A Novel Self-Distillation Architecture to Defeat Membership Inference Attacks

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, which aim to distinguish training members from non-members by exploiting differential behavior of the models on member and non-member inputs. We propose a new framework to train privacy-preserving models that induces similar behavior on member and non-member inputs to mitigate practical membership inference attacks. Our framework, called SELENA, has two major components. The first component and the core of our defense, called Split-AI, is a novel ensemble architecture for training. We prove that our Split-AI architecture defends against a large family of membership inference attacks, however, it is susceptible to new adaptive attacks. Therefore, we use a second component in our framework called Self-Distillation to protect against such stronger attacks, which (self-)distills the training dataset through our Split-AI ensemble and has no reliance on external public datasets. We perform extensive experiments on major benchmark datasets and the results show that our approach achieves a better trade-off between membership privacy and utility compared to previous defenses.

1 Introduction

Recent work has shown that ML models are prone to memorizing sensitive information of training data incurring serious privacy risks [23, 2, 3, 6, 20, 24, 7]. In this work, we focus on *membership inference attack (MIA)*, which tries to identify whether a target sample was used to train the target ML model or not based on model behavior[23]. MIAs pose a severe privacy threat by revealing private information about 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.

There are two main categories of membership inference defenses: *provable privacy* guaranteed by DP and *empirical membership privacy defenses*. The first category uses differential privacy mechanisms [1, 15, 28], which provide a *provable privacy guarantee* for all inputs. While provable privacy with high utility is more desirable, it has been a challenge to use provable privacy techniques guaranteed by DP like DP-SGD [1] to achieve high accuracy in many machine learning tasks. This motivates the second category of membership inference defenses, where privacy is empirically evaluated through practical MIAs to preserve high utility. However, none of the existing defenses in this category [16, 10, 22] are able to provide sufficient MIA protection and high utility simultaneously in the absence of public datasets [25, 5].

In this paper, we introduce a new defense, called SELENA,[1] to protect against black-box MIAs while also offering a high utility, which falls in the category of *empirical membership privacy defenses*.

[1] SELf ENsemble Architecture.

Submitted to NeurIPS Workshop on Privacy in Machine Learning (PriML) 2021. Do not distribute.

Our framework consists of two core components: *Split-AI*[2] and *Self-Distillation*. Our first component Split-AI trains multiple models (called sub-models) with random subsets from the training set, and applies adaptive inference to enable the model to have similar behavior on members and non-members: for any queried sample, no matter whether it is in training set, our defense only use sub-models which are not trained with it to get outputs; this ensures membership privacy for a large family of MIAs which we demonstrate through a formal analysis. Our second component Self-Distillation (self)-distills the exact same training sets by using the outputs from Split-AI as soft labels to train a new model, which solves the potential advanced attacks threat against Split-AI and computation overhead in inference.

