L_1 + \frac{L_5L_2}{2\eta L_4}).
\end{aligned}$$

Rewrite the inequality as $\frac{\alpha_t}{2} E[\|\nabla \mathcal{L}_{\text{defend}}(\theta^{(t)})\|^2] \leq E[\mathbb{V}^{(t)}] - E[\mathbb{V}^{(t+1)}] + c_1\alpha_t^2\sigma_2^2 + \alpha_t b_t^2 + c_2\alpha_t^2\tilde{\sigma}^2$ and apply telescoping from $t = 1$ to $t = T$, we have

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T} E[\|\nabla \mathcal{L}_{\text{defend}}(\theta^{(t)})\|^2] &\leq \frac{\mathbb{V}^{(1)} + \sum_{t=1}^{T}(c_1\alpha_t^2\sigma_2^2 + \alpha_t b_t^2 + c_2\alpha_t^2\tilde{\sigma}^2)}{\frac{1}{2}\sum_{t=1}^{T}\alpha_t} \\
&\leq \frac{2\mathbb{V}^{(1)}}{T\min\{\bar{\alpha}_1, \bar{\alpha}_2\}} + \frac{2\mathbb{V}^{(1)}}{\alpha\sqrt{T}} + \frac{2c_1\alpha}{\sqrt{T}}\sigma_2^2 + 2b_t^2 + \frac{2c_2\alpha}{\sqrt{T}}\tilde{\sigma}^2.
\end{aligned}$$

Since $\frac{2\mathbb{V}^{(1)}}{T\min\{\bar{\alpha}_1, \bar{\alpha}_2\}}$ is $\mathcal{O}(1/T)$, $\frac{2\mathbb{V}^{(1)}}{\alpha\sqrt{T}}$, $\frac{2c_1\alpha}{\sqrt{T}}\sigma_2^2$ and $\frac{2c_2\alpha}{\sqrt{T}}\tilde{\sigma}^2$ are $\mathcal{O}(1/\sqrt{T})$, we have $\frac{1}{T}\sum_{t=1}^{T} E[\|\nabla \mathcal{L}_{\text{defend}}(\theta^{(t)})\|^2] = \mathcal{O}(1/\sqrt{T})$.

A.4. Proof of Theorem 4

Notably, the parameter change from θ_k to θ'_k is related to the training loss of the shadow model. We therefore examine how changes in the training dataset (i.e., $Z_k \to Z'_k$) affect the training loss. Specifically, the parameters of the approximated shadow model on $D_k^{S'}$ can be expressed as

$$\theta'_k = \arg\min_{\theta} \; L_{\mathcal{T}}(\theta; D_k^S) + \sum_{z' \in Z'_k} l(\theta; z') \; - \sum_{z \in Z_k} l(\theta; z),$$

where $L_{\mathcal{T}}$ is the loss function of the reference shadow model f_k^{ref}, and l is the loss for each sample.

Instead of training the shadow model from scratch, we approximate θ'_k by introducing the influence of the perturbed samples Z'_k and removing the influence of the original samples Z_k, which is captured by a small parameter ϵ:

$$\theta'_{k,\epsilon} = \arg\min_{\theta} L_{\mathcal{T}}(\theta; D^S) + \epsilon \sum_{z' \in Z'_k} l(\theta; z') - \epsilon \sum_{z \in Z_k} l(\theta; z).$$

To solve for $\theta'_{k,\epsilon}$, we derive the optimality condition:

$$0 = \nabla L_{\mathcal{T}}(\theta'_{k,\epsilon}; D^S) + \epsilon \sum_{z' \in Z'_k} \nabla l(\theta'_{k,\epsilon}; z') - \epsilon \sum_{z \in Z_k} \nabla l(\theta'_{k,\epsilon}; z).$$

If ϵ is sufficiently small, then $\theta'_{k,\epsilon} \to \theta_k$. Therefore, we can perform a first-order Taylor expansion at θ_k of the right-hand side:

$$0 \approx \nabla L_{\mathcal{T}}(\theta_k; D^S) + \epsilon \sum_{z' \in Z'_k} \nabla l(\theta_k; z') - \epsilon \sum_{z \in Z} \nabla l(\theta_k; z) + (\theta'_{k,\epsilon} - \theta_k)[\nabla^2 L(\theta_k; D_k^S) + \epsilon \sum_{z' \in Z'_k} \nabla^2 l(\theta_k; z') - \epsilon \sum_{z \in Z} \nabla^2 l(\theta_k; z)].$$

