Privacy and Performance: Analysis of Biomedical Image Segmentation

Parth Shandilya
University of Basel, Switzerland

Abstract

Machine learning (ML) models are vulnerable to information leakage of their training data, which can be highly sensitive, for example, medical images. Privacy-enhancing technologies like differential privacy protect training ML models while bounding the risks of reconstructing the original data. Differentially private stochastic gradient descent (DP-SGD) integrates differential privacy into stochastic gradient descent (SGD) by adding noise to gradients during training to meet privacy standards. However, DP-SGD incurs a performance penalty, and several optimizations have been proposed to reduce the performance penalty. This paper aims to use optimization techniques proposed in the Opacus and Book-Keeping algorithms to evaluate the privacy and performance of training optical coherence tomography (OCT) scans of retina images using segmentation. We discovered that Book-keeping is more efficient in managing the privacy budget than Opacus. Regarding performance analysis, Book-keeping demonstrates better optimization and generalization in larger datasets.

Index Terms

Differential privacy, deep learning, optical coherence tomography (OCT), performance, privacy, segmentation.

I. Introduction

Privacy concerns in machine learning are rising as ML models are trained on sensitive data [1]. DP-SGD [2] offers a solution by modifying traditional SGD [3] to include mechanisms that protect user data. This paper analyzes two differentially private [4] optimization mechanisms, Opacus [5] and Book-Keeping [6], with DP-SGD by performing the segmentation of the OCT [7] scans image datasets using U-Net [8] convolutional neural network architecture to evaluate the optimization effects on privacy and performance. The goal is to understand both optimizations and their ability to integrate differential privacy in the training of ML models. Below, we highlight the key technical concepts relevant to this paper.

Segmentation is the process of grouping an image into coherent sub-regions based on extracted elements such as color or texture properties, and then categorizing each sub-region into one of the predetermined classes. Segmentation has extensive applications in precision medicine, particularly in the development of computer-aided diagnosis based on radiological images of various modalities, such as magnetic resonance imaging (MRI), optical coherence tomography (OCT), computed tomography (CT), or colonoscopy images. [9]. U-Net [8] is a convolutional network architecture designed for fast and precise segmentation of images which is tailored explicitly for biomedical semantic segmentation.

An optical coherence tomography (OCT) scan is a procedure that reflects visible light from a low-power laser to obtain retina images. The principle is similar to ultrasound but uses light instead of sound [7]. Retinal layer segmentation detects and prognoses diseases in OCT scan images. These images are critical for the early detection of eye-related diseases such as diabetic retinopathy and age-related macular degeneration.

II. Importance of Privacy in Medical Images

Privacy is a contextual word and is interpreted differently depending on the context. When it comes to the privacy of our data, privacy means we have the right to control how our personal information is collected and used. On the other hand, differential privacy is defined using a before-and-after approach, which means the researcher should not know more about any individual after analyzing data, and any adversary who already has an individual's information should not have a different view after accessing a database [4]. The following reasons depict the significance of privacy in medical images.

As with any digital data, retinal images stored or transmitted electronically are vulnerable to unauthorized access through data breaches. If not adequately protected, these images could be misused for identity theft, unauthorized tracking, or discrimination. For our segmentation task, we consider OCT scans taken from retinal images, which are unique to individuals and much like fingerprints. These scans contain sensitive personal information that could reveal medical conditions related to the eye or other health issues. Retinal images have applications beyond their primary use in medical diagnosis, including research and training AI models. Notably, a study on chest X-ray data [10] is the first to demonstrate that a well-trained deep-learning system can successfully recover patient identities from chest X-ray images.

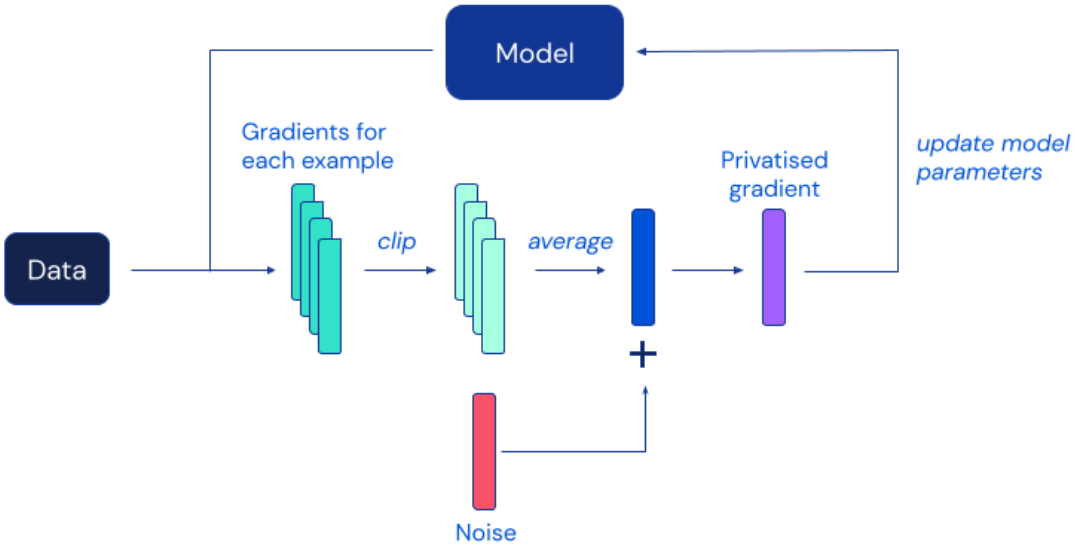

Fig. 1. DP-SGD generates model updates with privatised gradients by performing gradients processing from each example and introducing noise.

III. DP-SGD and Associated Challenges

Previously, we mentioned that DP-SGD [2] integrates differential privacy into stochastic gradient descent [3] during training to meet privacy standards as shown in Figure 1. However, DP-SGD [2] incurs a performance penalty because of the mechanism it uses to ensure privacy.

The following are the primary challenges in this step, which are the main focus of analysis for this paper:

1) Gradient clipping is a key algorithmic step in DP-SGD [2]. The gradients are clipped to a predefined threshold to limit the influence of any single data point. A too-high threshold might not sufficiently protect privacy, while a too-low threshold can degrade the model's performance by excessively clipping gradients.
2) Gaussian noise is added to the clipped gradients before updating the model parameters. We must calibrate the noise added after clipping properly to balance privacy and model performance.

IV. Methodology

This section outlines two differential privacy optimization techniques used during training, highlighting key aspects relevant to our analysis and providing an understanding of the hyperparameters in the context of each optimization. We evaluate the Opacus and Book-keeping optimization on a segmentation task, which has yet to be evaluated in either of the papers mentioned below.