2 Our defense

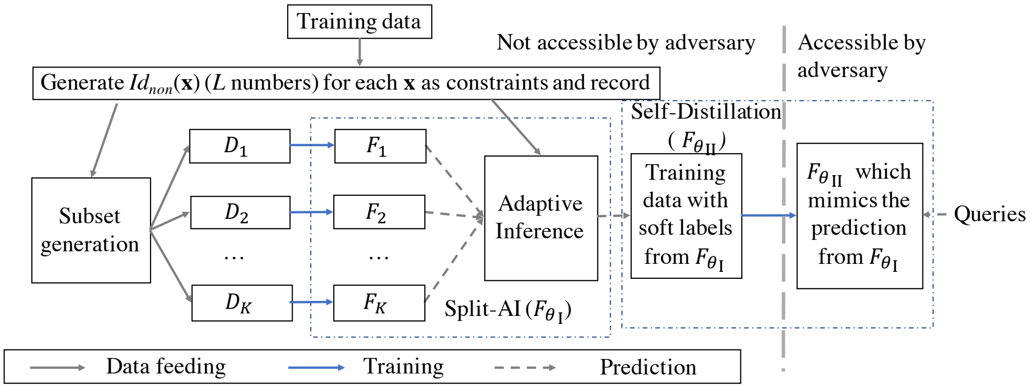

Training data
Not accessible by adversary
Accessible by adversary
Generate $Id_{non}(\mathbf{x})$ (L numbers) for each $\mathbf{x}$ as constraints and record
Self-Distillation ($F_{\theta_{\mathrm{II}}}$)
Subset generation
D_1
D_2
…
D_K
F_1
F_2
…
F_K
Adaptive Inference
Split-AI ($F_{\theta_{\mathrm{I}}}$)
Training data with soft labels from $F_{\theta_{\mathrm{I}}}$
$F_{\theta_{\mathrm{II}}}$ which mimics the prediction from $F_{\theta_{\mathrm{I}}}$
Queries
Data feeding
Training
Prediction

Figure 1: Our end-to-end defense framework with the Split-AI and Self-Distillation components.

Figure 1 gives an overview of our defense, where we denote Split-AI as $F_{\theta_{\mathrm{I}}}$ and protected model from Self-Distillation as $F_{\theta_{\mathrm{II}}}$.[3] We next detail Split-AI and Self-Distillation separately.

First component Split-AI. 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, such as accuracy [21], confidence [29, 26, 25] and robustness [5, 13], on member and non-member data. MIAs leverage these differences to obtain an attack advantage that is better than a random guess even in the black-box setting. Our Split-AI design is based on the following intuition: *if a training sample is not used to train a sub-model, that sub-model will have similar behavior on that training sample and non-members.*

Split-AI's training. Specifically, for each data point $\mathbf{x}$ 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 $\mathbf{x}$ and record the identification numbers of these L non-model indices (denoted as $Id_{non}(\mathbf{x})$[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. We then train K sub-models F_i, one for each subset 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, i.e., $\frac{1}{L}\sum_{i\in Id_{non}(\mathbf{x})} F_i(\mathbf{x})$;
- If not, the defender will randomly use non-member 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, i.e., $\frac{1}{L}\sum_{i\in Id_{non}(\mathbf{x}')} F_i(\mathbf{x})$.

[2] Split Adaptive Inference Ensemble.

[3] PATE[17, 18] also trains multiple sub-models to provide privacy but with a public dataset, difference detailed in Appendix A.

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

We formally prove that Split-AI strategy can reduce the accuracy of *direct* single-query MIAs (typical been used in most previous MI defenses [16, 10, 25], see Appendix B.1 for more details) to a random guess (See Theorem 2 in Appendix C). 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.

Limitations of Split-AI. While our Split-AI strategy is resilient to direct single-query MIAs, an adversary can leverage more advanced attacks including indirect attacks [14] an replay attacks. For indirect attacks, attacker can make a single *indirect* query by adding a small noise to the target sample. 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. For replay attacks: Split-AI has one possible output for member sample, while there are multiple possible outputs for non-members. Furthermore, Split-AI imposes a computational overhead in the inference phase as Split-AI needs to perform inference on L models for each queried sample.

Second component: Self-Distillation. To overcome these limitations, we leverage distillation [9]. To be more specific, here we term our second component as *Self-Distillation* because we use features in the exact same training set as Split-AI along with the prediction vectors from Split-AI as soft labels to train a new model using conventional training. The new protected model benefits from distillation to largely preserve Split-AI's defense ability against direct single-query attack (See Theorem 5 and Corollary 6 in Appendix C) while maintaining a good classification accuracy. For queried samples, the defender now just need to do the inference on the new protected model $F_{\theta_{\text{II}}}$ distilled from the Split-AI.

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 and multi-query indirect attacks. Self-Distillation also solves the computational overhead limitation of the Split-AI at inference time: the defender now only needs to make inference on a single Self-Distilled model.

3 Evaluations

Experimental setup. We follow the setting in previous work [16] that the attacker knows half members and non-members, i.e., the number of members and non-members used to train and evaluate the attack model are the same and the random guess baseline attack accuracy is 50%. We use three benchmark datasets and target models which are widely used in prior works on MI attacks and defenses [23, 16, 10]: Purchase100 [19], Texas100 [27] and CIFAR100 [12]. We use $K = 25$, $L = 10$ for all three datasets. Additional experimental details are in Appendix D. We systematically evaluate our end-to-end defense framework by direct single-query attacks, indirect label-only attacks (see Appendix B.1 for more details), and adaptive attacks (explained in next paragraph) and make a comparison with previous MI defenses: MemGuard [10] and adversarial regularization [16].

