Mitigating Membership Inference Attacks by Self-Distillation Through a Novel Ensemble Architecture

Xinyu Tang, Saeed Mahloujifar, and Liwei Song, *Princeton University;*
Virat Shejwalkar, Milad Nasr, and Amir Houmansadr, *University of Massachusetts Amherst;* Prateek Mittal, *Princeton University*

https://www.usenix.org/conference/usenixsecurity22/presentation/tang

This paper is included in the Proceedings of the 31st USENIX Security Symposium.

August 10–12, 2022 • Boston, MA, USA

978-1-939133-31-1

Open access to the Proceedings of the 31st USENIX Security Symposium is sponsored by USENIX.

Mitigating Membership Inference Attacks by Self-Distillation Through a Novel Ensemble Architecture

Xinyu Tang[1] Saeed Mahloujifar[1] Liwei Song[1] Virat Shejwalkar[2] Milad Nasr[2]
Amir Houmansadr[2] Prateek Mittal[1]
[1]*Princeton University* [2]*University of Massachusetts Amherst*
{xinyut, sfar, liweis, pmittal}@princeton.edu, {vshejwalkar, milad, amir}@cs.umass.edu

Abstract

Membership inference attacks are a key measure to evaluate privacy leakage in machine learning (ML) models. It is important to train ML models that have high membership privacy while largely preserving their utility. In this work, we propose a new framework to train privacy-preserving models that induce similar behavior on member and non-member inputs to mitigate membership inference attacks. Our framework, called SELENA, has two major components. The first component and the core of our defense is a novel ensemble architecture for training. This architecture, which we call Split-AI, splits the training data into random subsets, and trains a model on each subset of the data. We use an adaptive inference strategy at test time: our ensemble architecture aggregates the outputs of only those models that did not contain the input sample in their training data. Our second component, Self-Distillation, (self-)distills the training dataset through our Split-AI ensemble, without using any external public datasets. We prove that our Split-AI architecture defends against a family of membership inference attacks, however, our defense does not provide provable guarantees against all possible attackers as opposed to differential privacy. This enables us to improve the utility of models compared to DP. Through extensive experiments on major benchmark datasets we show that SELENA presents a superior trade-off between (empirical) membership privacy and utility compared to the state of the art empirical privacy defenses. In particular, SELENA incurs no more than 3.9% drop in classification accuracy compared to the undefended model while reducing the membership inference attack advantage by a factor of up to 3.7 compared to MemGuard and a factor of up to 2.1 compared to adversarial regularization.

1 Introduction

Machine learning has achieved tremendous success in many areas, but it requires access to data that may be sensitive. Recent work has shown that machine learning models are prone to memorizing sensitive information of training data incurring serious privacy risks [3,4,12,13,38,41,42]. Even if the model provider is trusted and only provides query services via an API, i.e., black-box model access in which only prediction vectors are available, private information can still be obtained by attackers. The *membership inference attack (MIA)* is one such threat in which an adversary tries to identify whether a target sample was used to train the target machine learning model or not based on model behavior [41]. MIAs pose a severe privacy threat by revealing private information about the training data. For example, knowing the victim's presence in the hospital health analytic training set reveals that the victim was once a patient in the hospital.

Shokri et al. [41] conducted MIAs against machine learning models in the black-box manner. They formalize the attack as a binary classification task and utilize a neural network (NN) model along with shadow training technique to distinguish members of training set from non-members. Following this work, many MIAs have been proposed which can be divided into two categories: direct attacks [30,31,43,44,51,52], which directly query the target sample and typically utilize only a single query; indirect attacks [8, 26, 27], which query samples that are in the neighborhood of the target sample to infer membership, and typically utilize multiple queries. The research community has further extended the MIA to federated settings [29, 31] and generative models [14]. MIAs have also provided a foundation for more advanced data extraction attacks [4] and for benchmarking privacy-preserving mechanisms [19, 33].

The effectiveness of MIAs and the resulting privacy threat have motivated the research community to design several defense mechanisms against these attacks [1, 21, 30, 40]. As MIAs distinguish members and non-members of the target model based on the difference in model's behavior on members, defense mechanisms need to enforce similar model behavior on members and non-members. There exist two main categories of membership inference defenses, as shown in Table 1: techniques that offer *provable privacy*, and defenses that offer *empirical membership privacy*. The first category

mainly uses differential privacy mechanisms [1, 28, 49] to be able to provide a *provable privacy guarantee* for all inputs. However, the use of DP (e.g., in DP-SGD [1]) is shown to significantly reduce the utility of the underlying models in many machine learning tasks (see Section 7.3 and Section 8). This has motivated the second category of membership inference defenses, where privacy is empirically evaluated through practical MIAs with the aim of preserving model utility. Our work in this paper falls in the second category, and as we will show, our technique offers a superior trade-off between MIA protection and model utility compared to the state-of-the-art empirical privacy defenses [21, 30, 40] (see Section 6 for more details).

	Low utility	High utility
Provable privacy	DP-based: DP-SGD [1]	Desired (No method achieves this goal so far)
Empirical membership privacy	Not considered	Adversarial Regularization [30], MemGuard [21], SELENA(Our work)

Table 1: Two categories of membership inference defenses: provable privacy with low utility vs. empirical membership privacy with high utility.

Our Framework. In this paper, we introduce a novel empirical MIA defense framework, called SELENA,[1] whose goal is to protect against practical black-box MIAs while also achieving high classification accuracy. Our framework consists of two core components: *Split-AI* and *Self-Distillation*.

Split Adaptive Inference Ensemble (Split-AI): Our first component, called Split-AI, is proposed to enable the model to have similar behavior on members and non-members. We obtain this goal by training multiple models (called sub-models) with random subsets from the training set. While such ensemble architectures have been considered in different ML contexts, our framework's novelty lies in the particular adaptive approach it uses to respond to the queries. The key intuition is that for a training sample, if one of sub-models is not trained with it, this sub-model will have similar behavior on this training sample and other non-members. We use this intuition in our adaptive inference strategy. When the queried sample is in the training set, the adaptive inference procedure will only call the sub-models that have not used the queried sample in their training set. When the queried sample is not in the training set, we query a particular subset of sub-models (as explained later in Section 4). Our approach provides an intuitive foundation for membership privacy: no matter if the queried sample is a member or a non-member, the adaptive inference will always use only those sub-models which have not used this sample for their training; this ensures the membership privacy which we demonstrate through a formal analysis.

[1] SELf ENsemble Architecture.

Self-Distillation: Our Split-AI shows a promising performance against the traditional type of MIA, i.e., the direct single-query attack [43, 44, 51, 52]. However, it falls short in protecting against recent adaptive MIAs, which work by crafting multiple, particularly-fabricated queries [8, 26]. Moreover, Split-AI has a high computational overhead, as it needs to search for each queried sample within the training set and perform inference on multiple sub-models. To protect Split-AI against adaptive attacks and to reduce its computational overhead, we use the second component of our framework, which we call Self-Distillation. Our Self-Distillation component performs a novel form of knowledge transfer on the model created by Split-AI to produce a final *protected model*. Specifically, it first queries Split-AI with its exact training samples to get their corresponding prediction vectors. Then, it uses these prediction vectors as the soft labels of the training set to train the protected model, which will be used for inference. During the inference stage, the protected model only needs to perform a single computation for each queried sample, therefore it has a much lower computation overhead compared to Split-AI's model. Furthermore, the protected model protects not only against traditional direct single-query MIA attacks, but also against adaptive MIA attacks as shown in our analysis in Section 6. Note that, unlike conventional uses of distillation for membership privacy [40], our Self-Distillation component does not need a public dataset for knowledge transfer as it uses its own training dataset for (self-)distillation.[2]

Evaluation. We evaluate our SELENA on three benchmark datasets (CIFAR100, Purchase100, Texas100) and compare it with existing defenses [21, 30, 43] rigorously using two types of existing attacks and one type of adaptive attack. (1) We first analyze our defense against *direct single-query attacks*, which have been typically used in most previous MI attacks and defenses. (2) We next evaluate our framework against *label-only attacks*, which infer membership information only based on labels and hence simply obfuscating prediction confidence vector cannot protect against such attacks. (3) We finally study *adaptive attacks*, which are tailored to our defense mechanism. Overall, SELENA achieves a better trade-off between the utility, i.e., classification accuracy, and the empirical membership privacy without requiring additional public data. For utility, SELENA incurs only a little drop in classification accuracy compared to the undefended model (no more than 3.9%), and outperforms adversarial regularization [30] by up to 7.0% (on Texas100). For membership privacy risks, SELENA reduces the MIA advantage over a random guess by a factor of up to 4.0 compared to the undefended model, a factor of up to 3.7 compared to MemGuard [21] and a factor of up to 2.1 compared to adversarial regularization [30]. Unlike DP-SGD that offers a provable privacy guarantee, our approach only provides an empirical membership inference defense (similar to MemGuard and adversarial regularization). However, our

[2] Note that our usage of the term self-distillation is different from what Zhang et al. [53] refer to as self-distillation.

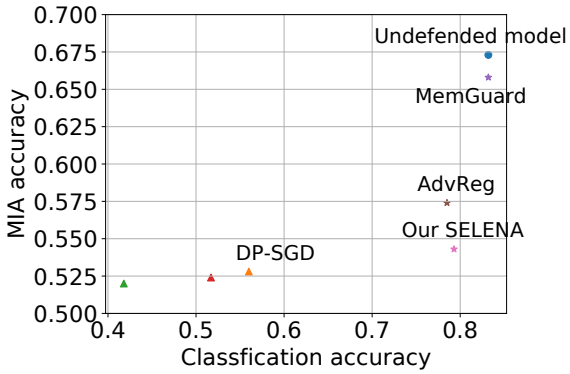


Figure 1: Comparison of our method with undefended model, DP-SGD [1] ($\varepsilon = 4$), MemGuard [21] and adversarial regularization [30] with respect to classification accuracy and MIA accuracy on Purchase100 dataset. Our SELENA outperforms adversarial regularization in both classification and MIA accuracy. Our SELENA significantly reduces MIA accuracy compared to undefended model and MemGuard while incurring little classification accuracy drop. Our SELENA achieves much higher classification accuracy compared to DP-SGD while only incurs little additional practical membership risks.

evaluation shows that SELENA achieves a much better utility than DP-SGD (See Figure 1).