A. Opacus: User-Friendly Differential Privacy Library in PyTorch [5]

Opacus's paper highlights the high cost of training machine learning with differential privacy and its impact on performance while ensuring privacy. Opacus's proposed solution made implementing differential privacy in the machine learning pipeline easier for researchers focusing on simplicity, flexibility and speed. Opacus computes batched per-sample gradients and optimized vector computations, providing higher efficiency. We focus on privacy accounting and noise scheduler hyperparameters for our OCT scan analysis. Privacy accounting enables users to set a target (ϵ, δ) privacy budget and automatically adjust the noise level (σ) internally to meet this budget. On the other hand, the noise scheduler adjusts the noise multiplier during training according to predefined schedules. The proposed solution is modular and integrates perfectly with the Pytorch [11] library, simplifying the coding of complex networks and reducing boilerplate code. Opacus's paper evaluates image classification tasks and claims the solution enables differentially private training with manageable overheads for most tasks and layers. However, challenges remain for complex models and larger batch sizes, mainly due to the inherent requirements of DP-SGD [2]. The proposed solution is in detail in the referenced paper [5].

B. Differentially Private Optimization on Large Model at Small Cost [6]

The paper underlines that per-sample gradient norms are computationally intensive. It also mentions that existing solutions slow training because of multiple rounds of back-propagation and are $2 - 1000\times$ more costly in time and space complexity than the standard (non-private) training. In order to make the differentially private training as efficient as the standard training,

the Book-Keeping technique is introduced, which requires only a single round of back-propagation and never instantiates per-sample gradients like Opacus and the Non-DP training. Book-Keeping algorithm automatically switches the standard training of any model to its differentially private version with the Pytorch [11] library by adding a single piece of code. We focus on the clipping function and clipping mode hyperparameter for our OCT scan analysis. The clipping function uses automatic clipping [12], and clipping mode uses hybrid Book-Keeping to overcome the computational challenge of training large models with high-dimensional data. The Book-Keeping algorithm is described in detail in the referenced paper [6]. The paper evaluates the Book-Keeping technique for image classification tasks. It claims that the solution reduces the time and space complexity of differentially private training to a similar level to standard training. The Book-Keeping technique claims to be well suited for training large models, saving memory up to $10\times$ and boosting the speed by $30\% \sim 5\times$ than previous differentially private implementations.

C. Datasets

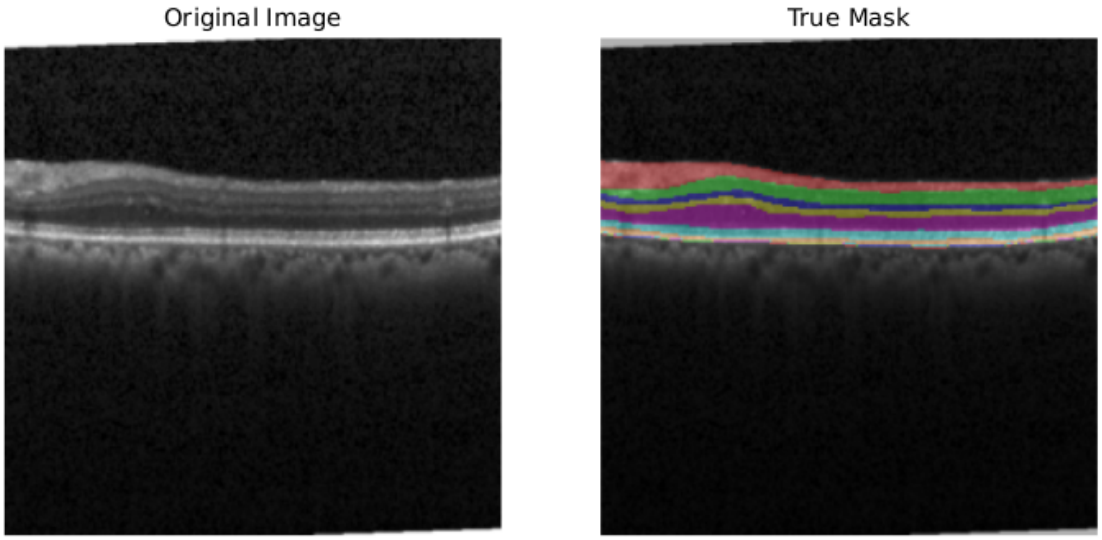

Fig. 2. Duke Dataset: The original image refers to the raw OCT B-scan images, while the true mask corresponds to the annotated segmentation maps.

1) **Duke**: The Duke dataset, visible in Figure 2, is a public dataset provided by Duke University, featuring 110 annotated OCT B-scans from patients with severe diabetic macular edema. The scans are annotated with eight retinal layer boundaries, aiding the training and testing of segmentation algorithms.
2) **UMN**: The local diabetic macular edema (DME) OCT dataset, as visible in Figure 3, known as the University of Minnesota (UMN) DME dataset, includes OCT scans from 29 DME subjects, each with 25 B-scans, and two expert manual segmentation. The UMN dataset is $6\times$ larger than the Duke dataset.

V. Implementation

Previously, we mentioned that U-Net [8] is explicitly designed for semantic segmentation of bio-medical images. We begin by implementing the network architecture for training in Python 3, guided by the architecture diagram below in Figure 4.

After the network architecture implementation, we implement the model's training module in a non-differential private manner. The training module accepts different optimization hyperparameters for the model. We focus on batch size, learning rate, number of epochs and weight decay for performance analysis. We extend our training module and add support for Opacus [5] and Book-Keeping [6] techniques. The claim of simplicity in making the non-differential private training differential private holds since the changes in lines of code are minimal in both cases. In the privacy analysis, we focus on noise multipliers as a common hyperparameter for Opacus and Book-Keeping algorithms.

VI. Experimental Setup

The experimental setup is designed to evaluate the performance of the training process under three different cases: training without differential privacy, training with differential privacy using Opacus, and training with differential privacy using Book-Keeping (BK). We use the same set of hyperparameters for a fair comparison of all cases. The batch size is set to 10, the learning rate is 5×10^{-4}, the number of epochs is 100, and the weight decay is 1×10^{-4}. These parameters optimize the model's performance while maintaining consistency across experiments. When training with differential privacy, we introduce standard differential privacy-specific hyperparameters. The noise multiplier varies between 0.3 and 1.0, and the maximum

