DIET-PATE: KNOWLEDGE TRANSFER IN PATE WITHOUT PUBLIC DATA

Michel Meintz, Adam Dziedzic & Franziska Boenisch
CISPA Helmholtz Center for Information Security
{michel.meintz, adam.dziedzic, boenisch}@cispa.de

ABSTRACT

The PATE algorithm is one of the canonical approaches to private machine learning. It leverages a private dataset to label a public dataset, enabling knowledge transfer from teachers to a student model under differential privacy guarantees. However, PATE's reliance on public data from the same distribution as the private data poses a fundamental limitation, particularly in domains such as healthcare and finance, where such public data is typically unavailable. In this work, we propose DIET-PATE which overcomes this limitation by identifying a synergy between programmatically generated data and data-free knowledge distillation. The programmatically generated data serves two critical purposes: first, pretraining both the teacher ensemble and the student model on this data significantly enhances overall performance, as it removes the need to learn generic feature representations solely from the private dataset. Second, by substituting for the public dataset during knowledge transfer, it entirely removes the need for in-distribution data. To correct the resulting distributional shift in the models' hidden layer activations, we incorporate data-free knowledge distillation, which aligns these activations and ensures reliable knowledge transfer. Our experiments demonstrate that DIET-PATE closely matches the performance of standard PATE, despite the absence of in-distribution public data. Furthermore, we show that our approach seamlessly extends to a distributed setting, where each teacher model is trained by a different entity. By eliminating its need for public data, we make PATE and its distributed derivatives practically applicable to sensitive domains.

1 INTRODUCTION

Data privacy is crucial in machine learning since large amounts of data, including datasets from medical, financial, and other sensitive domains, are used to train models. The PATE algorithm Papernot et al. (2017; 2018) is one of the canonical algorithms to obtain privacy-preserving machine learning. PATE transfers the knowledge from a private teacher ensemble to a student model via a privacy-preserving labeling of public data. By limiting the impact of a given teacher on a final label and adding noise to the aggregated labels from the ensemble, the algorithm establishes rigorous (ε, δ)-differential privacy guarantees Dwork et al. (2006). A key challenge of applying PATE to real world settings is the necessity of available public data from the same distribution as the private data, which is often unavailable especially in the medical or financial fields. To eliminate the dependence of PATE on public data, we introduce DIET-PATE for **D**ata-free **I**nformation **E**xtraction and **T**ransfer. Our method leverages two recent advances in data generation and knowledge transfer, which individually provide marginal gains, as we demonstrate in the empirical section. However, when combined, they deliver significant improvements through their synergy.

The first key ingredient in DIET-PATE is the synthetic data that is generated by using a collection of large-scale procedural image programs (Baradad et al., 2022). We use the programmatically generated data to pretrain the teacher models and the student as well as to transfer the knowledge from the teachers to the student. The advantage of using the programmatically generated data for pretraining lies in its ability to enable the model to learn generic features without consuming any privacy budget allocated to the private data. This approach allows to dedicate the entire privacy budget for the private data to teaching the models private-specific features, maximizing its effectiveness. Using the same programmatically generated data as for pretraining also to transfer the knowledge to the student

removes PATE's need of public data. However, this introduces a distribution shift as the teachers are trained on the private data.

To overcome the distribution shift, we rely on the second key ingredient in our DIET-PATE, namely the data free knowledge distillation to reduce the distribution shift. Raikwar & Mishra (2022) identified that the key limitation of distillation with data from a different distribution than the teacher's training set is the covariate shift in the distribution of hidden layer activations of the teacher model. They proposed to effectively reduce the covariate shift by using the current statistics instead of running statistics of the original data in the teacher's batch normalization layers. Applying the data-free knowledge distillation (DataFreeKD) together with the programmatically generated data to PATE opens a new way to perform a private knowledge distillation without any need for public data from the same domain.

Our DIET-PATE also seamlessly extends to a distributed setting, where each teacher is trained by a different party. The PATE algorithm was extended into the CaPC (Confidential and Private Collaborative) learning framework (Choquette-Choo et al., 2021). In CaPC, a distributed network of teacher models collaborates by exchanging predictions in the PATE style, enabling each teacher to enhance its local model's performance while maintaining privacy. The primary bottlenecks of the CaPC framework are the resource-intensive private inference, where teacher models perform inference on encrypted private samples, and the fragmentation of the privacy budget across multiple distributed teachers, which results in modest improvements to several models rather than a significant enhancement of a single one. We demonstrate that our distributed version of DIET-PATE effectively eliminates these two major bottlenecks. First, by leveraging the synthetic data, we replace the costly private inference on encrypted data with the orders of magnitude faster standard inference. Second, since the data used for knowledge transfer is synthetic and in the plain form rather than encrypted as in CaPC, we can publicly release the answers from the teacher ensemble and create a new student model for all collaborating parties. Thus, our distributed DIET-PATE simultaneously improves efficiency and performance for all the collaborating parties.

In summary, we make the following contributions:

1. We introduce the DIET-PATE framework that eliminates the dependence of the canonical PATE framework on the availability of public data from the same distribution as the teacher models' private training data.
2. We show a synergy between the *programmatically generated data*, which we use to pretrain a student and teachers as well as transfer teachers' knowledge, and *data-free distillation* that aligns the activation distributions in private teacher ensemble during knowledge extraction.
3. We demonstrate that our DIET-PATE can be extended to a distributed setting. It significantly improves efficiency, by removing the need for costly private inference, and enables creation of a shared student model that simultaneously provides higher performance across all collaborating parties rather than only partially improving their local models.
4. We conduct extensive empirical evaluations, demonstrating the effectiveness of DIET-PATE for central and distributed differentially private machine learning.

2 Background and Related Work

Differential Privacy (DP) Dwork et al. (2006) is a mathematical framework that provides theoretical upper bounds on the privacy leakage that is incurred by running a randomized algorithm, such as training a model, on private data. Intuitively, DP ensures that no individual's data point significantly impacts the outcome of a computation. Formally, a privacy parameter ε is used to specify the privacy guarantee. In (ε, δ)-DP Dwork et al. (2006), the parameter δ represents the probability, that ε may not hold.

