Accuracy-Privacy Tradeoffs in SOTA Weak-Supervision Models

Prashanthi Ramachandran, Shashidhar Pai

December 31, 2022

Abstract

Weak supervision is a powerful tool to train robust machine learning models in use cases where labeled data is expensive or scarce. Labels can also be of highly private/sensitive nature depending on the dataset. While fitting models on labeled data, there arises the risk of models memorizing data points, which impacts not just the generalizability of the model, but also compromises privacy. Further, recent techniques to learn with privacy have shown to improve the generalizability of models. However, the addition of noise to enhance the privacy of models causes the models to be poor and inefficient. In this paper, we add a layer of differential privacy to two existing weak-supervision frameworks, Ratner et al. [2017] and Karamanolakis et al. [2021] and analyze the privacy-accuracy trade-offs with the added layer of privacy. We also probe into the privacy benefits that noisy weak labels can introduce that can be adapted for scalable private learning.

1 Introduction

Most machine learning tasks require access to a large pool of labeled data to train robust and accurate models. A reliable pool of labeled data is not only laborious and expensive to obtain, but this data is also often of highly sensitive nature. While fitting on labeled data, models have a tendency to memorize this private data, thereby negatively impacting the generalization capability. For instance, in few-shot or one-shot frameworks, this becomes even more pronounced as the model could potentially overfit to sensitive training data. Further, in models such as Brown et al. [2020] and Marcus et al. [2022] with 12-17 billion parameters, memorization becomes a huge threat to privacy (illustrated in Figure 1). Therefore, there arises a need to train machine learning models privately. Learning with privacy has also been shown to improve the generalizability of models. Training models to be differentially private is essentially a form of regularization, which ensures that models do not overfit on training data. However, the addition of noise for achieving better privacy may cause the model to perform worse and increase the training time (Berthelot et al. [2019]).

In this project, we have applied a layer of privacy to two state-of-the-art weak supervision frameworks, Ratner et al. [2017] (Snorkel) and Karamanolakis et al. [2021] (ASTRA). We first analyze the existing privacy guarantees in these frameworks. Then, we adapted concepts from Papernot et al. [2018a], Long et al. [2019], and Abadi et al. [2016] and incorporated them into these frameworks. We analyzed the effects of weak-supervision learning with privacy, specifically, accuracy-privacy trade off, training time, and other hyper-parameters. From the privacy perspective, frameworks such as Ratner et al. [2017] and Karamanolakis et al. [2021] benefit from their minimal usage of labeled data while training. Hence, finally, we also give an account of the privacy benefits a model can achieve with the help of noisy labels that are used in weak supervision.

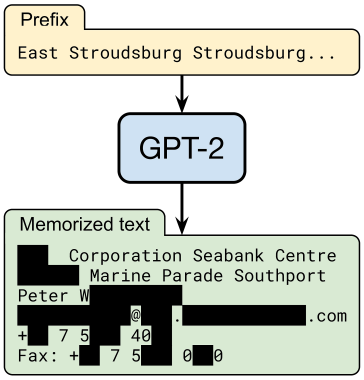

Figure 1: The ability of models such as GPT-2 to accurately predict personal information of specific people when prompted with a short snippet of internet text

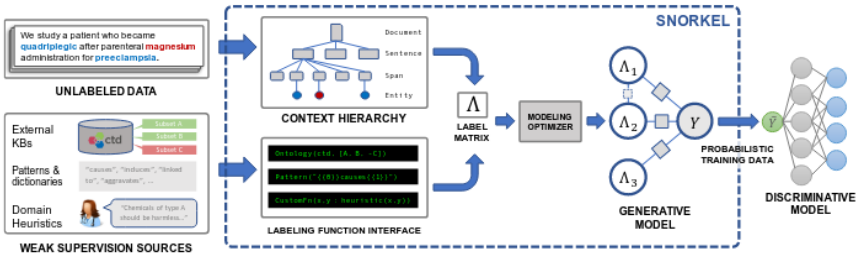

Figure 2: Snorkel Framework

2 Background

2.1 Snorkel

Weak-supervision can be very useful in situations where obtaining labels for data can be difficult or expensive. One way to perform weak-supervision is to use Snorkel (Ratner et al. [2017]), a framework for training machine learning models using weak supervision. As opposed to hand-labeling training data, Snorkel allows you to define a set of deterministic labeling functions, which are simple heuristics or rules that can be used to label data. Snorkel uses these functions to label a large amount of data to then train a machine learning model.

