Preserving privacy with PATE for heterogeneous data

Akshay Dodwadmath[1,2] **Sebastian U. Stich**[2]

[1]Saarland University, Germany
[2]CISPA Helmholtz Center for Information Security, Germany

Abstract

Differential privacy has become the standard system to provide privacy guarantees for user data in machine learning models. One of the popular techniques to ensure privacy is the *Private Aggregation of Teacher Ensembles* (PATE) framework. PATE trains an ensemble of teacher models on private data and transfers the knowledge to a student model, with rigorous privacy guarantees derived using differential privacy. So far, PATE has been shown to work assuming the public and private data are distributed homogeneously. We show that in the case of high mismatch (non iid-ness) in these distributions, the teachers suffer from high variance in their individual training updates, causing them to converge to vastly different optimum states. This leads to lower consensus and accuracy for data labelling. To address this, we propose a modification to the teacher training process in PATE, that incorporates teacher averaging and update correction which reduces the variance in teacher updates. Our technique leads to improved prediction accuracy of the teacher aggregation mechanism, especially for highly heterogeneous data. Furthermore, our evaluation shows our technique is necessary to sustain the student model performance, and allows it to achieve considerable gains over the original PATE in the utility-privacy metric.

1 Introduction

Machine learning (ML) has become ubiquitous and is being used in a vast number of domains. The range of deployment of machine learning models has expanded to include even sensitive domains such as healthcare [3, 24] and job interviews [14]. In these domains, the models are trained on sensitive data such as patient records or candidate profiles, the disclosure of which could be harmful to the individuals concerned. This might even lead to refusal of consent of data storage which would in turn hinder the usage of advances in ML. As a result, ensuring privacy and security of data used to train machine learning models has become an important area of research.

Recently, there have been efforts to use differential privacy [5] for ML models. In particular, two of the proposed methods have been vastly popular: the model-agnostic method of PATE [22], and the model-aware method of DP-SGD [1]. We revisit one of the approaches, PATE which uses disjoint subsets of sensitive private data to train a large ensemble of teacher models; the ensemble is then used along with Laplacian noise to predict labels on a set of public data. A student model is trained using these predictions and only this model is made publicly available. This approach has shown to provide guaranteed levels of (ϵ, δ)-differential privacy [7]. However, the framework has been show to work for disjoint subsets that are randomly obtained and are homogeneously distributed. This might not be true always, like in the case of specific, fixed partitions of data provided to the teacher models due to reasons of secrecy, constraints or the need for isolated training [9]. For example, if the teacher models are trained separately in different hospitals, the patient records at each hospital might be biased towards the demographics of the particular region. Thereby, there will be high variance in each of the teacher models' update, and if these models are asked to predict labels for records in a central public repository, they will be in severe disagreement for most of the cases leading to lower prediction accuracy.

Workshop on Distribution Shifts, 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

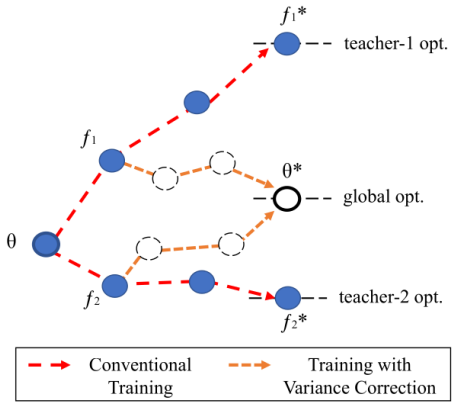

Figure 1: Training of two teachers f_1 and f_2 with the same initialization θ will lead them to converge to different optimums $f_1^\star$ and $f_2^\star$ for heterogeneous data. Adding variance correction during training will help them to converge to the same global optimum θ^*.

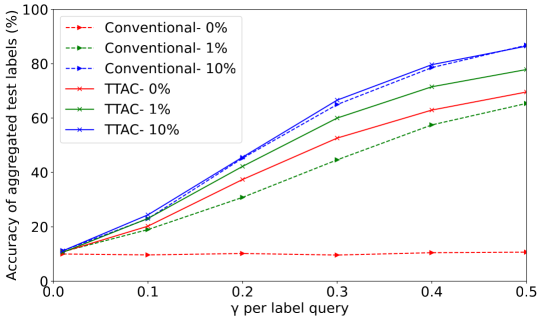

Figure 2: Accuracy of the aggregated teacher models trained using Conventional training (dashed lines) and TTAC (solid lines) for three different data similarity levels, and varying γ per query. Noise scale is inversely proportional to the γ value, thereby a low γ corresponds to high noise and vice versa.

