Semi-Leak: Membership Inference Attacks Against Semi-supervised Learning

Xinlei He[1], Hongbin Liu[2], Neil Zhenqiang Gong[2], and Yang Zhang[1]

[1] CISPA Helmholz Center for Information Security
[2] Duke University

Abstract. Semi-supervised learning (SSL) leverages both labeled and unlabeled data to train machine learning (ML) models. State-of-the-art SSL methods can achieve comparable performance to supervised learning by leveraging much fewer labeled data. However, most existing works focus on improving the performance of SSL. In this work, we take a different angle by studying the training data privacy of SSL. Specifically, we propose the first data augmentation-based membership inference attacks against ML models trained by SSL. Given a data sample and the black-box access to a model, the goal of membership inference attack is to determine whether the data sample belongs to the training dataset of the model. Our evaluation shows that the proposed attack can consistently outperform existing membership inference attacks and achieves the best performance against the model trained by SSL. Moreover, we uncover that the reason for membership leakage in SSL is different from the commonly believed one in supervised learning, i.e., overfitting (the gap between training and testing accuracy). We observe that the SSL model is well generalized to the testing data (with almost 0 overfitting) but "memorizes" the training data by giving a more confident prediction regardless of its correctness. We also explore early stopping as a countermeasure to prevent membership inference attacks against SSL. The results show that early stopping can mitigate the membership inference attack, but with the cost of model's utility degradation. [3]

Keywords: Membership Inference Attack, Semi-Supervised Learning

1 Introduction

Machine learning (ML) has made tremendous progress in the past decade. One of the key reasons for the great success of ML models can be credited to the large-scale labeled data. However, such labeled datasets are often hard to collect as they rely on human annotations and expertise in the specific domain. Meanwhile, unlabeled datasets are easy to obtain. To better leverage the unlabeled data, semi-supervised learning (SSL) has been proposed. Concretely, SSL uses a small set of labeled data and a large set of unlabeled data to jointly train the ML model. In recent years, SSL shows its effectiveness on different tasks by leveraging much

[3] Our code is available at https://github.com/xinleihe/Semi-Leak.

fewer labeled data [29,36,39]. For instance, by only using 250 labeled samples, FlexMatch [39] can achieve about 95% accuracy on CIFAR10.

Different from supervised learning where every data sample is treated equally in the training procedure, SSL takes different ways to handle the labeled and unlabeled data samples during the training. Concretely, the state-of-the-art SSL methods [29,36,39] leverage weak augmentation to the labeled samples and trains them in a supervised manner. For each unlabeled sample, it would generate a weakly-augmented view and a strongly-augmented view (by weak and strong augmentations), and the goal is to leverage the model's prediction probability (referred to as prediction or posteriors) of the weakly-augmented view to guide the training of the strongly-augmented view of the sample. Instead of directly using the posteriors as a "soft" label, those SSL methods switch the posteriors into a "sharpen" [36] or "hard" label [29,39]. Note that the sample is not used to train the model until the highest probability of the prediction on the weakly-augmented view exceeds a pre-defined threshold τ. In this way, the model trained by SSL can gradually learn more accurate predictions.

Despite being powerful, ML models are shown to be vulnerable to various privacy attacks [8,28,30], represented by membership inference attacks [28,27,22,31]. The goal of membership inference attack is to determine whether a data sample is used to train a target ML model. Successful membership inference attacks can raise privacy concerns as they may reveal sensitive information of people. For instance, if an ML model is trained on the data for people with a certain sensitive attribute (e.g., diseases), identifying the person in the training dataset directly reveals this individual's sensitive attribute. So far, most of the efforts on membership inference attacks concentrate on models trained by supervised learning. Also, there are some exploratory researches investigating the privacy risks in self-supervised learning [20,10]. However, in SSL, the labeled and unlabeled samples are treated differently during the training. It is important to quantify whether this unique training paradigm would lead to different privacy risks for labeled and unlabeled samples. Also, as the different augmented views instead of the original samples are used to train the model, we are curious whether a more effective membership inference attack mechanism can be proposed against SSL. To be best of our knowledge, this is largely unexplored.

In this work, we fill the gap by proposing the first data augmentation-based membership inference attack method against SSL. A key advantage for SSL is that it only needs a small amount of labeled data and leverages the unlabeled data itself to guide the training. Concretely, for the labeled data, the model is trained in a supervised manner. For the unlabeled data, SSL leverage the data itself as the supervision. In particular, for each unlabeled training sample, a weakly augmented and a strongly augmented views will be fed into the target model and the training objective is to minimize the distance of the model's prediction on these two views. Our proposed data augmentation-based attack is based on the intuition that the model's prediction of these two views should be more similar if the sample belongs to the model's training set.

We conduct our evaluation on three SSL methods (FixMatch, FlexMatch, and UDA) and three commonly used SSL datasets (SVHN, CIFAR10, and CIFAR100). Our empirical results show that our proposed attack can consistently outperform baseline attacks and reaches the best performance. For instance, for FixMatch trained on CIFAR10 with 500 labeled samples, our attack achieves 0.780 AUC while the best baseline attack only has 0.722 AUC. This indicates that our attack can better unleash the membership information in SSL.

Moreover, we find that, unlike supervised learning where the membership leakage can be credited to the overfitting nature of the model [28,27](i.e., the model predicts the training data more accurately than the testing data), models trained by SSL methods are well generalized and have almost no overfitting but still suffer high membership inference risk. Our analysis reveals that the model indeed "memorizes" the training data, but such memorization does not present as a more accurate prediction, but a more confident prediction. We show that the prediction entropy distribution of members and non-members has a large gap in models trained by SSL (measured by Jason-Shannon (JS) Distance).