In summary, we propose a membership inference defense to achieve high classification accuracy and highly mitigate practical MIAs. Our key contributions are as follows:

- We propose Split-AI as the first component of our framework that enforces the model to have a similar behavior on members and non-members while maintaining a good classification accuracy using sub-models trained on overlapping subsets of data and an adaptive inference strategy. We further prove that the direct single-query attack cannot achieve higher attack accuracy than a random guess against this component.
- We introduce Self-Distillation of the training set as the second component of our framework to overcome the limitations of Split-AI while largely preserving its defense abilities without relying on an additional public dataset.
- We systematically evaluate our framework on three benchmark datasets including Purchase100, Texas100, and CIFAR100 against a suite of MIAs including direct single-query attacks, label-only (indirect multi-query) attacks and adaptive attacks to show that our framework outperforms prior defenses.

2 Preliminaries and Problem Formulation

In this section, we introduce the machine learning concepts and notation relevant to our work, as well as our threat model and design goals.

2.1 ML Preliminaries and Notation

In this paper, we consider supervised machine learning for classification. Let $F_\theta : \mathbb{R}^d \mapsto \mathbb{R}^k$ be a classification model with d input features and k classes, which is parameterized by θ. For a given example $\mathbf{z} = (\mathbf{x}, y)$, $F_\theta(\mathbf{x})$ is the classifier's confidence vector for k classes and the predicted label is the corresponding class which has the largest confidence score, i.e., $\hat{y} = \text{argmax}_i F_\theta(\mathbf{x})$.

The goal of supervised machine learning is to learn the relationship between training data and labels and generalize this ability to unseen data. The model learns this relationship by minimizing the predicted loss across the training set D_{tr}:

$$\min_\theta \frac{1}{|D_{tr}|} \sum_{\mathbf{z} \in D_{tr}} l(F_\theta, \mathbf{z})$$

Here $|D_{tr}|$ is the size of the training set and $l(F_\theta, \mathbf{z})$ is the loss function. When clear from the context, we use F, instead of F_θ, to denote the target model.

2.2 Threat Model

Black-box attack: In this paper, we follow previous defenses [21,30] and assume the attacker has black-box access to the target model, i.e., the attacker can only make queries to the model provider and obtain corresponding prediction vectors or predicted labels, instead of having access to target model's parameters. Therefore, the adversary can perform standard black-box attacks, in particular the *direct single-query* attacks, which directly query the target sample *one time* and is the typical benchmarking technique, and the *label-only attacks*, which *make multiple queries* for a single target sample and exploit predicted label information. We also introduce a third type of black-box attack which is an adaptive attack tailored to our system. See Section 3 for a detailed explanation of direct single-query attacks and label-only attacks and Section 5 for the adaptive attacks.

Partial knowledge of membership for members: Like previous defenses [30,43], we assume the adversary knows a small ratio of samples from the training set, i.e., it knows some members. The goal of the adversary is to identify any other member sample.

2.3 Design Goals

In this paper, we aim to overcome the limitations of existing membership inference defenses [21,30,40], which estimate the membership risk through practical MIAs: none of these defenses are able to provide sufficient MIA protection and high utility simultaneously in the absence of public datasets.

Low MIA accuracy: Our goal is to design a practical defense against MIAs. We will evaluate our defense in a systematic and rigorous manner to ensure that it achieves low MIA accuracy (i.e., high membership privacy) across a broad class of attacks, instead of only one specific family of attacks.

High classification accuracy: We aim to protect membership privacy without significantly decreasing the classification accuracy (model utility).

No additional public data required for defense: Some prior works [34, 40] have proposed to preserve membership privacy by knowledge distillation using publicly available datasets. However, this is a limiting assumption since public datasets may not be available in many real-world ML training scenarios such as healthcare data. In this paper, we consider a more realistic scenario, where the model provider does not have access to external public dataset.

3 Existing Attacks and Defenses

Next, we overview prior MI attacks and MI defenses.

3.1 Membership Inference Attacks (MIAs)

MIAs can utilize the prediction vector as a feature using a neural-network-based model, called *NN-based attacks*, or can compute a range of custom metrics (such as correctness, confidence, entropy) over the prediction vector to infer membership, called *metric-based attacks*. These attacks can be mounted either by knowing a subset of the training set [30] or by knowing a dataset from the same distribution of the training set and constructing shadow models [41].

Let us denote D_{tr} as the training set for the target model, i.e., members, and D_{te} as the test set, i.e., non-members. D^A_{tr} and D^A_{te} are, respectively, the sets of members and non-members that the attacker knows. $I(\mathbf{x}, y, F(\mathbf{x}))$ is the binary membership inference classifier in the range of $\{0,1\}$ which codes members as 1, and non-members as 0. The literature typically measures MIA efficacy as the attack accuracy:

$$\frac{\sum_{(\mathbf{x},y)\in D_{tr}\setminus D^A_{tr}} I(\mathbf{x},y,F(\mathbf{x})) + \sum_{(\mathbf{x},y)\in D_{te}\setminus D^A_{te}} (1-I(\mathbf{x},y,F(\mathbf{x})))}{|D_{tr}\setminus D^A_{tr}| + |D_{te}\setminus D^A_{te}|}$$

In most previous attacks [30, 41, 43, 52], the number of members and non-members used to train and evaluate the attack model are the same. With this approach, the prior probability of a sample being either a member or a non-member is 50% (corresponding to a random guess).

Next, we summarize black-box MIAs in the following two categories: **direct** attacks and **indirect** attacks.

Direct single-query attacks: Most existing MIAs directly query the target sample and utilize the resulting prediction vector. Since ML models typically have only one output for each queried sample, just a single query is sufficient.

NN-based attack [30, 41]: The attacker can use the prediction vectors from the target model along with the one-hot encoded ground truth labels as inputs and build a NN model [30] I_{NN} for the membership inference task.

Correctness-based attack [52]: The generalization gap (i.e., the difference between training accuracy and test accuracy) is a simple baseline for MIA, as samples with correct prediction are more likely to be training members.

$$I_{\text{corr}}(F(\mathbf{x}),y) = \mathbb{1}\{\operatorname*{argmax}_i F(\mathbf{x})_i = y\}$$

Confidence-based attack [43, 44, 51]: Prediction confidence corresponding to training samples $F(\mathbf{x})_y$ is typically higher than prediction confidence for test samples. Therefore, confidence-based attack will only regard the queried sample as a member when the prediction confidence is higher than either a class-dependent threshold τ_y or a class-independent threshold τ.

$$I_{\text{conf}}(F(\mathbf{x}),y) = \mathbb{1}\{F(\mathbf{x})_y \geq \tau_{(y)}\}$$

Entropy-based attack [41, 43]: The prediction entropy of a training sample is typically lower than the prediction entropy of a test sample. Therefore, entropy-based attack will only regard the queried sample as a member when the prediction entropy is lower than either a class-dependent threshold τ_y or a class-independent threshold τ.

$$I_{\text{entr}}(F(\mathbf{x}),y) = \mathbb{1}\{-\sum_i F(\mathbf{x})_i \log(F(\mathbf{x})_i) \leq \tau_{(y)}\}$$

Modified entropy-based attack [43]: Song et al. [43] propose the modified prediction entropy metric which combines the information in the entropy metric and ground truth labels:

$$\begin{aligned}\text{Mentr}(F(\mathbf{x}),y) = &-(1-F(\mathbf{x})_y)\log(F(\mathbf{x})_y) \\ &-\sum_{i\neq y} F(\mathbf{x})_i \log(1-F(\mathbf{x})_i)\end{aligned}$$

Training samples typically have lower values of modified entropy metric than test samples and either a class-dependent threshold τ_y or a class-independent threshold τ attack is applied to infer membership.

$$I_{\text{Mentr}}(F(\mathbf{x}),y) = \mathbb{1}\{\text{Mentr}(F(\mathbf{x}),y) \leq \tau_{(y)}\}$$

Indirect multi-query attacks (label-only attacks): Long et al. [27] state that indirect attacks can make queries that are related to the target sample $\mathbf{x}$ to extract additional membership information, as a training sample influences the model prediction both on itself and other samples in its neighborhood. These indirect attacks usually make multiple queries for a single target sample [8, 26, 27]. For example, multi-query *label-only attacks* leverage the predicted label of the queried data as features, and are thus immune to defenses that only

obfuscate prediction confidences, e.g., MemGuard [21]. The key idea in label-only attacks is that the model should be more likely to correctly classify the samples around the training data than the samples around test data, i.e., members are more likely to exhibit high robustness than non-members [8, 26]. Simply obfuscating a model's confidence scores cannot hide label information to defend against such label-only attacks.

Boundary estimation attacks [8, 26]: As the target model is more likely to correctly classify samples around training samples than those around test samples, the distance to classification boundary for the training samples should be larger than that for the test samples. An attacker can either leverage techniques for finding adversarial examples under the black-box assumption [2, 7] or add noise to find the adversarial examples that change the predicted label with minimum perturbation. Such attacks should not achieve higher attack accuracy than the white-box adversarial example attacks such as Carlini-Wagner attack [5], which have full access to the model parameters and can find the adversarial example with the least distance for each target sample.

Data augmentation attacks [8]: In computer vision tasks, data augmentation techniques based on translation, rotation, and flipping help improve test accuracy. However, such data augmentation techniques pose a privacy threat: the target model is more likely to correctly classify the augmented data of training samples. An attacker can query the augmented data of the target sample and use the percentage of correct predictions to identify membership of the target sample.

3.2 Existing Defenses

Multiple defenses have been proposed to mitigate MIAs. We summarize them below. Section 8 gives a more comprehensive summary of prior defenses.

Adversarial Regularization [30]: Nasr et al. [30] include the estimation of membership threat in the training process of the ML model. They optimize a min-max game to train a privacy-preserving target classifier, which aims to reduce the prediction loss while also minimizing the MIA accuracy.