Recently, Karimireddy et al. [10] have shown in distributed learning for different client models trained on heterogeneous sets of data, the drift in the updates of each client model can be reduced by adding a correction to their updates using control variates (variance reduction). We show that making use of such a correction during the training of teacher models leads to improved consensus among the teachers and improved labelling, which in turn improves the student model performance.

Our contributions in this paper are as follows:

1. We show that the performance of the teacher aggregation mechanism in the PATE framework worsens with increased heterogeneity in the private data subsets. We propose to add a global averager model and variance correction to the teacher training process that can mitigate such a distribution shift and sustain the aggregation mechanism performance, even for high levels of heterogeneity.
2. We further compare the student model performance, with knowledge transferred from teachers trained using the original PATE framework and also our updated training procedure, for different data heterogeneity and noise levels. We show that our approach consistently provides better privacy-utility trade-off.

Related Works. There have been recent works to add differential privacy in distributed learning [16, 20], which is similar to our problem setting. However, we believe ours is the first approach that makes PATE work on heterogeneous datasets. Though we analyze in terms of general teacher-student knowledge distillation framework, our method can be extended to the distributed setting.

2 Methodology

We describe the components of our framework built using the student-teacher paradigm of PATE, with a modified teacher training approach. Similar to the original PATE proposal [22], our framework consists of three parts: (1) teacher training, (2) an aggregation mechanism and (3) student training.

Teacher training. We assume that the private training dataset (X, Y) is divided into N partitions where X is the set of inputs and Y is the set of labels. A separate model is trained on each partition to obtain N classifiers or teachers, denoted as $f_i(x, \theta_i)$, where x denotes an input and θ_i the model parameters. Any learning technique can be used to train each f_i, such as stochastic gradient descent (SGD), momentum SGD or ADAM [see e.g. 11]. The conventional approach, also used by the original PATE work is the SGD, in which mini-batches are drawn randomly and for each step $t = 1, 2, ...T$ the following update is done to the parameters θ_i of each teacher i:

$$\theta_i \leftarrow \theta_i - \eta_f \nabla_t f_i(\theta_i)\,, \tag{1}$$

where η_f denotes the step size (shared among all the teacher models) and $\nabla_t f_i(\theta_i)$ a mini-batch gradient computed at iteration t on the data of client i. Suppose each teacher starts from the same initialisation θ. Then each teacher will drift towards the respective optimum $\theta_i^\star$ of its loss function. In case of heterogeneous datasets, each of these $\theta_i^\star$ will most likely be far from each other as well as the global optimum $\theta^\star$ as shown in Figure 1.

In order to reduce the variance in the teacher updates, we make use of: (i) **Averaging** by using an auxiliary global model θ that averages the teachers' updates after every round of training and

Algorithm 1 - Teacher Training with Averaging and Correction (TTAC): an approach to reduce variance in teachers' updates due to data heterogeneity. The colored font indicates the updates done over the original teacher training framework in PATE [22].

averaging model inputs: initial θ, step size η_θ and control variate c.
teacher i's inputs: initial parameters θ_i, dataset partition (X_i, Y_i), step size η_f, decay rate γ and control variate c_i.
for each round $1, ..., R$ **do**
  for each teacher model $1, ..., N$ **do** ▷ training in parallel
    $\theta_i \leftarrow \theta$ ▷ initialize teacher model with averaging model
    for $t \in [T]$ **do**
      compute mini-batch gradient $\nabla_t f_i(\theta_i)$
      $\theta_i \leftarrow \theta_i - \eta_f(\nabla_t f_i(\theta_i) - c_i + c)$ ▷ update teacher model
    end for
    $c_i^+ \leftarrow c_i - c + \frac{1}{T\eta_f}(\theta - \theta_i)$ ▷ update teacher control variates
    $(\Delta\theta_i, \Delta c_i) \leftarrow (\theta_i - \theta, c_i^+ - c_i)$
    $c_i \leftarrow c_i^+$
  end for
  $\eta_f \leftarrow \eta_f \cdot \gamma$ ▷ decrease stepsize
  $(\Delta\theta, \Delta c) \leftarrow \frac{1}{N}\sum_{i=1}^{N}(\Delta\theta_i, \Delta c_i)$
  $\theta \leftarrow \theta + \eta_\theta \Delta\theta$ and $c \leftarrow c + \Delta c$ ▷ update averaging model
end for

