LOSSCONTROL: DEFENDING MEMBERSHIP INFERENCE ATTACKS BY CONTROLLING THE LOSS SUPPLEMENTARY MATERIAL

Editors: Vu Nguyen and Hsuan-Tien Lin

1. INTRODUCTION

This document provides supplementary material complementing the main paper.

Appendix A. DESINE INTUTION

Specific potent membership inference attacks leverage distinctions in model behavior between member and non-member samples, notably manifesting as substantial disparities in loss values (i.e., loss distribution). This discrepancy can be succinctly expressed as the difference between expected loss values: $\mathbb{E}[\ell]_{non} - \mathbb{E}[\ell]_{mem}$.

In the realm of machine learning, leveraging the posterior distribution of parameters θ conditioned on observations $z_1, z_2, \cdots, z_n$, the scheme (Sablayrolles et al., 2019) derives an optimal Bayesian attack strategy. This strategy is intricately linked solely to sample loss and is succinctly captured by the following formula:

$$\mathcal{A}_{opt}(f(\cdot, \boldsymbol{\theta}), \boldsymbol{z}_i) = \mathbb{I}[-\ell(\boldsymbol{\theta}, \boldsymbol{z}_i) > \tau(\boldsymbol{z}_i)], \tag{1}$$

where τ represents a threshold function. Notably, it is imperative to highlight that this insightful conclusion is predicated upon mild assumptions. Intuitively, Eq.(1) underscores that $\boldsymbol{z}_i$ is more likely to be utilized for training if the target model showcases minimal loss values on it. From the conclusion above, it becomes evident that this attack strategy does not necessitate the model's internal parameters, characteristic of a white-box attack. Essentially, this implies that the white-box setting does not confer any added advantages compared to black-box membership inference. This observation significantly substantiates why the attack method detailed in the paper Choquette-Choo et al. (2021) can succeed, even when the adversary only gains access to the model's prediction category, devoid of the prediction vector or loss value. These findings underpin our approach to mitigate MIAs by diminishing the discernibility between member and non-member loss distributions.

Leveraging the insights drawn from the abovementioned analysis, prior research (Chen et al., 2022) has employed these findings to develop an initial defense framework known as RelaxLoss. Notably, this framework marks the pioneering effort to safeguard member privacy effectively without undermining the model's utility.

Appendix B. THEORETICAL ANALYSIS

In this section, we illustrate through theoretical analysis: 1) how the soft labels generated during the Soft-label Training limit the model's overconfidence in training samples, ensuring the correct training of the model. Specifically, the confidence in the correct class is no longer close to 1 but is constrained by the hyperparameter β, thereby controlling the lower limit of the training sample loss to a level greater than 0. 2) how the two-stage model update process involved in Loss Ascent increases the loss of training samples, indirectly controlling the mean loss of samples within the training framework.

Theorem 1 *For a given sample $\boldsymbol{z}_i$ paired with its corresponding soft label $\boldsymbol{t}_i$, the cross-entropy loss, denoted as $\mathcal{L}_{CE}(\boldsymbol{s}_i, \boldsymbol{t}_i)$, attains its minimum value when the model's predicted vector $\boldsymbol{s}_i$ aligns perfectly with the provided soft label $\boldsymbol{t}_i$.*

Proof Since $\sum_{c=1}^{C} t_i^c = 1$ and $\sum_{c=1}^{C} s_i^c = 1$, both vectors $\boldsymbol{t}_i$ and $\boldsymbol{s}_i$ can be represented as probability distributions. We express the Kullback-Leibler (KL) divergence between these distributions as follows:

$$\begin{aligned} D_{KL}(\boldsymbol{t}_i \| \boldsymbol{s}_i) &= \sum_{c=1}^{C} t_i^c \log \frac{t_i^c}{s_i^c} && (2) \\ &= \sum_{c=1}^{C} t_i^c \log t_i^c - \sum_{c=1}^{C} t_i^c \log s_i^c && (3) \\ &= \sum_{c=1}^{C} t_i^c \log t_i^c + \mathcal{L}_{CE}(\boldsymbol{s}_i, \boldsymbol{t}_i) && (4) \end{aligned}$$

Here, $D_{KL}(\boldsymbol{t}_i \| \boldsymbol{s}_i)$ denotes the KL divergence between the distributions $\boldsymbol{t}_i$ and $\boldsymbol{s}_i$. Since $\sum_{c=1}^{C} t_i^c \log t_i^c$ is constant, the non-negativity of the KL divergence implies that the minimum value of $\mathcal{L}_{CE}(\boldsymbol{s}_i, \boldsymbol{t}_i)$ occurs when the KL divergence is minimized. The KL divergence is minimized at $\boldsymbol{s}_i = \boldsymbol{t}_i$, resulting in a minimum value of 0. Therefore, $\mathcal{L}_{CE}(\boldsymbol{s}_i, \boldsymbol{t}_i)$ achieves its minimum when $\boldsymbol{s}_i = \boldsymbol{z}_i$. ■

The implications of Theorem 1 highlight a distinction from conventional machine learning training, which centers on ground truth labels. The learning objective introduced by Soft-label Training acts as a safeguard, preventing the model from unwarranted overconfidence in its training samples. Even in scenarios prone to overfitting, the model will only produce predictions with maximum confidence of β, ensuring that the model's loss on training samples reaches a minimum threshold. This stands in contrast to traditional machine learning models, which would exhibit predictions with maximum confidence of 1, potentially leading to increased overconfidence and suboptimal generalization.