Early Stopping [6, 43]: During the training process, the model may learn too much information in the training samples thus the difference between its behavior on members and non-members becomes larger and larger, and the model becomes more vulnerable to MIAs. Therefore, early stopping, which is a general technique to prevent model overfitting by stopping model training before the whole training process ends, can mitigate MIA accuracy with a sacrifice of model utility. Song et al. [43] find that adversarial regularization is not better than early stopping [6] when evaluated by a suite of attacks including both NN-based attacks and metric-based attacks. They recommend that any defense that trades off a reduction in MIA accuracy at the cost of a reduction in utility should be compared with early stopping as a baseline.

MemGuard [21]: Jia et al. [21] obfuscate the prediction vector with a well-designed noise vector using the perspective of adversarial examples to confuse the membership inference classifier. Since MemGuard doesn't change prediction results, and only obfuscates confidence information, it maintains the original classification accuracy of the undefended model. Song et al. [43] show that MemGuard lacks consideration of strategic adversaries. Though resistant to NN-based attack, MemGuard underestimates the threat of metric-based attacks.

DP-based defenses: Differential privacy [9] is a formal framework that provides a rigorous privacy guarantee. In machine learning, DP-based defenses, such as DP-SGD [1], add noise to the training process of a classifier. However, it is challenging to perform machine learning with differential privacy while achieving acceptable utility loss and privacy guarantees [19, 37] (See Section 8).

4 Our Defense Architecture

In this section, we first present an overview of our defense framework and then describe the two key framework components: Split-AI and Self-Distillation.

4.1 Overview

MIAs aim to distinguish members and non-members of the private training data of a model. These attacks use the fact that the trained model has a different behavior on member and non-member data. This difference in behavior can appear in different forms, for example, the accuracy of model might be different on members and non-members [39], or the confidence might be higher on member inputs [43, 44, 51]. Similarly, the model might be more likely to correctly classify the samples around the member examples compared to those around non-member examples [8, 26]. MIAs leverage these differences to obtain an attack advantage that is better than a random guess even in the black-box setting. To mitigate these differences, we propose a framework to defend against MIAs by training multiple sub-models using subsets from whole training set and introducing a specific adaptive inference technique that exploits the following intuition: *if a training sample is not used to train a sub-model, this sub-model will have similar behavior on this training sample and non-members.* Section 6.2 shows the advantage of our defense in improving the trade-off between membership privacy and utility, which is based on this intuition, over existing membership inference defenses (MemGuard [21] in Section 6.2.2 and adversarial regularization [30] in Section 6.2.3).

Based on this intuition, we propose a framework, SELENA, composed of two components to defend against MIAs. The first component, which we call *Split-AI*, trains an ensemble of K sub-models with overlapping subsets of the training dataset. The constraint for each subset is as follows: for a given training sample, there are L sub-models which are not trained with

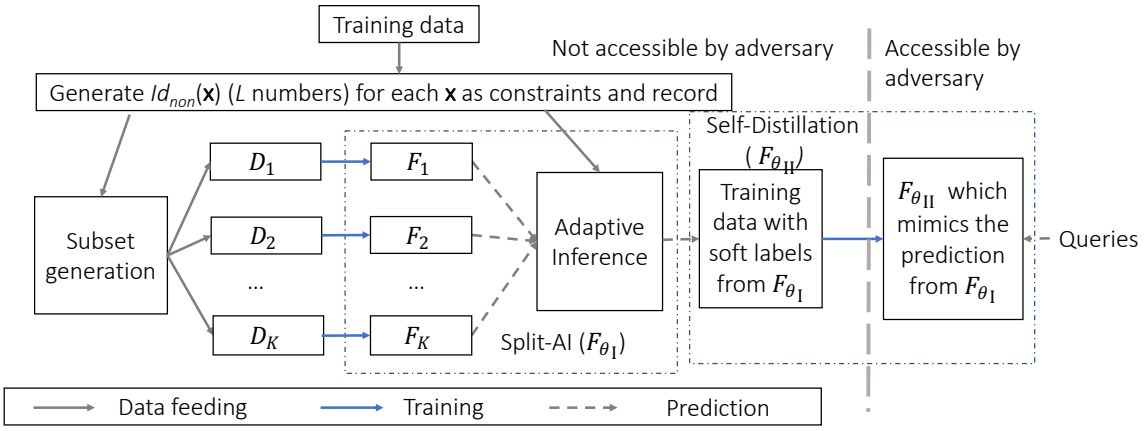


Figure 2: Our end-to-end defense framework with the Split-AI and Self-Distillation components.

this training sample, and therefore, they will behave similarly on this training sample and the non-member samples. Split-AI applies adaptive inference for members and non-members: For a member sample, Split-AI computes L predictions of the L sub-models which are not trained with the member sample, and outputs the average of the L predictions as the final prediction. For a non-member sample, the adaptive inference randomly samples L sub-models from the K total sub-models, subject to a specific distribution, and returns the average of the L predictions on the non-member as the final prediction. We detail our algorithm and explain why it preserves membership privacy in Section 4.2.

The second component, which we call *Self-Distillation*, addresses the two weaknesses of Split-AI: its potential privacy risks due to replay/indirect attacks and its high computation overhead in inference. Specifically, the Self-Distillation component transfers the knowledge of the model obtained by Split-AI into a new model by using the soft labels of the training set from Split-AI. We call this Self-Distillation because it does not require any public dataset for distillation. As we will demonstrate, this protected model from Self-Distillation has similar classification performances as Split-AI with significantly lower computation overhead, and can protect against replay and indirect MIA attacks.

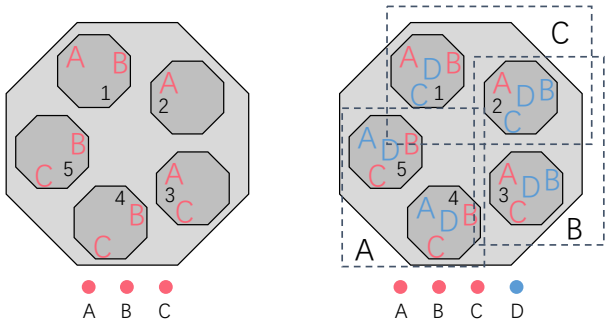
As we study the black-box MIAs, only the final prediction vectors or predicted labels of the protected model from Self-Distillation are available to the attacker. Figure 2 gives an overview of our defense, where we denote Split-AI as F_{θ_I} and protected model from Self-Distillation as $F_{\theta_{II}}$.[3] Next, we detail Split-AI and Self-Distillation.

4.2 Our Split-AI Ensemble Architecture

Here we describe Split-AI, the first component of our system.

Split-AI's training: Following the intuition in Section 4.1: we train K sub-models and ensure that each training sample is not used to train L sub-models such that these L sub-models will have similar behavior on this training sample and other non-members. We accomplish this via a specific data partitioning strategy:

For each data point $\boldsymbol{x}$ *in the training set, we randomly generate L non-model indices from {1, 2, ..., K} to denote the L non-models that are not trained with the data point and record the identification numbers of these L non-model indices (denoted as* $Id_{non}(\boldsymbol{x})$[4]*). We then generate the dataset partition based on these non-model indices. For each subset* D_i*, we will only use those training samples which do not include i in their non-model indices.*



(a) Split-AI's training (b) Split-AI's inference

Figure 3: Illustration of Split-AI's data partition for training and adaptive inference in $K = 5, L = 2$ for member samples A, B, C and non-member sample D (red color for members and blue color for non-members). $Id_{non}(A) = (4,5), Id_{non}(B) = (2,3), Id_{non}(C) = (1,2)$.

Figure 3(a) illustrates this partition strategy for three train-

[3] PATE [34, 35] also trains multiple sub-models to provide privacy but with a public dataset. We detailed the difference between our SELENA and PATE in Section 7.2.

[4] $Id_{non}(\mathbf{x})$ records L sub-model indices which are not trained with $\mathbf{x}$.

ing samples (A, B, C) under the setting of $K = 5, L = 2$. We randomly generate non-model sub-model indices: $Id_{non}(A) = (4,5)$, $Id_{non}(B) = (2,3)$, $Id_{non}(C) = (1,2)$. Therefore, A is used to train sub-model 1, 2, 3. B is used to train sub-model 1, 4, 5. C is used to train sub-model 3, 4, 5.

This specific data partition strategy ensures that for each data point, we have L sub-models which are not trained with it. This facilitates our key intuition in Split-AI: we use models that are not trained with a data point to estimate its soft label while protecting the membership information. K and L are parameters of our framework. The approximate size of each subset is $((K-L)/K) \times |D_{tr}|$. We then train K sub-models F_i, one for each subset of the training data D_i, which have the same architecture and hyper-parameter settings.

Split-AI's inference: We now describe the adaptive inference based ensemble strategy for members and non-members. For each queried sample $\mathbf{x}$, the ensemble will check whether there is an exact match of $\mathbf{x}$ in the training set:

- If so, which indicates that $\mathbf{x}$ is a member, the defender will average the prediction vectors on $\mathbf{x}$ from L models which are not trained with $\mathbf{x}$ as the output;
- If not, the defender will randomly use non-model indices of a member sample $\mathbf{x}'$ and average the prediction vectors on $\mathbf{x}$ from L models of $Id_{non}(\mathbf{x}')$ as the output.

Figure 3(b) illustrates the adaptive inference for three member samples (A, B, C) and one non-member sample D following the setting in Figure 3(a). A is non-member for sub-model 4, 5; B is non-member for sub-model 2, 3; C is non-member for sub-model 1, 2; and D is non-member for all sub-models. Adaptive inference will average on non-model indices in sub-models for A, B, C and randomly select one member sample's non-model indices for non-member sample D.

Algorithm 1 presents the entire procedure for Split-AI. We next provide our theoretical analysis of Split-AI's defense capability, which proves that Split-AI strategy is resilient to direct single-query MIAs (discussed in Section 3.1) and can reduce the accuracy of such attacks to a random guess. We first define the direct single-query membership inference game in Definition 1 and then we present our Theorem 2.