The Snorkel framework (refer to Figure 2) comprises a labeling function interface, a modelling optimizer, a generative model, and a discriminative model. After the framework is trained, the latter is deployed for public usage. Snorkel allows subject matter experts to define labeling functions through the labeling function interface using unlabeled data and other weak supervision sources. It then learns a modeling optimizer that learns to model the correlation between various labeling functions to address the "double-counting" problem. It then learns a generative model over the labeling functions which allows it to estimate their accuracies. The output is a set of probabilistic labels that can be used to train a variety of machine learning models.

2.2 ASTRA

Self-training is a method of training a machine learning model using a combination of labeled and unlabeled data. It is often used in the context of weak supervision, where the goal is to train a

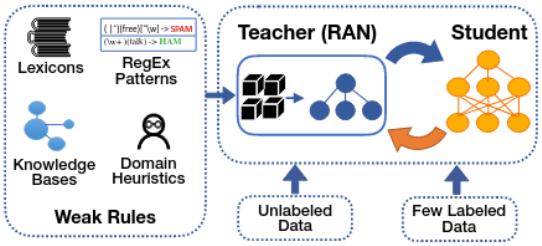

Figure 3: ASTRA Framework

model using large amounts of noisy or incompletely labeled data. In self-training, the model is first trained on a small amount of labeled data. The model is then used to label a larger amount of unlabeled data, and the labeled data is used to train a new version of the model. This process is repeated until the model reaches a satisfactory level of accuracy.

Self-training can be implemented using the ASTRA (weAkly-supervised Self-TRAining) framework, which is a toolkit for self-training machine learning models using weak supervision. ASTRA allows you to define a set of labeling functions, which are simple heuristics or rules that can be used to label data, much like Snorkel. However, it also also addresses a key problem with Snorkel where heuristic have very low coverage (most labeling functions abstain) and a lot of the unlabeled data can therefore be wasted. The ASTRA Framework (Figure 3) consists of a teacher model, also known as a RAN (rule attention network) and a student model. The student model is first trained with the small set of labeled data. The teacher then aggregates weak rules and student-assigned pseudo-labels with trainable weights that capture their fidelity towards an instance x. The teacher and student iteratively train this way until the student reaches convergence on the small set of labeled data.

2.3 Learning with differential privacy

A learning algorithm is said to be differentially private if its output does not tell an observer whether a certain data point was used in the training computation or not. In other words, adding, removing or modifying one data point should ensure that the final result is statistically indistinguishable from the original result. Mathematically:

$$\log \frac{M(D) \in S}{M(D') \in S} \leq \epsilon \tag{1}$$

where,

- $M(D)$ is a model trained on dataset D
- $M(D')$ is a model trained on dataset D'
- S is a set of outcomes of the model
- and ϵ is a small value, known as the *privacy budget*

To ensure differential privacy, we need to add noise somewhere in the training process. In DP-SGD (Abadi et al. [2016]), the noise is adding to the training data so that the model parameters learn to generalize well. The noise could alternatively be added to the outputs of the model, but the