In this scenario, considering the sample $\boldsymbol{z}_i$, the loss derived from the ground truth labels is denoted as $\mathcal{L}_{CE}(\boldsymbol{s}_i, \boldsymbol{y}_i) = -\log \beta$. This implies that through Soft-label Training, the lower boundary of the training sample's loss is controlled, and the loss value escalates with a decrease in β.

Theorem 2 *Given a model parameter $\boldsymbol{\theta}$, a confidence threshold β, and a sample $\boldsymbol{z}_i$, executing gradient ascent on $\boldsymbol{z}_i$ results in an elevation of the loss $\mathcal{L}_{CE}(\boldsymbol{s}_i, \boldsymbol{y}_i)$ concerning the ground truth label $\boldsymbol{y}_i$.*

Proof To illustrate the gradient ascent impact, let us assume, without loss of generality, that $\boldsymbol{y_i} \in \mathbb{R}^3$, and focus on the scenario where sample $\boldsymbol{z}_i$ belongs to the first class. The gradient of the cross-entropy loss function based on soft labels can be expressed as:

$$\begin{aligned}
\nabla \mathcal{L}_{CE}(\boldsymbol{s}_i, \boldsymbol{t}_i) &= J_{\boldsymbol{\theta_i}}(\boldsymbol{s_i} - \boldsymbol{t_i}) && (5)\\
&= J_{\boldsymbol{\theta_i}} \begin{pmatrix} s_i^1 - \beta \\ s_i^2 - (1-\beta)/2 \\ s_i^3 - (1-\beta)/2 \end{pmatrix} && (6)\\
&= \beta \times J_{\boldsymbol{\theta_i}} \begin{pmatrix} \frac{s_i^1}{\beta} - 1 \\ 0 \\ 0 \end{pmatrix} + J_{\boldsymbol{\theta_i}} \begin{pmatrix} 0 \\ s_i^2 - (1-\beta)/2 \\ s_i^3 - (1-\beta)/2 \end{pmatrix} && (7)
\end{aligned}$$

Here, $J_{\boldsymbol{\theta_i}}$ denotes the Jacobian matrix of logits (the model's output before the final softmax layer) with respect to model parameters $\boldsymbol{\theta}$. Assuming $s_i^1 \leq \beta$, signifying the model's confidence in the correct class is below the confidence threshold, the gradient can be decomposed per class. The component related to the increase in loss based on the ground truth label is in the direction of the first class. This aligns with the direction of the loss increase for the sample's ground truth label, considering $p_i^1/\beta \in [0, 1]$, and the β term acts as part of the learning rate. ■

When updating model parameters along the gradient ascent derived from the cross-entropy loss using soft labels, there is a component that contributes to an elevated gradient of the cross-entropy loss based on the ground truth label. This implies that executing gradient ascent on $\boldsymbol{z}_i$ will result in an increased loss $\mathcal{L}_{CE}(\boldsymbol{s}_i, \boldsymbol{y}_i)$ with respect to the ground truth label $\boldsymbol{y}_i$.

Theorem 3 *Given a model parameter $\boldsymbol{\theta}$, a sample $\boldsymbol{z}_i$, and a temperature coefficient T where $T > 1$, let $\boldsymbol{p}_i$ and $\boldsymbol{s}_i$ represent the logit and predicted vectors corresponding to $\boldsymbol{z}_i$. If* $\arg\max \boldsymbol{p_i} = y$, *indicating correct model prediction, then the cross-entropy loss based on the ground truth label satisfies $\mathcal{L}_{CE}(\mathit{Softmax}(\boldsymbol{p}_i, 1), \boldsymbol{y}_i) < \mathcal{L}_{CE}(\mathit{Softmax}(\boldsymbol{p}_i, T), \boldsymbol{y}_i)$.*

Proof We begin by establishing the inequality:

$$\begin{aligned}
&\mathcal{L}_{CE}(\text{Softmax}(\boldsymbol{p}_i, 1), \boldsymbol{y}_i) < \mathcal{L}_{CE}(\text{Softmax}(\boldsymbol{p}_i, T), \boldsymbol{y}_i)\\
\Rightarrow &\text{Softmax}(\boldsymbol{p}_i, 1)^y > \text{Softmax}(\boldsymbol{p}_i, T)^y\\
\Rightarrow &\frac{\exp(p_i^y)}{\sum_{c=1}^{C} \exp(p_i^c)} > \frac{\exp\left(\frac{p_i^y}{T}\right)}{\sum_{c=1}^{C} \exp\left(\frac{p_i^c}{T}\right)} && (8)
\end{aligned}$$

Now, let's consider Eq.(8). Defining $\text{Softmax}(\boldsymbol{p}_i, T)^y$ as $f(T)$, we aim to demonstrate that $f(T)$ decreases as T increases, i.e., for $T \in [1, \infty)$, $f(T)$ is a decreasing function.