Definition 1 (Direct, Single-Query Membership Inference)**.** *The single-query membership inference game is defined between an attacker A and a learner C and is parameterized by a number n which is the number of training examples.*

1. *The attacker selects a dataset $X = \{x_1, \ldots, x_{2n}\}$ and sends it to the learner.*
2. *Learner selects a uniformly random Boolean vector $b = b_1, \ldots, b_{2n}$ such that the Hamming weight of b is exactly n.*
3. *Learner constructs a dataset $S = \{x_i; \forall i \in [2n], b_i = 1\}$ and learns a model F_{θ_l} using S as training set.*
4. *Learner selects a random $i \in [2n]$ and sends $(x_i, F_{\theta_l}(x_i))$ to the adversary*
5. *Adversary outputs a bit b'_i.*

The advantage of A in breaking the security game above is $\mathsf{SQMI}(A,C,n) = \mathbf{E}[1 - |b_i - b'_i|]$ where the expectation is taken over the randomness of the adversary and learner.

Algorithm 1 Split-AI Model F_{θ_I}

Initialize:
K: total number of sub-models $F_1, F_2, ..., F_K$.
L: for each training sample, the number of sub-models which are not trained with it.
(X_{train}, Y_{train}): training data and labels.
Training Phase:
Randomly generate the L non-model indices for each training sample $Id_{non}(\mathbf{x})$.
for $i = 1$ to K **do**
  Construct subset $(X^i_{train}, Y^i_{train})$ for model F_i based on the recorded Id_{non}s of non-model indices: $\{(\mathbf{x}, y)$: $(\mathbf{x}, y) \in (X_{train}, Y_{train})$, i not in $Id_{non}(\mathbf{x})\}$.
  for number of the training epochs **do**
    Update F_i by descending its stochastic gradients over $l(F_i(X^i_{train}), Y^i_{train})$.
  end for
end for
Inference Phase: $F_{\theta_I}(\mathbf{x})$
Given $\mathbf{x}$
if $\mathbf{x}$ in X_{train} **then**

$$F_{\theta_I}(\mathbf{x}) = \frac{1}{L} \sum_{i \in Id_{non}(\mathbf{x})} F_i(\mathbf{x})$$

else
  Randomly select $\mathbf{x}'$ in the training set,

$$F_{\theta_I}(\mathbf{x}) = \frac{1}{L} \sum_{i \in Id_{non}(\mathbf{x}')} F_i(\mathbf{x})$$

end if

Remark 1. *We can define a variant of the security game of Definition 1 for a fixed dataset X. That is, instead of X being chosen by adversary, we define the game for a given X. We use $\mathsf{SQMI}(A,C,X)$ to denote the success of adversary in the security game with the dataset fixed to X.*

Theorem 2. *Consider a learner C_{ST} that uses Algorithm 1. For any direct, single-query membership inference adversary A we have*

$$\mathsf{SQMI}(A, C_{ST}, n) = 50\%$$

We detail the proof of Theorem 2 in Appendix A.1. The intuitive explanation for this proof is that for each data point, the distribution of output of this algorithm on this given point $\mathbf{x}$ is independent of the presence of $\mathbf{x}$ in the training set. This

is because, we will not use models that are trained with $\mathbf{x}$ to answer queries, even if $\mathbf{x}$ is in the training set.

4.3 Our Self-Distillation Mechanism

Limitations of Split-AI. While Split-AI is resilient to direct single-query MIAs, an adversary can leverage more advanced attacks. For example, instead of direct query, attacker may do an indirect query [27] for the target sample (see Section 3.1 for definitions) or may do multiple queries for one target sample to identify membership information, as suggested in recent work [8]. Split-AI suffers from severe privacy risks under the setting of such aggressive attacks that exploit the matching process for training samples: (1) An adaptive attacker can make a single *indirect* query by adding a small noise to the target sample. Such adaptive attacks can fool the matching process used in the inference strategy that checks if the input is a training member or non-member. Split-AI will recognize noisy training samples as non-members and may end up using sub-models trained with the target sample, thus leaking membership information. (2) Attacker can perform replay attacks by making multiple queries for the same target sample: Split-AI will only have one possible prediction vector for members, while approximately C_K^L possible prediction vectors for non-members. Furthermore, Split-AI incurs a computational overhead during inference: For each queried sample, Split-AI first needs to identify whether it is in the training set, thus incurring overhead for this matching process. Second, Split-AI needs to perform inference on L models for each queried sample, while conventional approaches only perform inference on a single model.

Self-Distillation. To overcome the above limitations, we need a more sophisticated defense mechanism and we correspondingly introduce the second component of our framework. We leverage distillation, which is proposed by Hinton et al. [17] to reduce the size of NN architectures or ensemble of NN architectures. To be more specific, here we use a method which we call *Self-Distillation*: we first apply Split-AI to get the prediction vectors for the training samples. We then use the same training set along with the prediction vectors (obtained from Split-AI) as soft labels to train a new model using conventional training. The new protected model benefits from distillation to maintain a good classification accuracy. For queried samples, the defender now just needs to do the inference on the new protected model $F_{\theta_{\text{II}}}$ distilled from the Split-AI. For defense capability, we prove that this new model largely preserves Split-AI's defense ability against direct single-query attack by Theorem 5 and Corollary 6 under mild stability assumptions (Definition 2) in Appendix A.2. Note that our theoretical analysis of SELENA is only valid for direct single-query attacks. In fact, there exist some datasets that SELENA cannot obtain provable privacy for under multi-query attacks. This includes settings with similar data points that have different labels (see Appendix A.3).

Self-Distillation overcomes the privacy limitations of Split-AI and mitigates advanced MIAs. The defender controls the Self-Distillation component and ensures that Self-Distillation only queries each exact training sample once. The attacker only has black-box access to the protected output model of Self-Distillation, but cannot access the Split-AI model. Hence, the attacker cannot exploit the soft labels computation of Split-AI as discussed before. Hence, the final protected model from Self-Distillation effectively mitigates the replay attack and indirect single-query attacks:

1. For replay attack: each sample is only queried once during the Self-Distillation process, while replay attack requires at least two queries of each sample to obtain advantage over random guess. In addition, the final protected model has a deterministic behavior with only one possible prediction vector for each queried sample.

2. For indirect single-query attacks: each exact sample is queried during the Self-Distillation process and noisy samples around the training sample are not queried. In addition, the attacker only has black-box access to the protected model from Self-Distillation (and no access to defender's Split-AI): indirect query attacks are thus limited in obtaining additional membership information.

Self-Distillation also solves the computational overhead of the Split-AI in inference: the defender now does not need to check whether the queried sample is a training sample and it only needs to make inference on a single Self-Distilled model.

In Section 6, we will evaluate the effectiveness of our whole framework via rigorous experimental analysis including direct single-query attacks, label-only attacks and adaptive attacks.

5 Membership Inference Attacks Evaluated

To (empirically) demonstrate the privacy of our system, we evaluate it against three main classes of MIA attacks. First, we evaluate our defense against the direct single-query attacks and label-only attacks introduced earlier. Specifically, we evaluate SELENA against the direct single-query attacks discussed in Section 3.1, and for label-only attacks, we use the boundary estimation attack for all three datasets and the data augmentation attack for CIFAR100.

Additionally, we evaluate our system against *adaptive membership inference attacks*, as introduced in the following. Song and Mittal [43] emphasize the importance of placing the attacker in the last step of the arms race between attacks and defenses: the defender should consider adaptive attackers with knowledge of the defense to rigorously evaluate the performance of the defenses. Therefore, here we consider attacks that are tailored to our defense. As our defense leverages soft labels from the Split-AI ensemble to train a new model $F_{\theta_{\text{II}}}$ in Self-Distillation, we need to analyze whether and how an attacker can also leverage the information about soft labels.

We first note that an attacker is unable to directly interact with our Split-AI to directly estimate soft labels, since the prediction API executes queries on the model produced by the Self-Distillation component. Second, we expect that when the model provider finishes training the protected model $F_{\theta_{\text{II}}}$ with soft labels obtained from Split-AI, it can safely delete the sub-models and soft labels of the training set to avoid inadvertently leaking information about the soft labels. However, an attacker can still aim to indirectly *estimate* soft labels.

As we assume that the attacker knows partial membership of the exact training set in evaluating membership privacy risks (specifically, half of the whole training set) and attacker cannot have access to the defender's non-model indices $Id_{non}(\mathbf{x})$ for training set, the attacker will generate new non-model indices $Id_{non}(\mathbf{x})'$ for these known member samples to train a new shadow Split-AI ensemble and use the shadow Split-AI to estimate soft labels of the target samples. The attacker can then use such soft labels as an additional feature to learn the difference in target model's behavior on members and non-members, and launch MIAs on $F_{\theta_{\text{II}}}$. The shadow Split-AI discussed in our paper is stronger than original shadow models [41] since it is trained with exact knowledge of the partial training dataset.

We design four adaptive direct single-query attacks[5] including two NN-based attacks and two metric-based attacks to leverage the information in the estimated soft labels. To clarify, $F_{\theta_{\text{II}}}$ denotes the protected target model which answers the attacker's queries and $F'_{\theta_{\text{I}}}$ denotes attacker's shadow Split-AI.

MIAs based on NN and soft labels: The first NN-based attack concatenates the soft labels obtained from $F'_{\theta_{\text{I}}}$, the predicted confidence from $F_{\theta_{\text{II}}}$ and the one-hot encoded class labels as features to train a neural network attack model (denoted as I_{NN1}). The second attack utilizes the difference between the estimated soft labels from $F'_{\theta_{\text{I}}}$ and outputs from $F_{\theta_{\text{II}}}$, and uses this difference as an input to the neural network architecture used by Nasr et al. [30] (denoted as I_{NN2}).