PATE. In this work, we achieve DP by post-processing the outputs of an ensemble of models trained on private data and using the noisy argmax mechanism introduced by Dwork et al. (2014), following the approach of *Private Aggregation of Teacher Ensembles* (PATE) (Papernot et al., 2017). In the noisy argmax mechanism, when queried on unlabeled public data, the teacher ensemble performs a private voting, followed by adding noise to the histogram of vote counts and returning the noisy label with the most votes. As an outcome from the PATE, a public student model is trained on the public

data points with their corresponding noisy labels returned by the teacher ensemble. To compute the privacy guarantees (i.e., a bound on ε), PATE leverages the privacy analysis based on RDP (Mironov, 2017) as introduced by (Papernot et al., 2018). The key limitation of PATE is its dependence on the availability of public samples from the same distribution as the data used to train the teacher ensemble. In this work, we focus on how to overcome this main problem in PATE.

Confidential and Private Collaboration (CaPC). The PATE algorithm was subsequently extended to the CaPC (Confidential and Private Collaborative) learning framework (Choquette-Choo et al., 2021), where a distributed set of teachers exchange the model predictions in the PATE style to improve their own local models. In CaPC, a given participant Q (querying party) encrypts a new unlabeled private example and sends it to all other collaborating (answering) parties, which in this case act like teachers in PATE. Each of the teachers runs private inference on the encrypted example $\hat{x}$. The encrypted labels from all the answering parties are aggregated within multi-party computation (MPC) into a histogram, where we add noise using the "Content Service Provider" and then release the final outcome to the querying party.

Programmatically Generated Data. Programmatically generated data are created using procedural image programs—a collection of large-scale, parameterized programs designed to produce unique outputs with diverse characteristics (Baradad et al., 2022). The generated outputs typically feature simple shapes (*e.g.,* circles, squares, triangles) and textures, which are systematically varied by modifying attributes such as color, size, orientation, patterns, and combinations of shapes. This systematic variation ensures a wide range of visual appearances, enabling the generation of highly diverse datasets. In Figure 4, we plot some example data points generated through the different methods used throughout the paper.

Knowledge Distillation (KD). The transfer of knowledge from one model, called teacher, to another one, denoted as a student, is commonly referred to as knowledge distillation. This strategy is mainly utilized to compress a large teacher model Bucila et al. (2006). However, originally it utilizes the teacher's training data Hinton et al. (2015), which can lead to privacy risks when the training data is private. Instead of using the original data, Raikwar & Mishra (2022) propose *data free knowledge distillation* using Gaussian noise. Yet, utilizing Gaussian noise to transfer knowledge from one model to another poses challenges, due to the covariate shift in the inner activations of the neurons. This shift can be mitigated, by utilizing the batch norm layers Ioffe & Szegedy (2015) and adjusting them to the statistics of the current mini-batch.

3 Our DIET-PATE Framework

The DIET-PATE framework combines the strengths of the PATE approach with programmatically generated data and data-free distillation. Our method comprises four main stages, which are also illustrated in Figure 1:

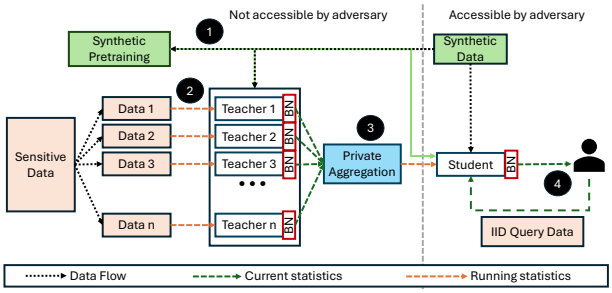


Figure 1: **Overview of our DIET-PATE framework.**

❶ **Initialization:** In the initial stage, the teacher models and the student model are initialized with the same weights. Rather than relying solely on random initialization, we enhance this process with pretraining. Specifically, we pretrain a single model on programmatically generated datasets.The pretrained model is then assigned to all teacher models and the student model, providing a strong starting point for the framework.

❷ **Teacher Fine-Tuning:** In this stage, we partition the sensitive data into non-overlapping subsets, with each subset used to fine-tune an individual teacher model. The important part in our framework is the decision on which statistics, the mean and standard deviation used in the batch normalization layer (Ioffe & Szegedy, 2015), should be used to mitigate the covariate shift (Raikwar & Mishra, 2022). We distinguish between *current statistics* computed dynamically from the current mini-batch

of data, and *running statistics* aggregated over the original sensitive dataset. For the training of the teacher ensemble, we use the running statistics from the respective data partition.

❸ **Private Label Aggregation:** In the third stage, we use synthetic samples and infer them one by one in the teacher ensemble. Each teacher does inference on a given programmatically generated sample using the *current statistics*, since the synthetic data is from a different distribution than the private data used to fine-tune the teachers. We observe that using the pretraining on the synthetic data enables us to obtain a high boost in performance since the teachers' and student statistics are aligned closer to the statistics of the synthetic data. The usage of the current statistics is also crucial to obtain higher agreement between teachers' predictions. The predictions from teachers are aggregated into a histogram as individual labels. Then, we add Gaussian noise to the label counts as in the scalable PATE (Papernot et al., 2018) and return their noisy argmax in a private manner, which we denote as the private aggregation in Figure 1. The final noisy label is released and subsequently used to fine-tune the student model.

❹ **Student Model Deployment:** In the final stage, the fine-tuned student model is made publicly available. Users can privately query the model with new data (IID Query Data) from the same distribution as the original sensitive dataset. During these queries, the student model uses its current statistics to ensure accurate predictions.

3.1 Synergy in DIET-PATE

In DIET-PATE, we combine the effects of programmatically generated data and the data-free knowledge distillation. Our proposed approach stands out for its flexibility in selecting both programmatically generated datasets and knowledge distillation techniques. Enhancements such as creating more advanced synthetic datasets or employing more effective knowledge distillation methods can further improve the performance of the publicly available student model, which is the output from PATE. Thereby, the core innovation of DIET-PATE lies in these two key components, which individually provide marginal gains, as we demonstrate in the empirical section, however, when combined, they deliver significant improvements through their synergy.