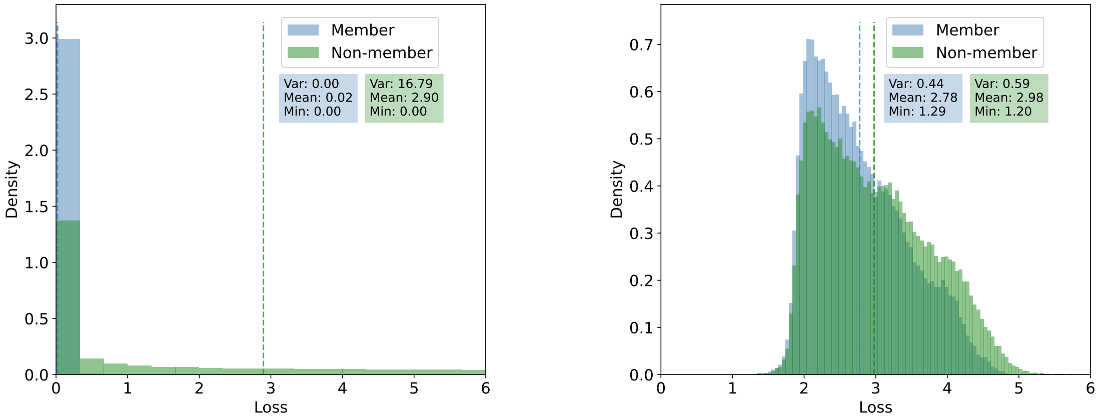
$$
\begin{aligned}
f(T)' = & \frac{-\exp\left(\frac{p_i^y}{T}\right) \times \frac{p_i^y}{T^2} \times \sum_{c=1}^{C} \exp\left(\frac{p_i^c}{T}\right)}{\left(\sum_{c=1}^{C} \exp\left(\frac{p_i^c}{T}\right)\right)^2} \quad + \\
& \frac{\exp\left(\frac{p_i^y}{T}\right) \times \sum_{c=1}^{C} \left[\frac{p_i^c}{T^2} \times \exp\left(\frac{p_i^c}{T}\right)\right]}{\left(\sum_{c=1}^{C} \exp\left(\frac{p_i^c}{T}\right)\right)^2} \\
= & \frac{\exp\left(\frac{p_i^y}{T}\right) \times \sum_{c=1}^{C} \left[\frac{p_i^c}{T^2} \times \exp\left(\frac{p_i^c}{T}\right) - \frac{p_i^y}{T^2} \times \exp\left(\frac{p_i^c}{T}\right)\right]}{\left(\sum_{c=1}^{C} \exp\left(\frac{p_i^c}{T}\right)\right)^2} \\
= & \frac{\exp\left(\frac{p_i^y}{T}\right) \times \sum_{c=1}^{C} \left[\left(\frac{p_i^c}{T^2} - \frac{p_i^y}{T^2}\right) \times \exp\left(\frac{p_i^c}{T}\right)\right]}{\left(\sum_{c=1}^{C} \exp\left(\frac{p_i^c}{T}\right)\right)^2} \\
< & \; 0
\end{aligned}
\tag{9}
$$

Since $\arg\max \boldsymbol{p_i} = y$ with $p_i^y \geq p_i^c$, the numerator in Eq.(9) is negative, indicating $f(T)$ is indeed a decreasing function. ■

This theorem elucidates that in cases where the model makes a correct prediction, employing Softmax with a temperature $T > 1$ results in the generation of soft labels that diminish the confidence of the training sample in the correct class. According to Theorem 1, the learning process guided by these soft labels drives the predicted vector $\boldsymbol{p_i}$ towards the provided soft label $\boldsymbol{t_i}$, consequently escalating the cross-entropy loss based on the ground truth labels.

Appendix C. ANALYTICAL INSIGHTS

In this section, we analyze the fundamental properties that explain the effectiveness of LossControl. We provide empirical evidence that LossControl can 1) indeed indirectly affect the loss distribution by limiting the loss lower bound and controlling the loss mean, 2) reduce the generalization gap, and 3) shorten the distance between member samples and the prediction boundary, all of which help alleviate MIAs.