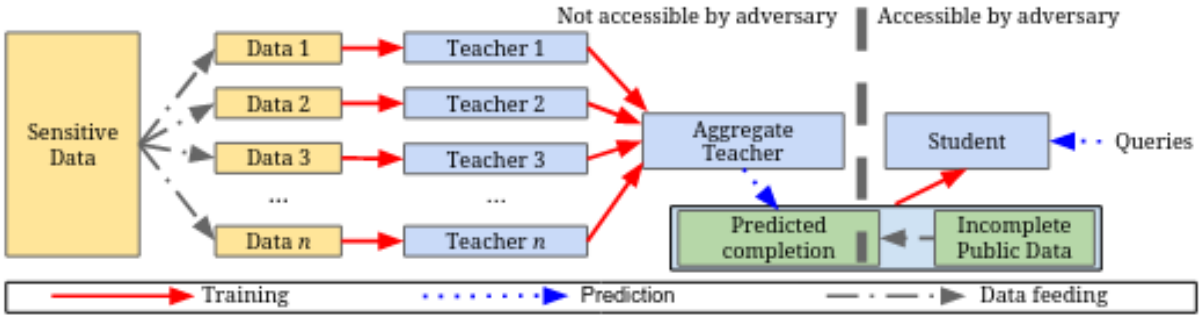

Figure 4: The PATE Framework

amount of noise required to ensure privacy may overpower the output of the model. A third way to guarantee privacy would be to train an *ensemble* of models, similar to federated learning, where each model is trained on a disjoint subset of the training data. If all the models agree on a certain label output, it must mean that they have learned to generalize well. This is because all the models in the ensemble cannot memorize a single data point, since they are trained on disjoint subsets of the training data. As a result, we do not need to add external noise to this architecture. The different ways to add noise to the training process have been illustrated in Figure 5.

PATE's (Papernot et al. [2018a]) architecture uses an ensemble of teachers with access to labeled data and a deployable student model with access to the teachers and unlabeled data. Each teacher in the ensemble is trained with a disjoint subset of the sensitive labeled dataset. The student is trained on the outputs of the ensemble of teachers on unseen data. For each unseen data point, the student queries the teacher ensemble and the outputs of the teachers are aggregated using a differentially-private algorithm known as `NoisyMax` that adds noise to votes of each teacher corresponding to the number of possible labels. This aggregation is analogous to the majority voting mechanism adopted in Snorkel for aggregating the outputs of the labeling functions, except PATE's algorithm is differentially private, owing to the added noise.

3 Methods

To set up basic Snorkel and ASTRA, we initially hoped to work with the source code provided by these two frameworks on easily available datasets such as MNIST[1], SVHN[2], UCI Adult[3], UCI Diabetes[4] datasets to test our hypothesis. However, we chose to set up WRENCH or Weak supeRvision bENCHmark (Zhang et al. [2021]), a benchmarking toolkit for evaluating the performance of ML models trained using weak supervision following Prof. Bach's recommendation. WRENCH provides a standardized way to compare the performance of different weak supervision approaches on a variety of datasets. It includes a variety of datasets (BasketBall, Commercial, ChemProt, Youtube, Census, etc.) tools for generating weak supervision labels (weak labels from pre-defined labeling functions), and evaluating model performance. Using WRENCH, we were able to run Snorkel and ASTRA on many of these datasets.

We incorporated the private learning techniques in PATE into Snorkel by replacing the major-

[1] `http://yann.lecun.com/exdb/mnist/`
[2] `http://ufldl.stanford.edu/housenumbers/`
[3] `https://archive.ics.uci.edu/ml/datasets/Adult`
[4] `https://archive.ics.uci.edu/ml/datasets/Diabetes`

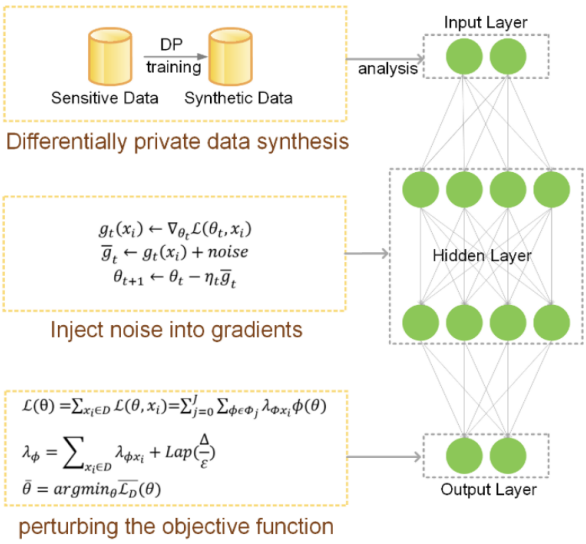

Figure 5: Adding noise to different aspects of the training process to ensure differential privacy

Data point #	Label count before adding noise	Label count after adding noise
1	[2.0, 0.0]	[-29.93, -6.25]
2	[2.0, 0.0]	[-8.48, -9.70]
3	[1.0, 1.0]	[8.45, -5.18]
4	[0.0, 1.0]	[1.47, -0.06]
5	[1.0, 1.0]	[2.02, 1.47]

Table 1: Before and after adding noise to label counts in `NoisyMax` for the Basketball dataset with two classes and $\epsilon = 0.1$

ity voting mechanism described in Snorkel for aggregating weak labels with `NoisyMax` algorithm described in PATE. This meant that we modified the discriminator model in Snorkel is trained. The weak labels assigned to each data sample by the pre-defined labeling functions are aggregated with a certain amount of noise added to them. The amount of noise added is dictated by ϵ or the privacy budget that we define. For instance, for the Basketball dataset (Fu et al. [2020]) with two classes and 17970 data points and $\epsilon = 0.1$, the aggregation of labels before and after adding Laplacian noise have been described in Tables 1 and 2.