Since θ_k minimizes $L_{\mathcal{T}}(\theta; D_k^S)$, we have $\nabla L_{\mathcal{T}}(\theta_k; D^S) = 0$. Solving for $\theta'_{k,\epsilon} - \theta_k$, we get:

$$\theta'_{k,\epsilon} - \theta_k \approx -[\nabla^2 L_{\mathcal{T}}(\theta_k; D^S) + \epsilon \sum_{z' \in Z'_k} \nabla^2 l(\theta_k; z') - \epsilon \sum_{z \in Z_k} \nabla^2 l(\theta_k; z)]^{-1} [\epsilon \sum_{z' \in Z'_k} \nabla_\theta l(\theta_k; z') - \epsilon \sum_{z \in Z_k} \nabla l(\theta_k; z)],$$

where $\nabla^2 L_{\mathcal{T}}(\theta_k; D_k^S)$ is the Hessian H_{θ_k}. Dropping all terms in $\mathcal{O}(\epsilon)$, the equation simplifies to:

$$\theta'_{k,\epsilon} - \theta_k \approx -H_{\theta_k}^{-1} [\sum_{z' \in Z'_k} \nabla_\theta l(\theta_k; z') - \epsilon \sum_{z \in Z_k} \nabla l(\theta_k; z)] \epsilon.$$

Since θ'_k and $\theta'_{k,\epsilon}$ are equivalent when $\epsilon = \frac{1}{|D_k^S|}$, we can linearly approximate the parameter change $\Delta(Z_k, Z'_k)$ as

$$\Delta(Z_k, Z'_k) = \frac{1}{|D_k^S|} \frac{d\theta'_{k,\epsilon}}{d\epsilon}\bigg|_{\epsilon=0} = -\frac{1}{|D_k^S|} H_{\theta_k}^{-1} [\sum_{z' \in Z'_k} \nabla_\theta l(\theta_k; z') - \epsilon \sum_{z \in Z_k} \nabla l(\theta_k; z)].$$

Appendix B: Complexity Analysis Details

The time complexity of D-S^{2}HARE mainly involves three components: training reference shadow models, estimating parameters of approximated shadow models and executing the attack-defense arms race iteratively. Suppose the number of shadow model and target model parameters is p, the size of the shadow model training data is $|D^S|$, the size of the target model training data is $|D|$. T_s is the number of training iterations for target and shadow models, T_a is the number of training iterations for the attack model, and T is the number of iterations of the attack-defense arms race.

- *Training reference shadow models*. Suppose the input dimension of reference shadow model is d. For simplicity, suppose that reference shadow models are neural networks with L hidden layers, each containing l neurons. The output dimension of reference shadow models is c. The computational complexity of training a reference shadow model is linear in the number of model parameters, which is given by $\mathcal{O}((dl+l+(L-1)(l^2+l)+lc+c)|D^S|T_s)$, where $(dl+l+(L-1)(l^2+l)+lc+c)$ represents the number of model parameters p, $|D^S|$ is the size of training dataset and T_s is the number of iterations. Therefore, the time complexity of training K reference shadow models is $\mathcal{O}(Kp|D^S|T_s)$.
- *Estimating parameters of approximated shadow models*. We estimate the parameters of approximated shadow models using Eq. (8), which involves calculating the inverse Hessian and the gradient terms. Following Koh and Liang (2017), the inverse Hessian can be approximated in $\mathcal{O}(p|B|)$ time, where $|B|$ denotes the number of training samples uniformly sampled from the dataset for approximation. Calculating the gradient terms requires $\mathcal{O}(p|Z|)$ time, where $|Z|$ represents the number of perturbed samples in the training dataset for the reference shadow model. Since $|Z|$ is typically small, the $\mathcal{O}(p|Z|)$ term can be ignored. Thus, the time complexity of estimating parameters for $N-K$ approximated shadow models simplifies to $\mathcal{O}((N-K)p|B|)$.
- *Executing the attack-defense arm-race*. The attack-defense arms race framework involves two main processes: Responsive CPI attack model refinement and target model update. Similar to the training of reference shadow models, training the Responsive CPI attack model over N training samples requires $\mathcal{O}(pNT_a)$, and training the target model over a training dataset with $|D|$ samples requires $\mathcal{O}(p|D|T_s)$. These processes are executed over T iterations, yielding a total complexity of $\mathcal{O}(T(pNT_a+p|D|T_s))$.

Summing the three components, the overall time complexity of D-S^{2}HARE is $\mathcal{O}(Kp|D^S|T_s+(N-K)p|B|+T(pNT_a+p|D|T_s))$.

Appendix C: Implementation Details

Implementation details of trade-Off parameters. We provide additional details regarding the confidentiality-utility trade-off parameters used for various defense methods when evaluating their impact on the performance of confidentiality preservation and utility retention in Section 5.2. Specifically, for our method, the red points from left to right in the figure correspond to the performance under the trade-off hyperparameter $\lambda \in \{0.1, 0.3, 0.5, 0.7\}$. For Noisy Label, we flipped 10%, 20%, 30%, 40% of the labels for the target task, with the blue points in the figure arranged from right to left accordingly. In DP-SGD, we set the privacy budget ϵ used in its original paper (Abadi et al. 2016) to the values in $\{0.1, 0.5, 1, 2\}$, corresponding to the green points from left to right in the figure. For Resampling, the purple points from right to left reflect down-sampling 20%, 40%, 60%, 80% of training samples with confidential property attribute $X_p = 1$. For Adversarial Defense, the orange points from right to left in the figure represent the performance under the trade-off parameter between model optimization and defense in its original paper (Stock et al. 2024), with values in $\{0.2, 0.4, 0.6, 0.8\}$.