Contributions. (1) We are the first to study the privacy risk of SSL through the lens of membership inference attacks and we propose a data augmentation-based attack that is tailored to SSL methods. (2) We conduct extensive experiments on SVHN, CIFAR10, and CIFAR100 datasets. Our results show that our proposed attack outperforms baseline attacks that are extended from existing works to SSL settings. (3) We show that the effectiveness of membership inference attacks against SSL is not credited to the model's overfitting level but credited to the model prediction's distinguishable entropy distributions for members and non-members (measured by Jason-Shannon Distance). (4) We study an early-stopping-based defense against our proposed attack. We show that this defense can decrease the attack AUC of our attack but sacrifice the testing accuracy of the trained models.

2 Preliminary and Related Work

2.1 Semi-Supervised Learning

Semi-supervised learning (SSL) [16,21,3,29,36,39] aims to train accurate models via exploiting a large amount of unlabeled data when the labeled data is scarce. In this paper, we focus on the vision domain since most advanced SSL methods are designed for it. Generally speaking, state-of-the-art SSL techniques [29,36,39] produce "pseudo labels" for the unlabeled samples when the model's predictions are confident enough based on pre-defined threshold strategies. For example, Lee [16] first proposed to produce the class label that has the highest confidence score output by the classifier for unlabeled samples during training. After assigning pseudo labels to unlabeled samples, they can train classifiers in a supervised fashion with labeled and unlabeled samples. Recently, FixMatch [29] achieves state-of-the-art classification accuracy via assigning the strongly augmented unlabeled samples with the pseudo labels produced from

the corresponding weakly augmented samples when the highest confidence score exceeds a certain threshold. While UDA [36] was proposed to treat the classifier's "sharpen" output confidence scores as the 'pseudo labels' rather than one class label. Similar to FixMatch, UDA trains strongly augmented unlabeled samples with the pseudo labels produced from the corresponding weakly augmented samples. FlexMatch [39] updates FixMatch by introducing the curriculum learning-based method to flexibly adjust the threshold for different classes during the training. Existing studies on SSL mainly focus on how to improve the performance, however, we are the first to show that state-of-the-art SSL methods are vulnerable to our tailored membership inference attacks, which exploit the strong/weak data augmentations used by state-of-the-art SSL methods.

2.2 Membership Inference Attacks

The goal of membership inference attack [28,27,22,25,5,11,26,12,20,10,31,9,19,34,4] is to determine whether a given data sample is used to train a target model. Multiple works studied the membership inference attacks against the supervised learning [28,27,23,19,6,9]. Shokri et al. [28] proposed the first black-box membership inference attack against machine learning models by leveraging multiple shadow models and attack models. The attack model takes a sample's posteriors generated from the target model as the input and predicts whether it is a member or not. Salem et al. [27] relaxed the assumption from Shokri et al. [28] and proposed novel model-independent and dataset-independent membership inference attacks. Nasr et al. [23] studied the white-box membership inference attacks in both centralized and federated learning settings. Li and Zhang [19] and Choo et al. [6] concentrated on a more restricted attack scenario (called label-only attack) where the target model only returns the predicted labels instead of posteriors when the adversary queries the target model with given samples. Roughly speaking, their proposed label-only attacks aim to infer a given sample's membership status via comparing a pre-defined threshold with the scale of adversarial perturbation that needs to be added to the given sample to change the target model's predicted label. However, these membership inference attacks are tailored to supervised learning and we show that semi-supervised learning is more vulnerable to our proposed data augmentation-based membership inference attack compared with existing membership inference attacks.

3 Conventional Membership Inference Attacks

In membership inference attack, the adversary aims to determine whether a given data sample x belongs to the target model $\mathcal{T}$'s training dataset or not given the adversary's background knowledge $\mathcal{K}$. A data sample x is called *member* (or *non-member*) if it belongs to (or does not belong to) the training dataset of the target model $\mathcal{T}$. Formally, we define the membership inference attack as $\mathcal{A} : x, \mathcal{T}, \mathcal{K} \rightarrow \{0, 1\}$, where the attack $\mathcal{A}$ is essentially a mapping function and 1 (or 0) means the data sample x is a member (or non-member).

3.1 Threat Model

Given a target model $\mathcal{T}$, we first assume that the adversary only has black-box access to it, which means that the adversary can only query the target model with a data sample x and obtain the target model's prediction on it (denoted as posteriors). Note that in this paper we consider the black-box attack since it is the most difficult and practical real-world scenario.

Following previous work [28,10,31], we assume that the adversary has a *shadow dataset* $\mathcal{D}_{shadow}$ that has the same distribution as the target model $\mathcal{T}$'s training dataset $\mathcal{D}_{target}^{train}$. The adversary can use the shadow dataset $\mathcal{D}_{shadow}$ to train a *shadow model* $\mathcal{S}$, which mimics the behavior of the target model $\mathcal{T}$ to better conduct the attacks. Also, we assume that the shadow model $\mathcal{S}$ has the same architecture as the target model. Such assumption is realistic as: (1) The adversary can leverage the same machine learning service to train the shadow model and (2) The adversary can perform hyperparameter stealing attacks [24,33] to obtain the target model's architecture.

3.2 Methodology

Generally speaking, the membership inference attack pipeline usually consists of three major components, i.e., shadow training, constructing attack training dataset, and attack model training or performing the attack.

Shadow Training. Shadow training [28,22,27] aims to train shadow models to mimic the behavior of the target model based on the adversary's background knowledge. Specifically, the adversary first evenly splits the shadow dataset $\mathcal{D}_{shadow}$ into two disjoint parts, i.e., shadow training data $\mathcal{D}_{shadow}^{train}$ and shadow testing data $\mathcal{D}_{shadow}^{test}$. The adversary then uses the $\mathcal{D}_{shadow}^{train}$ to train a shadow model $\mathcal{S}$ that mimics the behavior of the target model $\mathcal{T}$.