3.1 Challenges

Before proceeding to measure the privacy score of the baseline implementations, we grappled with the mathematical formulation of privacy score and understanding ϵ and δ in the DP frameworks. There were several sources that were useful for this, such as the Interactive Counting app [5] and Understanding DP with PATE [6]. We played around with the implementation of PATE in the tutorial and got a deeper understanding of the calculation of data-dependent and data-independent ϵ. Through this tutorial, we were also able to understand the relationship between accuracy and ϵ

[5] https://georgianpartners.shinyapps.io/interactive_counting/
[6] https://towardsdatascience.com/understanding-differential-privacy-85ce191e198a

Data point #	Label count before adding noise	Label count after adding noise
1	[0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]	[0.08134712 -0.04773839 1.1647048 0.0663271 -0.07054743 -0.01966151 -0.04219786 -0.15686153 0.23780519 0.03346664]
2	[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]	[-9.74e-02 -4.37e-02 7.70e-02 8.14e-02 1.24e-05 9.25e-02 -4.16e-02 -2.00e-01 -6.36e-02 5.46e-02]
3	[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]	[0.13181995 0.00661608 0.07122173 -0.03086705 0.12316332 -0.17880009 -0.05490708 -0.21607319 0.02321637 0.04180721]
4	[0. 1. 0. 0. 0. 0. 0. 0. 1. 0.]	[0.42 1.08 0.05 0.07 0.07 -0.20 0.18 -0.07 0.95 0.02]
5	[0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]	[-0.05 -0.26 0.08 1.07 0.02 -0.09 -0.04 0.24 0.11 -0.12]

Table 2: Before and after adding noise to label counts in `NoisyMax` for the ChemProt dataset with ten classes and $\epsilon = 0.1$

in PATE and PATE-G. We also explored another paper(Papernot et al. [2018b]) that touches upon the accuracy-privacy tradeoff in PATE. We think it is worthy to discuss the technical challenges we faced during the deployment of the methods described above:

- One knowledge gap that we had to fill was the exact details of how PATE or PATE-G can be applied to ASTRA and Snorkel. PATE and PATE-G frameworks operate on a teacher-student model and through our exploration into these frameworks, we drew parallels between the frameworks as follows:

  (a) ASTRA has a similar structure to PATE where it utilizes a RAN (teacher) and a student model, and

  (b) while Snorkel does not explicitly follow a teacher-student model, we were able to draw parallels between a teacher model that provides useful and more-or-less accurate labels and labeling functions that achieve the same.

  This allowed us to formalize the application of the differentially-private frameworks onto weakly-supervised models.

- A significant portion of our time was spent in setting up repositories and installing dependencies. For instance, there were some packages that are incompatible with the M1 chip, and as a result, we switched to a Linux machine with a GPU.

- To enable Snorkel and PATE running together, we needed to install them in the same Conda environment. As per recent documentation of package that make PATE available, we chose to use PySyft's implementation of PATE to run PATE. However, the latest version of PySyft only supports Python 3.7 or higher and WRENCH and Snorkel work with Python 3.6. As a result, we had to mitigate several package compatibility issues while setting up the combined environment.

- Then, it turned out that the latest version of PySyft did not support PATE 2017. To mitigate this, we had to syft[7] through the documentation and Github PRs and issues. We learned that

[7] Get it? :)

only the 0.2.x versions of PySyft supported easy access to methods used in PATE. Therefore, we used the 0.2.x branch of PySyft for our purposes.