Adaptive attacks. The systematic evaluation of existing defenses by Song et al. [25] emphasizes that the defender should consider adaptive attackers with knowledge of the defense to rigorously evaluate the performance of the defenses. Here we consider the attacker to construct a shadow Split-AI using the known training samples to provide additional information (More details in Appendix B.2).

Results. Table 1 summarizes the classification accuracy and best attack accuracy for each attack type, including comparison with previous defenses [16, 10]. We also includes undefended models as a baseline. We use $\text{acc}_{\text{train}}$ and acc_{test} to denote the model classification accuracy on training set and test set. We use acc_{dsq}, acc_{lo}, acc_{ada}, and acc_{best} to denote accuracy for direct single-query attacks, label-only attacks, adaptive attacks and best attack accuracy among all attacks respectively.

Comparison with MemGuard. While the test accuracy of our defense is a little lower (at most 3.9%) 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

Table 1: 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 the best attack accuracy, lower the 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 and best attack.

dataset	defense	$\text{acc}_{\text{train}}$	acc_{test}	acc_{dsq}	acc_{lo}	acc_{ada}	acc_{best}
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%**

random guess, which is a factor of $2.4 \sim 3.7$ higher than our defense. In general, MemGuard does not have any defense against MIAs that do not rely on confidence information: attacker can use label-only attacks as adaptive attacks since MemGuard only obfuscates confidence.

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 defenses 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.

We also include a comparison of our defense with early stopping [25] and DP-SGD [1] in Appendix E. In addition, we also highlight the following two points from Table 1:

Our SELENA effectively induces similar behaviors including generalization, confidence, robustness for member and non-member samples and therefore the MIA attack accuracy is largely reduced. Let us take the generalization gap g as an example: the generalization gap in undefended models/MemGuard is 16.78% on Purchase100, 27.0% on Texas100, 22.98% on CIFAR100; the generalization gap in adversarial regularization is 13.4% on Purchase100, 10.2% on Texas100 and 15.4% on CIFAR100. In contrast, the generalization gap in our defense 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.

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 attack and label-only attacks. However, even considering the strong adaptive attacks, SELENA still achieves lower attack accuracy in comparison to previous defenses.

Computation overhead in SELENA. 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 in the training phase. However, our SELENA does not incur additional computation overhead in inference compared to undefended model. Here we argue that the cost of computing resources in the training phase is acceptable as the improvement in GPU technology are making the computing resources cheap while the privacy threat remains severe. We note that if multiple GPUs are available, our approach can easily benefit from parallelization by training the K sub-models in parallel.

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] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[12] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.

[19] Purchase. Acquire valued shoppers challenge. https://www.kaggle.com/c/acquire-valued-shoppers-challenge/data. [Online; accessed 22-March-2020].

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

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

[26] 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.

[27] Texas. Texas hospital stays dataset. https://www.dshs.texas.gov/THCIC/Hospitals/Download.shtm. [Online; accessed 22-March-2020].

[28] 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.

[29] 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.

[30] 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.

Appendix

A Comparison With PATE

PATE [17, 18] 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 noisy 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 test accuracy [18], while the test accuracy drop in our defense is no more than 3.9%.

B Membership inference attacks

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 [16] or by knowing a dataset from the same distribution of the training set and constructing shadow models [23].

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_{tr}^{A} and D_{te}^{A} 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 which codes members as 1, and non-members as 0. The literature typically measures MIA efficacy as the attack accuracy:

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

In most previous attacks [23, 16, 30, 25], 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 existing black-box MIAs in the following two categories: **direct** attacks and **indirect** attacks, as well as explain adaptive attacks against our SELENA defense.