3.2 Distributed Learning with DIET-PATE

While the standard PATE framework assumes a centralized party that collects all the data, real-world scenarios often prohibit direct data sharing due to regulations such as the GDPR (General Data Protection Regulation). To address this limitation, we extend DIET-PATE to a distributed setting. Our distributed DIET-PATE overcomes two major limitations of CaPC: (1) Unlike CaPC, which requires resource-intensive private inference on encrypted data, distributed DIET-PATE performs efficient standard inference on programmatically generated synthetic data. (2) Instead of limiting private predictions to a single teacher querying others, distributed DIET-PATE enables the training of a shared student model. This is achieved by aggregating predictions from all collaborating parties on programmatically generated data, ensuring a more effective learning process. Thus, while CaPC spends the privacy budget over all teachers separately, we aggregate the privacy budget into a single student model to achieve a higher utility at the same privacy cost. Distributed DIET-PATE also performs the MPC (Secure Multi-Party Computation) between all the teachers and the "Content Service Provider", as in CaPC, to privately aggregate the answers. An overview of distributed learning with DIET-PATE is presented in Figure 2. In the following, we detail further how distributed DIET-PATE addresses the key challenges of CaPC.

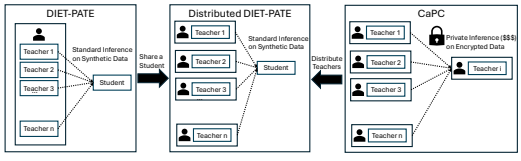


Figure 2: **Distributed DIET-PATE** combines the strengths of both PATE and CaPC frameworks.

Efficient Inference. In the CaPC framework, only a given teacher model i can partially improve its local model by privately labeling its own new private data. Note that, for example, in the medical domain with many collaborating hospitals, such new data points are assumed to be a new patients' records, *i.e.,* highly sensitive data. In the private labeling process with CaPC, teacher i chooses a private sample to be labeled, then the sample is encrypted, and sent to all collaborating parties for private inference on the encrypted data. Each teacher (apart from i) performs

inference on the encrypted sample, and then the teachers' encrypted responses are aggregated to produce a final prediction, which is eventually returned to the querying teacher i. Only the querying teacher i decrypts the final prediction to learn the single label for its private data point. By repeating this process for multiple private samples, teacher i can further fine-tune its local model and improve its performance.

Distributed DIET-PATE solves the main issue in CaPC, which is the costly private inference on encrypted data, by leveraging the programmatically generated synthetic data. Since this synthetic data is non-private, it does not have to be encrypted. To train the student model, we simply generate a new synthetic data sample, and then **all teachers** (no exceptions) can **perform standard inference directly on this unencrypted sample**, which is presented to all the teachers in a plain form. Similarly to CaPC, we aggregate the predicted labels. In distributed DIET-PATE, the final noisy label for the generated synthetic sample is publicly released to train the standard student model, in the same way as in standard PATE.

Shared Student. The newly labeled samples can be used to train a *shared* student model, following a process similar to the standard PATE framework. The privacy budget in distributed DIET-PATE for all the private data is *jointly spent* across the teachers to label the newly generated synthetic data. In contrast, in CaPC, the privacy budget is *divided* among the teachers to answer their individual queries. This division of the privacy budget in CaPC is inefficient as it serves to marginally improve many local models instead of significantly improving a single model. For example, if we assign the total privacy budget of ε to n teachers that allows them to answer q queries in total for the whole ensemble and assume that data and the budget are evenly distributed, then in CaPC, each teacher can spent only up to certain fraction of the privacy budget ε and answer q/n queries to improve its local model. Contrary, in DIET-PATE, the whole ensemble of teachers consumes the full budget of ε and transfers the entire resulting knowledge to the student model using all q queries, hence benefiting from the main advantage of the full ensemble to obtain a higher performance than in any teacher. Thus, in all cases, our distributed DIET-PATE provides more benefits to the whole collaboration.

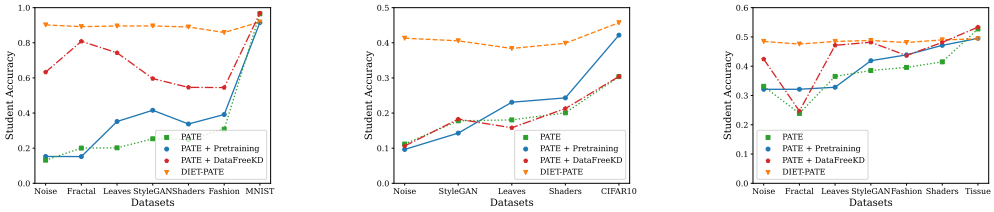
4 Empirical Evaluation

Experimental Setup. We evaluate DIET-PATE using the ResNet18 architecture for all teachers and the student. We also run ablation studies where we follow the standard PATE and CaPC frameworks and use the ResNet10 architecture. The synthetic datasets considered in the experiments are Dead leaves mixed Baradad et al. (2021), StyleGAN-Oriented Baradad et al. (2021), FractalDB Kataoka et al. (2021) and Shaders 21k MixUp Baradad et al. (2022). We use MNIST, CIFAR10, and TissueMNIST (a collection of standardized biomedical images from Kidney Cortex Microscope with 236386 samples) (Yang et al., 2023) as private datasets. A full setup description for our experiments is included in Appendix A. We use the code from Raikwar & Mishra (2022) as the base for the student training while the training of teachers follows the standard PATE setup, including their privacy accounting (Papernot et al., 2018).

The programmatically generated datasets are used to pretrain a ResNet18 model, which serves as the starting point for training both the teachers and the student in DIET-PATE. Pretraining is conducted for 75 epochs on the entire generated datasets using the SimCLR (Chen et al., 2020) self-supervised learning framework, with a learning rate of 3×10^{-4}, two views per image, a batch size of 256, and a feature dimension of 128. The data is resized to match the final task dimensions before training. Based on performance, we select the Shaders21k-pretrained backbone for CIFAR10 and the StyleGAN-pretrained backbone for MNIST and TissueMNIST. Specifically, for CIFAR10, we resize the Shaders21k data to 32×32×3, while for the grayscale MNIST task, we resized the StyleGAN data to 28×28×1. After pretraining, only the last layer is fine-tuned on the sensitive private data for teachers and the transfer data + private label for the student.