- The link to the Tensorflow repository with the official code for PATE are outdated, but eventually we were able to find the correct link/path to that as well. For our results, we used a combination of the methods in the official Tensorflow repository[8] and PySyft 0.2.x branch[9].
- To accommodate the correct version of Python in our combined Conda environment, we also had to make changes to the original Tensorflow repository. These are general changes that need to be fixed for any new Python3 version. For instance, `TypeError: can only concatenate list (not "range") to list:` and import statements for utilizing mthods from other Python files.
- We also observed that in the original Tensorflow code for ICLR 2017, there are several parameters that are hard-coded. It took us a while to correctly understand the code since they were not readable owing to hard-coded values for number of teachers in the ensemble and the number of classes in the dataset. We were only able to catch this because we used PATE in combination with WRENCH, which allowed us to run a variety of datasets with varying number of labeling functions (analogous to teachers in the ensemble in PATE) and the number of classes in the dataset.

4 Experiments

In this project, we conducted a series of experiments to evaluate the performance of our weak supervision algorithm, Snorkel with an added layer of PATE. We compared our approach to vanilla Snorkel as well as vanilla PATE. All experiments were conducted on the various datasets made available to us in WRENCH. To ensure the robustness of our results, we performed 3 rounds of experimentation, each with a different randomly selected train-test split. All experiments were implemented in Python (WRENCH and Tensorflow) and run on a 64-bit AMD Ryzen 5 3600 6-Core Processor with an x86_64 architecture. The following is an account of the experiments we performed:

- We trained a simple MLP model on soft labels obtained by labeling functions in vanilla Snorkel. Since we do not add any noise to the weak labels coming from the labeling functions, we assume that the ϵ is ∞ for vanilla Snorkel. We also train models on Snorkel + PATE (weak labels aggregated using PATE's `NoisyMax` algorithm for varying values of ϵ and consequently, Laplacian noise. We perform these experiments on two different datasets, Youtube[10] and Census/Adult UCI[11]. Our results have been summarized in Table 5. It is interesting to note that in cases where the ϵ is very low. the accuracy is worse than random guessing. This must be the result of adding a high amount of noise to already noisy weak labels.

  As expected, the loss value goes down and the accuracy increases as ϵ increases (noise decreases; privacy budget goes up). We have plotted this trend for accuracy vs noise in Figure 6.
- We also conducted an analysis of how the privacy loss changes as we increase the number of queries the student model makes to the teacher ensemble. We have summarized our results for the Youtube dataset with $\epsilon = 0.1$ in Table 4.

[8] https://github.com/tensorflow/privacy/tree/master/research/pate_2017
[9] https://github.com/OpenMined/PySyft/tree/syft_0.2.x
[10] https://archive.ics.uci.edu/ml/datasets/YouTube+Spam+Collection
[11] http://archive.ics.uci.edu/ml/datasets/Census+Income

Run #	Loss	Final val. acc.	Best val. acc.
1	0.141	0.793	0.801
2	0.256	0.794	0.800
3	0.183	0.795	0.798
Average		0.794	0.79

Table 3: Three different runs of training a simple MLP on Snorkel with noisy aggregation vs PATE on the Adult UCI (Census) dataset with fixed noise $\epsilon = 1.33$

Num. of queries	Privacy loss
100	11.75
500	31.51
1000	51.51

Table 4: Trend of how the privacy loss varies with the number of queries made by the student model to the teacher ensemble ($\epsilon = 0.1$)

- Finally, we conducted an experiment to analyze the performance of Snorkel + Noisy PATE Aggregation as opposed to PATE itself. We trained a simple MLP using the noisily aggregated weak labels for the same amount of noise ($\epsilon = 1.33$) value described in the PATE paper for the Census (or Adult UCI dataset). Our Snorkel + PATE aggregation model achieved approximately 80% accuracy, which is almost as much as PATE's benchmark (83%). Our results have been summarized in Table 3. This goes to show that weak supervision methods with noisy aggregation can yield very similar accuracy scores for the same task as a scalable private learning framework such as PATE with the same privacy budget. This goes to show the potential of weak labels as opposed to a fully labeled dataset trained in a special way to enhance privacy, not just in terms of performance, but also in terms of differential privacy. It would be very useful to probe further into this question and perform more comprehensive experiments. At this point, it is also important to note that WRENCH used only a subset of data points (10083 samples) that PATE (16281 samples) used for these results.

5 Related Work

Privacy enabled learning is a well-studied problem, but most work has been evaluated only on simple classification tasks like MNIST, leaving unclear its utility when applied to larger-scale learning tasks and real-world data sets. Move over there are been little work to understand the the privacy implications of SOTA machine learning models trained on sensitive data.