B.1 Existing membership inference 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. This category of MIAs includes NN-based attack [23, 16], correctness-based attack [30], confidence-based attack [29, 26, 25], entropy-based attack [23, 25], modified entropy-based attack [25]. We consider all these attacks across three datasets and report the best direct single-query attack accuracy in Table 1.

Indirect multi-query attacks (label-only attacks): Long et al. [14] stated that indirect attacks can make queries that are related to target sample **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 [14, 13, 5]. 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 [10]. 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 [13, 5]. Simply obfuscating a model's confidence scores can not hide label information to defend against such label-only attacks. This category of MIAs includes boundary estimation attacks [13, 5] and data augmentation attacks [5]. We consider boundary estimation attacks for all three datasets and data augmentation attacks on CIFAR100 as only CIFAR100 uses data augmentation during the training process.

B.2 Adaptive attacks

The systematic evaluation of existing defenses by Song et al. [25] emphasizes 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 ensemble 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 ensemble, 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-member model indices $Id_{non}(\mathbf{x})$ for training set, the attacker will generate new non-member 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 [23] since it is trained with exact knowledge of the partial training dataset.

We design four adaptive attacks including two NN-based attacks and two metric-based attacks to leverage the estimated soft labels to attack our defense. To clarify, $F_{\theta_{\text{II}}}$ denotes the protected target model which answers the attacker's queries and $F'_{\theta_{\text{I}}}$ denotes the shadow Split-AI ensemble constructed by attacker.

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 NN architecture used by Nasr et al. [16] (denoted as I_{NN2}).

MIAs based on distance between soft labels and predicted confidence: Similar to previous metric-based attacks [25], 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).

C Proof for Split-AI 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 $TV(X, X') = \sup_{S \subset \text{Supp}(X) \cup \text{Supp}(X')} \Pr[X \in S] - \Pr[X' \in S]$. We present our Split-AI algorithm in Algorithm 1.

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_I} using S as training set.*
4. *Learner selects a random $i \in [2n]$ and sends $(x_i, F_{\theta_I}(x_i))$ 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.

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*

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

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 distribution 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.

Algorithm 1 Split-AI Model $F_{\theta_{\mathrm{I}}}$

```
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 $L$ identification numbers of sub-models 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 for models: $\{(\mathbf{x}, y)$: $(\mathbf{x}, y) \in (X_{train}, Y_{train})$, $i$ 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: $F_{\theta_{\mathrm{I}}}(\mathbf{x})$
Given $\mathbf{x}$
if $\mathbf{x}$ in $X_{train}$ then
    $$F_{\theta_{\mathrm{I}}}(\mathbf{x}) = \frac{1}{L} \sum_{i \in Id_{non}(\mathbf{x})} F_i(\mathbf{x})$$
else
    Randomly select $\mathbf{x}'$ in the training set,
    $$F_{\theta_{\mathrm{I}}}(\mathbf{x}) = \frac{1}{L} \sum_{i \in Id_{non}(\mathbf{x}')} F_i(\mathbf{x})$$
end if
```

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 $T \subset \{x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_{2n}\}$ 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 select $i \in [2n]$. This is a stronger security game as the adversary can select the worst example in the dataset. However, Theorem 2 remain 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.*

Definition 2 (stable distillation)**.** *A distillation algorithm $Q\colon M_s \times \mathrm{AUX} \to M_o$ 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 $M_s \times \mathrm{AUX}$ 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 consist 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

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

Therefore, we have

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

□

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

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

Remark 7 (How private is SELENA against multi-query attacks?)**.** *The above theoretical analysis of SELENA is only valid for single-query direct 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 $(x', 1)$ in the dataset 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 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 satisfy certain assumptions to future work.*

D Experimental setup

Here we introduce the datasets, the model architectures, and the hyper-parameter settings in more detail.

D.1 Dataset

We use three benchmark datasets widely used in prior works on MIAs:

CIFAR100 [12]: This is a benchmark dataset used to evaluate image classification algorithms. CIFAR100 is composed of 32 × 32 color images in 100 classes, with 600 images per class. For each class label, 500 images are used as training samples, and remaining 100 images are used as test samples.