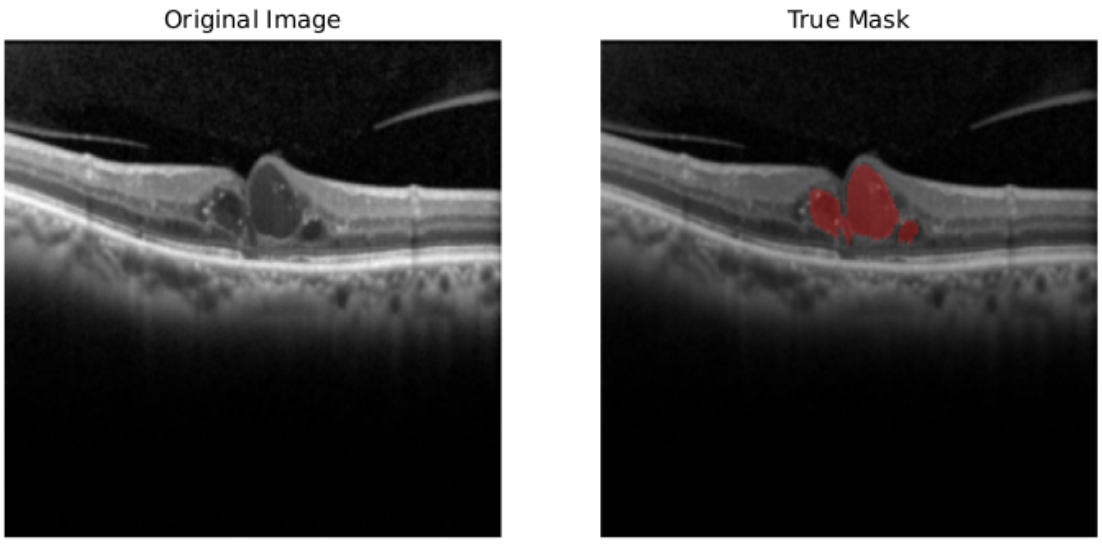

Fig. 3. UMN Dataset: The original image refers to the raw OCT B-scan images, while the true mask is focusing on identifying fluid regions in the scans.

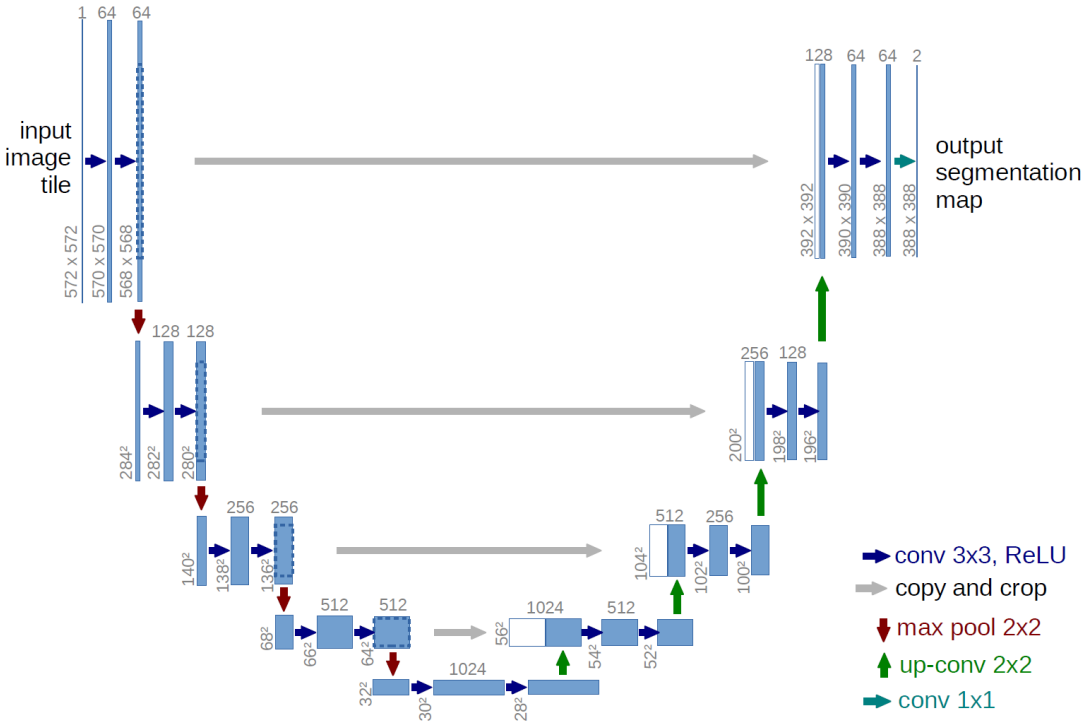

Fig. 4. U-net design (for example, 32x32 pixels in the lowest resolution). Each blue box represents a multi-channel feature map. The number of channels is displayed at the top of the box. The x-y-size is specified at the box's lower left edge. The white boxes indicate copied feature maps. The arrows denote the various operations. [8]

gradient or clipping threshold (R) is fixed at 1.0. These parameters control the level of privacy and gradient clipping applied during training. For training using Opacus, additional hyperparameters include a target delta value of 1×10^{-5} and a per-layer clipping style for fine-grained privacy control. On the other hand, we employ an automatic clipping function combined with a hybrid clipping mode for training using the Book-Keeping technique. This mode leverages layer-wise and global clipping strategies to optimize performance while preserving privacy. The data is split into three subsets: training (80%), validation (10%), and test (10%). We compare the non-differential private training against differential private training using Opacus and Book-Keeping algorithms.

VII. Evaluation

The evaluation process collects key metrics and hyperparameters during each training run to assess model performance. The recorded parameters include batch size, number of epochs, learning rate, noise multiplier, maximum gradient norm (clipping threshold), clipping mode, algorithm used, overall privacy budget spent, dataset utilized, model name, device used for training, iteration-wise training times, training losses, validation losses, validation dice scores, and total training time. We implement a visualization script that generates plots to analyze the collected data. We implement another script to load the trained model and generate segmentation masks for the dataset. This script is applied to the test set to assess the model's performance on unseen data. We combine metric tracking during training with post-training evaluation on the test set to analyze model performance and privacy trade-offs under different experimental conditions.

VIII. Results

We evaluate the results based on key metrics such as training time, loss, validation accuracy, and privacy budget consumption and analyze the effect on privacy and performance.

A. *Training Time Over Iterations*

The plots in Figures 5 and 6 provide insights into the computational efficiency of the training process under different noise multiplier settings for Opacus and Book-Keeping (bk).