Implementation of robustness checks. When implementing unresponsive CPI attacks, we trained 500 shadow models. For logistic regression target model, we set the learning rate to 0.1 and L2 regularization coefficient to 1.0. The kernel width σ^2 was set to 0.015 when estimating training sample weight r_i. The trade-off parameters in robustness checks were kept the same as those in the main experiments, and all other implementation details followed the settings described in Section 5.1.

Appendix D: Sensitivity Analysis

We conducted a sensitivity analysis to assess how the performance of our method is influenced by key hyperparameters: the number of reference shadow models K used to simulate adversaries, and the perturbation budget δ used in the approximation strategy. For this analysis, we kept all other parameters consistent with the implementation details outlined in Section 5.1. We evaluated our method on the binary classification task of loan default rate.

Number of reference shadow models. We varied the number of reference shadow models K from 50 to 400, while keeping the total number of shadow models (reference and approximated) fixed at 500. The results can be found in Table 3. When the number of reference shadow models is too small (e.g., $K = 50$), the simulated adversary's ability to effectively infer the confidential property is significantly compromised. This limitation undermines our defense method's CPI-attack robustness, as it is not equipped with necessary defenses against adversaries with effective inference capabilities. Conversely, when the number of reference shadow models is too large (e.g., $K = 400$), our method's CPI-attack robustness improves. However, this enhancement comes at the cost of efficiency, as the runtime increases due to the need to train more reference shadow models from scratch.

Table 3 Sensitivity analysis of the number of reference shadow models.

	White-box			Black-box		
	Attack success rate	Target accuracy	Run time	Attack success rate	Target accuracy	Run time
$K = 50$	0.60	0.898	119.24s	0.53	0.897	225.32s
$K = 100$	0.55	0.897	141.86s	0.52	0.896	258.81s
$K = 200$	0.51	0.894	258.19s	0.51	0.896	354.68s
$K = 300$	0.51	0.894	374.67s	0.50	0.895	458.84s
$K = 400$	0.50	0.891	477.18s	0.50	0.893	575.51s

Perturbation budget. We also varied the perturbation budget δ used in our approximation strategy from 100 to 1500. Table 4 shows the results. When the perturbation budget is too small, the approximated shadow models are nearly identical to the reference shadow models, limiting the effectiveness in simulating adversaries. Conversely, when the perturbation budget is too large, the approximation error of the model parameters increases, also hindering the effectiveness of simulating adversaries. This ineffectiveness in simulating adversaries, in turn, undermines our method's ability to defend against CPI attacks. As a result, our model struggles to maintain confidentiality when the perturbation budget is too small or too large.

Table 4 Sensitivity analysis of perturbation budget.

	White-box		Black-box	
	Attack success rate	Target accuracy	Attack success rate	Target accuracy
$\delta = 100$	0.67	0.896	0.63	0.902
$\delta = 200$	0.58	0.895	0.57	0.895
$\delta = 500$	0.50	0.892	0.50	0.890
$\delta = 1000$	0.55	0.897	0.52	0.896
$\delta = 1500$	0.66	0.898	0.60	0.900

Appendix E: Notations

Table 5 Summary of notations.

Notation	Description
$D = (X, Y)$	Target model training dataset with attribute matrix X and target task label Y
X_p	Vector of the confidential property-related attribute p
D^{adv}	Adversary's auxiliary dataset
f_θ, f_i^S	Target model with parameters θ, shadow model
$\mathcal{F}_\theta$, $\mathcal{F}_i^S$	Target model information, shadow model information
P, P_i^S	Confidential property of target model, confidential property of shadow model f_i^S
h_ϕ	Attack model with parameters ϕ
$\mathcal{D}_S$, $\mathcal{D}_T$	Distribution of $\mathcal{F}_i^S$, $\mathcal{F}_\theta$
$\mathcal{L}_\mathcal{A}$, $\mathcal{L}_\mathcal{P}$	Training loss, test error of Responsive CPI attack model
$\mathcal{L}_\mathcal{T}$	Loss function of the target model
r_i	Training sample weight of Responsive CPI attack model
λ	Trade-off parameter
N, K	Number of all shadow models, number of reference shadow models
f_k^{ref}	Reference shadow model
D_k^S, $D_k^{S'}$	Reference shadow model training dataset, perturbed dataset
θ_k, θ_k'	Reference shadow model parameters, approximated shadow model parameters
Z_k, Z_k'	Original set of samples in D_k^S, perturbed set of samples in $D_k^{S'}$
δ	Perturbation budget