(ii) **Correction** by introducing control variates to correct for the drift in the teachers' updates. Specifically, we estimate the update direction of the averaging model (c) and the update direction for each teacher (c_i). This mimics the SCAFFOLD training procedure [10]. The variance of each teacher update is reduced using the difference ($c - c_i$).

Procedure. All the control variate parameters are initialized to zero. In each round of training, every teacher f_i is initialized to the averaging model $f_i \leftarrow \theta$. Then the SGD update for each f_i is modified using the control variates:

$$\theta_i \leftarrow \theta_i - \eta_f(\nabla_t f_i(\theta_i) - c_i + c)\,. \tag{2}$$

This is followed by the teacher control variate c_i update,

$$c_i^+ \leftarrow c_i - c + \tfrac{1}{T\eta_f}(\theta - \theta_i)\,, \tag{3}$$

where T denotes the number of (local) steps. After one round of training is completed for all the teachers, the teacher updates are accumulated to update the averaging model parameters:

$$\theta \leftarrow \theta + \tfrac{\eta_\theta}{N}\textstyle\sum_{i=1}^{N}(\theta_i - \theta)\,, \tag{4}$$

$$c \leftarrow c + \tfrac{1}{N}\textstyle\sum_{i=1}^{N}(c_i^+ - c_i)\,, \tag{5}$$

where η_θ denotes the step size of the averaging model. The stepsize η_f is decayed in every round of training. The proposed algorithm for teacher training with variance correction is shown in Alg. 1.

Aggregation mechanism and student training. We keep the remaining parts of our framework same as the original PATE proposal. Once the teachers are trained, they are deployed as an ensemble to make predictions on unseen inputs x from the public dataset. Each teacher model is queried for a label prediction $f_i(x)$, the labels are counted for every class and random noise is added to arrive at the final prediction for the given sample:

$$f(x) = \arg\max_j \big\{ n_j(x) + \mathrm{Lap}\big(\tfrac{1}{\gamma}\big) \big\}\,, \tag{6}$$

where $n_j(x)$ denotes the vote count for the j-th class (i.e., $n_j(x) = |\{j : f_i(x) = j\}|$) and $\mathrm{Lap}\big(\frac{1}{\gamma}\big)$ denotes the Laplacian distribution with location 0 and scale $\frac{1}{\gamma}$ [following the notation in 22].

Each prediction the aggregation mechanism makes induces a privacy cost. A limited number of labelling is done to limit the privacy cost and the labelled dataset is used to train a student model. Once the training is completed, only the student model is made to be publicly available.

Table 1: Public model (student) accuracy scores for different number of queries, inverted noise scale γ, and data similarity levels. The privacy bound ϵ for respective γ is indicated (for fixed $\delta = 10^{-5}$). We average the results across 10 runs. **TTAC results in a large improvement of student accuracies when data similarity is low (0% or 1%).** This is consistently seen across all values of γ. With increasing similarity levels, both methods result in comparable student performance.