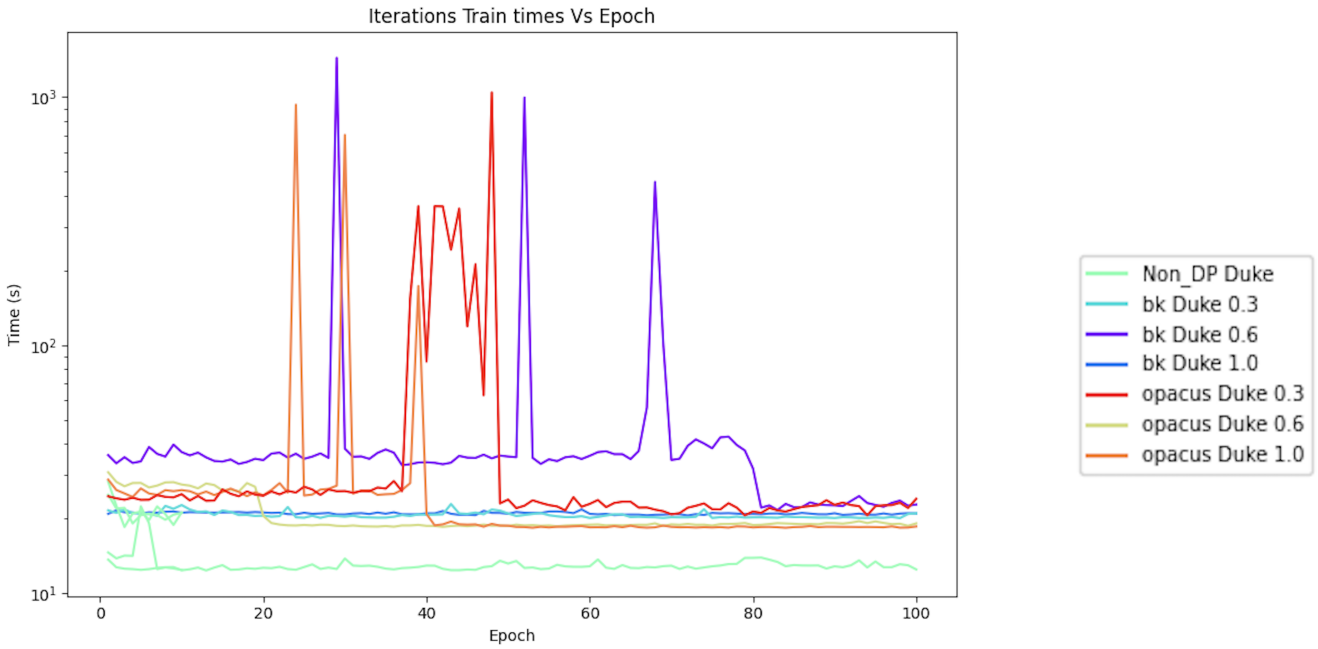

Fig. 5. Training time per iteration for the Duke dataset, with epochs on the x-axis and time (in seconds) on the y-axis. An epoch represents a complete pass through the training dataset by the learning algorithm.

For the Duke dataset with Book-keeping, mid-range noise multipliers resulted in longer training times than lower and higher noise multipliers. Lower noise multipliers for the Duke dataset with Opacus led to longer training times. For the UMN dataset with Book-keeping, mid-range noise multipliers resulted in shorter training times than lower and higher noise multipliers. For the UMN dataset with Opacus, mid-range noise multipliers also resulted in shorter training times than lower and higher noise multipliers.

The increased computational time for mid-range noise multipliers with Book-keeping is attributed to a smaller dataset size, where computational overhead becomes visible. Opacus shows increased training time with lower noise multipliers due to additional privacy-preserving computations. Both Book-keeping and Opacus demonstrate more efficient training times with mid-range noise multipliers. The larger dataset size allows for better distribution of computational overhead, leading to more efficient updates.

B. *Training Loss Over Iterations*

The plots in Figures 7 and 8 provide insights into the performance of the training process under different noise multiplier settings for Opacus and Book-Keeping (bk).

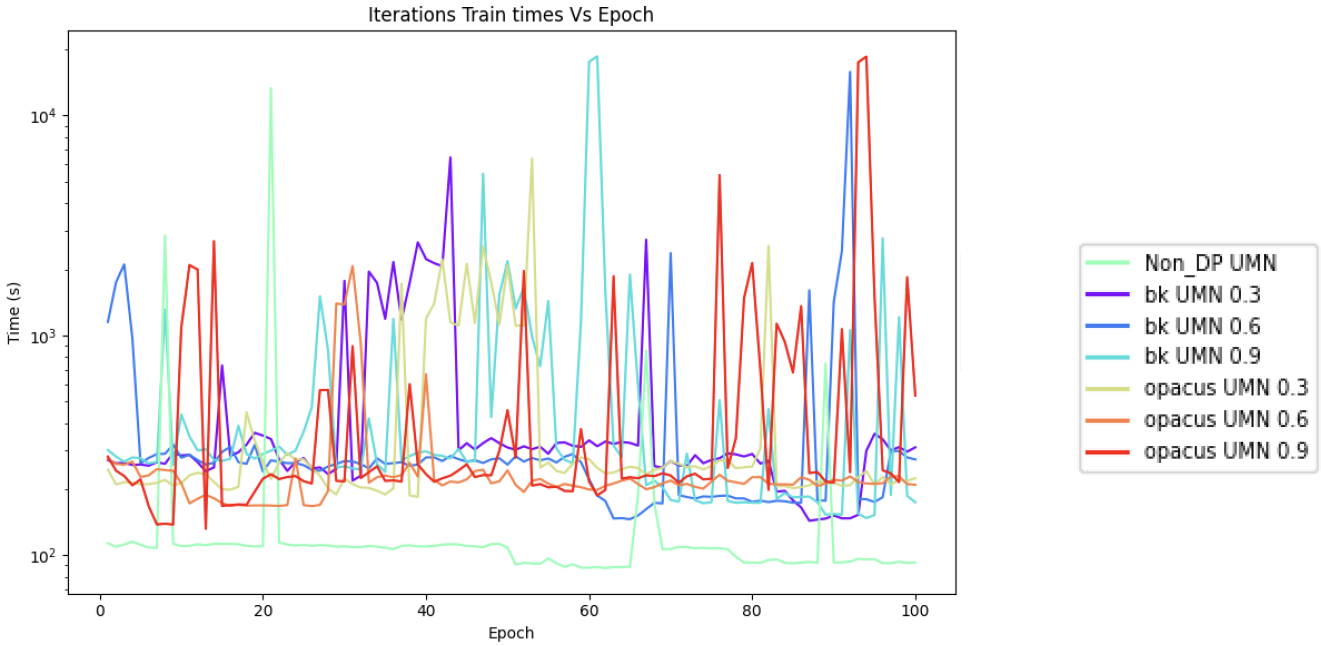

Fig. 6. Training time per iteration for the UMN dataset, with epochs on the x-axis and time (in seconds) on the y-axis. An epoch represents a complete pass through the training dataset by the learning algorithm.