MIAs based on distance between soft labels and predicted confidence: Similar to previous metric-based attacks [43], an attacker may try to distinguish between members and non-members by leveraging the distance between estimated soft labels from $F'_{\theta_{\text{I}}}$, and the prediction confidence vectors from $F_{\theta_{\text{II}}}$. We have:

$$I_{\text{dist}}(F_{\theta_{\text{II}}}(\mathbf{x}), F'_{\theta_{\text{I}}}(\mathbf{x}), y) = \mathbb{1}\{Dist(F_{\theta_{\text{II}}}(\mathbf{x}), F'_{\theta_{\text{I}}}(\mathbf{x})) \leq \tau_{(y)}\}$$

$$\text{or, } I_{\text{dist}}(F_{\theta_{\text{II}}}(\mathbf{x}), F'_{\theta_{\text{I}}}(\mathbf{x}), y) = \mathbb{1}\{Dist(F_{\theta_{\text{II}}}(\mathbf{x}), F'_{\theta_{\text{I}}}(\mathbf{x})) \geq \tau_{(y)}\}$$

where we apply both class-dependent threshold τ_y and class-independent threshold τ and we will report the highest MIA accuracy. In this work we consider L_2 distance $I_{\text{L}_2\text{-}dist}$ and cross-entropy loss $I_{\text{CE-}dist}$ (since the cross-entropy loss function is used for training our defense models).

[5] Our Table 2 shows that label-only attacks are weaker than direct single-query attacks on undefended model. We have also designed adaptive multi-query label-only attacks against SELENA and evaluated on Purchase100 dataset, which are better than original label-only attacks, but weaker than adaptive direct single-query attacks.

6 Evaluations

In this section, we first briefly introduce the datasets and model architectures used to train the classification models in Section 6.1.

Next in Section 6.2 we systematically evaluate our end-to-end defense framework including its efficacy against (1) direct single-query attacks, (2) indirect label-only attacks, and (3) adaptive attacks. We also make a comparison with undefended model, MemGuard [21], adversarial regularization [30] and early stopping [43] by considering both the utility and membership privacy risks.

6.1 Experimental Setup

We use three benchmark datasets and target models which are widely used in prior works on MI attacks and defenses.

Datasets. Purchase100, Texas100 and CIFAR100. We follow Nasr et al. [30] to determine the partition between training data and test data and to determine the subset of the training and test data that constitutes attacker's prior knowledge. Specifically, the attacker's knowledge corresponds to half of the training and test data, and the MIA success is evaluated over the remaining half.

Target Models. For CIFAR100, we use ResNet-18 [15], which is a benchmark machine learning model widely used in computer vision tasks. For Purchase100 and Texas100, we follow previous work [30] to use a 4-layer fully connected neural network with layer sizes $[1024, 512, 256, 100]$.

In our defense, we set $K = 25$ and $L = 10$ for all three datasets. We will release code to reproduce all our experiments.

6.2 Results

Table 2 summarizes the classification accuracy and best attack accuracy for each attack type, including comparison with both undefended models (in Section 6.2.1) and previous defenses (MemGuard in Section 6.2.2 and adversarial regularization in Section 6.2.3). In addition, we also compare our SELENA with early stopping in Section 6.2.4.

6.2.1 Comparison with Undefended Model

We first compare our SELENA with undefended model on both membership privacy threats and classification accuracy.

SELENA significantly reduces membership inference risks. From Table 2, we can see that our defense leads to a significant reduction in privacy risks. Across three types of attacks, the MIA accuracy against our defense is no higher than 54.3% on Purchase100, 55.1% on Texas100 and 58.3%

Table 2: Comparison of membership privacy and accuracy on training/test set of undefended model, previous defenses and SELENA on three different datasets. AdvReg refers to adversarial regularization. The last column is the highest attack accuracy for each row, i.e. for a specific defense on one dataset, the highest attack accuracy that MIAs can achieve. The last column gives an overview of comparison: the lower best attack accuracy, the lower membership inference threat. For each dataset, the defense which has the lowest corresponding attack accuracy is bold in the column of best direct single-query attack, best label-only attack and best attack.

dataset	defense	acc on training set	acc on test set	best direct single-query attack	best label-only attack	best adaptive attack	best attack
Purchase100	None	99.98%	83.2%	67.3%	65.8%	N/A	67.3%
	MemGuard	99.98%	83.2%	58.7%	65.8%	N/A	65.8%
	AdvReg	91.9%	78.5%	57.3%	57.4%	N/A	57.4%
	SELENA	82.7%	79.3%	**53.3%**	**53.2%**	54.3%	**54.3%**
Texas100	None	79.3%	52.3%	66.0%	64.7%	N/A	66.0%
	MemGuard	79.3%	52.3%	63.0%	64.7%	N/A	64.7%
	AdvReg	55.8%	45.6%	60.5%	56.6%	N/A	60.5%
	SELENA	58.8%	52.6%	**54.8%**	**55.1%**	54.9%	**55.1%**
CIFAR100	None	99.98%	77.0%	74.8%	69.9%	N/A	74.8%
	MemGuard	99.98%	77.0%	68.7%	69.9%	N/A	69.9%
	AdvReg	86.9%	71.5%	58.6%	59.0%	N/A	59.0%
	SELENA	78.1%	74.6%	**55.1%**	**54.0%**	58.3%	**58.3%**

on CIFAR100. On the other hand, MIA accuracy against undefended models (in the absence of our defense) is much higher: such MIA advantage over a random guess is a factor of $3.0 \sim 4.0$ higher than our defense.

SELENA achieves its privacy benefits at the cost of a small drop in utility. Compared with undefended models, our defense only has a small utility loss (while providing substantial privacy benefits). Compared to undefended models, the classification(test) accuracy of our defense incurs at most 3.9% accuracy drop (on Purchase100), and even no accuracy drop on Texas100.

We also note that even though our approach has a small loss in utility, it achieves a better utility-privacy trade-off compared to prior defenses like MemGuard, adversarial regularization and early stopping, which we discuss next.

6.2.2 Comparison with MemGuard

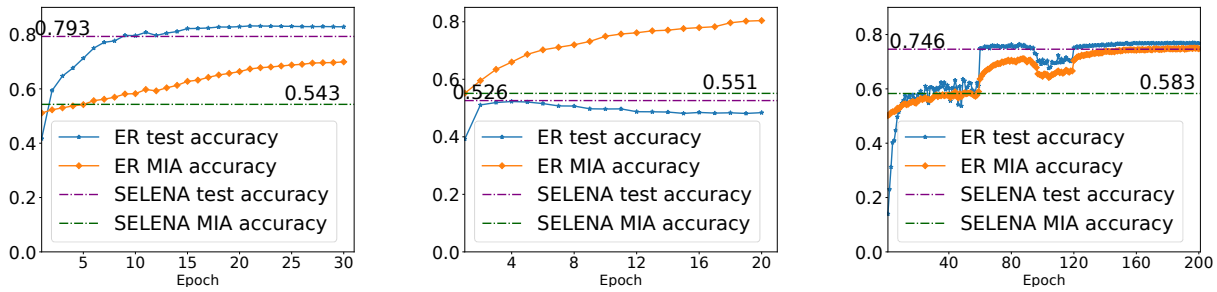
While the test accuracy of our defense is a little lower than MemGuard (MemGuard has the same test accuracy as the undefended model), the MIA accuracy against MemGuard is much higher than our defense. Compared to a random guess, which achieves 50% attack accuracy, the best attacks on MemGuard can achieve $14.7\% \sim 19.9\%$ advantage over a random guess, which is a factor of $2.4 \sim 3.7$ higher than our defense. In general, MemGuard does not provide any additional defense compared to the undefended model against MIAs that do not rely on confidence information: attacker can use label-only attacks as adaptive attacks since MemGuard only obfuscates confidence.

6.2.3 Comparison with Adversarial Regularization

Our defense achieves higher classification accuracy and lower MIA accuracy compared with adversarial regularization. The classification accuracy of our defense is higher than adversarial regularization across all three datasets, and as high as 7.0% for the Texas100 dataset. For MIAs, our defense achieves significantly lower attack accuracy than adversarial regularization. MIA attacks against adversarial regularization is higher than our defense across all three datasets, and its advantage over random guess is at most a factor of 2.1 than our defense (on Texas100). Besides, adversarial regularization is much harder to tune and can also take more training time (by a factor up to 7.8) compared to our defense when multiple GPUs are used in parallel (see Section 7.1).

6.2.4 Comparison with early stopping

We further compare our defense with early stopping, which can also help in minimizing the difference in model behavior on members and non-members [43]. Specifically, we will compare the model performance of an undefended model in each epoch during the training process and our final protected model $F_{\theta_{\text{II}}}$. For early stopping, we only consider direct single-query attacks (due to their strong performance on undefended models). Figure 4 shows a detailed comparison between our defense $F_{\theta_{\text{II}}}$ and early stopping. The dashed lines are the



(a) Purchase100 (b) Texas100 (c) CIFAR100

Figure 4: Detailed comparison of SELENA with early stopping. From left to right are results for Purchase100, Texas100 and CIFAR100. The solid curve lines are the test accuracy and MIA accuracy with corresponding training epochs. ER denotes early stopping. The dashed lines are the test accuracy and MIA accuracy of SELENA, which is shown in Table 2. Our defense achieves a better privacy-utility trade-off than all epochs in the conventional training.

classification accuracy on test set and the best MIA accuracy of our defense, which is already reported in Table 2. The solid curve lines correspond to classification accuracy on test set and MIA accuracy using the undefended model as a function of the training epochs. As we can see from Figure 4, *our defense significantly outperforms early stopping.*

Comparison at similar attack accuracy. The undefended model will only have same level of MIA accuracy as the dashed line of our defense at the very beginning of the training process. However the test accuracy of the undefended model at that point is far lower than that of our defense. For example, approximately, *for Texas100, when MIA accuracy against the conventional trained model is 55.1%, the test accuracy of the undefended model is 13.4% lower than that of our defense.* For other two dataset, when the MIA accuracy against the undefended model achieves similar attack accuracy as our defense, the test accuracy is 8.0% lower on Purchase100 and 11.0% lower on CIFAR100 compared to our defense.

Comparison at similar classification accuracy. When the undefended model achieves the same classification accuracy on the test set as SELENA, the MIA accuracy against the undefended model is significantly higher than our defense. For example, *when the test accuracy of the conventional model reaches 74.6% on CIFAR100 (similar to our defense), the attack accuracy is 63.6%, compared to the best attack accuracy of 58.3% for our defense (which is 5.3% lower).* We can see similar results on other datasets: when the test accuracy of undefended models achieves similar classification accuracy as our defense on Purchase100 and Texas100, the attack accuracy is 58.1% on Purchase100 and 66.0% on Texas100, which is 3.8% and 10.9% higher than SELENA respectively.