4.1 Insights into DIET-PATE

DIET-PATE significantly improves performance. We observe that DIET-PATE outperforms or matches the performance of PATE without using public data. We present the main results in Figure 3, which depicts the accuracy of the student model when using different datasets for the knowledge transfer. The ordering of the synthetic datasets is based on the *Kernel Inception Distance* (KID)



(a) MNIST (b) CIFAR10 (c) TissueMNIST

Figure 3: **DIET-PATE outperforms or matches the performance of PATE without using public data.** This figure shows the knowledge transfer capabilities of the different datasets with a privacy budget of $\varepsilon = 6$, $\delta = 10^{-5}$ for MNIST and $\varepsilon = 10$, $\delta = 10^{-5}$ for CIFAR10 and TissueMNIST.

proposed by Bińkowski et al. (2021). The KID scores are computed between the original private data and the respective dataset (leftmost is the synthetic dataset, namely Gaussian noise, that is the furthest away from the private dataset and rightmost is the data from the same dataset). The values can be found in Table 3. In the figure, we observe multiple clear trends. The pretraining on the synthetic data alone hardly yields any improvements over standard PATE. Similarly, the data free KD yields some improvements over standard PATE, however, its performance still varies widely with respect to the transfer dataset and type of private data. In contrast, our DIET-PATE achieves the best constant performance over all transfer datasets when leveraging the synergy between data free KD and pretraining on programmatically generated data. Successful knowledge transfer is now even possible using only random Gaussian noise, which completely removes the necessity of using any type of structured data.

4.2 DISTRIBUTED DIET-PATE

Efficient Inference. We demonstrate that standard inference on unencrypted data in distributed DIET-PATE is orders of magnitude faster than private inference on encrypted data in CaPC (see Table 1). In these experiments, we use the same small CryptoNet-ReLU model used in CaPC (with two convolutional layers), along with ResNet10 (also used in CaPC) and ResNet18, to evaluate inference speed. Even for significantly larger models than those used in CaPC, such as ResNet18, our approach achieves much faster inference by avoiding the computational overhead of private inference on encrypted data. Moreover, DIET-PATE imposes no restrictions on the models that can be used in a distributed setting. Our results in Table 1 highlight that DIET-PATE enables the use of larger models for more complex tasks in collaborative learning with dramatically faster computation.

Table 1: **The standard inference in distributed DIET-PATE is orders of magnitude faster than private inference in CaPC.** We measure the wall-clock time (sec) for private inference in CaPC vs standard inference in distributed DIET-PATE. We vary the modulus range, N, which denotes the maximum value of a given plain text number to increase the maximum security level possible in CaPC (based on its HE-transformer library (Boemer et al., 2020), which supports private inference only on CPUs). We use the CryptoNet-ReLU model provided by HE-transformer and standard ResNet10 and ResNet18 architectures.

Method (format)	Compute	Model	Batch Size	Forward pass (sec)
CaPC (encrypt N=8k)	CPU	CryptoNet-ReLU	1	14.22 ± 0.11
CaPC (encrypt N=16k)	CPU	CryptoNet-ReLU	1	29.46 ± 2.34
CaPC (encrypt N=32k)	CPU	CryptoNet-ReLU	1	57.26 ± 0.39
DIET-PATE (plain)	CPU	CryptoNet-ReLU	1	0.00038 ± 0.0006
DIET-PATE (plain)	GPU	CryptoNet-ReLU	1	0.00017 ± 0.0008
DIET-PATE (plain)	GPU	ResNet10	1	0.0027 ± 0.0066
DIET-PATE (plain)	GPU	ResNet10	32	0.0045 ± 0.0075
DIET-PATE (plain)	GPU	ResNet18	1	0.0041 ± 0.0065
DIET-PATE (plain)	GPU	ResNet18	32	0.0048 ± 0.0049

Improved Privacy-Utility Trade-Offs. We also compare our distributed DIET-PATE to CaPC in terms of privacy-utility trade-offs for both the **Greedy Teacher** setup, where a single teacher consumes the entire privacy budget to improve their local model, and the **Fair Teachers** setup, where the privacy budget is equally split between all teachers. Our full experimental setup is specified in Appendix A.4.

CaPC: Greedy Teacher. In this case, we assume that one teacher uses the entire privacy budget. This yields the maximum accuracy that can be achieved by a single party in CaPC. Note that while this scenario *can* occur in practice, as CaPC lacks a built-in mechanism to track the privacy budget consumed by each teacher, it is very *unrealistic* because no other party would benefit from the collaboration and, hence, there would be a lack of incentive to participate in it all together. Yet, the scenario can serve as a theoretical upper bound on utility that can be achieved. In our experiments, we give the greedy teacher all additional private query samples and let them query until they exhaust the privacy budget. We then use the labeled data from the collaboration to further fine-tune their model. We report the mean accuracy over the 10 random seeds and the standard deviation.

CaPC: Fair Teachers. In this more realistic setup, we equally split the private query data over the teachers. Each teacher obtains the same fraction of the privacy budget ε. Since the privacy budget does not linearly compose in (ε, δ)-DP, we divide the budget inside Rényi-DP (Mironov, 2017). Since this is the notion that is used in the CaPC and PATE internal privacy accounting, this does not add any overhead. Each teacher can query until they reach their fraction of privacy budget in Rényi-DP. At this point, they have to stop, even when they still have unanswered queries.

We report our results in Table 2. They highlight that our distributed DIET-PATE *significantly* outperforms the fair teachers in all cases. The fact that DIET-PATE does not match the upper bound performance of the greedy teacher for MNIST and TissueMNIST stems from the discrepancy of training data of their respective evaluated models. In the case of CaPC, the evaluated *teacher* model has been trained on its initial private data and the additional new private query data from the same distribution (up to 9k data points for MNIST and CIFAR10, and 42k for TissueMNIST). In contrast, our DIET-PATE *student* model has not seen any single data points from the private training distribution. Instead, it was trained purely on the labeled programatically generated data, and performs inference at test time using the activation alignment, which refers to the alignment of the activation distributions in the models during inference, i.e. the principle behind data-free KD Hinton et al. (2015).Overall, the results highlight that distributed DIET-PATE is a better choice for distributed learning than CaPC in realistic collaborative setups.

Table 2: **The performance of distributed DIET-PATE against CaPC.** We set the $\varepsilon = 6$, $\delta = 10^{-5}$ for MNIST and $\varepsilon = 10$, $\delta = 10^{-5}$ for both CIFAR10 and TissueMNIST. We pretrain all MNIST and TissueMNIST teachers on StyleGAN oriented for distributed DIET-PATE, the CIFAR10 teachers are pretrained on Shaders21k, and the models in CaPC are trained from scratch, following Choquette-Choo et al. (2021). For distributed DIET-PATE, we report the student test accuracy.