Constructing Attack Training Dataset. To construct the training dataset for the attack model, the adversary first uses $\mathcal{D}_{shadow}^{train}$ (contains members) and $\mathcal{D}_{shadow}^{test}$ (contains non-members) to query the shadow model $\mathcal{S}$ and obtain the corresponding posteriors. Following Salem et al. [27], we leverage the descendingly sorted posteriors as the inputting features for the attack model. Finally, we assign the membership status 1/0 for members/non-members as labels.

Training Neural Network-based Attack Model. For neural network-based attacks [28,27] (denoted as $\mathcal{A}_{NN}$), the adversary aims to train a neural network-based attack model to distinguish members and non-members given the posteriors generated by the target model $\mathcal{T}$. After constructing the attack training dataset, the adversary trains an NN-based attack model on the constructed training dataset. Following previous works [28,27,10,20], we consider a multi-layer perceptron (MLP) as the neural network architecture for the attack model. Once the attack model is trained, it can be used by the adversary to predict whether a given data sample x is a member or non-member.

Metric-based Attacks. Metric-based attacks [37,32,17,31] also require the adversary to train a shadow model $\mathcal{S}$. Unlike NN-based attacks that require training

an attack model, metric-based attacks design a specific metric and calculate a threshold over the metrics by querying the shadow model $\mathcal{S}$ with $\mathcal{D}^{train}_{shadow}$ and $\mathcal{D}^{test}_{shadow}$. We adopt four state-of-the-art metric-based attacks following Song and Mittal. [31]: (1) Prediction correctness ($\mathcal{A}_{Corr}$) which considers a sample as a member if the label is correctly predicted by the target model; (2) Prediction confidence ($\mathcal{A}_{Conf}$) which judges a sample as a member if the prediction probability at the ground truth class is larger than a pre-defined threshold (learned from the shadow model); (3) prediction entropy ($\mathcal{A}_{Ent}$) which considers a sample as a member if the entropy of the prediction is smaller than a pre-defined threshold (learned from the shadow model); and (4) Modified prediction entropy ($\mathcal{A}_{Ment}$) which is similar to (3) but modifies the entropy function and combines the ground truth label as a new metric.

4 Our Method

The main difference between SSL methods and supervised learning methods is that SSL methods leverage a large amount of unlabeled samples together with a small amount of labeled samples to train the model. Recall that state-of-the-art SSL methods [35,29,39] leverage both weak and strong data augmentations to the unlabeled samples during the training. The key idea of these SSL methods is to train the model that maximizes the model's prediction agreement on weakly and strongly augmented views that come from the same unlabeled sample. In other words, for an unlabeled training sample, the trained model may tend to output more similar posteriors for its weakly and strongly augmented views. While for labeled training samples, the trained model may output similar posteriors for different weakly augmented views from the same sample since those posteriors result in the same predicted label. This observation may also hold for unlabeled samples since the posteriors of the same training unlabeled sample tend to produce the same "pseudo label". Intuitively speaking, the target model $\mathcal{T}$ may output similar (or dissimilar) posteriors for different weakly and/or strongly augmented views of member (or non-member).

Based on the above intuition, we propose a data augmentation-based membership inference attack (denoted as $\mathcal{A}_{DA}$) tailored to state-of-the-art SSL methods. $\mathcal{A}_{DA}$ follows the similar pipeline as NN-based attack $\mathcal{A}_{NN}$, i.e., shadow training and training an NN-based attack model.

However, our attack $\mathcal{A}_{DA}$ extracts membership features (i.e. the input for the attack model) in a different way from the attack $\mathcal{A}_{NN}$. Specifically, given a data sample x, we first generate K weakly augmented and K strongly augmented views of it, respectively. Then we use the augmented views to query the shadow model to obtain output posteriors. After that, we calculate three similarity matrices among: (1) K posteriors of weakly augmented views themselves, (2) K posteriors of strongly augmented views themselves, and (3) K posteriors of weakly augmented views and K posteriors of strongly augmented views, based on a predefined similarity metrics (e.g., JS Distance, Cosine Distance, etc.). Then we obtain three similarity matrices where each of them contains K^2 similarity

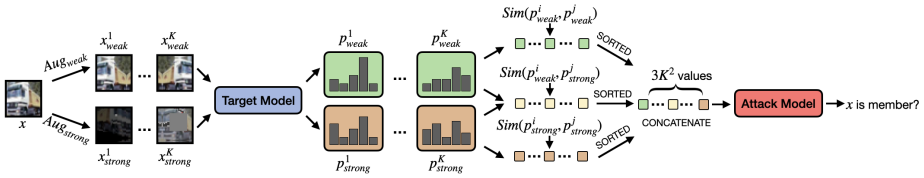


Fig. 1: Overview of our data augmentation based attack $\mathcal{A}_{DA}$.

values. We expand each similarity matrix into a vector and sort the values in each vector in descending order, respectively. Then we concatenate them together, and finally obtain a vector with $3K^2$ values. The obtained vectors are then assigned with the membership status as the labels. Once the attack model is trained, to determine whether a sample belongs to the target model's training dataset, we again generate K weakly and K strongly augmented views of it to query the target model, generate the attack input to query the attack model, and obtain its membership prediction. Figure 1 shows the overview of $\mathcal{A}_{DA}$ and the detailed algorithm is shown in Algorithm 1 in the supplemental material.