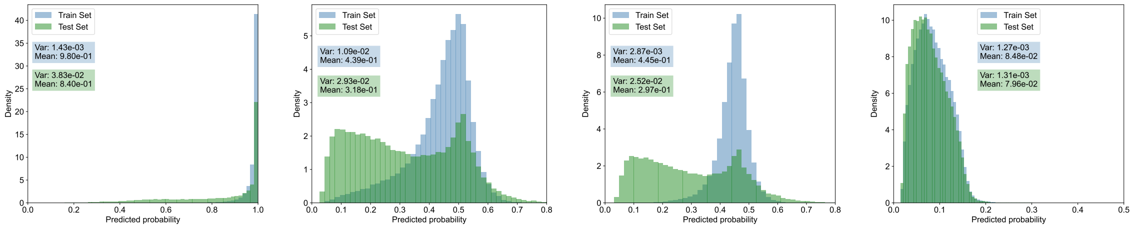


(a) Vanilla

(b) LossControl

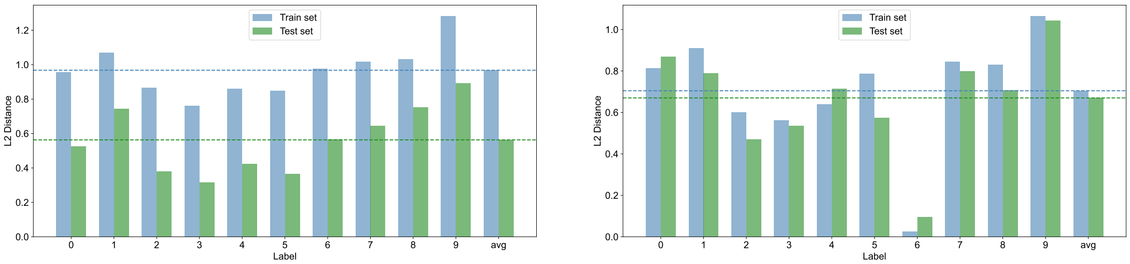
Figure 2: Loss histograms on Texas100 when applying (a) vanilla training and (b) LossControl, where α=0.5 and β=2.8. The empirical mean and minimum values of the loss distribution are depicted in the figure. The AUC resulting from a loss thresholding attack equals 0.78 in (a) and 0.57 in (b). Our observations indicate that our method effectively matches the target mean and constrains the lower bound of loss (as discussed in Section 4.1)

LossControl regulate the loss distribution We compare our LossControl with vanilla training on the Texas100 dataset and visualize the resulting loss histograms in Figure 2. Upon examining Figure 2(b), our framework successfully regulates the average loss values for members and non-members, maintaining them at 2.78 and 2.98, respectively. The minimum loss values are also constrained to 1.29 for members and 1.20 for non-members. In contrast, the outcomes depicted in Figure 2(a) reveal that vanilla training results in an average loss of 0.02 for members, with a minimum value close to 0. Meanwhile, the average loss for non-members is 2.90, and the minimum value is also close to 0.



(a) Vanilla (b) Label smoothing (c) Confidence penalty (d) LossControl

Figure 3: Prediction confidence histograms on the Texas100 when applying (a) Vanilla Training, (b) Label Smoothing with $\alpha = \mathbf{0.5}$, (c) Confidence Penalty with $\alpha = \mathbf{0.5}$, and (d) LossControl with $\alpha = \mathbf{2.8}$ and $\beta = \mathbf{0.7}$. The empirical mean and variance of the loss distribution are also depicted in the figure.



(a) Vanilla (b) LossControl

Figure 4: L_2 distance on CIFAR-10 with ResNet20 architecture when applying (a) vanilla training and (b) LossControl, where L_2 distance represents the L_2 distance between the original sample and its perturbed sample generated by the HopSkipJump attack. The x-axis represents different categories, 'avg' represents the average value across all categories, and the y-axis represents the L_2 distance.

LossControl reduce the generalization gap We compare our LossControl with vanilla training, Label-smoothing and Confidence-penalty on the Texas100 dataset and visualize the resulting prediction confidence histogram in Figure 3. The outcomes from Figure 3(b) and Figure 3(c) suggest that Label-smoothing and Confidence-penalty, functioning as regularization techniques, effectively address overfitting in the training set by incorporating regularization terms into the training objective. These techniques prevent excessively confident predictions or regularize the model by penalizing low-entropy output distributions. Notably, both techniques display remarkably similar prediction confidence distributions, affirming the conceptual similarities in their designs. Furthermore, our approach, with more achievable learning objectives, blurs the distinction between member and non-member prediction confidence distributions compared to the aforementioned regularization techniques. This results in smaller means and more analogous distributions between the training and test sets, naturally leading to a narrowed generalization gap and reduced privacy leakage (Yeom et al., 2018).

LossControl shorten the distance between member samples and the prediction boundary We apply LossControl on the CIFAR-10 dataset using the ResNet20 architecture and visualize the resulting L_2 distances in Figure 4. Precisely, we measure the L_2 distance between adversarial samples generated by the HopSkipJump attack and the original samples for each category in both the training and test sets. This distance metric characterizes the proximity of samples to the decision boundary, indicating potential vulnerabilities to MIAs, as proposed by Choquette-Choo et al. (2021); Li and Zhang (2021). Observations reveal that, whether for specific categories or overall, vanilla training exhibits a significantly larger L_2 distance on the training set compared to the test set. Our approach effectively narrows the distance gap between the training and test sets. Notably, in some cases (such as Category 0, 4, and 6), the L_2 distance on the training set is even smaller than on the test set. These results suggest that our defense mechanism successfully mitigates boundary attacks.

Appendix D. EXPERIMENT SETUP

D.1. Datasets

CIFAR-10 CIFAR-10 Krizhevsky et al. (2009) is a dataset comprising 60,000 color images, each sized at 32 × 32 pixels and containing 3 color channels. These images cover 10 different categories of objects, with each image labeled to indicate its category. During standard preprocessing [1], the pixel values of the images are normalized to have zero mean and unit standard deviation, ensuring they are better suited for training models and performing image classification tasks.

CIFAR-100 CIFAR-100 Krizhevsky et al. (2009) is a dataset containing 60,000 color images, each measuring 32×32 pixels with 3 color channels, spread across 100 different classes. Similar to the preprocessing done for the CIFAR-10 dataset, we apply mean subtraction and standardization to the pixel values of these images.

Texas100 Texas100 dataset comprises medical data from 67,330 patients made available by the Texas Department of State Health Services [2]. Each patient's record is characterized by 6,169 binary attributes, encompassing details like diagnoses, general information, and procedures undertaken by the patient. The dataset is categorized by assigning each record the most appropriate procedure from the 100 most commonly occurring procedures. We utilize the preprocessed data provided by Shokri et al. (2017); Song and Mittal (2021) [3].