Setup	MNIST	CIFAR10	TissueMNIST
CaPC: Greedy Teacher (%)	94.79 ± 0.0070	40.11 ± 0.0124	53.9 ± 0.0055
CaPC: Fair Teachers (%)	85.59 ± 0.0390	39.54 ± 0.0122	35.48 ± 0.0340
Distributed DIET-PATE (%)	89.83 ± 0.0010	41.90 ± 0.0130	48.99 ± 0.0001

5 CONCLUSIONS

In this work, we introduced DIET-PATE, a novel framework that redefines the privacy-preserving machine learning based on the canonical PATE method. DIET-PATE overcomes the primary limitation of the standard PATE, which relies on public data from the same distribution as the private dataset to perform the knowledge transfer. The key to the success of DIET-PATE is the synergy between programmatically generated synthetic data and data free knowledge distillation. Furthermore, DIET-PATE solves the main problems of private and confidential collaborative learning (CaPC) by addressing the high computational cost of private inference and the limited scope of improvements. Through efficient standard inference on GPUs and training of a shared student model, DIET-PATE enables faster, more effective, and scalable collaboration. This advancements enable practical applications of privacy-preserving machine learning in diverse real-world scenarios.

REFERENCES

Manel Baradad, Jonas Wulff, Tongzhou Wang, Phillip Isola, and Antonio Torralba. Learning to see by looking at noise. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21. Curran Associates Inc., 2021. ISBN 9781713845393.

Manel Baradad, Richard Chen, Jonas Wulff, Tongzhou Wang, Rogerio Feris, Antonio Torralba, and Phillip Isola. Procedural image programs for representation learning. *Advances in Neural Information Processing Systems*, 35:6450–6462, 2022.

Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans, 2021. URL https://arxiv.org/abs/1801.01401.

Fabian Boemer, Rosario Cammarota, Daniel Demmler, Thomas Schneider, and Hossein Yalame. Mp2ml: A mixed-protocol machine learning framework for private inference. In *Proceedings of the 15th international conference on availability, reliability and security*, pp. 1–10, 2020.

Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Knowledge Discovery and Data Mining*, 2006. URL https://api.semanticscholar.org/CorpusID:11253972.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Christopher A. Choquette-Choo, Natalie Dullerud, Adam Dziedzic, Yunxiang Zhang, Somesh Jha, Nicolas Papernot, and Xiao Wang. Capc learning: Confidential and private collaborative learning. In *ICLR (International Conference on Learning Representations)*, 2021.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data ourselves: privacy via distributed noise generation. In *Advances in Cryptology-EUROCRYPT*, 2006.

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

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

Hirokatsu Kataoka, Kazushige Okayasu, Asato Matsumoto, Eisuke Yamagata, Ryosuke Yamada, Nakamasa Inoue, Akio Nakamura, and Yutaka Satoh. Pre-training without natural images, 2021. URL https://arxiv.org/abs/2101.08515.

Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, pp. 263–275. IEEE, 2017.

Nicolas Papernot, Martin Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *Proceedings of the 2017 ICLR conference*, 2017.

Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Ulfar Erlingsson. Scalable private learning with pate. In *Proceedings of the 2018 ICLR conference*, 2018.

Piyush Raikwar and Deepak Mishra. Discovering and overcoming limitations of noise-engineered data-free knowledge distillation. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 4902–4912. Curran Associates, Inc., 2022.

Jiaqi Wang, Roei Schuster, Ilia Shumailov, David Lie, and Nicolas Papernot. In differential privacy, there is truth: on vote-histogram leakage in ensemble private learning. *Advances in Neural Information Processing Systems*, 35:29026–29037, 2022.

Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

Table 3: **KID scores.** We compute the scores for all programmatically generated data that we utilize for the knowledge transfer.

Dataset	StyleGAN	Dead Leaves	Shaders21k	FractalDB	Gaussian noise
MNIST	0.196	0.266	0.271	0.504	0.53
CIFAR10	0.149	0.307	0.085	0.4116	0.7599
TissueMNIST	0.174	0.2963	0.1526	0.464	0.62

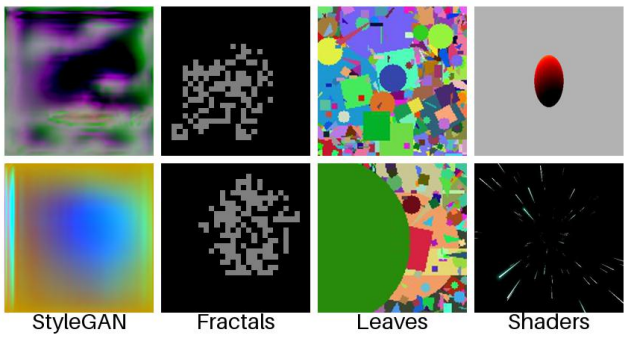


Figure 4: **Examples of programmatically generated data.**

A Experimental Setup

A.1 Datasets

We present the additional examples of programmatically generated data used as public data in Figure 5.

A.2 Models

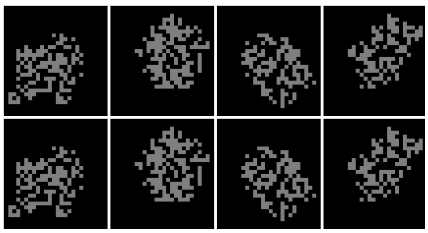
In line with Standard PATE (Papernot et al., 2018), we also experiment with ResNet10. Additionally, we include ResNet18. Teacher accuracies for MNIST are presented in Table 4.