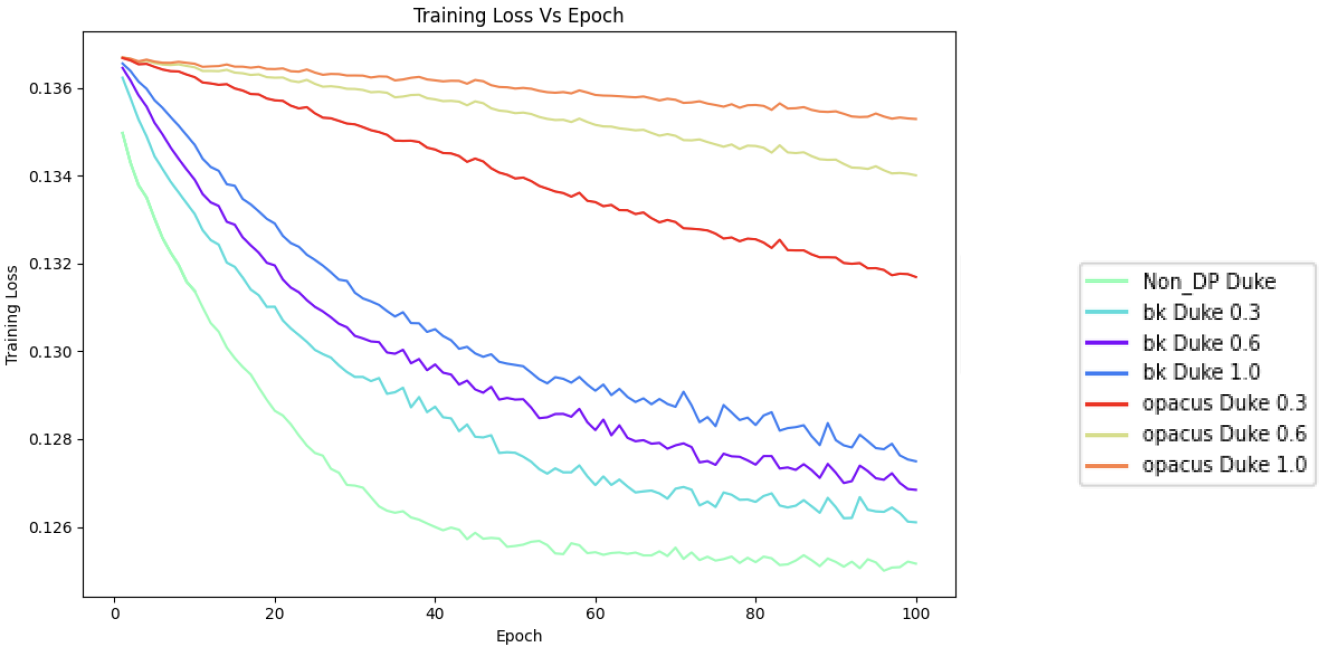

Fig. 7. Training loss per iteration for the Duke dataset, with epochs on the x-axis and training loss on the y-axis. The model's performance is evaluated every epoch using a loss function that measures the difference between predicted and true values.

Book-keeping and Opacus exhibit higher training losses for the Duke dataset with higher noise multipliers. For the UMN dataset with Book-keeping, training loss is comparable to non-DP training. For the UMN dataset with Opacus, training loss increases as the noise multiplier increases.

Higher noise multipliers introduce more randomness into gradient updates, leading to higher training losses. The smaller dataset amplifies this effect for both Book-keeping and Opacus. Book-keeping demonstrates better noise absorption, resulting in training losses similar to non-DP training. In contrast, Opacus shows increased losses with higher noise multipliers due to its sensitivity to privacy-preserving computations.

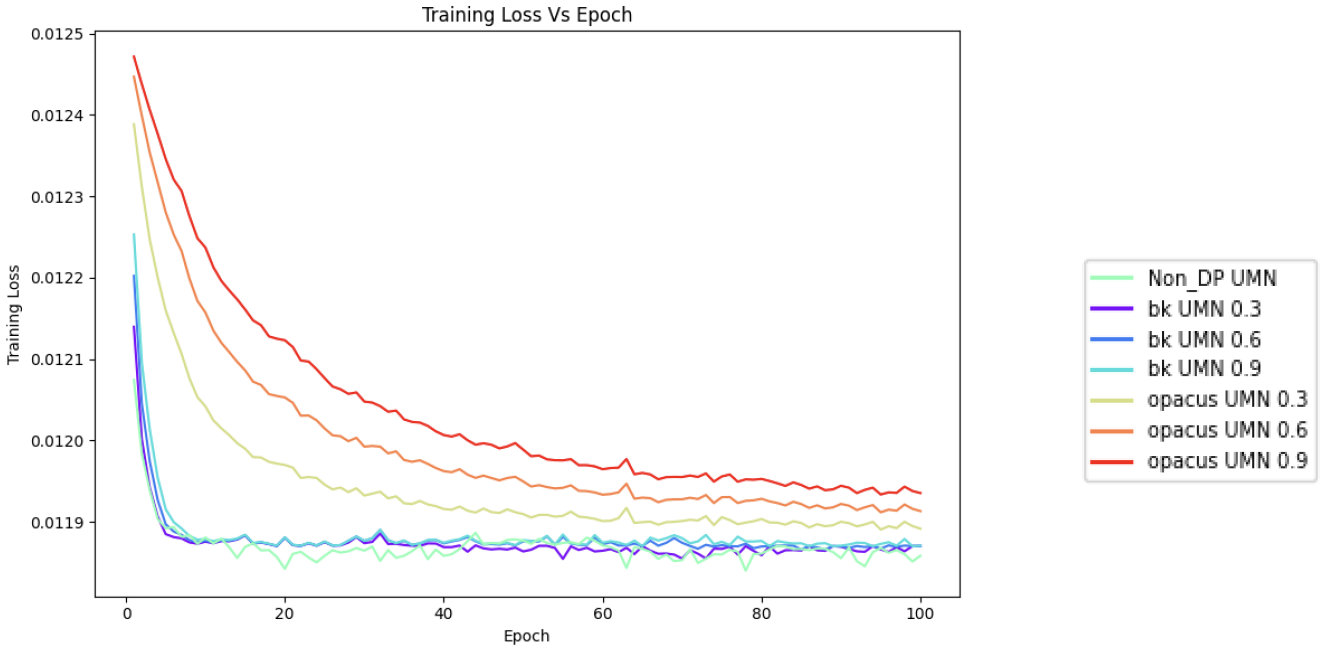

Fig. 8. Training loss per iteration for the UMN dataset, with epochs on the x-axis and training loss on the y-axis. The model's performance is evaluated every epoch using a loss function that measures the difference between predicted and true values.