Abadi et al. [2016] develop new algorithmic techniques called deferentially-private stochastic gradient descent (DP-SGD) that modifies stochastic gradient descent to become deferentially private. The algorithm limits the privacy loss per gradient update, rather than updating the weights with the raw gradients, the gradients are clipped, limiting the amount of information learned from any given example. They also define the method to tune the hyper parameters (c - clip and σ - standard deviation of noise sampled from a Gaussian distribution) to get the privacy guarantees for each step in the gradient descent.

The Private Aggregation of Teacher Ensembles, or PATE (Papernot et al. [2018a]), provides an foundational teacher student framework to integrate differential guarantees. PATE transfers

Dataset	Framework	ϵ	DD-ϵ	DI-ϵ	Final loss	Final val. acc.	Best val. acc.
Youtube	Snorkel	∞	–	–	0.274	0.875	0.90
	Snorkel + PATE	0.0001	1.439	1.439	0.661	0.375	0.50
		0.001	1.467	1.457	0.678	0.50	0.70
		0.01	4.139	3.239	0.615	0.334	0.534
		0.1	74.952	51.512	0.648	0.65	0.741
		1.0	56.504	2001.439	0.553	0.75	0.884
		2.0	28.387	4001.439	0.347	0.842	0.90
		5.0	23.370	10001.439	0.316	0.817	0.884
		10.0	34.839	20001.439	0.231	0.792	0.90
Census	Snorkel	∞	–	–	0.061	0.791	0.796
	Snorkel + PATE	0.0001	1.440	1.439	0.597	0.453	0.725
		0.001	1.620	1.457	0.62	0.492	0.695
		0.01	11.806	3.239	0.603	0.526	0.731
		0.1	10.610	51.512	0.595	0.677	0.779
		1.0	1.439	2001.439	0.259	0.793	0.803
		2.0	1.439	4001.439	0.0892	0.796	0.798
		5.0	1.439	10001.439	0.0789	0.792	0.797
		10.0	1.439	20001.439	0.0625	0.785	0.798

Table 5: The privacy metrics (data-dependent and data-independent epsilon) and performance of Snorkel and Snorkel+PATE (noisy aggregation of weak labels) on two different datasets Youtube and Census.

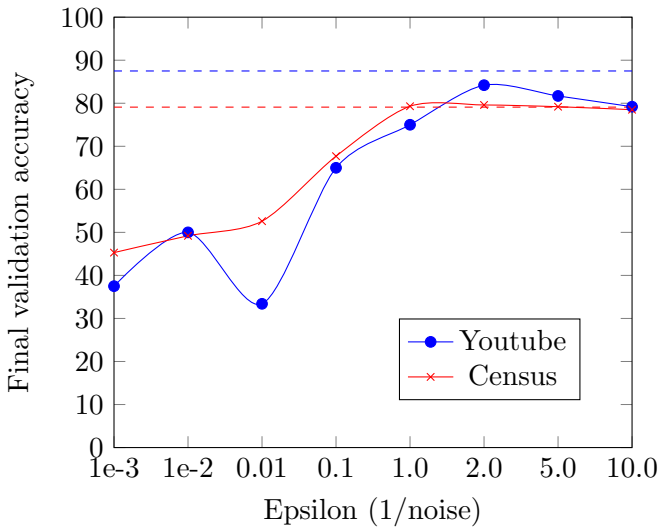

Figure 6: A plot showing how the final validation accuracy of simple MLPs trained on weak labels aggregated using the `NoisyMax` mechanisms from PATE on two different datasets (Youtube and Census) as the level of noise changes (the higher the ϵ, the lower the noise) against the baseline values for vanilla Snorkel (dashed lines).

the knowledge learnt from an ensemble of 'teachers' models to the student, with intuitive privacy provided by training teachers on disjoint data and strong privacy guaranteed by noisy aggregation of teachers' answers. However, PATE is only evaluated on simple classification tasks like MNIST[1]. It is unclear if its utility holds when applied to larger-scale deep learning tasks which use real-world data. PATE-G (Long et al. [2019]), an advancement on PATE, leverages generative adversarial nets to generate data, combined with private aggregation among different discriminators to ensure strong privacy guarantees.