5 Evaluation

5.1 Experimental Setup

Dataset Configuration. We evaluate the performance of target models and membership inference attacks on three commonly used SSL datasets, i.e., SVHN, CIFAR10, and CIFAR100. For each dataset, we first randomly split it into four equal parts, i.e., $\mathcal{D}^{train}_{target}$, $\mathcal{D}^{test}_{target}$, $\mathcal{D}^{train}_{shadow}$, and $\mathcal{D}^{test}_{shadow}$. We leverage $\mathcal{D}^{train}_{target}$ to train the target model and consider the samples from $\mathcal{D}^{train}_{target}$ as the members of the target model. Samples in $\mathcal{D}^{test}_{target}$ are considered as the non-members of the target model. $\mathcal{D}^{train}_{shadow}$ is used to build up the shadow model. Both $\mathcal{D}^{train}_{shadow}$ and $\mathcal{D}^{test}_{shadow}$ are used to train the attack model. Note that the $\mathcal{D}^{train}_{target}$ is smaller than the original training dataset (e.g., for CIFAR10, $\mathcal{D}^{train}_{target}$ contains 15,000 samples while the original training dataset contains 50,000 samples), which may lead to lower target model performance.

Metric. We follow previous work [29,36,39,28,10] and adopt testing accuracy as the evaluation metric for target model performance. Regarding the attack, we leverage AUC as the evaluation metric [34,19] as we aim to quantify both the general membership privacy risk for members vs. non-members and the separate privacy risks for labeled/unlabeled members vs. non-members (unbalanced).

Target Model. For a fair comparison, we apply the same hyperparameters for FixMatch, UDA, and FlexMatch. Specifically, we apply SGD optimizer. The initial learning rate is set to 0.03 with a cosine learning rate decay which sets the learning rate to $\eta \cos(\frac{\pi k}{2N})$, where η is the initial learning rate, k is the current training step, and N is the total number of training steps. We set $N = 100 \times 2^{10}$.

We leverage an exponential moving average of model parameters with the momentum of 0.999. The labeled batch size (i.e., the batch size of the labeled data) is set to 64 and the ratio of unlabeled batch size to the labeled batch size is set to 7. Note that the threshold τ is set to 0.8 for UDA and 0.95 for FixMatch and FlexMatch following the original papers. We apply RandAugment [7] as the strong augmentation method in our experiments (see Section 8.1 in the supplementary material). Regarding the model architectures, we leverage Wide ResNet (WRN) [38] with a widen factor of 2 as the target model architecture and we also investigate different widen factors in our ablation studies (see Section 5.6).

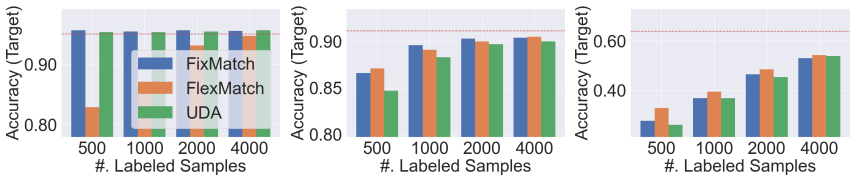
Attack Model. We apply a 3-layer MLP with 64, 32, and 2 hidden neurons for each layer as the attack model's architecture. We train the attack model for 100 epochs using Adam optimizer with the learning rate of 0.001 and the batch size of 256. For our proposed attack, we set the number of augmented views used to query the target model to 10 and leverage JS Distance as the similarity function. Note that we also evaluate different numbers of augmented views and different similarity functions in our ablation studies (see Section 5.5).

5.2 Target Model Performance

We first evaluate the performance of the supervised models and the SSL models on the original classification tasks using $\mathcal{D}^{test}_{target}$. We use the full $\mathcal{D}^{train}_{target}$ to train the supervised models, while we use a small portion of labeled samples and treat the remaining samples as unlabeled ones in $\mathcal{D}^{train}_{target}$ when training the SSL models. We observe that SSL with more labeled samples can achieve better performance on the original classification tasks. For instance, on Figure 2b, when the target model is FixMatch trained on CIFAR10, the classification accuracy is 0.866, 0.896, 0.903, and 0.904 with 500, 1,000, 2,000, and 4,000 labeled samples, respectively. This is expected as more labeled samples help the target model to better learn the decision boundary at the early stage. Another observation is that for a more complicated task, it may require more labeled samples to achieve comparable performance as the supervised models. We consider SVHN, CIFAR10, and CIFAR100 have increasing difficulty levels. Take models trained by UDA as a case study (green bar in Figure 2), on SVHN, with only 500 labeled samples, the testing accuracy is 0.953, which is even better than the supervised model (0.951). We suspect the reason is that 500 labeled samples is enough to learn a relatively accurate decision boundary and the strong data augmentation used in SSL methods can better help the model to generalize to the unseen data. On the other hand, on CIFAR10 and CIFAR100, it may require 1,000 and 4,000 labeled samples to catch up with the performance of the supervised model. Such observation indicates that a larger portion of labeled data is still helpful for a more complicated task.

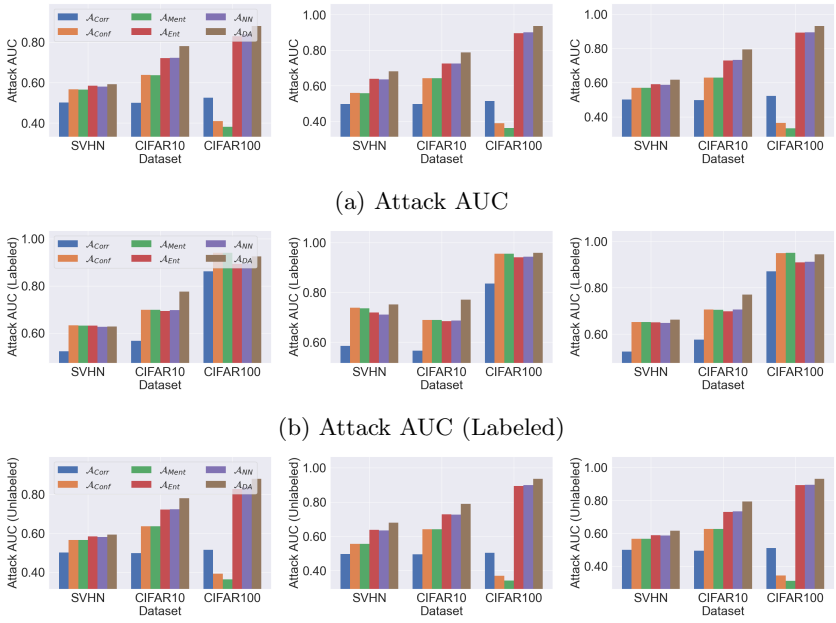
5.3 Membership Inference Attack Performance

We then evaluate the performance of different membership inference attacks on SSL models. The results are summarized in Figure 3. Note that we leverage



(a) SVHN (b) CIFAR10 (c) CIFAR100

Fig. 2: Testing accuracy on the original classification tasks. Note that the red dash line denotes the performance of supervised models.