Purchase100 Purchase100 dataset is derived from customers' shopping data released as part of the Kaggle Acquire Valued Shoppers Challenge [4]. This dataset, consisting of 197,324 data samples, represents individual customers' purchase histories. Each sample comprises 600 binary features, where each feature signifies the presence or absence of a specific product in the respective user's purchase history. The dataset has been categorized into 100 distinct classes based on various purchase styles. The objective of the classification

[1] https://pytorch.org/hub/pytorch_vision_resnet/
[2] https://www.dshs.texas.gov/THCIC/Hospitals/Download.shtm
[3] https://github.com/inspire-group/membership-inference-evaluation
[4] https://www.kaggle.com/c/acquire-valued-shoppers-challenge

task is to predict the purchase style associated with the 600 binary features for each user. This dataset's preprocessed version has been utilized from the work conducted by Shokri et al. (2017); Song and Mittal (2021) [4].

We summarize all datasets in details in Table 3.

Table 3: Detail of datasets. N_{total} denotes the total dataset size. N_{train} and N_{test} are the size of the training and testing set, respectively.

Dataset	Data type	Feature dimension	N_{total}	N_{train}/N_{test} (target model)	N_{train}/N_{test} (shadow model)
CIFAR-10	color image	$32 \times 32 \times 3$	60,000	12,000	12,000
CIFAR-100	color image	$32 \times 32 \times 3$	60,000	12,000	12,000
Texas100	medical record	6,169	67,330	13,466	13,466
Purchase100	purchase record	600	197,324	39,465	39,465

D.2. Model Architectures

We employ distinct neural network architectures tailored to different datasets. For CIFAR-10 and CIFAR-100, our models utilize a 20-layer ResNet and an 11-layer VGG architecture. We adopt architectures akin to Nasr et al. Nasr et al. (2018) when handling non-image datasets. Specifically, for Purchase100, we utilize a 4-layer fully-connected neural network structured with layer sizes of [1024, 512, 256, 100], while for Texas100, a 5-layer fully-connected neural network with layer sizes of [2048, 1024, 512, 256, 100] is employed.

D.3. Implementation Details

Our default optimization strategy involves utilizing Stochastic Gradient Descent (SGD) with specific parameters: momentum set to 0.9 and weight decay set to 1e-4. We initialize the learning rate τ at 0.1 and perform a learning rate decay by a factor of 10 at predetermined epochs. We list below the decay epochs in square brackets, and the total number of training epochs are marked in parentheses: CIFAR-10 and CIFAR-100 [100,150] (200); Texas100 and Purchase100 [50,100] (120). Additionally, we employ the following technique to enhance performance across heterogeneous data modalities: we perform the second parameter update exclusively for samples in the natural image datasets (CIFAR-10 and CIFAR-100) with sample losses falling within the interval $[\alpha, \ln(C)]$.

D.4. Defense Methods

Relaxloss RelaxLoss (Chen et al., 2022) introduces a novel training technique that effectively defends target models against privacy attacks while concurrently enhancing their utility. The core idea revolves around minimizing the gap between loss distributions, thereby mitigating membership privacy risks. Our experiments follow the parameter settings in the original paper and apply different α in the interval around the recommended α value to fully evaluate its privacy-utility trade-off [5].

[5] https://github.com/DingfanChen/RelaxLoss

Dropout Dropout (Srivastava et al., 2014) is a technique that helps prevent the interdependence of specific feature detectors in neural networks by randomly deactivating a subset of neurons during training. This method effectively reduces the risk of overfitting in the models. In our experiments, we specifically apply dropout to the final fully-connected layer of each target model and assess its impact using a predefined set of dropout rates: {0.1, 0.3, 0.5, 0.7, 0.9}.

Label-smoothing Label smoothing (Guo et al., 2017; Müller et al., 2019) helps in training by curbing the model's overconfidence in its predictions. It achieves this by penalizing predictions that deviate significantly from a more uniform distribution, measured using KL divergence. The formula expressing this concept is:

$$\mathcal{L} = \alpha \cdot D_{KL}(\mathcal{U} \mid\mid p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})) + (1-\alpha) \cdot \mathcal{L}_{CE}(\boldsymbol{s}, \boldsymbol{y}), \tag{10}$$

where D_{KL} represents KL-divergence, $\mathcal{U}$ is the uniform distribution, and $p_{\theta}(y \mid x)$ signifies the model's predicted output. In our experiments, we specifically assign values to α from the set {0.1, 0.3, 0.5, 0.7, 0.9}. This deliberate exploration across this spectrum enables us to thoroughly assess how varying α influences the model's behavior, facilitating a detailed observation of shifts in the privacy-utility curve.

Confidence-penalty The confidence-penalty (Pereyra et al., 2017) technique works by adding a regularization term to the model's loss function. This term evaluates the entropy of the output prediction, penalizing high-confidence predictions. The loss function, as expressed below, combines the standard cross-entropy loss with an additional term that penalizes confident predictions:

$$\mathcal{L} = \mathcal{L}_{CE}(\boldsymbol{s}, \boldsymbol{y}) + \alpha \cdot \sum_{c=1}^{C} p_{\boldsymbol{\theta}}(y^c|\boldsymbol{x}) \log(p_{\boldsymbol{\theta}}(y^c|\boldsymbol{x})), \tag{11}$$

where α acts as a hyperparameter regulating the impact of the entropy-based regularization. In line with Pereyra et al.'s Pereyra et al. (2017) approach, we explore a range of values for α, specifically [0.1, 0.3, 0.5, 1.0, 2.0, 4.0, 8.0], to assess its effect on the model's behavior.