A.3 Hyperparameters

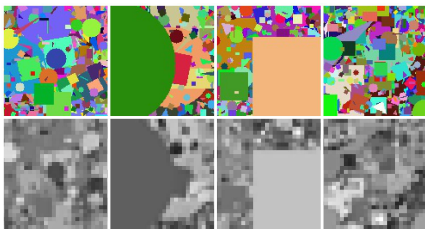
We detail all hyperparameters of our experiments in Table 6. Through extensive analysis of the privacy budget, we derived different privacy parameters for each dataset, depending on the number of teachers and the consensus of the transfer dataset. For the synthetic datasets we used the available GitHub repositories to download pre-generated data, if available, or generated them ourselves. The number of samples per synthetic dataset are detailed in Table 6b. To generate the two views in the SimCLR Chen et al. (2020) framework, we use the following setup: *RandomResizedCrop* to the final size, i.e. 28x28 for the pre-training for MNIST and TissueMNIST as well as 32x32 for the pretraining of CIFAR10. Then a *RandomHorizontalFlip*, a *RandomColorJitter* with a probability of 0.8, a *GaussianBlur* with a kernel size of 0.1 * image_size. For the pretraining for CIFAR10 we also apply a *RandomGrayscale* with probability of 0.2

A.4 CaPC Utility Evaluation Setup

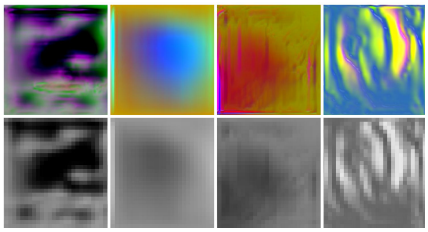
MNIST. For the evaluation of MNIST, we assume 200 teachers, each instantiated as a ResNet18 model. The 60k training data points are evenly split into 200 subsets of 300 samples each, serving as individual training data for the teachers. We fine-tune only the last layer of each model using these subsets. Following CaPC, we designate 9k samples from the test set as additional private data, allowing teachers to query each other, obtain labels, and refine their own models. The remaining 1k test samples are used for performance assessment. Since the order of queries in PATE and CaPC influences the number of queries that can be answered within a given privacy budget (*e.g.,* queries rejected due to low consensus incur lower costs, allowing more queries overall), we repeat our



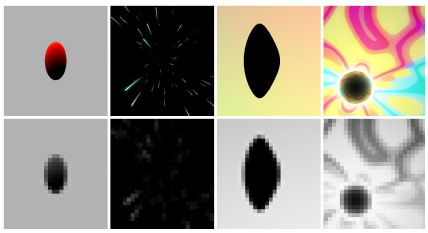
(a) Original FractalDB images Kataoka et al. (2021), Bottom row: Rescaled to be (28px, 28px).



(b) Top row: Original Dead leaves mixed images Baradad et al. (2021), Bottom row: Rescaled to be (28px, 28px).



(c) Top row: Original StyleGAN-oriented images Baradad et al. (2021), Bottom row: Rescaled to be (28px, 28px).



(d) Top row: Original Shaders21k images Baradad et al. (2022), Bottom row: Rescaled to be (28px, 28px).

Figure 5: Examples of programmatically generated data.

Table 4: **Teacher accuracy on MNIST.** We report the average teacher performance on MNIST.

Architecture	Number of teachers	Average Teacher Validation Accuracy
ResNet10	200	85.91%
ResNet18	200	89.34%

experiment across 10 different random seeds. These seeds correspond to different random orderings of private samples for the greedy teacher baseline and varying assignments of the 9k samples across the 200 teachers for the fair baseline. For distributed DIET-PATE, we discard the 9k in-distribution queries and instead use StyleGAN-generated synthetic data while maintaining the same privacy budget ($\varepsilon = 6$). This additional in-distribution data provides CaPC with a practical advantage over DIET-PATE. Distributed DIET-PATE utilizes ResNet18 models pretrained on StyleGAN. We then evaluate two CaPC baselines.

TissueMNIST. TissueMNIST is evaluated on 250 teachers, instantiated as ResNet18 model. The 165k training data points are split evenly among the teachers. We assume 90% (42k samples) of the test set as private data, on which the teachers can query each other. The rest is used to evaluate

Table 5: Accuracies of teachers on different pretraining backbones

Dataset	Number of teachers	Pretraining dataset	Avg. Teacher Accuracy
MNIST	200	Dead Leaves Mixed	84.46%
MNIST	200	StyleGAN oriented	85.13%
MNIST	200	Shaders21k MixUp	76.14%
CIFAR10	50	Shaders21k	46.72%
TissueMNIST	250	Dead Leaves Mixed	46.5%
TissueMNIST	250	StyleGAN oriented	48.8%
TissueMNIST	250	Shaders21k MixUp	43.57%

	Num Teachers	T	σ_1	σ_2
MNIST	200	150	120	40
CIFAR10	50	50	30	15
TissueMNIST	250	170	100	40

(a) PATE Hyperparameters

	Num Gen Samples
Shaders21k	1300000
StyleGAN	105000
Dead Leaves	105000

(b) Training Hyperparameters (for pre-training)

Table 6: Hyperparameters used in our experiments.

	Epochs	LR	Optimizer	Batch Size	Pretraining Dataset	δ
MNIST	50	10^{-3}	Adam	256	StyleGAN-oriented	10^{-5}
CIFAR10	50	10^{-3}	Adam	128	Shaders21k MixUp	10^{-5}
TissueMNIST	50	10^{-3}	Adam	256	StyleGAN-oriented	10^{-5}

(a) Training Hyperparameters (from scratch)

the performance of the models. For distributed DIET-PATE we use ResNet18 models trained on StyleGAN and discard the 42k in-distribution samples. Instead we use StyleGAN-oriented synthetic data with the same privacy budget.

CIFAR10. For the evaluation of CIFAR10, we assume 50 teachers, each instantiated as a ResNet18 model. We split the 50k training datapoints evenly into 50 subsets of 1k samples each, serving as individual training data for the teachers. 9k samples of the test set are chosen as additional private data and the remaining 1k are left for performance assesment. For distributed DIET-PATE we use ResNet18 models pretrained on Shaders21k and utilize Shaders21k synthetic data for the student training instead of the in-distribution data.