C. *Validation Loss Over Iterations*

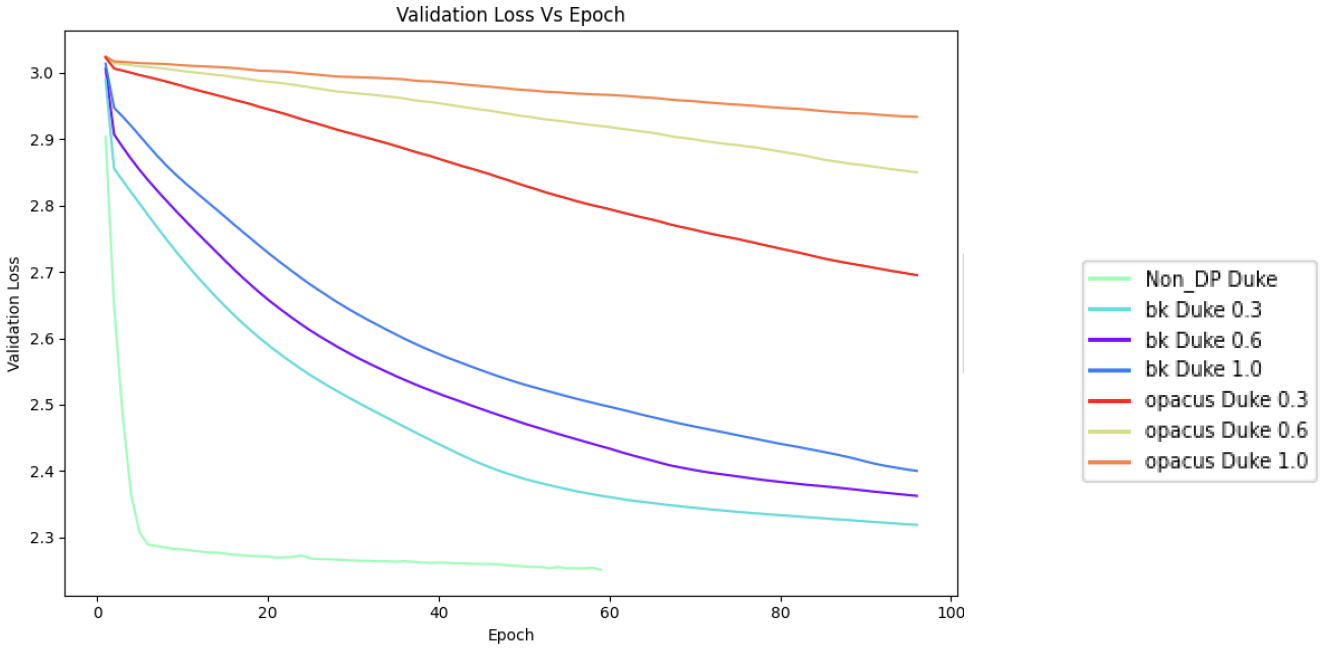

Fig. 9. Validation loss per iteration for the Duke dataset, with epochs on the x-axis and validation loss on the y-axis. The validation loss is measured every epoch to evaluate the model's performance on unseen validation data, indicating how well it generalizes to new inputs.

The plots in Figures 9 and 10 provide validation loss trends for both datasets under different noise multiplier settings for Opacus and Book-Keeping (bk).

For the Duke dataset, higher noise multipliers result in higher validation losses for both Book-keeping and Opacus. For the UMN dataset with Book-keeping, validation losses are comparable to those of non-DP training. For the UMN dataset with Opacus, validation losses increase as noise multipliers increase.

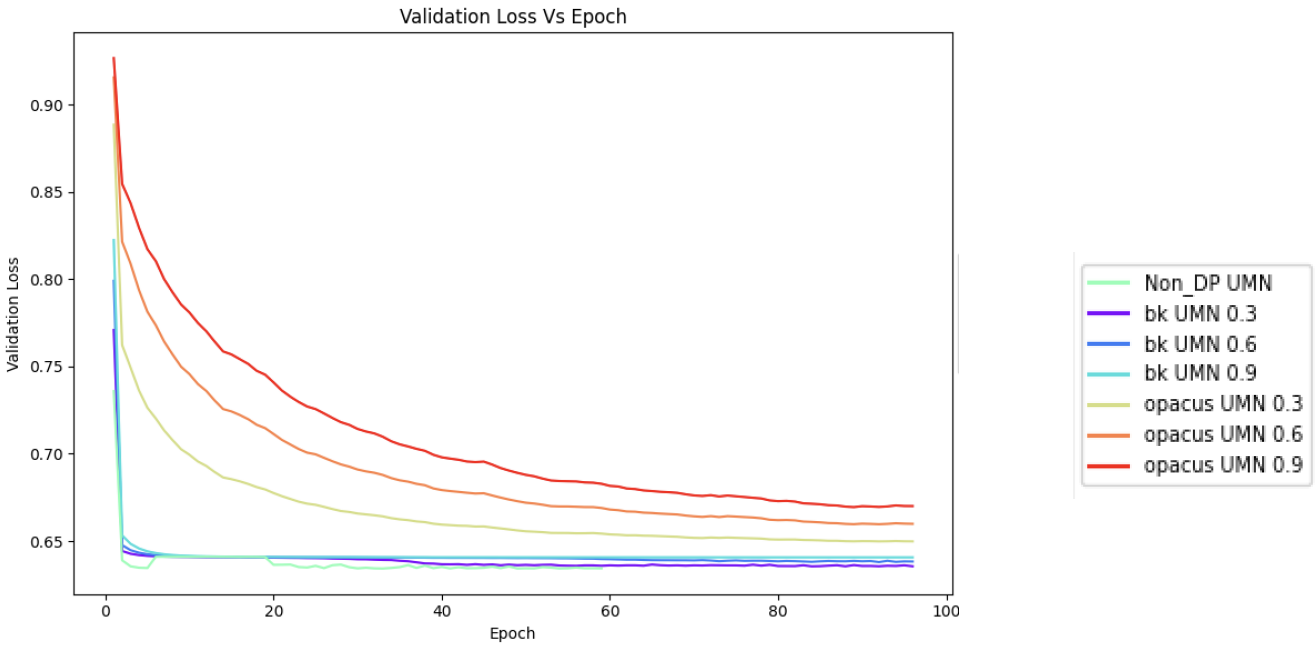

Fig. 10. Validation loss per iteration for the UMN dataset, with epochs on the x-axis and validation loss on the y-axis. The validation loss is measured every epoch to evaluate the model's performance on unseen validation data, indicating how well it generalizes to new inputs.

The smaller size of the Duke dataset amplifies the impact of higher noise multipliers on generalization performance, resulting in increased validation losses. Book-keeping demonstrates better generalization capabilities by maintaining validation losses similar to non-DP training. Opacus exhibits increased validation losses due to its sensitivity to high noise levels.