Distillation Knowledge distillation represents the general approach of transferring knowledge from a group of teacher models to a student model. To emphasize the impact of the distillation process itself, we employed self-distillation Zhang et al. (2019) as our experimental methodology. This means we trained the student model to emulate a single teacher model with an identical architecture. The objective in training the student model is:

$$\begin{aligned} &\mathcal{L} = (1-\alpha) \cdot \mathcal{L}_{CE}(\boldsymbol{s}, \boldsymbol{y}) + \alpha T^2 \cdot D_{KL}(\widetilde{p}_{\boldsymbol{\theta}_s}(\boldsymbol{y}|\boldsymbol{x}) \mid\mid \widetilde{p}_{\boldsymbol{\theta}_t}(\boldsymbol{y}|\boldsymbol{x})) \\ &where \quad \widetilde{p}_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})^c = \frac{exp(\frac{f(\boldsymbol{\theta},\boldsymbol{x})^c}{T})}{\sum_{c'} exp(\frac{f(\boldsymbol{\theta},\boldsymbol{x})^{c'}}{T})} \end{aligned} \quad . \tag{12}$$

The term D_{KL}, representing KL-divergence, aims to minimize the gap between the softened student predictions $\widetilde{p}_{\boldsymbol{\theta}s}(\boldsymbol{y}|\boldsymbol{x})$ and the teacher predictions $\widetilde{p}_{\boldsymbol{\theta}t}(\boldsymbol{y}|\boldsymbol{x})$. Here, T signifies the temperature scaling factor controlling the degree of softening, while α serves as a hyperparameter balancing the KL-divergence and the normal cross-entropy $\mathcal{L}_{CE}$ term. We set $\alpha = 0.5$ and varied the temperature T across the set {1, 2, 5, 10, 20, 50, 100} to plot the privacy-utility curves.

DP-SGD DP-SGD (Abadi et al., 2016) is a method ensuring privacy in optimization, involves two steps: gradient clipping, which limits gradient norms to a maximum value C, and adding random noise before updating parameters. We focus on adjusting the clipping bound C while keeping the noise scale fixed at 0.1, following official recommendations. In exploring privacy-utility trade-offs, we manipulate the noise scale while maintaining a consistent clipping bound for the plotted privacy-utility curves.

Adv-Reg Adv-Reg (Nasr et al., 2018) integrates an adversarial objective during the target model's training process: the model aims to minimize a combined loss, which consists of the cross-entropy loss and an adversarial loss obtained from a surrogate attack. Each surrogate attack model undergoes training using the target model's training data and a distinct hold-out dataset. Adhering to the official implementation [6], we vary the weight α (defaulted to 1.0) of the adversarial loss over the set $\{0.8,\ 1.0,\ 1.2,\ 1.4,\ 1.6,\ 1.8\}$ to generate the privacy-utility curves.

Appendix E. ADDITIONAL RESULTS

E.1. Impact of Hyperparameters on Privacy-Utility Trade-off

We analyze the influence of critical hyperparameters of our approach. We categorize them into two groups for efficient exploration: 1) mean of the loss target (α) and temperature coefficient (T) and 2) lower limit of loss (β) and fraction (p). First, we investigate the impact of α and T on the Purchase100 dataset by fixing β and p to 1. We vary α from 0.1 to 1 and T from 1 to 10. As shown in Figure 5, both utility (accuracy) and privacy (NN AUC) improve as we move towards the lower right corner. This indicates that increasing α and T lead to better performance in terms of both utility and privacy. Therefore, in practice, setting α and T to sufficiently high values might be preferable. Next, we explore the impact of β and p on the Purchase100 dataset by fixing $T = 10$ and selecting a suitable α value of 2. In this experiment, β ranges from 0.1 to 1 and p ranges from 0 to 1. Observing the results in Figure 6, we notice that utility (accuracy) improves as we move towards the lower right corner of the figure, indicating an increase in β and p. However, privacy (NN AUC) shows a weaker resistance. This suggests that there is a trade-off between utility and privacy when selecting β and p. Based on the observations above, we narrow down the hyperparameter search space for our experiments by fixing the global temperature coefficient T to 10. We then select appropriate α values specific to different datasets. Finally, we conduct parameter search experiments to find more effective settings for β and p. These experiments involve varying β within the range of [0.1, 1] and p within the range of [0, 1].

Supplementary to Table 1 in the main paper, Table 4 shows the parameter values selected for different datasets and model architectures.