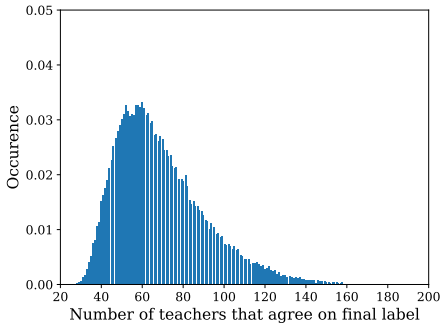
B Additional Insights

B.1 Towards Putting PATE on a Data Diet

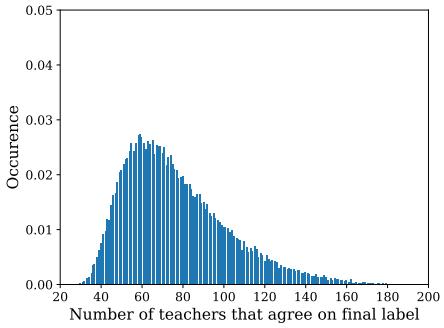
There are two fundamental differences between PATE and KD that affect the extent to which PATE can benefit from the data free KD and inform how to best leverage the approach in our setup.

The first difference is that during standard KD (Hinton et al., 2015), the student has *access to the full logits* output by the teachers. Then, using a KD-loss—that aligns every component of the student and teacher's output vector—leads to a strong alignment between their predictive behavior. However, in PATE, teachers only output labels to respect the DP guarantees, since sharing per-class outputs has been shown to leak additional privacy (Wang et al., 2022). Despite the difference in the teacher outputs from standard KD vs PATE, we find that the data free KD helps to obtain a higher consensus in the predictions from the teacher ensemble and more effectively transfer the knowledge to the student.

The second difference is that PATE operates with multiple teachers instead of a single teacher as in KD. When independently training each teacher from a different random seed using different private subsets, we observe a *limited consensus* between the teachers, see Figure 6a. This is disadvantageous



(a) Consensus density plot for teachers with different initializations of weights. $\mu = 69.3$, $\sigma = 22.19$.



(b) Consensus density plot for teachers with the same initialization of weights. $\mu = 76.88$, $\sigma = 25.5$.

Figure 6: **Consensus of teachers.** A comparison between the consensus density of teachers with different vs same weight initialization when using DataFreeKD to adjust the model statistics. μ is the mean number of teachers that agree on a given label and σ is the standard deviation of the number. We use 200 teachers. We observe that significantly fewer teachers with different initial weights ($\mu = 69.3$ in plot a) agree on a label compared to teachers initialized with the same weights ($\mu = 76.88$ in plot b). Please see Figure 8 for more details (and a larger visualization).

	Teacher: Same	Teacher: Different
Student: Same	**63.6%**	51.5%
Student: Different	59.2%	51.4%

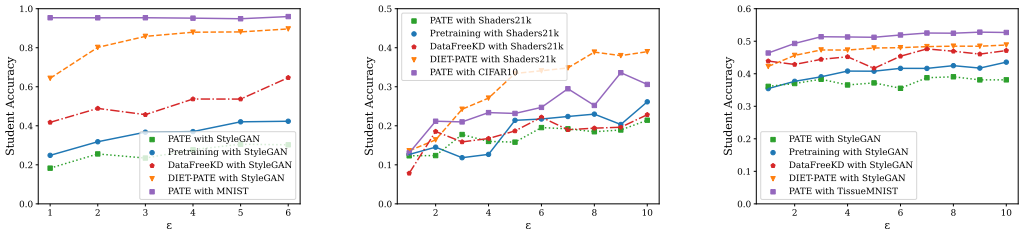
Table 8: **Accuracy for different initializations of weights in teacher models.** Accuracies for ($\varepsilon = 10$, $\delta = 10^{-5}$)-DP using Gaussian noise and DataFreeKD to transfer knowledge for the MNIST dataset using ResNet10 as the base architecture for all models. Teachers either all have the same initialization or different random initialization.

for PATE where the privacy costs incurred by every query depend on the consensus with higher consensus leading to lower privacy costs per query and the possibility to answer more queries—yielding more training data for the student. We hypothesize that the low consensus is caused by querying random data where every teacher has, despite the activation calibration, a slightly different behavior, and the fact that PATE uses hard labels, where small differences in class probabilities can already cause a full label flip. When exploring solutions, we identified a key component of effectively applying data free KD on PATE, namely initializing the teachers with the same weights. This aligns their low-level behavior and leads to a higher consensus, as we show in Figure 6b, and ultimately yields higher student accuracy. For example, when initializing teachers and student with the same random weights, using MNIST as private data, and transferring with Gaussian noise, we observe that we can increase performance to 63.6% from 51.4% when they are initialized differently (see Table 8).

The unique insight on the required alignment of weights suggests a further improvement: Instead of initializing student and teachers with the same *random* weights and training all weights from scratch, which might still cause a large deviation from the initially similar behavior, we can also rely on *transfer learning*. In transfer learning, a model is pretrained on one dataset. To apply the pretrained model to another dataset, solely the last layer(s) have to be fine-tuned on the new data. The transfer learning only incurs minimal changes to the general model behavior. We find that pretraining on programmatically generated synthetic data increases alignment between teachers significantly.With all these insights, that inform our unique and beneficial design choices, we are ready to introduce DIET-PATE.

B.2 Privacy Utility Trade-Offs in DIET-PATE

Private-Utility Trade-offs. DIET-PATE achieves a significantly better privacy-utility trade-off than PATE when leveraging the programmatically generated data. We present the main results



(a) MNIST (b) CIFAR10 (c) TissueMNIST

Figure 7: **DIET-PATE exhibits much better privacy-utility trade-off than PATE when using only the programmatically generated data.** We present how the increased privacy budget corresponds to a higher accuracy of the student model.

of our method for different ε values in Figure 7. Already for MNIST and TissueMNIST, the synergy between pretraining and data-free knowledge distillation (KD) is clearly evident. For more challenging tasks, such as CIFAR10, where a small number of samples is insufficient to enhance performance, DIET-PATE even provides *substantial* improvements by incorporating more pretraining data and mitigating distribution shift. Furthermore, in this case, DIET-PATE even outperforms the standard PATE, which relies on the in-distribution public data. For simpler tasks such as MNIST and TissueMNIST, a sufficient number of publicly available samples from the same distribution as the private data can already boost the performance, sometimes even slightly exceeding the effectiveness of programmatically generated data. Here, the reduction in distribution shift alone is a bit more beneficial than the use of pretraining datasets.