Training Method	Data Similarity ($s\%$)	Number of Queries = 100		Number of Queries = 1000	
		$\gamma = 0.1$ ($\epsilon = 11.75$)	$\gamma = 0.5$ ($\epsilon > 20$)	$\gamma = 0.1$ ($\epsilon > 20$)	$\gamma = 0.5$ ($\epsilon \gg 20$)
Conventional [22]	0%	8.77%	9.46%	9.53%	10.54%
TTAC [ours]	0%	**21.05%**	**61.41%**	**23.36%**	**70.95%**
Conventional [22]	1%	18.41%	56.80%	20.57%	69.59%
TTAC [ours]	1%	**25.57%**	**74.02%**	**22.64%**	**82.83%**
Conventional [22]	100%	**21.91%**	**77.86%**	**26.94%**	**90.73%**
TTAC [ours]	100%	21.37%	75.82%	24.69%	90.06%

3 Experimental Evaluation

Experimental setup. We evaluate using the standard MNIST dataset, consisting of 60,000 training examples and 10,000 testing examples [13]. We create partitions of the training dataset with $s\%$ similar data among the teachers, where $s\%$ is i.i.d. data and the remaining (100 - s)% is sorted according to label [8, 10], with $s \in [0, 1, 10, 100]$. For the test dataset we follow the original PATE work, and use a subset of the first 9000 samples for aggregation mechanism labelling and student model training, and the remaining 1000 samples to evaluate student model performance.

Our experiments reuse the teacher models for MNIST from the original PATE, consisting of convolutional layers with max-pooling and one fully connected layer with ReLUs. To reduce the training complexity, we use the same network of the teacher model for student as well as our averaging model, and restrict the number of teachers to N = 10. We use SGD to train the student model. The original PATE method made use of a student model trained using semi-supervised learning with GANs [23] and experimented upto N = 250 teachers, but we do not need such complexity to show the benefits of our approach. However, due to these simplifications, we do not report the same student utility and privacy levels as in the original PATE work. Further, we compute a data-independent and a data-dependent privacy bound for each prediction, and use composition theorem [6] to arrive at a (ϵ, δ) differential privacy guarantee [7] (refer Appendix A.3). We report these values in Table 1 for each γ.

Aggregation mechanism evaluation. The labelling accuracies of the teacher aggregation trained using Conventional method in original PATE and our proposed method TTAC are compared on the MNIST test set for different $s\%$. We experiment with Laplacian noise of inverted scale γ ranging from 0.01 to 0.5 (similar values evaluated in [22]) and report the corresponding accuracies in Figure 2. It can be seen that the teachers never agree for s = 0 in Conventional method, resulting in the labelling accuracy being equivalent to random guessing ($\sim$10%). This indicates that there is indeed a large drift in the teachers' updates. For highly heterogeneous datasets ($s \in [0, 1]$), aggregation mechanism trained using TTAC consistently outperforms Conventional method, while they remain close for less heterogeneous datasets ($s \in [10, 100]$).

Student model performance. We compare the student accuracy scores in Table 1. For $s \in [0, 1]$, TTAC outperforms Conventional training for the same number of queries (100 or 1000) experimented in the original PATE proposal; this is irrespective of the (ϵ, δ) value. The student model in the Conventional method completely fails for s = 0 with a maximum accuracy score of 10.54%, while TTAC achieves a maximum accuracy score of 70.95%, indicating the importance of our variance correction approach. Even for a low privacy budget (ϵ =11.75), TTAC achieves 21.05% accuracy, meaning our approach can improve the privacy-utility trade off. Though the Conventional method gives a slight improvement over TTAC in the performance levels for s = 100, it is negligible compared to TTAC superiority for heterogeneous datasets.

4 Conclusion

Our work revisited the PATE framework and studied the influence of heterogeneity in teacher data splits. Our experiments showed that such a distribution shift can cause severe issues throughout the

PATE framework, as we observed random guessing performance of aggregation mechanism and student model for highly heterogeneous datasets. We then proposed a modification to the teacher training framework to add correction to the teachers' updates and reduce variance. Our approach is able to overcome the distribution shift and improve student performance even for low privacy budgets. Note that, there could be an additional privacy loss due to the teacher averaging and correction we do in our method, as these are influenced by teachers being trained on private datasets. Though in this work, we verified privacy empirically, we do not do it formally. We plan to investigate this further in future work. Finally, we believe that our work can contribute to real world applications, by making knowledge transfer possible from teachers trained at separate locations or under different constraints, while preserving privacy.

References

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

[2] R. Anil, B. Ghazi, V. Gupta, R. Kumar, and P. Manurangsi. Large-scale differentially private BERT. *arXiv preprint arXiv:2108.01624*, 2021.