D. Privacy Spent Over Iterations

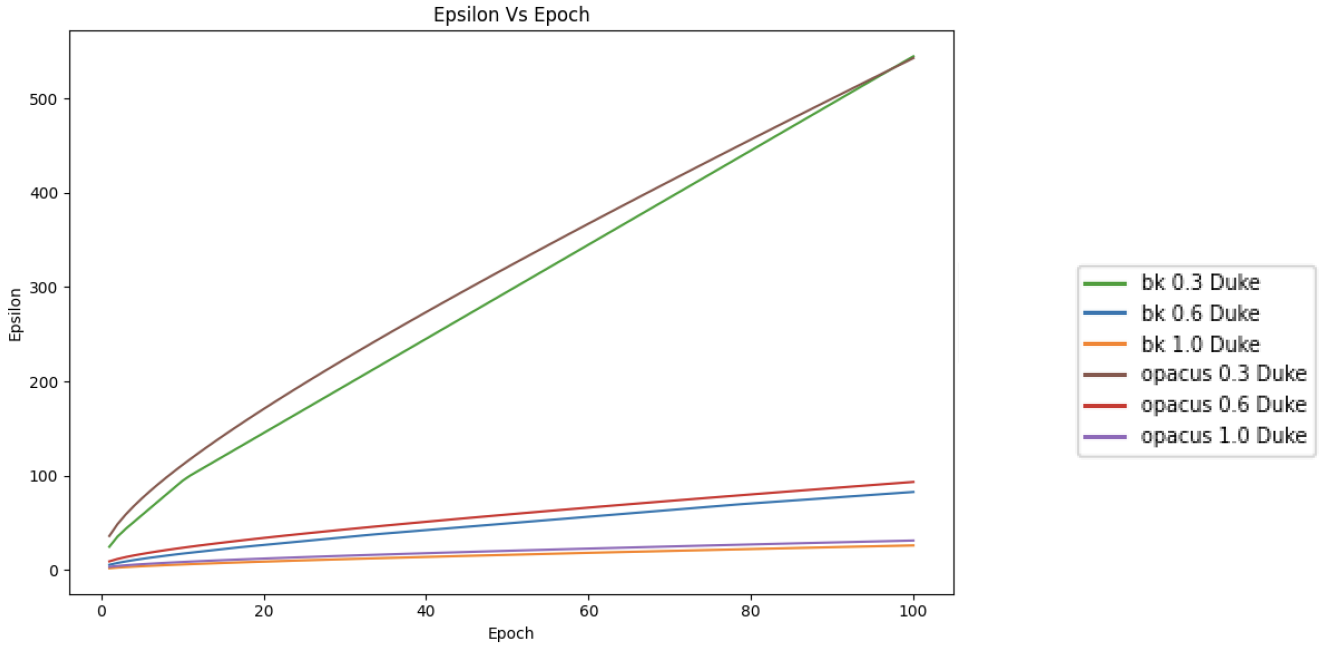

Fig. 11. Privacy spent per iteration for the Duke dataset, with epochs on the x-axis and ϵ (epsilon) on the y-axis. ϵ quantifies the privacy loss in differential privacy, with smaller values indicating stronger privacy guarantees.

Privacy budget consumption (epsilon) is analyzed across iterations for both datasets in Figures 11 and 12. Lower noise multipliers result in higher privacy budget consumption for the Duke dataset, with epsilon values reaching up to 500. Both

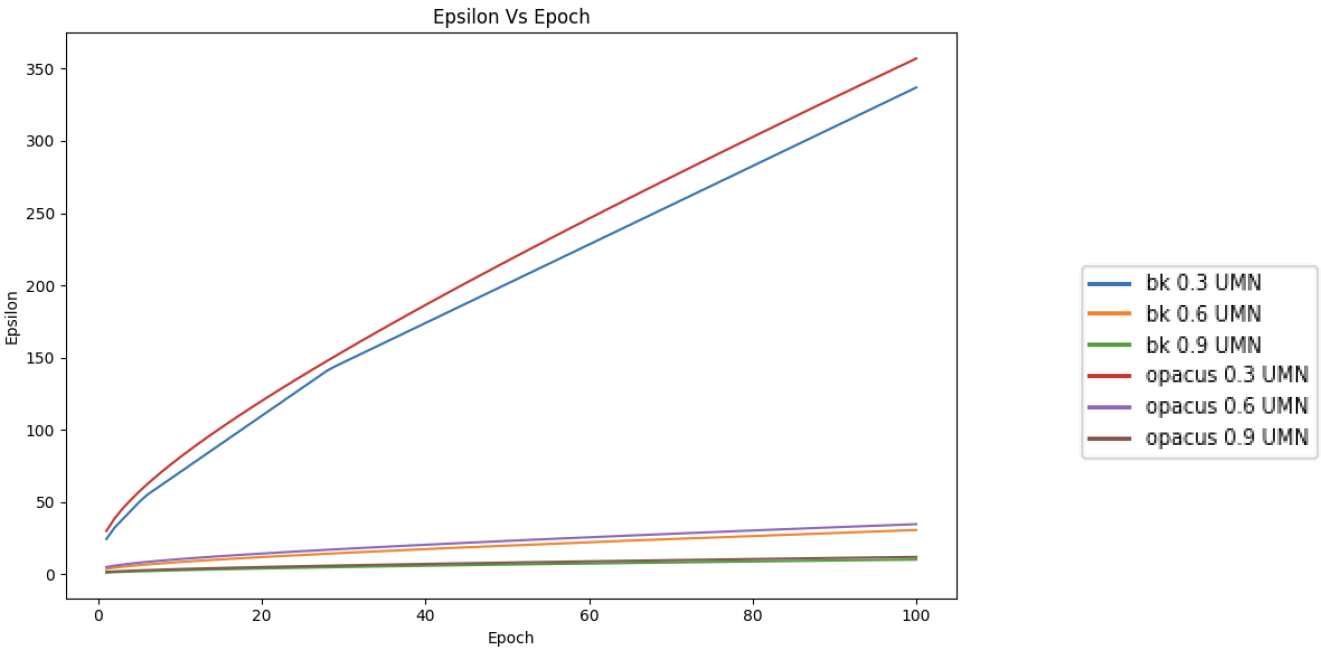

Fig. 12. Privacy spent per iteration for the UMN dataset, with epochs on the x-axis and ϵ (epsilon) on the y-axis. ϵ quantifies the privacy loss in differential privacy, with smaller values indicating stronger privacy guarantees.

Opacus and Book-keeping exhibit similar privacy budget consumption trends. For the UMN dataset, Lower noise multipliers also lead to higher privacy budget consumption, with epsilon values reaching 350. However, Opacus surpasses Book-keeping in terms of overall privacy budget consumption.

Lower noise multipliers inherently lead to higher privacy loss due to reduced obfuscation of gradients during training, resulting in greater consumption of the privacy budget. The similarity in privacy budget consumption between Opacus and Book-keeping can be attributed to the smaller dataset size, limiting noise multiplier differences' impact on overall privacy loss. The larger dataset size amplifies differences between mechanisms, with Opacus consuming more privacy budget than Book-keeping. Opacus is more sensitive to noise multiplier variations when applied to larger datasets.