(a) Attack AUC

(b) Attack AUC (Labeled)

(c) Attack AUC (Unlabeled)

Fig. 3: The AUC of membership inference attacks against models trained by different SSL methods with 500 labeled samples. The first to third columns denote the models trained by FixMatch, FlexMatch, and UDA, respectively.

AUC as the attack evaluation metric to better quantify the privacy leakage of all training data (first row) as well as the separate privacy leakage of labeled (second row) and unlabeled (third row) training data. We find that for the baseline attacks (i.e., except our $\mathcal{A}_{DA}$), $\mathcal{A}_{NN}$ and $\mathcal{A}_{Ent}$ perform the best, while other attacks like $\mathcal{A}_{Corr}$, $\mathcal{A}_{Conf}$, and $\mathcal{A}_{Ment}$ are less effective. For instance, on FlexMatch trained on CIFAR10 with 500 labeled samples (the middle one of Figure 3a), the attack AUC is 0.726 for both $\mathcal{A}_{NN}$ and $\mathcal{A}_{Ent}$, while only

0.497, 0.643, and 0.642 for $\mathcal{A}_{Corr}$, $\mathcal{A}_{Conf}$, and $\mathcal{A}_{Ment}$. To better investigate the reason behind this, we further measure the attack AUC for labeled data and unlabeled data, respectively. We find that $\mathcal{A}_{Conf}$ and $\mathcal{A}_{Ment}$ achieve even better performance on labeled training samples than $\mathcal{A}_{NN}$ and $\mathcal{A}_{Ent}$. For instance, for FlexMatch trained on CIFAR100, the AUC (labeled) for $\mathcal{A}_{Conf}$ and $\mathcal{A}_{Ment}$ are both 0.955, while only 0.944 and 0.941 for $\mathcal{A}_{NN}$ and $\mathcal{A}_{Ent}$. This is expected as the labeled sample has a higher confidence score on its ground-truth label, which facilitates the attacks that leverage such information. However, this is not the case for the unlabeled samples. As we can observe that, for FlexMatch trained on CIFAR100, the AUC (unlabeled) is only 0.370 and 0.341 for $\mathcal{A}_{Conf}$ and $\mathcal{A}_{Ment}$, but 0.899 and 0.894 for $\mathcal{A}_{NN}$ and $\mathcal{A}_{Ent}$. This indicates that, for the unlabeled samples, the model may give similar correctness predictions on both unlabeled training samples and testing samples, which makes it harder to differentiate them. However, the model will give more confident predictions on unlabeled training samples than on testing samples, which results in better performance for $\mathcal{A}_{NN}$ and $\mathcal{A}_{Ent}$.

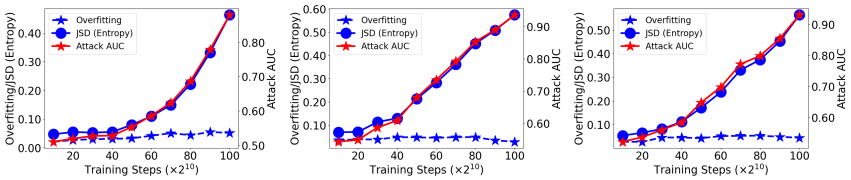
On the other hand, we also observe that our proposed data augmentation-based attack $\mathcal{A}_{DA}$ achieves consistently better overall performance on all datasets and SSL methods than those baseline attacks. Moreover, $\mathcal{A}_{DA}$ works better in determining the membership of unlabeled training samples. For instance, on FixMatch trained on CIFAR10, the unlabeled AUC is 0.780 for $\mathcal{A}_{DA}$ while only 0.722 for the best baseline attack ($\mathcal{A}_{NN}$). This is because $\mathcal{A}_{DA}$ unveils the pattern that the predictions of a sample's weak and strong augmented views should be closer if the sample is an unlabeled sample used during the training.

5.4 What Determines Membership Inference Attack in SSL

The effectiveness of membership inference attacks has been largely credited to the intrinsic overfitting phenomenon of the ML model [28,27]. Here overfitting denotes the model's training accuracy minus its testing accuracy. Such assumption has been verified on various ML models [28,22,27,10]. However, it is unclear whether such assumption still holds for SSL. If not, what is the reason for models trained by SSL to be vulnerable to membership inference attacks?

From Figure 3, we find that $\mathcal{A}_{Ent}$ achieves good performance in predicting the membership status of a sample, which gives us the hint that the members' and non-members' predictions may have different entropy distributions. Here we leverage the JS Distance to quantify the difference between the entropy distribution of members' and non-members' predictions (we denote this measure as JS Distance (Entropy)).

To better quantify the correlation between different factors (e.g., overfitting, JS Distance (Entropy)) and the attack performance, we measure them under different training steps of the target models. Note that here we consider the $\mathcal{A}_{DA}$ as it performs the best in membership inference. Figure 4 shows the results of models trained by different SSL methods on the CIFAR100 with 500 labeled samples. The results for models trained on different datasets and with different numbers of labeled samples are shown in Section 8.3 in supplementary materials.