We also highlight the following two points from Table 2:

1. Our SELENA effectively induces the similar behaviors including generalization, confidence, robustness for member and non-member samples and therefore the MIA attack accuracy is significantly reduced. Let us take the generalization gap g as an example: in undefended models/MemGuard, g is 16.78% on Purchase100, 27.0% on Texas100, 22.98% on CIFAR100; in adversarial regularization, g is 13.4% on Purchase100, 10.2% on Texas100 and 15.4% on CIFAR100. In contrast, in our defense, g is 3.4% on Purchase100, 6.2% on Texas100 and 3.5% on CIFAR100: Our mechanism reduces the total generalization gap by a factor of up to 6.6 compared to undefended models/MemGuard, and a factor of up to 4.4 compared to adversarial regularization.

2. The additional estimation of soft labels provided by shadow Split-AI (using the entirety of the attacker's knowledge) provides additional information to the attacker which enhances the accuracy of our adaptive attacks: attack has more advantage over random guess than direct single-query attacks and label-only attacks. However, even considering the strong adaptive attacks, SELENA still achieves lower attack accuracy in comparison to previous defenses, which validates the defense effectiveness of our SELENA.

In conclusion, using direct single-query attacks, label-only attacks, as well as adaptive attacks with estimated soft labels, we show that our approach outperforms previous defenses and achieves a better trade-off between utility and empirical membership privacy.

7 Discussions

In this section, we will discuss the computation overhead of our defense, as well as comparison with PATE [34,35] (which uses a disjoint training set partition for sub-models and differential privacy to protect privacy), and DP-SGD [1] (which is a general framework to provide a provable privacy guarantee for neural networks).

7.1 Efficiency

One cost that our framework needs to pay is the use of additional computing resources in the training process as we train multiple sub-models for Split-AI. Table 3 and Table 4 present the comparison of the training time cost and inference time cost between our SELENA and previous defenses. As MemGuard [21] focuses on post-processing techniques for prediction vectors of undefended models in the inference phase, we omit MemGuard in Table 3 for training time and compare our SELENA with MemGuard in Table 4 for inference time. For time comparison, all experiments are tested on a single NVIDIA Tesla-P100 GPU. We separately set the batch size as 512, 128, 256 during the training process for Purchase100, Texas100 and CIFAR100 (note that the batch size might also impact the running time, and here we maintain the same batch size for each dataset across different defenses). For undefended model, adversarial regularization, and our Split-AI, we train 30, 20, and 200 epochs for Purchase100, Texas100, and CIFAR100. For Self-Distillation, we train 60, 30, and 200 epochs for Purchase100, Texas100, and CIFAR100 to ensure convergence. All running times are tested three times and we report the average of the three runs.

Table 3: Comparison of training time.

Dataset	None	AdvReg	SELENA sequential	SELENA parallel
Purchase 100	9.5s	55.7s	359.4s	73.5s
Texas100	10.7s	111.6s	343.0s	68.0s
CIFAR100	1.78h	23.5h	29.6h	3.0h

Comparison of training time in Table 3: our defense (SELENA sequential: sequentially train each sub-model on a single GPU) costs up to 6.1h more computation time than adversarial regularization (CIFAR100). However, we can simply accelerate training Split-AI by training several sub-models parallelly. For example, if we train all K sub-models simultaneously (SELENA parallel), the training time for SELENA is 73.5s for Purchase100, 68.0s for Texas100 and 3.0h for CIFAR100. In contrast, adversarial regularization cannot benefit from parallel training: there is only one model during training.

Comparison of inference time in Table 4: MemGuard costs three orders of magnitude more per inference compared to SELENA since it has to solve a complex optimization problem to obfuscate prediction vectors for every query while SELENA only needs to perform computation on a single model.

In conclusion, we argue that the cost of computing resources in the training phase and no additional computation in inference phase is acceptable as the improvement in GPU technology is making the computing resources cheap while the privacy threat remains severe. If multiple GPUs are available, our approach can easily benefit from parallelization by training the K sub-models in parallel. Finally, we can also tune the system parameters K and L to control the trade-off between the computation cost, model utility and privacy.

Table 4: Comparison of inference time. Tests are done on 1000 samples: 500 members and 500 non-members. Batch size is 1.

Dataset	MemGuard	SELENA
Purchase100	702.7s	0.7s
Texas100	668.6s	0.7s
CIFAR100	768.5s	8.6s

7.2 Comparison with PATE

PATE [34,35] is a framework composed of teacher-student distillation and leverages public data to achieve a better privacy-utility trade-off for differential privacy. PATE uses a disjoint training set partition for sub-models in the teacher component. To get the private label of the public dataset to train the student model, PATE applies noise-added count among sub-models.

There are three major differences between our work and PATE:

1. PATE requires a *public dataset* to provide the provable end-to-end privacy guarantee, which is not possible in certain practical scenarios such as healthcare. Our defense does not need public datasets and provides a strong empirical defense against MIAs.
2. We apply a novel *adaptive inference strategy* to defend against MIAs: for each training sample, we only use prediction of sub-models in Split-AI that are not trained with it as these sub-models will not leak membership information for it. PATE does not use adaptive inference and relies on majority voting over all sub-models.
3. We use *overlapping* subsets to train sub-models. This allows our approach to obtain high accuracy for each sub-model with sufficient subset size. PATE faces the limitation of each sub-model being trained with much reduced subset size due to disjoint subsets.

In addition, PATE incurs a $0.7\% \sim 6.7\%$ drop in classification accuracy [35], while the classification accuracy drop in our defense is no more than 3.9%.

7.3 Comparison with DP-SGD

In this work, we use the canonical implementation of DP-SGD and its associated analysis from the TensorFlow Privacy library.[6] We vary the parameter *noise_multiplier* in the range

[6] https://github.com/tensorflow/privacy.

of [1, 3] on Purchase100 and [1, 2] on Texas100 with a step size 0.2. We set the privacy budget $\varepsilon = 4$ and report the best classification accuracy for these two datasets.

The test accuracy on Purchase100 is 56.0% and the best direct single-query MIA accuracy is 52.8%. The test accuracy on Texas100 is 39.1%, and the best direct single-query MIA accuracy is 53.8%. Note that though DP-SGD provides a differential privacy guarantee and the best direct single-query MIA accuracy is 0.5% $\sim$ 1% lower than that against our SELENA, DP-SGD suffers from a significant loss in utility: compared to the undefended model DP-SGD incurs 13.2% $\sim$ 27.5% drop in classification accuracy, while our defense incurs no more than 3.9% drop in classification accuracy.

8 Related Work

Membership inference attacks against machine learning. MIAs are usually studied in a black-box manner [30, 39, 41]: an attacker either leverages the shadow training technique or utilizes knowledge of partial membership information of training set. Most MIAs are direct single-query attacks [43, 44, 51, 52]. A more recent line of MIA research has considered indirect multi-query attacks which leverage multiple queries around the target sample to extract additional information [8, 20, 26, 27]. Jayaraman et al. [20] analyze MIA in more realistic assumptions by relaxing proportion of training set size and test set size in the MIA set up to be any positive value instead of 1. Hui et al. [18] study MIA in a practical scenario, assuming no true labels of target samples are known and utilizing differential comparison for MIAs. Another threat model for MIAs is that of a white-box setting, i.e., the attacker has full access to the model [24, 31], which can exploit model parameters to infer membership information.

Membership inference defenses for machine learning. Membership inference defenses can be divided into two main categories. One category of defenses are specifically designed to mitigate such attacks. It has been shown that techniques to improve a model's generalization ability, including regularization [23] and dropout [45], can decrease the MIA success [39, 41] limitedly. Several defenses [25, 30] propose adding a specific constraint during the training process to mitigate the difference of model behavior on members and non-members. Post-processing techniques on prediction vectors are also applied on membership inference defenses [21, 50]. Note that the defenses which obfuscate prediction vectors cannot defend against label-only attacks [8, 26]. Moreover, Song et al. [43] re-evaluate two state-of-the-art defenses (adversarial regularization [30] and MemGuard [21]) and find that both of them underestimated metric-based attacks. Shejwalkar et al. [40] propose distillation of public data to protect membership privacy. However, public dataset is not usually available in many practical scenarios. Another category of defenses uses differential privacy mechanisms [9–11], which provide a provable privacy guarantee for users. A general framework combining deep learning and differential privacy is DP-SGD [1, 28, 49]. However, machine learning with differential privacy suffers from the challenge of achieving acceptable utility loss and privacy guarantee [19, 37]. Several methods have been proposed to improve the classification accuracy under an acceptable ε guarantee, which is still an active area of research. Current state-of-the-art approaches still incur significant drop in classification accuracy (around 25%) on benchmark datasets with acceptable $\varepsilon \leq 3$ [32, 36, 46].

Other Attacks Against Machine Learning Privacy. Fredrikson et al. [12] propose model inversion attacks, which can infer missing values of an input feature from the classifier's prediction. Ganju et al. [13] study property inference attacks aiming to infer properties of the target model's training set. Salem et al. [38] propose the dataset reconstruction attack in the online learning setting. Another line of works studies model extraction attacks [16, 22, 47], i.e., stealing the ML model's learned parameters through the prediction API. Besides model parameters, other works also focus on stealing the target model's hyperparameters [48]. Recently Carlini et al. [3, 4] study the memorization and data extraction attacks on natural language processing models, which shows that machine learning models suffer from severe privacy threats.

9 Conclusions

In this paper we introduce a new practical membership inference defense using Split-AI and Self-Distillation. We first split the training set into K subsets to train K sub-models. We ensure each training sample is not used to train L sub-models, and apply an adaptive inference strategy for members and non-members. Split-AI will only use the average of a particular subset of L sub-models which are not trained with the queried samples. Hence Split-AI can defend against direct single-query attacks. We apply Self-Distillation from Split-AI to defend against stronger attacks and avoid additional computing resources in inference. We perform a rigorous evaluation through MIAs including direct single-query attacks, label-only attacks and adaptive attacks to show that our defense outperforms previous defenses to achieve a better trade-off between the utility and empirical membership privacy. Future work includes understanding the adaptation of Split-AI in other privacy tasks such as provable private mechanisms, analyzing the defense performance against white-box MIAs, and extending our defense from classification models to generative models.