Zhao et al. [2019] and Gong et al. [2020] highlight the three aspect of privacy attacks faced by modern deep learning models: membership inference, training data extraction, and model extracting Figure 7. They also describe the mechanism to introduce differential privacy at various layers in the model such as input layer, hidden layer, and output layer, and discuss their advantages and disadvantages. Although the team provides an excellent theoretical explanation on the making deep learning models deferentially private, they do not present experimental results comparing accuracy privacy trade off in applying these methodologies to the current SOTA models.

Figure 7: An image recovered using a new model inversion attack (left) and a training set image of the victim (right).

The privacy of the training data comes at the cost of the resulting ML models' utility. Boenisch et al. [2022] introduce a new technique built on the top of the PATE framework to support training deep learning models with individualized privacy guarantees. Different data records and different modalities of data have different privacy requirements and suggest that data with lower privacy requirements can contribute more information to the training process of the deep learning algorithms models. They provide a theoretical analysis of their privacy bounds, and experimentally evaluate their effect on the final model's utility using the MNIST, SVHN, and Adult income data sets.

6 Future Work

In conclusion, while weak-supervision models have the potential to significantly improve the efficiency and scalability of learning with limited labels, but they raise important questions about privacy. It is important to consider the tradeoffs between accuracy and privacy introduced by the noise in the aggregation. In addition, it may also be worthwhile to think about the noise introduced by weak labels and how they might potentially enhance the differential-privacy of models.

Going forward, we would like to work on figuring out how exactly to incorporate noisy aggregation in ASTRA as well, since we currently cannot conceptually figure out which step requires the noisy aggregation to be analogous to PATE. This is owing to the iterative self-training nature of ASTRA. Once we are able to draw parallels between ASTRA (with just one teacher and one

student model that depend each other) and PATE (with an ensemble of teacher models that the student model depends on), we should be able to run further experiments to evaluate the relation between accuracy and privacy. We think it is worthy to note here that we were able to set up and run some tests on ASTRA. However, we did not think they were significant to mention in the report because they did not make sense conceptually.

In the future, we would also like to understand the distinctions between the various privacy metrics in the PATE paper better and perform more ablation studies to truly understand the relationship between these metrics. We would also like to play around with the GAN-version of PATE described in the paper as PATE-G and analyze that model in combination with weak-supervision models.

References

M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, page 308–318, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450341394. doi: 10.1145/2976749.2978318. URL `https://doi.org/10.1145/2976749.2978318`.

D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019.

F. Boenisch, C. Mühl, R. Rinberg, J. Ihrig, and A. Dziedzic. Individualized pate: Differentially private machine learning with individual privacy guarantees, 2022. URL `https://arxiv.org/abs/2202.10517`.

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

D. Y. Fu, M. F. Chen, F. Sala, S. M. Hooper, K. Fatahalian, and C. Ré. Fast and three-rious: Speeding up weak supervision with triplet methods. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

M. Gong, Y. Xie, K. Pan, K. Feng, and A. Qin. A survey on differentially private machine learning [review article]. *IEEE Computational Intelligence Magazine*, 15(2):49–64, 2020. doi: 10.1109/MCI.2020.2976185.

G. Karamanolakis, S. Mukherjee, G. Zheng, and A. H. Awadallah. Self-training with weak supervision. *arXiv preprint arXiv:2104.05514*, 2021.

Y. Long, S. Lin, Z. Yang, C. A. Gunter, and B. Li. Scalable differentially private generative student model via PATE. *CoRR*, abs/1906.09338, 2019. URL `http://arxiv.org/abs/1906.09338`.

G. Marcus, E. Davis, and S. Aaronson. A very preliminary analysis of dall-e 2. *arXiv preprint arXiv:2204.13807*, 2022.

N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and Ú. Erlingsson. Scalable private learning with pate. *ArXiv*, abs/1802.08908, 2018a.

N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and Ú. Erlingsson. Scalable private learning with pate. *arXiv preprint arXiv:1802.08908*, 2018b.

A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, volume 11, page 269. NIH Public Access, 2017.

J. Zhang, Y. Yu, Y. Li, Y. Wang, Y. Yang, M. Yang, and A. Ratner. WRENCH: A comprehensive benchmark for weak supervision. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. URL `https://openreview.net/forum?id=Q9SKS5k8io`.

J. Zhao, Y. Chen, and W. Zhang. Differential privacy preservation in deep learning: Challenges, opportunities and solutions. *IEEE Access*, 7:48901–48911, 2019. doi: 10.1109/ACCESS.2019.2909559.