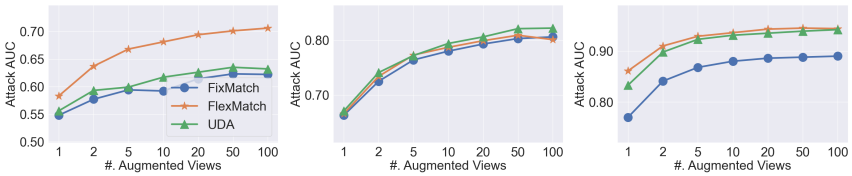
(a) FixMatch (b) FlexMatch (c) UDA

Fig. 4: The overfitting/JS Distance (Entropy) and attack AUC with respect to different training steps. The target model is trained on CIFAR100 with 500 labeled samples. Note that we consider the attack AUC of $\mathcal{A}_{DA}$, which is the strongest attack.

In Figure 4, we observe that during the whole training procedure, the models trained by SSL have nearly 0 overfitting, which means that the models can always generalize well to the unseen data. However, we find that the attack AUC keeps increasing during the training. This indicates that the success of membership inference attacks is not necessarily related to the high overfitting level, which is overlooked by previous research. On the other hand, we observe that the JS Distance (Entropy) does increase during the training, which means that although the model does not predict more accurately to the member samples (mainly unlabeled samples) than the non-member samples, the model indeed makes a more confident prediction on member samples (i.e., with lower entropy of prediction). Our observation reveals that the models trained by SSL indeed "memorize" the training data. However, such memorization does not reflect in the overfitting, i.e., the gap between training and testing accuracy. Instead, it reflects in the more confident prediction of the members than the non-members.

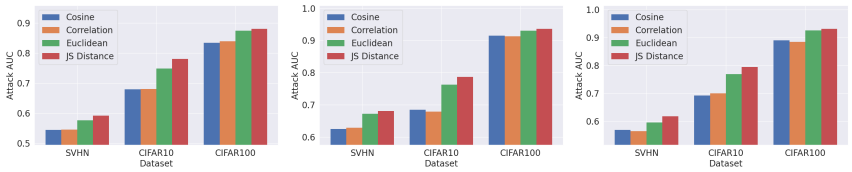
5.5 Ablation Study (Attack Model)

Number of Views. We first investigate how the attack performance would be affected by different numbers of views generated by the weak and strong augmentations to query the target model. To this end, for the SSL methods trained on different datasets with only 500 labeled samples, we range the number of views from 1 to 100 and the attack performance is shown in Figure 5. Note that we also show the results with 1,000, 2,000, and 4,000 labeled samples in Section 8.4 in the supplementary material. A clear trend is that more views lead to better attack performance. For instance, for FixMatch trained on CIFAR10 with 500 labeled samples (Figure 5b), the attack AUC is 0.780 with 10 augmented views, while 0.806 for 100 augmented views. However, we find that the attack performance increases rapidly when the number of augmented views increases from 1 to 10, but plateaus from 10 to 100. Moreover, more views lead to more queries to the target model and higher computational cost. We consider 10 as a suitable number of views since it achieves comparable performance to 100 while spending less query budget.



(a) SVHN (b) CIFAR10 (c) CIFAR100

Fig. 5: The attack AUC of $\mathcal{A}_{DA}$ with different numbers of augmented views to query the target model. The target model is trained with 500 labeled samples.



(a) FixMatch (b) FlexMatch (c) UDA

Fig. 6: The attack AUC of $\mathcal{A}_{DA}$ with different similarity functions. The target model is trained with 500 labeled samples.

Similarity Function. Note that in our attack $\mathcal{A}_{DA}$, we can apply different similarity functions to measure the distance between the posteriors generated from different augmented views. Here we evaluate 4 distance metrics, i.e., Cosine Distance, Correlation Distance, Euclidean Distance, and JS Distance. The result for FixMatch, FlexMatch, and UDA trained on three different datasets with 500 labeled samples are summarized in Figure 6. Note that we also show the results with 1,000, 2,000, and 4,000 labeled samples in Section 8.5 in the supplementary material. We find that JS Distance consistently outperforms the other three distance metrics and achieves the best performance. For instance, FixMatch trained on CIFAR10, the attack AUC is 0.679, 0.682, 0.749, and 0.780 for Cosine Distance, Correlation Distance, Euclidean Distance, and JS Distance. We suspect the reason is that JS Distance is designed to calculate the difference between two probabilities' distributions, which may better fit our scenario as the prediction posteriors are probability as well.

Moreover, we also find that the magnitude of data augmentation and the shadow model architecture only have limited impact on the attack performance (see Section 8.6 in the supplementary material for more details).

5.6 Ablation Study (Target Model)

We also investigate whether the target model's capacity and the unlabeled ratio (i.e., $\frac{batchsize(unlabeled)}{batchsize(labeled)}$ during each training step) would affect the performance. Note that here we select FixMatch trained on CIFAR100 with 500 labeled data

Table 1: The target model performance and attack performance ($\mathcal{A}_{DA}$) when the target model has different capacities. The target model is trained by FixMatch on CIFAR100 with 500 labeled samples. ($\star$) denotes the default setting.

Architecture	Test ACC	Attack AUC	Attack AUC (Labeled)	Attack AUC (Unlabeled)
WRN28-1	0.217	0.726	0.954	0.718
WRN28-2 ($\star$)	0.276	0.874	0.896	0.873
WRN28-4	0.299	0.917	0.910	0.917
WRN28-8	0.305	0.927	0.918	0.927

Table 2: The target model performance and attack performance ($\mathcal{A}_{DA}$) when the target model leverages different unlabeled ratios during each training step. The target model is trained by FixMatch on CIFAR100 with 500 labeled samples. ($\star$) denotes the default setting.