Acknowledgements

We are grateful to anonymous reviewers at USENIX Security and Esfandiar Mohammadi for valuable feedback. This work was supported in part by the National Science Foundation under grants CNS-1553437, CNS-1704105, and CNS-

1953786, the ARL's Army Artificial Intelligence Innovation Institute (A2I2), the Office of Naval Research Young Investigator Award, the Army Research Office Young Investigator Prize, Schmidt DataX award, and Princeton E-ffiliates Award.

References

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

[2] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *International Conference on Learning Representations*, 2018.

[3] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *USENIX Security Symposium*, pages 267–284, 2019.

[4] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *USENIX Security Symposium*, 2021.

[5] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.

[6] Rich Caruana, Steve Lawrence, and C Lee Giles. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Advances in Neural Information Processing Systems*, pages 402–408, 2001.

[7] Jianbo Chen, Michael I Jordan, and Martin J Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1277–1294. IEEE, 2020.

[8] Christopher A Choquette Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

[9] Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.

[10] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[11] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[12] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333, 2015.

[13] Karan Ganju, Qi Wang, Wei Yang, Carl A Gunter, and Nikita Borisov. Property inference attacks on fully connected neural networks using permutation invariant representations. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 619–633, 2018.

[14] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. Logan: Membership inference attacks against generative models. In *Proceedings on Privacy Enhancing Technologies (PoPETs)*, number 1, pages 133–152. Sciendo, 2019.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[16] Xinlei He, Jinyuan Jia, Michael Backes, Neil Zhenqiang Gong, and Yang Zhang. Stealing links from graph neural networks. In *USENIX Security Symposium*, 2021.

[17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[18] Bo Hui, Yuchen Yang, Haolin Yuan, Philippe Burlina, Neil Zhenqiang Gong, and Yinzhi Cao. Practical blind membership inference attack via differential comparisons. In *Network and Distributed Systems Security Symposium (NDSS)*, 2021.

[19] Bargav Jayaraman and David Evans. Evaluating differentially private machine learning in practice. In *USENIX Security Symposium*, pages 1895–1912, 2019.

[20] Bargav Jayaraman, Lingxiao Wang, David Evans, and Quanquan Gu. Revisiting membership inference under realistic assumptions. In *Proceedings on Privacy Enhancing Technologies (PoPETs)*, 2021.

[21] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. Memguard: Defending against black-box membership inference attacks via

adversarial examples. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 259–274, 2019.

[22] Kalpesh Krishna, Gaurav Singh Tomar, Ankur P Parikh, Nicolas Papernot, and Mohit Iyyer. Thieves on sesame street! model extraction of bert-based apis. *International Conference on Learning Representations*, 2020.

[23] Anders Krogh and John A Hertz. A simple weight decay can improve generalization. In *Advances in Neural Information Processing Systems*, pages 950–957, 1992.

[24] Klas Leino and Matt Fredrikson. Stolen memories: Leveraging model memorization for calibrated white-box membership inference. In *USENIX Security Symposium*, pages 1605–1622, 2020.

[25] Jiacheng Li, Ninghui Li, and Bruno Ribeiro. Membership inference attacks and defenses in classification models. In *Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy*, 2021.

[26] Zheng Li and Yang Zhang. Membership leakage in label-only exposures. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021.

[27] Yunhui Long, Vincent Bindschaedler, Lei Wang, Diyue Bu, Xiaofeng Wang, Haixu Tang, Carl A Gunter, and Kai Chen. Understanding membership inferences on well-generalized learning models. *arXiv preprint arXiv:1802.04889*, 2018.

[28] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations*, 2018.

[29] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 691–706. IEEE, 2019.

[30] Milad Nasr, Reza Shokri, and Amir Houmansadr. Machine learning with membership privacy using adversarial regularization. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 634–646, 2018.

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

[32] Milad Nasr, Reza Shokri, and Amir Houmansadr. Improving deep learning with differential privacy using gradient encoding and denoising. *arXiv preprint arXiv:2007.11524*, 2020.

[33] Milad Nasr, Shuang Song, Abhradeep Thakurta, Nicolas Papernot, and Nicholas Carlini. Adversary instantiation: Lower bounds for differentially private machine learning. In *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2021.

[34] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *International Conference on Learning Representations*, 2017.

[35] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with pate. In *International Conference on Learning Representations*, 2018.

[36] Nicolas Papernot, Abhradeep Thakurta, Shuang Song, Steve Chien, and Úlfar Erlingsson. Tempered sigmoid activations for deep learning with differential privacy. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[37] Md Atiqur Rahman, Tanzila Rahman, Robert Laganière, Noman Mohammed, and Yang Wang. Membership inference attack against differentially private deep learning model. *Trans. Data Priv.*, 11(1):61–79, 2018.

[38] Ahmed Salem, Apratim Bhattacharya, Michael Backes, Mario Fritz, and Yang Zhang. Updates-leak: Data set inference and reconstruction attacks in online learning. In *USENIX Security Symposium*, pages 1291–1308, 2020.

[39] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. In *Network and Distributed Systems Security Symposium (NDSS)*, 2019.

[40] Virat Shejwalkar and Amir Houmansadr. Membership privacy for machine learning models through knowledge transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[41] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

[42] Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. Machine learning models that remember too much. In *Proceedings of the 2017 ACM SIGSAC*

Conference on Computer and Communications Security, pages 587–601, 2017.

[43] Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In *USENIX Security Symposium*, 2021.

[44] Liwei Song, Reza Shokri, and Prateek Mittal. Privacy risks of securing machine learning models against adversarial examples. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 241–257, 2019.

[45] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[46] Florian Tramèr and Dan Boneh. Differentially private learning needs better features (or much more data). In *International Conference on Learning Representations*, 2021.

[47] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In *USENIX Security Symposium*, pages 601–618, 2016.

[48] Binghui Wang and Neil Zhenqiang Gong. Stealing hyperparameters in machine learning. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 36–52. IEEE, 2018.

[49] Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled rényi differential privacy and analytical moments accountant. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1226–1235. PMLR, 2019.

[50] Ziqi Yang, Bin Shao, Bohan Xuan, Ee-Chien Chang, and Fan Zhang. Defending model inversion and membership inference attacks via prediction purification. *arXiv preprint arXiv:2005.03915*, 2020.

[51] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282. IEEE, 2018.

[52] Samuel Yeom, Irene Giacomelli, Alan Menaged, Matt Fredrikson, and Somesh Jha. Overfitting, robustness, and malicious algorithms: A study of potential causes of privacy risk in machine learning. *Journal of Computer Security*, 28(1):35–70, 2020.

[53] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3713–3722, 2019.

A Proof for Split-AI and SELENA against Direct, Single-Query Membership Inference Attack

Notation. In this section, we use $x \leftarrow X$ to denote that x is sampled from a distribution X. We use $\text{Supp}(X)$ to denote the support set of a random variable X. By $TV(X,X')$ we denote the total variation distance between X and X', that is $TV(X,X') = \sup_{S \subset \text{Supp}(X) \cup \text{Supp}(X')} \Pr[X \in S] - \Pr[X' \in S]$.

A.1 Split-AI's Privacy under Direct Single-query Attacks

Here we provide the detailed proof for Theorem 2.

Proof. We show that for any adversary's choice of $i \in [2n]$ in step 4 of the security game, the view of adversary in two cases when $b_i = 0$ and when $b_i = 1$ are statistically identical. Note that the only information that the adversary receives is $r_i = F_{\theta_I}(x_i)$. We show that the distributions of two random variables $r_i \mid b_i = 0$ and $r_i \mid b_i = 1$ are identical. Let U_i be a random variable corresponding to the subset of trained models that do not contain x_i in their training set (in particular $|U_i| = L$ if $b_i = 1$ and $|U_i| = K$ when $b_i = 0$). Also, let U denote a random variable corresponding to a subset of L models that do not contain a random x_k in their training data where k is selected from $\{j \in [2n]; b_j = 1\}$ uniformly at random.

We first note that $U \mid b_i = 0$ and $U_i \mid b_i = 1$ are identically distributed random variables. Specifically, they are both an ensemble of L models trained on a uniformly random subset of a dataset $T \subset \{x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{2n}\}$ where $|T| = n-1$.

Now, lets calculate the distribution of response when $b_i = 1$ and when $b_i = 0$. For $b_i = 1$ we have

$$(r_i \mid b_i = 1) \equiv (\frac{1}{L} \cdot \sum_{F \in U_i} F(x_i) \mid b_i = 1)$$

For $b_i = 0$ we have

$$(r_i \mid b_i = 0) \equiv (\frac{1}{L} \cdot \sum_{F \in U} F(x_i) \mid b_i = 0)$$

Now since $U_i \mid b_i = 1$ and $U \mid b_i = 0$ are distributed identically, the summation of the query points are also identically distributed. Therefore, $r_i \mid b_i = 0$ and $r_i \mid b_i = 1$ are identically distributed. Note that it is crucial that the adversary only queries the point x_i as otherwise we had to take the summation over $U \mid b_i = 1$ and $U \mid b_i = 0$ which are not identically

distributed (the case of $b_i = 1$ could have x_i in the training set of the L models).

Since we prove that $r_i \mid b_i = 1$ and $r_i \mid b_i = 0$ are identical, the adversary cannot distinguish them and the success probability of the adversary is exactly 0.5. The intuitive explanation for this proof is that for each data point, the distribution of output of this algorithm on a given point x is independent of the presence of x in the training set, as we will not use models that are trained with x to answer queries, even if x is in the training set.