E. Mean Validation Loss and Accuracy

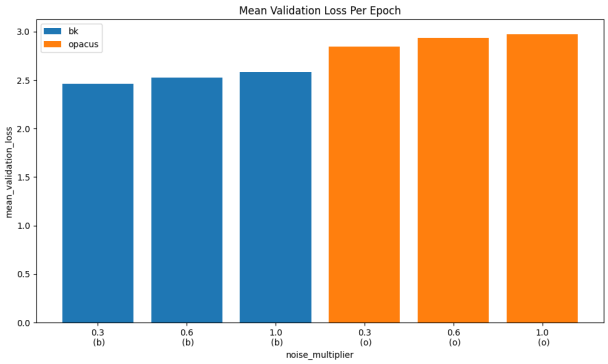

Fig. 13. Plot for mean validation loss for Duke dataset

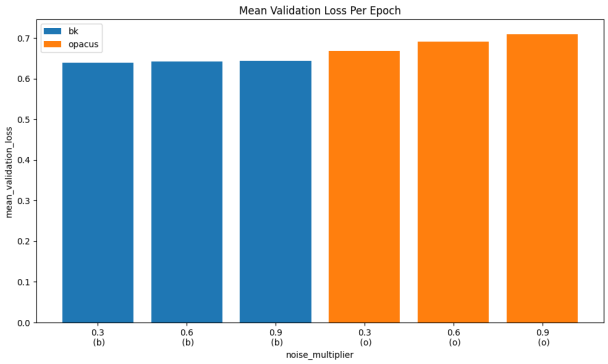

Fig. 14. Plot for mean validation loss for UMN dataset

The mean validation loss illustrated in Figures 13 and 14 is an inverse indicator of model accuracy on unseen data, with lower values reflecting better generalization and higher accuracy. Opacus exhibits a higher mean validation loss for Duke Dataset than Book-keeping. For the UMN Dataset, Opacus also shows a higher mean validation loss than Book-keeping; however, the range of loss is narrower compared to the Duke dataset.

For Duke Dataset, The higher mean validation loss observed with Opacus suggests that Book-keeping provides better generalization capabilities, resulting in potentially higher accuracy on unseen data. For UMN Dataset, While Opacus still demonstrates a higher mean validation loss than Book-keeping, the larger size of the UMN dataset helps stabilize performance, leading to a narrower range of losses compared to the Duke dataset.

IX. Conclusion

The selection of differential privacy mechanisms, such as Book-keeping or Opacus, depends on the dataset size and the desired trade-off between performance and privacy. Based on our analysis, Book-keeping aligns closely with the claims proposed in the paper [6] and is particularly well-suited for large datasets due to its efficiency and scalability.

When managing the privacy budget, Book-keeping is more efficient for smaller datasets during the initial stages of training. However, as the number of iterations increases, its efficiency declines. In contrast, Opacus starts with higher privacy budget consumption but tends to stabilize over time. For larger datasets such as UMN, Book-keeping demonstrates superior efficiency in managing the privacy budget compared to Opacus.

Book-keeping outperforms Opacus regarding training and validation loss for smaller datasets, such as Duke, demonstrating better optimization and generalization. On the other hand, larger datasets like UMN, Book-keeping and Opacus exhibit comparable performance in training and validation loss. However, Opacus incurs higher computational overhead per iteration than Book-keeping, highlighting its increased complexity when scaling to larger datasets.

X. Future Work

Future work includes but is not limited to implementing training with the K-fold split training method to observe the impact of data splits on differential privacy mechanisms. We could also implement automatic clipping [12] for Opacus and evaluate the impact on privacy and performance.

References

[1] A. Ziller, T. T. Mueller, S. Stieger, et al., "Reconciling privacy and accuracy in AI for medical imaging," *Nature Machine Intelligence*, vol. 6, pp. 764–774, 2024. [Online]. Available: https://doi.org/10.1038/s42256-024-00858-y

[2] 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*, Vienna, Austria, 2016, pp. 308–318.

[3] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016. [Online]. Available: https://arxiv.org/abs/1609.04747

[4] "What is Differential Privacy," IEEE Digital Privacy, [Online]. Available: https://digitalprivacy.ieee.org/publications/topics/what-is-differential-privacy. [Accessed: Nov. 28, 2024].

[5] A. Yousefpour, I. Shilov, A. Sablayrolles, D. Testuggine, K. Prasad, M. Malek, J. Nguyen, S. Ghosh, A. Bharadwaj, J. Zhao, G. Cormode, and I. Mironov, "Opacus: User-friendly differential privacy library in PyTorch," *arXiv preprint arXiv:2109.12298*, 2021. [Online]. Available: https://arxiv.org/abs/2109.12298

[6] Z. Bu, Y.-X. Wang, S. Zha, and G. Karypis, "Differentially private optimization on large models at small cost," *Proc. Mach. Learn. Res.*, vol. 202, pp. 3192–3218, Jul. 2023. [Online]. Available: https://proceedings.mlr.press/v202/bu23a.html

[7] M. Iwanicka, M. Sylwestrzak, A. Szkulmowska, and P. Targowski, "Methods and Techniques: Optical Coherence Tomography (OCT)," in *Book Title*, 2019, pp. 207–227. [Online]. Available: https://doi.org/10.1017/9789048550531.009

[8] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *arXiv preprint arXiv:1505.04597*, 2015. [Online]. Available: https://arxiv.org/abs/1505.04597

[9] H. Seo et al., "Machine learning techniques for biomedical image segmentation: An overview of technical aspects and introduction to state-of-art applications," Medical Physics, vol. 47, no. 5, pp. e148-e167, 2020, doi: 10.1002/mp.13649.

[10] K. Packhäuser, S. Gündel, N. Münster, et al., "Deep learning-based patient re-identification is able to exploit the biometric nature of medical chest X-ray data," *Scientific Reports*, vol. 12, no. 14851, 2022. [Online]. Available: https://doi.org/10.1038/s41598-022-19045-3.

[11] PyTorch, "PyTorch," 2023. [Online]. Available: https://pytorch.org. [Accessed: 05-Dec-2024].

[12] Z. Bu, Y.-X. Wang, S. Zha, and G. Karypis, "Automatic Clipping: Differentially Private Deep Learning Made Easier and Stronger," arXiv preprint arXiv:2206.07136, 2023. [Online]. Available: https://arxiv.org/abs/2206.07136

XI. Declaration of Originality

For the paper, I used the Grammarly tool to improve the language of texts that I had written myself. I have checked all the texts and take full responsibility for the results.