[3] Min Chen, Yixue Hao, Kai Hwang, Lu Wang, and Lin Wang. Disease prediction by machine learning over big data from healthcare communities. *IEEE Access*, 5:8869–8879, 2017.

[4] D. Desfontaines and B. Pejó. Differential privacies: a taxonomy of differential privacy variants and extensions (long version). *arXiv preprint arXiv:1906.01337*, 2019.

[5] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. *Theory of cryptography conference*, pages 265–284, 2006.

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

[7] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: privacy via distributed noise generation. *Advances in Cryptology-EUROCRYPT*, pages 486–503, 2006.

[8] T.-M. H. Hsu, H. Qi, and M. Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

[9] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14 (1–2):1–210, 2021.

[10] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. *International Conference on Machine Learning (ICML)*, 2020.

[11] Sai Praneeth Karimireddy, Martin Jaggi, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. Breaking the centralized barrier for cross-device federated learning. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*. Curran Associates, Inc., 2021.

[12] J. Konecny, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[13] Y. LeCun, L Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 1998.

[14] Ali A Mahmoud, Tahani AL Shawabkeh, Walid A Salameh, and Ibrahim Al Amro. Performance predicting in hiring process and performance appraisals using machine learning. *International Conference on Information and Communication Systems (ICICS)*, pages 110–115, 2019.

[15] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. *AISTATS*, pages 1273—1282, 2017.

[16] H. B. McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *ICLR*, 2018.

[17] I. Mironov. Rényi differential privacy. *Computer Security Foundations Symposium (CSF)*, pages 263–275, 2017.

[18] Christopher Mühl and Franziska Boenisch. Personalized PATE: Differential privacy for machine learning with individual privacy guarantees. *arXiv preprint arXiv:2202.10517*, 2022.

[19] I. E. Olatunji, T. Funke, and M. Khosla. Releasing graph neural networks with differential privacy guarantees. *arXiv:2109.08907v1*, 2021.

[20] Y. Pan, J. Ni, and Z. Su. Fl-pate: Differentially private federated learning with knowledge transfer. *2021 IEEE Global Communications Conference (GLOBECOM), 2021, pp. 1-6, doi: 10.1109/GLOBECOM46510.2021.9685079*, 2021.

[21] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and Úlfar Erlingsson. Scalable private learning with PATE. *International Conference on Learning Representations (ICLR)*, 2018.

[22] Nicolas Papernot, Martin Abadi, Ulfar Erlingsson, Ian J. Goodfellow, and Kunal Talwar. Semisupervised knowledge transfer for deep learning from private training data. *International Conference on Learning Representations (ICLR)*, 2017.

[23] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2234–2242, 2016.

[24] Jenna Wiens and Erica S Shenoy. Machine learning for healthcare: on the verge of a major shift in healthcare epidemiology. *Clinical Infectious Diseases*, 66(1):149–153, 2018.

A Appendix

A.1 Framework Overview

We show a complete overview of our approach in Figure 3. Teacher training with our variance correction approach and aggregation mechanism predictions are done privately and are not accessible to an external adversary. Only the student model is released publicly.

A.2 Differential Privacy

Differential privacy (DP) is a technique that allows sharing information about features or patterns of a dataset, without disclosing specific details about each user in the dataset. The outcome of a DP algorithm will roughly be the same with or without the presence of a specific user data, which makes DP a reliable standard for privacy. We recall the following popular form of DP [7]:

(ϵ, δ)-Differential Privacy: *A randomized algorithm $\mathcal{A}$ is (ϵ, δ)-differentially private if for any two neighboring datasets D and D' , and any subset S of possible outputs of $\mathcal{A}$,*

$$\Pr[\mathcal{A}(D) \in S] \leq e^{\epsilon} \Pr\left[\mathcal{A}\left(D'\right) \in S\right] + \delta\,. \tag{7}$$

In our case, neighboring datasets are the training datasets differing by one example, and the randomized algorithm is the model training algorithm. ϵ is the upper bound of the privacy loss for each prediction, and δ is the additional small density of probability for which the upper bound may not hold.

A.3 Privacy Analysis

In this section, we discuss the privacy guarantees provided by our framework. First we recall some quantities introduced in earlier work [1, 6, 22].