□

Remark 3 (A stronger security game and theorem)**.** *Note that there is a worst-case variant of Definition 1 where in step 4, instead of the challenger, the adversary selects $i \in [2n]$. This is a stronger security game as the adversary can select the worst example in the dataset. However, Theorem 2 remains unchanged in this game. This is because the proof applies to any $i \in [2n]$ and does not require i to be chosen at random. As we will see below, we have another theorem (Theorem 5) that considers the privacy of end-to-end SELENA for which the guarantee only holds for the weaker definition.*

A.2 SELENA's Privacy under Direct Single-query Attacks

Definition 2 (stable distillation)**.** *A distillation algorithm Q: $M_s \times \text{AUX} \to M_o$ is a potentially randomized algorithm with access to a source model $m_s \in M_s \subseteq Y^X$ and some auxiliary information and returns an output model $m_o \in M_o \subset Y^X$. We define the notion of stability for a distillation algorithm on a point $x \in X$, and joint distribution $\mathcal{M}$ on $M_s \times \text{AUX}$ as follows:*

$$\mathsf{stablity}(Q, \mathcal{M}, x) = 1 - TV(Q(\mathcal{M})[x], \mathcal{M}[x]).$$

Moreover, we say the algorithm Q has (α, β)-stability on a distribution $\mathcal{M}$ and a dataset X iff

$$\Pr_{x \leftarrow X}[\mathsf{stability}(Q, \mathcal{M}, x) \le 1 - \alpha] \le \beta$$

Example. If the distillation algorithm Q ensures that for a specific point x and for all $m_s \in M_s$ we have $Q(m_s)[x] = m_s[x]$, then Q has stability 1 on point x for all distributions $\mathcal{M}$ defined on M_s.

Remark 4. *The distillation algorithm Q could also depend on an additional dataset that is correlated with m_s as the auxiliary information. For instance, in our self-distillation algorithm, the distillation is done through the same training set that was used to train m_s. In this case, we are interested in the joint distribution $\mathcal{M}$ that consists of a model m_s as first element and a dataset D as the second element, so that m_s is a model trained on dataset D.*

Now we state a corollary of our Theorem 2 about the privacy of the distilled models from the output of the Split-AI operation.

Notation. For a learner C and a dataset X, we use $\mathcal{M}_{C,X}$ to denote a distribution of models that is obtained from the following process: First select a random subset S of size $|X|/2$ and then train a model m on that subset using learner C and output (m, S). For a learner C and a distillation model Q, we use QoC to denote a learner that first uses C to train a model and then uses distillation algorithm Q to distill that model and then returns the distilled model.

Theorem 5. *Let C be an arbitrary learner. Assume for a set of samples X the distillation algorithm Q has (α, β)-stability on distribution $\mathcal{M}_{C,X}$ and dataset X. Then, for any adversary A we have*

$$\mathsf{SQMI}(A, QoC, X) \le \mathsf{SQMI}(A, C, X) + \alpha + \beta.$$

Proof. Consider an adversary A that given a response $QoC[x_i]$ on query $x_i \in X$ outputs a bit $b'_i = A(QoC(x_i))$. Let E be an event defined on X such that $E(x) = 1$ iff

$$\mathsf{stability}(Q, \mathcal{M}_{C,X}, x) \ge 1 - \alpha.$$

For a point x_i such that $E(x_i) = 1$ we have

$$\begin{aligned}
&\Pr\left[A(QoC[x_i]) = b_i\right] \le \Pr\left[QoC[x_i] \neq C[x_i]\right] \\
&\quad + \Pr\left[A(C[x_i]) = b_i \mid C(x_i) = QoC[x_i]\right] \cdot \Pr\left[QoC[x_i] = C[x_i]\right] \\
&\le \alpha + \Pr\left[A(C[x_i]) = b_i\right]
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
&\Pr_{x_i \leftarrow X}\left[A(QoC[x_i]) = b_i\right] \\
&\le \Pr_{x_i \leftarrow X}\left[A(QoC[x_i]) = b_i \mid E(x_i)\right] \cdot \Pr_{x_i \leftarrow X}[E(x_i)] + \Pr_{x_i \leftarrow X}[\bar{E}(x_i)] \\
&\le \Pr_{x_i \leftarrow X}\left[A(QoC[x_i]) = b_i \mid E(x_i)\right] \cdot \Pr_{x_i \leftarrow X}[E(x_i)] + \beta \\
&\le \left(\Pr_{x_i \leftarrow X}\left[A(C[x_i]) = b_i \mid E[x_i]\right] + \alpha\right) \cdot \Pr_{x_i \leftarrow X}[E(x_i)] + \beta \\
&\le \Pr_{x_i \leftarrow X}\left[A(C[x_i]) = b_i\right] + \alpha + \beta \\
&= \mathsf{SQMI}(A, C, X) + \alpha + \beta.
\end{aligned}$$

□

Now we are ready to state a corollary of Theorems 5 and Theorem 2 for the full pipeline of Split-AI followed by Self-Distillation. The following Corollary directly follows from Theorems 5 and Theorem 2.

Corollary 6. *Let C_{ST} be a learner that uses the Split-AI algorithm 1. Also, let Q_{SD} be a distiller that uses self-distillation algorithm. If Q_{SD} is (α, β)-stable for a dataset X and distribution $\mathcal{M}_{C_{ST},X}$, then, for any adversary A we have*

$$\mathsf{SQMI}(A, Q_{SD} o C_{ST}, X) \le 0.5 + \alpha + \beta.$$

A.3 Discussion of Split-AI and SELENA for Correlated Points

Remark 7 (How private is SELENA against multi-query attacks?)**.** *The above theoretical analysis of SELENA is only valid for direct single-query attacks. But one might wonder if we can show a similar theory for privacy of SELENA against multi-query attacks. Unfortunately, we cannot prove a result as general as Corollary 6 for multi-query attacks. In fact, there exist some datasets that SELENA cannot obtain provable privacy for. For instance, imagine a dataset that contains two points* $(x,0)$ *and* $(x',1)$ *such that* x *and* x' *are almost the same points, i.e.* $x \approx x'$*, yet they are labeled differently in the training set (*x *is labeled as 0 and* x' *as 1). In this scenario, we can observe that the adversary can obtain information about membership of* x *and* x'*, when querying both points. In particular, if only one of* x *and* x' *are selected as members, then we expect the result of query on* x *and* x' *to be the same and equal to the label of the one that is selected as a member. However, we argue that this lack of privacy for certain datasets will not manifest in the real world examples as such high correlation does not frequently appear in real-world datasets. Our empirical analysis of SELENA is consistent with this claim. We defer the theoretical analysis of SELENA for multi-query attacks on datasets that satisfies certain assumptions to future work.*

Specific study of possible leakage in Remark 7. To study the possible leakage in Remark 7 on Split-AI, we investigate the effect of querying correlated points. In particular, we consider pairs (x,x'), where x is a member and x' is a close non-member. Then, we measure the difference between outputs from L sub-models in $Id_{non}(x)$ and random L sub-models for a non-member sample x'. This way, we obtain an attack which shows the magnitude of the privacy loss due to the leakage described in Remark 7.

Experiment setup. We design the following experiment on the CIFAR100 dataset. We use L_2 distance to measure the correlation between member samples and non-member samples. For each training sample x, we find the sample x' among test set which has the least L_2 distance to x but labeled differently. For each correlated pair (x,x'), we query Split-AI on x' twice, the first query uses L sub-model indices defined by $Id_{non}(x)$ and the second query uses random L sub-models. We denote these two queries by $F_{\theta_I}(x', Id_{non}(x))$ and $F_{\theta_I}(x', rnd)$ respectively. Now we can leverage the MIAs evaluated in Section 6: consider $F_{\theta_I}(x', Id_{non}(x))$ as a member and $F_{\theta_I}(x', rnd)$ as a non-member, we use these predictions along with the label of x' as input to the direct single-query attacks (due to their strong performance on undefended models).

Result. We present the result of the correlated point attack as a function of how close these correlated pairs are, i.e., the distance $L_2(x,x')$. For L_2 distance from 1 to 15,[7] we evaluate the ratio of member samples that satisfy this L_2 restriction and the corresponding attack success rate, and plot the result in Figure 5. We can see that for L_2 distance larger than 6, the attack performance is close to a random guess. For L_2 distance less than 6, we can see that as L_2 distance restriction becomes smaller, the attack accuracy tends to increase. This is consistent with what we discuss in Remark 7. However, we should also note that the ratio of such pairs that satisfy the restriction is close to 0. Specifically, for $L_2 = 1$. there are only 6 member samples out of 50000 member samples that satisfy this restriction, which is consistent with our discussion in Remark 7 that the presence of such highly correlated pairs in real-world datasets is small.

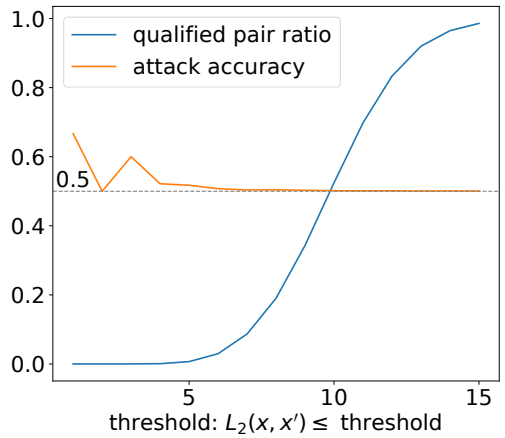


Figure 5: Given the L_2 distance threshold for correlated pairs (x,x') as the x-axis, we plot the fraction of pairs with distance less than that threshold. We also plot the average attack accuracy among paired queries within that distance.

Can our NN-based attacks (in Section 3 and Section 5) leverage the correlation leakage? We emphasize that our NN-based attacks described in Section 3 and Section 5 have all the required information for leveraging the correlation leakage described in this subsection. Our attacks have access to a large fraction of dataset together with their membership information and the prediction vector on the target model. Therefore, in principle, the NN-based attack could learn to perform the following: (1) On a given point x, finds the most correlated point x' in the provided dataset. (2) Calculates the expected prediction vector for querying x' on models and non-models of x. (3) Runs the attack described above in this subsection. We cannot prove that the neural network does all these steps, but it has all the power to do so.

[7] Image pixel in range [0,1].