C Additional Experiments

Table 9: The resulting number of labels returned and the student accuracy for the standard knowledge transfer from PATE (no pretraining) and ResNet10 using different transfer datasets for MNIST. We set the following parameters for PATE: $T = 150$, $\sigma_1 = 120$, $\sigma_2 = 40$, $\delta = 10^{-5}$ for all datasets. $T = 200$, $\sigma_1 = 100$, $\sigma_2 = 20$, $\delta = 10^{-5}$ for FMNIST (Fashion MNIST dataset). **There is not enough public data to fulfill the whole privacy budget,* $\varepsilon = 6.47$. Note that RS denotes running statistics and CS represents current statistics.

Dataset	ε =5	ε =8	ε =10	ε =20
MNIST + RS	2882	4631*	-	-
MNIST + CS	2927	4717*	-	-
Noise + RS	2059	4514	6578	19713
Noise + CS	1222	2732	3942	11967
FractalDB + RS	2381	5291	7682	23036
FractalDB + CS	1548	3465	5023	15033
Shaders21k + RS	1986	4454	6350	19351
Shaders21k + CS	1983	4400	6393	19232
Leaves + RS	1389	3191	4663	13907
Leaves + CS	1674	3700	5416	16316
StyleGAN + RS	1839	4054	5889	17843
StyleGAN + CS	2017	4468	6521	19673
FMNIST + RS	724	1569	2316	6989
FMNIST + CS	849	1885	2745	8268

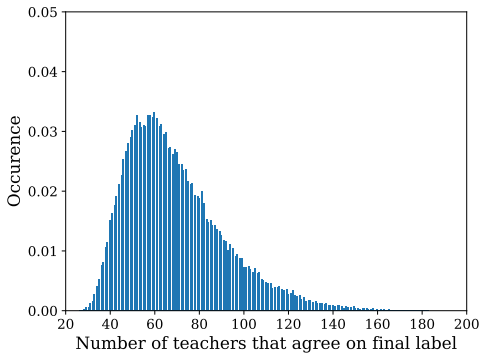
(a) Number of labels returned.

Dataset	ε =5	ε =8	ε =10	ε =20
MNIST + RS	95.2% ± 0.3%	95.8*% ± 0.8%	-	-
MNIST + CS	95.9% ± 0.4%	96.6*% ± 0.5%	-	-
Noise + RS	9.6% ± 0.4%	9.3% ± 0.1%	9.4% ± 0.1%	9.7% ± 0.4%
Noise + CS	34.8% ± 2.7%	52.6% ± 2.1%	61.8% ± 1.0%	76.7% ± 1.8%
FractalDB + RS	10.9% ± 0.8%	13.1% ± 1.2%	12.7% ± 1.6%	18.2% ± 1.0%
FractalDB + CS	58.5% ± 1.7%	72.2% ± 1.4%	75.9% ± 1.2%	85.3% ± 0.9%
Shaders21k + RS	25.7% ± 1.6%	27.4% ± 1.3%	30.3% ± 1.1%	33.0% ± 0.9%
Shaders21k + CS	45.9% ± 3.0%	55.1% ± 1.9%	57.1% ± 2.4%	65.1% ± 2.1%
Leaves + RS	34.1% ± 2.3%	37.9% ± 4.4%	40.1% ± 3.7%	44.4% ± 1.2%
Leaves + CS	58.7% ± 0.8%	68.1% ± 2.8%	72.7% ± 2.1%	82.7% ± 0.9%
SytleGAN + RS	36.2% ± 1.3%	36.9% ± 1.5%	40.1% ± 1.3%	44.6% ± 2.8%
SytleGAN + CS	50.1% ± 2.6%	58.3% ± 2.4%	62.5% ± 2.8%	71.1% ± 1.8%
FMNIST + RS	40.0% ± 0.7%	43.5% ± 1.0%	45.9% ± 1.6%	50.9% ± 1.1%
FMNIST + CS	54.2% ± 1.1%	59.8% ± 2.2%	63.7% ± 1.3%	75.2% ± 0.9%

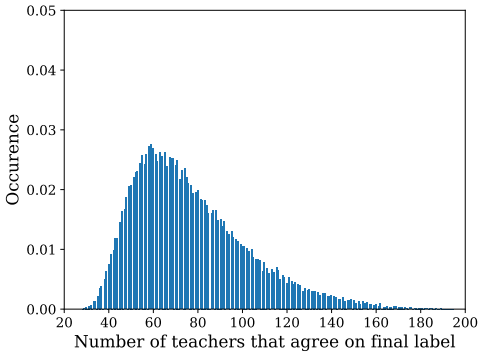
(b) Student accuracy.

C.1 Choice of Transfer Data

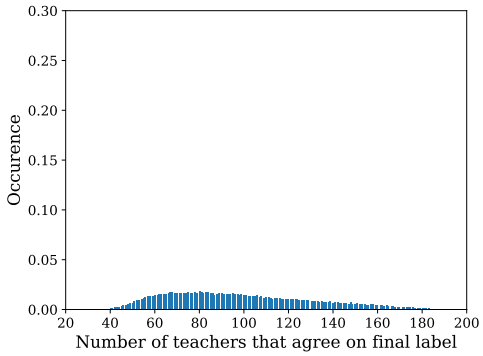
We perform an additional comparison between the consensus density of teachers in Figure 8.



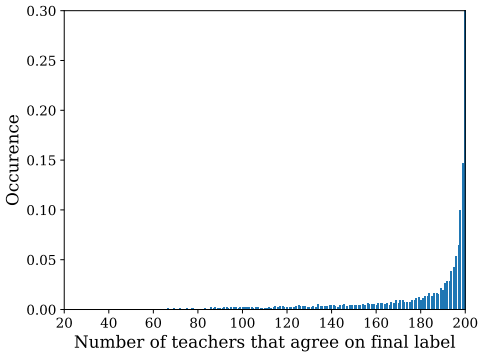
(a) Consensus density plot for teachers with different initializations of weights. $\mu = 69.3,\ \sigma = 22.19$.



(b) Consensus density plot for teachers on Gaussian noise with the same initialization of weights. $\mu = 76.88.6,\ \sigma = 25.5$



(c) Consensus density plot for teachers on public FashionMNIST data with the same initialization of weights. $\mu = 98.6,\ \sigma = 32.8$



(d) Consensus density plot for teachers on public MNIST data with the same initialization of weights. $\mu = 181.29,\ \sigma = 29.4$

Figure 8: **A full comparison between the consensus density of teachers.** We follow the notation from Figure 6.