Privacy-Preserving Synthetic Image Data Generation and Classification

by

Fahim Faisal

A thesis submitted to
The Faculty of Graduate Studies of
The University of Manitoba
in partial fulfillment of the requirements
of the degree of

Master of Science

Department of Computer Science
The University of Manitoba
Winnipeg, Manitoba, Canada
May 24, 2023

Thesis advisors

Dr. Carson K. Leung, Dr. Yang Wang

Author

Fahim Faisal

Privacy-Preserving Synthetic Image Data Generation and Classification

Abstract

Computer vision, generative models (e.g., ChatGPT, etc.) and deep learning are now widely used across various sectors, from large corporations to end devices, simplifying people's lives and improving the reliability of medical findings. Sensitive image data and deep learning's high memorization capacity pose privacy risks, particularly for medical images containing sensitive private information. De-anonymization does not work due to the re-identification risk and reduced utility. So, we developed a differentially private approach with selective noise addition to generate high-dimensional synthetic medical image data with guaranteed differential privacy. In addition to ensuring data privacy, protecting the classification model's privacy is crucial due to its vulnerability to "membership inference attacks". State-of-the-art (e.g., differential privacy, etc.) defenses compromised task accuracy to preserve privacy, and some methods reuse private data or require more public data, which is impractical in some domains. To address privacy concerns while maintaining utility, we propose a collaborative distillation approach that transfers knowledge using minimal synthetic data, resulting in a compact private classifier model.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to begin by expressing my appreciation toward my M.Sc. thesis advisors, Dr. Carson K. Leung (University of Manitoba) and Dr. Yang Wang (University of Manitoba & Concordia University), for their unwavering assistance, guidance, and encouragement. Their relentless efforts, supervision, and financial backing have been instrumental in making this thesis work a reality. I would also like to extend my heartfelt thanks to Dr. Noman Mohammed from Data and Security Lab for collaborating with me and offering his domain expertise in the area of private learning. Lastly, having Dr. Noman Mohammed (University of Manitoba) and Dr. Yiming Qian (formerly at University of Manitoba, and currently at Amazon) on my advisory committee and examination committee is an absolute privilege. Thanks Dr. Lorenzo Livi for chairing my M.Sc. defence.

I want to express my gratitude towards Dr. Carson K. Leung and Dr. Yang Wang for providing me with financial support, computational resources (Crane and Digital Research Alliance of Canada), and consistent encouragement throughout my academic journey, which proved to be essential for completing my thesis. The weekly meetings and frequent messaging opportunities in the dedicated Slack channel were particularly helpful in enabling me to make steady progress in my research.

My mentors' in-depth knowledge of the thesis topics and keen attention to detail has helped me refine my research problem and methodology and produce a final thesis. Their feedback and constructive criticism have been instrumental in shaping my arguments, experiments, and analysis. It is worth noting that they had a positive outlook toward experimental failures during the initial study period. Instead of considering them failures, they encouraged me by highlighting that I had ventured

into a new direction. This positive approach inspired me to remain optimistic and persistent. Additionally, they were always readily available whenever I sought his assistance, and responded promptly.

I also want to thank Faculty of Graduate Studies, Department of Computer Science, the department head, and the staff for providing me with the resources, central fellowship-based funding, and facilities necessary for my research. My gratitude also extends to my colleagues, lab mates, and friends who supported me during the writing process. In conclusion, I want to express my appreciation to my family and friends for their assistance; they have consistently been there for me when I required their help.

Fahim Faisal

B.Sc., BRAC University, Bangladesh, 2019

The University of Manitoba

May 24, 2023

This thesis is dedicated to my family and my beloved wife.
Their constant support and well wishes made this work possible.

Publication

Some ideas, materials, figures, and tables in my thesis have appeared in the following publications and submitted manuscripts.

- **Faisal, F.**, Mohammed, N., Leung, C.K., Wang, Y.: Generating privacy-preserving synthetic medical data. 2022 IEEE Ninth International Conference on Data Science and Advanced Analytics (DSAA). pp. 1003–1012 (2022).
- **Faisal, F.**, Leung, C.K., Mohammed, N., Wang, Y.: Privacy-preserving learning via data and knowledge distillation. 2023. (Currently under review for an international conference)

Chapter 1

Introduction

The popularity of deep-learning models' computation power encourages medical professionals to solve many disease detection problems. Notably, some medical sectors require much time and substantial human resources to do their job. Because of the high proficiency and computation power, in modern days, machine learning is taking part in this sector to solve disease analysis problems in much reduced time and more efficiently. One problem is that medical datasets are difficult to use in modeling as they are associated with personal information, and preserving such health data's privacy is crucial [Ali+20; BAZ20; YAC20]. We need a method that produces high-quality private synthetic data that do not reveal the identity of any user and we also need a model that is private and does not leak any trained member's information.

Our research has resulted in the development of a differential privacy-based data generator that can produce private data for training purposes while also providing theoretical guarantees. Additionally, we extend this work by developing a compact classifier model with a built-in defense mechanism that ensures membership privacy.

Using this private model to train synthetic data ensures empirical privacy, resulting in a two-step privacy mechanism that protects users' privacy from both a data and model perspective. By combining both empirical and theoretical bases, this dual protective training approach is anticipated to prioritize users' primary concern of privacy and thereby incentivize them to share sensitive data for training purposes.