[6] https://github.com/SPIN-UMass/ML-Privacy-Regulization

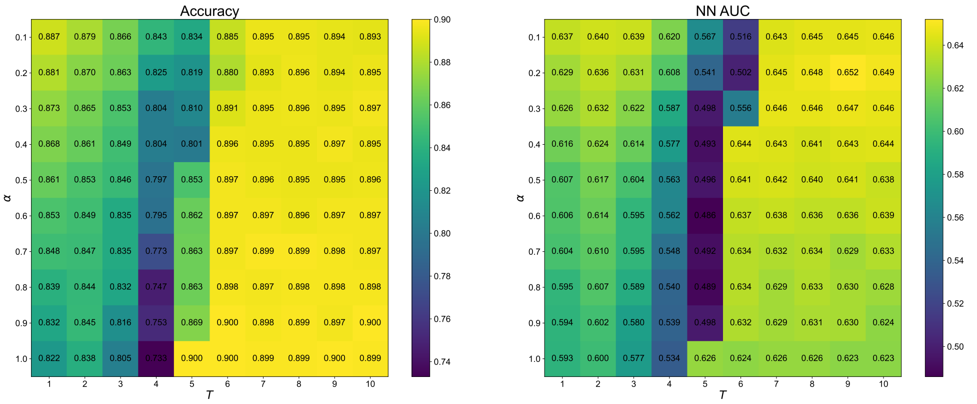


Figure 5: Accuracy and NN AUC on Purchase100, with α ranging from 0.1 to 1 and T ranging from 1 to 10, The parameter β and fraction p is set to be 1.

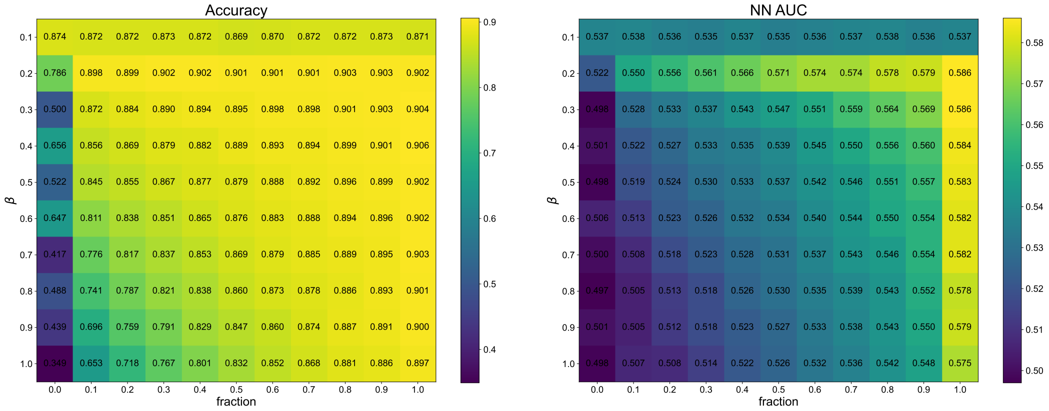


Figure 6: Accuracy and NN AUC on Purchase100, with β ranging from 0.1 to 1 and fraction p ranging from 0 to 1, The parameter α is set to be 2 for Purchase100, while the parameter T is set to be 10.

Table 4: The parameter values selected for different datasets and model architectures. This is supplementary to Table 1 in the main paper.

	CIFAR10 (ResNet20)	CIFAR100 (ResNet20)	CIFAR10 (VGG11)	CIFAR100 (VGG11)	Texas100	Purchase100
selected T	10					
selected α	1	3	1	2	2.8	2
selected β	0.9	1	0.8	0.9	1	0.3
selected p	0.2	0.6	0.2	0.3	0.9	0.4

Table 5: The attack accuracy (in %) evaluated on the target models with and without (w/o) applying our defense. Δ corresponds to the relative difference after applying our defend method (in %). The selected target models are the same as in Table 1.

		NN	Loss	Boundary	Entropy	M-Entropy	Grad-x ℓ_1	Grad-x ℓ_2	Grad-w ℓ_1	Grad-w ℓ_2
CIFAR10 (ResNet20)	w/o	80.8	83.7	70.0	82.9	83.9	84.2	84.2	84.6	84.7
	with	54.2	50.0	52.3	50.0	50.0	50.0	50.0	50.0	50.0
	Δ	-33.0	-40.3	-25.3	-40.0	-40.4	-40.6	-40.6	-40.9	-41.0
CIFAR100 (ResNet20)	w/o	96.1	96.8	84.9	96.1	96.8	97.2	97.1	97.6	97.6
	with	50.0	50.0	50.8	50.0	50.0	50.0	50.0	50.0	50.0
	Δ	-48.0	-48.3	-40.2	-48.0	-48.3	-48.6	-48.5	-48.8	-48.8
CIFAR10 (VGG11)	w/o	72.7	77.5	69.6	76.1	77.6	78.2	78.1	78.1	78.2
	with	54.6	50.0	59.2	50.0	50.0	50.0	50.0	50.0	50.0
	Δ	-24.9	-35.5	-14.9	-34.3	-35.6	-36.1	-36.0	-36.0	-36.1
CIFAR100 (VGG11)	w/o	96.7	96.8	86.6	96.7	97.6	97.6	97.6	97.8	97.9
	with	50.2	50.0	62.7	50.0	50.0	50.0	50.0	50.0	50.0
	Δ	-48.1	-48.3	-27.6	-48.3	-48.8	-48.8	-48.8	-48.9	-48.9
Texas100	w/o	65.9	79.3	———	70.7	79.4	80.0	80.1	80.2	80.0
	with	50.0	50.0	———	50.0	50.0	50.2	50.9	50.3	50.0
	Δ	-24.1	-37.0	———	-29.3	-37.0	-37.3	-36.5	-37.3	-37.5
Purchase100	w/o	60.6	64.7	———	63.8	64.8	65.8	65.8	65.8	65.7
	with	52.0	50.0	———	50.0	50.0	50.0	50.0	50.0	50.0
	Δ	-14.2	-22.7	———	-21.6	-22.8	-24.0	-24.0	-24.0	-23.9

Table 6: The accuracy (in %) of adaptive (a.) attacks evaluated on the target models with (w/) and without (w/o) applying our defense. Δ corresponds to the relative difference after applying our defend method (in %). The selected target models are the same as in Table 1.