Ratio	Test Acc	Attack AUC	Attack AUC (Labeled)	Attack AUC (Unlabeled)
1	0.210	0.578	0.965	0.565
2	0.263	0.646	0.942	0.636
4	0.273	0.785	0.946	0.779
7 ($\star$)	0.276	0.874	0.896	0.873
8	0.269	0.886	0.924	0.884
16	0.247	0.909	0.913	0.909

as a case study, since the target model's capacity and the unlabeled ratio are general to different SSL methods, and CIFAR100 with 500 labeled data is the most challenging setting to train the target model (see Figure 2). We consider an adaptive adversary [14] who is aware of the training details of the target model and can train the shadow model in the same way.

Model Capacity. The target model architecture we leverage in our paper is WRN28-2. To better quantify the impact of model capacity on the target and attack performance, we vary the width of WRN28 from 1 to 8 and the results are shown in Table 1. We can observe that a larger model capacity, in general, leads to a better target model's performance on the original classification task, but also increases the membership risk (especially for unlabeled samples). For instance, when the model capacity increase from WRN28-1 to WRN-28-8, the target testing accuracy increases from 0.217 to 0.305, while the attack AUC increases from 0.726 to 0.927. One reason is that, with larger model capacity, the model can "memorize" more different views of data samples, which not only facilitate target tasks, but also raise the membership risk.

Ratio of Unlabeled Samples in Each Training Step. We then investigate whether the unlabeled ratio (URatio) during each training step affects the attack performance. Concretely, we vary the unlabeled ratio from 1 to 16 while training the target model and Table 2 summarizes the results. We have two findings. First, the best target model performance reaches with the default setting (7).

Second, the membership inference risk, in particular for the unlabeled data, keeps increasing when the ratio increases. On the other hand, the membership inference risk for labeled data slightly decreases (but still in a high level) while increasing the ratio. Therefore, a better choice may be leveraging a relatively small unlabeled ratio to achieve good target performance while reducing the membership risk for unlabeled samples.

6 Discussion on Defenses

We observe that the attack performance increases sharply at the late training steps (see Figure 4), which indicates that early stopping may be a good strategy to mitigate membership inference attacks. We take CIFAR100 with 4,000 labeled samples as a case study and show the target/attack model performance with respect to different training steps in Figure 7 (in the supplementary material). We find that there is a trade-off between model utility and membership inference performance, i.e., it may reduce both the attack performance and the target model's utility. We note that previous work [20,31] also observe such a trade-off. Besides early stopping, we also evaluate three other defenses, i.e., top-k posteriors [28], model stacking [27], and DP-SGD [2]. Our case study (see Section 8.7 in the supplementary material) shows that early stopping achieves the best trade-off between model utility and membership inference performance.

7 Conclusion

In this paper, we perform the first training data privacy quantification against models trained by SSL through the lens of membership inference attack. Empirical evaluation shows that our proposed data augmentation-based attacks consistently outperform the baseline attacks, in particular for unlabeled training data. Moreover, we have an interesting finding that the reason leading to membership leakage in SSL is different from the commonly believed overfitting nature of ML models trained in supervised manners. The models trained by SSL are well generalized to the testing data (i.e., with almost 0 overfitting level). However, our attack can still successfully break the membership privacy. The reason is that the models trained by SSL "memorize" the training data by giving more confident predictions on them, regardless of the ground truth labels. We also find that early stopping can serve as a countermeasure against the attacks, but there is a trade-off between membership privacy and model utility.

Acknowledgments: This work is partially funded by the Helmholtz Association within the project "Trustworthy Federated Data Analytics" (TFDA) (funding number ZT-I-OO1 4) and National Science Foundation grant No. 1937786.

References

1. https://github.com/pytorch/opacus
2. Abadi, M., Chu, A., Goodfellow, I., McMahan, B., Mironov, I., Talwar, K., Zhang, L.: Deep Learning with Differential Privacy. In: ACM SIGSAC Conference on Computer and Communications Security (CCS). pp. 308–318. ACM (2016)
3. Berthelot, D., Carlini, N., Goodfellow, I.J., Papernot, N., Oliver, A., Raffel, C.: MixMatch: A Holistic Approach to Semi-Supervised Learning. In: Annual Conference on Neural Information Processing Systems (NeurIPS). NeurIPS (2019)
4. Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., Tramèr, F.: Membership Inference Attacks From First Principles. CoRR abs/2112.03570 (2021)
5. Chen, D., Yu, N., Zhang, Y., Fritz, M.: GAN-Leaks: A Taxonomy of Membership Inference Attacks against Generative Models. In: ACM SIGSAC Conference on Computer and Communications Security (CCS). pp. 343–362. ACM (2020)
6. Choo, C.A.C., Tramèr, F., Carlini, N., Papernot, N.: Label-Only Membership Inference Attacks. In: International Conference on Machine Learning (ICML). pp. 1964–1974. PMLR (2021)
7. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.: RandAugment: Practical Automated Data Augmentation with a Reduced Search Space. In: Annual Conference on Neural Information Processing Systems (NeurIPS). pp. 18613–18624. NeurIPS (2020)
8. Fredrikson, M., Jha, S., Ristenpart, T.: Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. In: ACM SIGSAC Conference on Computer and Communications Security (CCS). pp. 1322–1333. ACM (2015)
9. He, X., Wen, R., Wu, Y., Backes, M., Shen, Y., Zhang, Y.: Node-Level Membership Inference Attacks Against Graph Neural Networks. CoRR abs/2102.05429 (2021)
10. He, X., Zhang, Y.: Quantifying and Mitigating Privacy Risks of Contrastive Learning. In: ACM SIGSAC Conference on Computer and Communications Security (CCS). pp. 845–863. ACM (2021)
11. He, Y., Rahimian, S., Schiele1, B., Fritz, M.: Segmentations-Leak: Membership Inference Attacks and Defenses in Semantic Image Segmentation. In: European Conference on Computer Vision (ECCV). pp. 519–535. Springer (2020)
12. Hui, B., Yang, Y., Yuan, H., Burlina, P., Gong, N.Z., Cao, Y.: Practical Blind Membership Inference Attack via Differential Comparisons. In: Network and Distributed System Security Symposium (NDSS). Internet Society (2021)
13. Jayaraman, B., Evans, D.: Evaluating Differentially Private Machine Learning in Practice. In: USENIX Security Symposium (USENIX Security). pp. 1895–1912. USENIX (2019)
14. Jia, J., Salem, A., Backes, M., Zhang, Y., Gong, N.Z.: MemGuard: Defending against Black-Box Membership Inference Attacks via Adversarial Examples. In: ACM SIGSAC Conference on Computer and Communications Security (CCS). pp. 259–274. ACM (2019)
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks. In: Annual Conference on Neural Information Processing Systems (NIPS). pp. 1106–1114. NIPS (2012)
16. Lee, D.H.: Pseudo-Label: The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks. In: ICML Workshop on Challenges in Representation Learning (WREPL). ICML (2013)
17. Leino, K., Fredrikson, M.: Stolen Memories: Leveraging Model Memorization for Calibrated White-Box Membership Inference. In: USENIX Security Symposium (USENIX Security). pp. 1605–1622. USENIX (2020)