Privacy loss. For a randomized algorithm $\mathcal{A}$, neighboring databases D and D', auxilary input aux, and an outcome o of the algorithm, the privacy loss at o is defined as

$$\mathcal{L}(o;\mathcal{A};aux;D;D') \triangleq \log\left(\frac{\Pr[\mathcal{A}(aux;D)=o]}{\Pr[\mathcal{A}(aux;D')=o]}\right)\,. \tag{8}$$

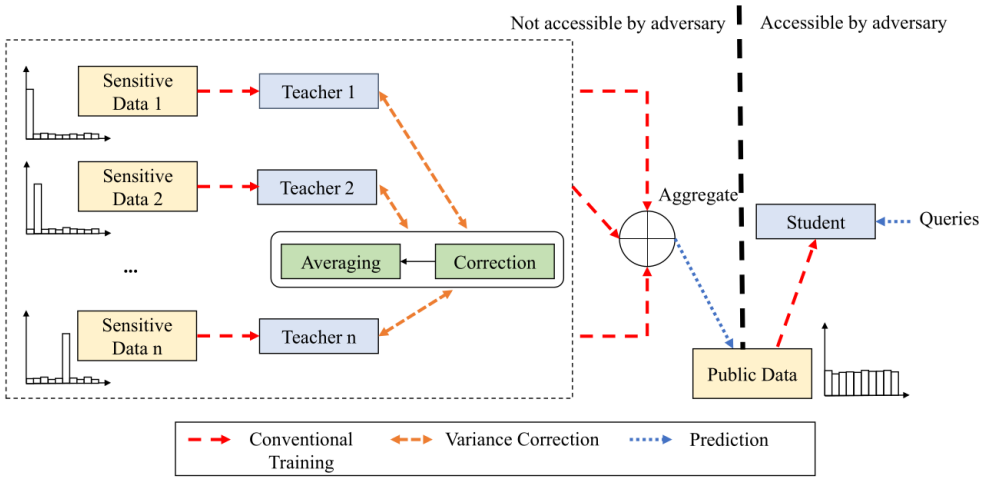

Figure 3: Overview of our approach for preserving privacy with PATE for heterogeneous data distributions among the teachers. We use teacher averaging and update correction, and modify the Conventional training method of PATE [22]. This is followed by the aggregation mechanism prediction on public data for student model training, similar to the original framework.

Privacy loss random variable. The privacy loss random variable $\mathcal{L}(\mathcal{A}; aux; D; D')$ is the random variable defined by evaluating the privacy loss at an outcome sampled from $\mathcal{A}(D)$ [22].

Moments accountant [1]: For a randomized algorithm $\mathcal{A}$, neighboring datasets D and D', auxiliary input aux, the moments accountant is defined as:

$$\alpha_{\mathcal{A}}(\lambda) \triangleq \max_{aux, D, D'} \alpha_{\mathcal{A}}\left(\lambda; aux, D, D'\right) , \tag{9}$$

where $\alpha_{\mathcal{A}}\left(\lambda; \mathrm{aux}, D, D'\right)$ is the moment generating function of $\mathcal{L}(\mathcal{A}; aux; D; D')$, and is termed as the λ^{th} moment.

The original PATE proposal [22], used proof from [6] to infer that their aggregation mechanism satisfies $(2\gamma, 0)$-differential privacy for each step. Because of our modifications to the training mechanism, this property might not hold true. In TTAC the teacher models are in general correlated and it is not possible to directly apply their theorem in our case. We leave this analysis for future work.

In the remainder of this section, we discuss the privacy guarantees under the assumption that $(2\gamma, 0)$-DP holds, and reuse these privacy bounds of Papernot et al. [22]:

Data-independent privacy bound: Suppose that on neighboring databases D, D', the label counts n_j differ by at most 1 in each coordinate. Let $\mathcal{A}$ be the mechanism that reports $\arg\max_j \left\{n_j + \mathrm{Lap}\left(\frac{1}{\gamma}\right)\right\}$. Then $\mathcal{A}$ satisfies $(2\gamma, 0)$-differential privacy. Moreover, for any l, aux, D and D',

$$\alpha(l; aux; D; D') \leq 2\gamma^2 l(l+1) . \tag{10}$$

Data-dependent privacy bound: Let $\mathcal{A}$ be $(2\gamma, 0)$-differentially private and $q \geq \Pr\left[\mathcal{A}(D) \neq o^{\star}\right]$ for some outcome $o^{\star}$. Let $l, \gamma \geq 0$ and $q < \frac{e^{2\gamma}-1}{e^{4\gamma}-1}$. Then for any aux and any neighbor D' of D, $\mathcal{A}$ satisfies

$$\alpha\left(l; aux, D, D'\right) \leq \log\left((1-q)\left(\frac{1-q}{1-e^{2\gamma}q}\right)^{l} + q\exp(2\gamma l)\right) . \tag{11}$$

Papernot et al. [22] further show that q for the aggregation mechanism can be upper bounded, depending on the teacher votes (cf. Appendix A in [22] for proof), using which certain moments are bounded.

Similar to Papernot et al. [22], we use the smaller of the two bounds from equation 10 and equation 11, and the moments are computed for λ values upto 8. We then use the composition theorem from [6] to calculate the bound over all the queries. Finally, we use the tail bound (cf. Theorem 2 in [1]) to convert the moments bound to an (ϵ, δ)-differential privacy guarantee. We report some of the (ϵ, δ) values obtained from this process in Table 1.

We note that it is difficult to quantify the extra privacy loss that can result due to our teacher averaging and correction in the training process, since they do not directly act on private data. However, we plan to study in future work, how our modifications might affect the derivation of privacy bounds.

A.4 Related Works

Differential Privacy in Machine Learning: Different variants of DP have been proposed which provide different advantages and varying levels of privacy guarantees [6, 4, 17], and can be chosen based on the requirements. Our approach as well as the original PATE proposal is based on the (ϵ, δ)-DP defined in Equation 7. There have been extensions of PATE which use other variants such as Rényi-DP [21] and Personalized DP [18].

The same authors from the original PATE work, proposed a modified framework [21] which could aggregate the teachers' answers that are more selective and add less noise. This offered better intuitive privacy, and incurred lower-differential privacy cost. Our approach could easily be extended to this framework. Moreover, there have been other works too to improve the balance between privacy and utility. Mühl and Boenisch [18] applied personalized privacy budgets to each training example, instead of using a single global privacy budget. As far as we know, ours is the first work that focuses on sustaining privacy-utility tradeoff for heterogeneous datasets.

Apart from PATE, another common approach to apply DP in ML models is the Differentially Private Stochastic Gradient Descent [1] or DP-SGD. This framework performs noise addition within the ML training procedure and augments the standard paradigm of gradient based training to be differentially private. The disadvantage of this approach is that it requires white-box access to the model parameters.

The popularity of these two approaches can be seen from the different works that have used them for a variety of problems, such as Anil et al. [2] who extend DP-SGD for large-scale pretraining of the language model BERT-Large, Olatunji et al. [19] who modify the PATE framework for learning on graph specific data using Graph Neural Networks(GNNs) and so on.

Federated Learning: This has become a popular approach in ML for distributed learning [12, 15]. A Federated Learning system consists of local client models trained on user devices, and a central server that aggregates user updates to learn a global model. One of the common problems in federated learning is the variance in client updates that occurs for heterogeneous local datasets, which results in the clients drifting away from the global objective . Karimireddy et al. [10] address this client-drift by using control variates on both server and clients. We use a similar variance correction strategy for our teacher models.

A.5 Algorithm for Conventional Training

The original PATE work uses Conventional training for teacher models, based on SGD update rule defined in equation 1. We provide an overview of this training process in Algorithm 2.

Algorithm 2 - Conventional training: training procedure for teacher models in the original PATE proposal [22].

teacher i**'s inputs**: initial parameters θ_i, dataset partition (X_i, Y_i) and step size η_f.
for each round $1, ..., R$ **do**
  for each teacher model $1, ..., N$ **do**
    for $t \in [T]$ **do**
      compute mini-batch gradient $\nabla_t f_i(\theta_i)$
      $\theta_i \leftarrow \theta_i - \eta_f(\nabla_t f_i(\theta_i))$ ▷ update teacher model
    end for
  end for
end for