		NN	Loss	Boundary	Entropy	M-Entropy	Grad-x ℓ_1	Grad-x ℓ_2	Grad-w ℓ_1	Grad-w ℓ_2
CIFAR10 (ResNet20)	w/o	80.8	83.7	70.0	82.9	83.9	84.2	84.2	84.6	84.7
	w/ (a.)	54.6	55.9	52.8	53.2	56.0	52.0	51.7	52.0	52.3
	Δ	-32.4	-33.2	-24.6	-35.8	-33.3	-38.2	-38.6	-38.5	-38.3
CIFAR100 (ResNet20)	w/o	96.1	96.8	84.9	96.1	96.8	97.2	97.1	97.6	97.6
	w/ (a.)	53.1	55.6	47.8	51.7	55.9	50.1	50.1	50.2	50.2
	Δ	-44.7	-42.6	-43.7	-46.2	-42.3	-48.5	-48.4	-48.6	-48.6
CIFAR10 (VGG11)	w/o	72.7	77.5	69.6	76.1	77.6	78.2	78.1	78.1	78.2
	w/ (a.)	58.0	59.0	56.8	56.1	59.0	54.3	54.4	54.6	54.9
	Δ	-20.2	-23.9	-18.4	-26.3	-24.0	-30.6	-30.3	-30.3	-29.8
CIFAR100 (VGG11)	w/o	96.7	96.8	86.6	96.7	97.6	97.6	97.6	97.8	97.9
	w/ (a.)	65.6	67.9	62.8	62.1	68.0	52.6	52.6	54.3	54.5
	Δ	-32.2	-29.9	-27.5	-35.8	-30.3	-46.1	-46.1	-44.5	-44.3
Texas100	w/o	65.9	79.3	———	70.7	79.4	80.0	80.1	80.2	80.0
	w/ (a.)	53.5	57.5	———	54.7	57.5	53.4	53.5	53.1	54.0
	Δ	-18.8	-27.5	———	-22.6	-27.6	-33.3	-33.2	-33.8	-32.5
Purchase100	w/o	60.6	64.7	———	63.8	64.8	65.8	65.8	65.8	65.7
	w/ (a.)	54.5	54.6	———	53.1	54.6	52.6	52.6	53.4	54.1
	Δ	-10.1	-15.6	———	-16.8	-15.7	-20.1	-20.1	-18.8	-17.7

E.2. Comparison to Undefended Method

Supplementary to Table 1 in the main paper, Table 5 shows the attack accuracy values on target models trained with and without applying our defense method. We observe that our approach effectively reduces the attack accuracy to a random-guessing level (around 50%) for most cases.

E.3. Adaptive Attack

Supplementary to Section 5.4 in the main paper, we show the accuracy of adaptive attacks on target models trained with and without applying our defense method in Table 6. In the case of threshold-based attacks, we determine the attack's decision threshold by selecting the threshold that achieves the highest attack accuracy on the shadow model (trained with the same configuration as our defensed target model in Table 1). For NN-based attacks, we employ the full logits predictions from a pre-trained shadow model as features for training the adaptive attack model. It is noteworthy that our approach consistently diminishes attack accuracy across all scenarios, albeit the reduction is not as conspicuous when compared to the non-adaptive attacks presented in Table 5.

References

Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proc. ACM Conf. Comput. Commun. Secur., CCS*, pages 308–318, 2016.

Dingfan Chen, Ning Yu, and Mario Fritz. Relaxloss: Defending membership inference attacks without losing utility. In *Int. Conf. Learn. Represent., ICLR*, pages 1–12, 2022.

Christopher A. Choquette-Choo, Florian Tramèr, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *Int. Conf. Mach. Learn., ICML*, pages 1964–1974, 2021.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Int. Conf. Mach. Learn., ICML*, pages 1321–1330, 2017.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. pages 1–58, 2009.

Zheng Li and Yang Zhang. Membership leakage in label-only exposures. In *Proc. ACM Conf. Comput. Commun. Secur., CCS*, pages 880–895, 2021.

Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. When does label smoothing help? In *Adv. neural inf. proces. syst., NeurIPS*, pages 4696–4705, 2019.

Milad Nasr, Reza Shokri, and Amir Houmansadr. Machine learning with membership privacy using adversarial regularization. In *Proc. ACM Conf. Comput. Commun. Secur., CCS*, pages 634–646, 2018.

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. Regularizing neural networks by penalizing confident output distributions. In *Int. Conf. Learn. Represent., ICLR*, pages 1–11, 2017.

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In *Int. Conf. Mach. Learn., ICML*, pages 5558–5567, 2019.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *Proc. IEEE Symp. Secur. Privacy, SP*, pages 3–18, 2017.

Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In Michael D. Bailey and Rachel Greenstadt, editors, *Proc. USENIX Secur. Symp., USENIX*, pages 2615–2632, 2021.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, 2014.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *Proc.IEEE Comput. Secur. Found. Symp., CSF*, pages 268–282, 2018.

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 *Proc. IEEE Int. Conf. Comput. Vis., ICCV*, pages 3712–3721, 2019.