18. Li, J., Li, N., Ribeiro, B.: Membership Inference Attacks and Defenses in Classification Models. In: ACM Conference on Data and Application Security and Privacy (CODASPY). pp. 5–16. ACM (2021)
19. Li, Z., Zhang, Y.: Membership Leakage in Label-Only Exposures. In: ACM SIGSAC Conference on Computer and Communications Security (CCS). pp. 880–895. ACM (2021)
20. Liu, H., Jia, J., Qu, W., Gong, N.Z.: EncoderMI: Membership Inference against Pre-trained Encoders in Contrastive Learning. In: ACM SIGSAC Conference on Computer and Communications Security (CCS). ACM (2021)
21. Miyato, T., Maeda, S., Koyama, M., Ishii, S.: Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. IEEE Transactions on Pattern Analysis and Machine Intelligence (2019)
22. Nasr, M., Shokri, R., Houmansadr, A.: Machine Learning with Membership Privacy using Adversarial Regularization. In: ACM SIGSAC Conference on Computer and Communications Security (CCS). pp. 634–646. ACM (2018)
23. Nasr, M., Shokri, R., Houmansadr, A.: Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning. In: IEEE Symposium on Security and Privacy (S&P). pp. 1021–1035. IEEE (2019)
24. Oh, S.J., Augustin, M., Schiele, B., Fritz, M.: Towards Reverse-Engineering Black-Box Neural Networks. In: International Conference on Learning Representations (ICLR) (2018)
25. Pyrgelis, A., Troncoso, C., Cristofaro, E.D.: Knock Knock, Who's There? Membership Inference on Aggregate Location Data. In: Network and Distributed System Security Symposium (NDSS). Internet Society (2018)
26. Rahimian, S., Orekondy, T., Fritz, M.: Differential Privacy Defenses and Sampling Attacks for Membership Inference. In: PriML Workshop (PriML). NeurIPS (2020)
27. Salem, A., Zhang, Y., Humbert, M., Berrang, P., Fritz, M., Backes, M.: ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. In: Network and Distributed System Security Symposium (NDSS). Internet Society (2019)
28. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership Inference Attacks Against Machine Learning Models. In: IEEE Symposium on Security and Privacy (S&P). pp. 3–18. IEEE (2017)
29. Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C., Cubuk, E.D., Kurakin, A., Li, C.: FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. In: Annual Conference on Neural Information Processing Systems (NeurIPS). NeurIPS (2020)
30. Song, C., Shmatikov, V.: Overlearning Reveals Sensitive Attributes. In: International Conference on Learning Representations (ICLR) (2020)
31. Song, L., Mittal, P.: Systematic Evaluation of Privacy Risks of Machine Learning Models. In: USENIX Security Symposium (USENIX Security). USENIX (2021)
32. Song, L., Shokri, R., Mittal, P.: Privacy Risks of Securing Machine Learning Models against Adversarial Examples. In: ACM SIGSAC Conference on Computer and Communications Security (CCS). pp. 241–257. ACM (2019)
33. Wang, B., Gong, N.Z.: Stealing Hyperparameters in Machine Learning. In: IEEE Symposium on Security and Privacy (S&P). pp. 36–52. IEEE (2018)
34. Watson, L., Guo, C., Cormode, G., Sablayrolles, A.: On the Importance of Difficulty Calibration in Membership Inference Attacks. CoRR abs/2111.08440 (2021)

35. Xie, Q., Dai, Z., Du, Y., Hovy, E.H., Neubig, G.: Controllable Invariance through Adversarial Feature Learning. In: Annual Conference on Neural Information Processing Systems (NIPS). pp. 585–596. NIPS (2017)
36. Xie, Q., Dai, Z., Hovy, E.H., Luong, T., Le, Q.: Unsupervised Data Augmentation for Consistency Training. In: Annual Conference on Neural Information Processing Systems (NeurIPS). NeurIPS (2020)
37. Yeom, S., Giacomelli, I., Fredrikson, M., Jha, S.: Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. In: IEEE Computer Security Foundations Symposium (CSF). pp. 268–282. IEEE (2018)
38. Zagoruyko, S., Komodakis, N.: Wide Residual Networks. In: Proceedings of the British Machine Vision Conference (BMVC). BMVA Press (2016)
39. Zhang, B., Wang, Y., Hou, W., Wu, H., Wang, J., Okumura, M., Shinozaki, T.: FlexMatch: Boosting Semi-Supervised Learning with Curriculum Pseudo Labeling. In: Annual Conference on Neural Information Processing Systems (NeurIPS). NeurIPS (2021)