Purchase100 [19]: This dataset is based on Kaggle's Acquire Valued Shopper Challenge, which contains shopping records of several thousand individuals. We obtained a prepossessed and simplified version provided by Shokri et al. [23]. This dataset is composed of 197,324 data samples with 600 binary features. Each feature corresponds to a product and represents whether the individual has purchased it or not. This dataset is clustered into 100 classes corresponding to purchase styles.

Texas100 [27]: This dataset is based on the Hospital Discharge Data public use files with information about inpatients stays in several health facilities released by the Texas Department of State Health Services from 2006 to 2009. Each data record contains external causes of injury, the diagnosis, the procedures the patient underwent and some generic information. We obtain a prepossessed and simplified version of this dataset provided by Shokri et al. [23], which is composed of 67,330 data samples with 6,170 binary features. This dataset is used to classify 100 most frequent used procedures.

D.2 Target Models

For CIFAR100, we use ResNet-18 [8], which is a benchmark machine learning model widely used in computer vision tasks. We adopt the cross-entropy loss function and use Stochastic Gradient Descent (SGD) to learn the model parameters. We train the model for 200 epochs with batch size of 256, initializing learning rate 0.1 with weight decay 0.0005 and Nesterov momentum of 0.9 and divide the learning rate by 5 at epoch 60, 120, 160.[5]

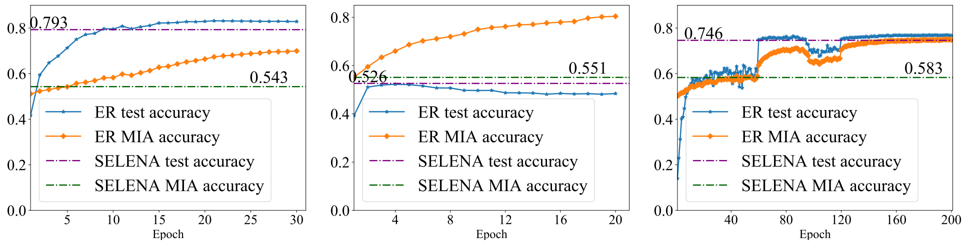
For Purchase100 and Texas100, we follow previous work [16] to use a 4-layer fully connected neural network with layer sizes $[1024, 512, 256, 100]$ and Tanh as the activation function. We use the cross-entropy loss function and Adam [11] optimizer to train the model on Purchase100 for 30 epochs and on Texas100 for 20 epochs with learning rate of 0.001. The batch size is 512 for Purchase100 and 128 for Texas100.

E Comparison with other defenses

E.1 Comparison with early stoppoing

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

[5] https://github.com/weiaicunzai/pytorch-cifar100



(a) Purchase100

(b) Texas100

(c) CIFAR100

Figure 2: Detailed comparison of SELENA with early stopping. From left to right are results for Purchase100, Texas100 and CIFAR100. The solid curves 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 1. Our defense achieves a better privacy-utility trade-off than all epochs in the conventional training.

vulnerable to membership inference attacks. 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. [25] find that adversarial regularization is not better than early stopping [4] when evaluated by a suite of attacks including both NN-based attacks and metric-based attacks. Therefore, we further compare our defense with early stopping.

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 attack (due to their strong performance on undefended models). Figure 2 shows a detailed comparison between our defense $F_{\theta_{\text{II}}}$ and early stopping. The dashed lines are the classification accuracy on test set and the best MIA accuracy of our defense, which is already reported in Table 1. The solid 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 2, *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 39.2%, which is 13.4% lower than that of our defense (52.6%).* 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 our defense, 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 our defense separately.

E.2 Comparison with DP-SGD

We use the canonical implementation of DP-SGD and its associated analysis from the TensorFlow Privacy library[6]. We varied the parameter $noise\_multiplier$ in the range of [1, 3] on Purchase100 and [1, 2] on Texas100 with a step size 0.2. We set the privacy budget $\epsilon = 4$ and report the best classification accuracy for these two datasets.

The test accuracy on Purchase100 is 56.0% and the corresponding best direct single-query MIA accuracy is 52.8%. The test accuracy on Texas100 is 39.1%, and the corresponding 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 test accuracy.

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