1.1 Private synthetic data generation

However, in the first part of the thesis, we focus on privacy concerns for using sensitive datasets—e.g., magnetic resonance imaging (MRI), computerized tomography (CT) scans, X-rays, or breast cancer datasets and come up with a differential privacy based medical data generator. Because medical data can also have a marker indicating who the actual person could be. For example, X-ray images in digital imaging and communications in medicine (DICOM) files are structured so that the cover sheets baked into DICOM files include patients' sensitive information for identification purposes. They consist of the patient's date of birth, sensitive diagnosis information, the name of clinical institutions, etc. Sometimes, hospital databases use patients' social security numbers (SSNs) to identify those files in the system. Such sensitive information leakage could link them to another sensitive dataset using those identifiers. This kind of privacy breach will be harmful to the patients. Some follow de-anonymization or content removal approaches (e.g., skull stripping for MRI data), which remove such identifying attributes from the image header in X-ray images. Yet, as the protocol for X-ray images differs worldwide, standardizing such an approach is not feasible, and it losses utility with the risk of re-identification.

Machine learning attack models can even identify the person using that re-identified image. On the other hand, deep-learning models are data-hungry; if we provide enough data, the model can learn efficiently. In these circumstances, my goal is to build a generative model where generated data will be efficiently used for the diagnosis of radiology and X-ray images using synthetic data without having any direct dataset from the medical institutes. The authority will only provide a deep generative model trained from their pneumonia detection datasets. My generator is differentially private, and the used artificial data does not belong to real patients. In this framework, I add noise to the generator's gradient only but not to the discriminator so that generated data is differentially private. Such private modeling will ensure that data privacy is preserved. A third party cannot access the model's features or weights for exploiting learned weights to reconstruct source data. Reverse engineering utilizing that privacy-preserving model will not be feasible anymore because the noise will be injected into the encoded weights, so the model itself is private. Such an approach will encourage medical institutions to share more data, and such synthetic data are less expensive to collect and can be larger in quantity than real data. Previous differential private generative adversarial network (GAN)'s [Bea+19; Fri+19; TKP19; ZJW18; Li+20] recent performance encouraged me to use GANs for differentially private synthetic data generation using differential privacy-stochastic gradient descent. However, most of those methods fail to produce high-dimensional data, which is precisely what I aim to do in this work.

1.2 Privacy-preserving model training

In the second part of my work, I also want to ensure protection in the target classification model so that it has some built-in Membership Inference Attack defense and can come up with comparable accuracy privacy trade-off. Previous approaches require public reference data and repeated access to private data, which is risky, and there is a slight utility degradation. Thus, our comprehensive model-based data distillation approach can sanitize the knowledge transfer process to achieve a better trade-off than all the existing state-of-the-art methods. Also, it can restrict access to private data and does not require public data or data with a specific property. So, such a private classification model can be equipped with the first part's generated private data to ensure protection against Membership inference also.

In recent times, the use of machine learning and computer vision has become increasingly popular in medical diagnostic and image analysis tasks. However, models trained using Deep Learning techniques are susceptible to *membership inference attacks (MIA)* due to their high capacity to memorize training data. MIA is the simplest attack that enables an adversary to gain information about an individual by knowing that their data was used to train a predictive model. This attack can lead to more advanced attacks, such as attribute inference and feature extraction attacks. The consequences of MIA can be severe, particularly when the data involved is sensitive, as it may result in linking attacks or the de-anonymization of private data. Additionally, even trusted service APIs such as Google/Amazon machine learning platforms are not immune to MIA attacks, which pose a risk to users of such services. black box MIA [Sho+17] exploits prediction confidence and true label, while white

box [NSH19] MIA exploits gradient information leakage and other internal features to predict membership.

Defenses mainly focus on mitigating whitebox and blackbox MIA based on provable and empirical approaches. Empirical works can provide optimal performance as they are optimized for utility; On the contrary, provable methods can give a theoretical guarantee with a significant toll on accuracy. Even though theoretical methods are not designed for a particular dataset and are not optimized for utility, previous provable approaches like DP-SGD or PATE [Aba+16; Pap+16] can assure reasonable trade-off. But, due to colossal noise addition, they cannot provide acceptable utility. On the other hand, white box defenses based on empirical evidence like adversarial regularization and regular regularization ensure optimal accuracy with slightly reduced privacy. As they are mainly optimized for utility, increasing privacy requires higher regularization, which takes a toll on utility compared to privacy failing to provide an acceptable trade-off. Previous empirical knowledge transfer-based solutions showed some hope, but an approach like DMP [SH21] requires reference data with specific properties like low entropy besides private data. Tuning or managing public data can be problematic for some privacy-sensitive tasks, and synthetic data-based DMP provides very low accuracy. On the other hand, models like KCD [Cho+22], and SELENA [Tan+22] require repetitive access to private data while knowledge transfer, which leaks a lot of private information. This requirement of additional public data and repeated access to confidential data for model training, with utility degradation, make these approaches infeasible. Again, they use a larger model to improve accuracy. Large capacity encourages them to memorize data, degrading privacy due to

overfitting. So, appropriate model capacity can help to avoid memorization and privacy leakage. Thus, we need an efficient model that will overcome the limitations of overfitting and the need for public data comparable to non-private ones. This work will be the first to analyze a framework that can encourage privacy, efficiency, and utility at the same time. We develop a comprehensive model-based data distillation, Privacy-Preserving Learning via Data, and Knowledge Distillation (PLDK) to sanitize the training process to achieve a better empirical trade-off than all the existing state-of-the-art methods with lower MIA attack risk close to random guess.

Our approach in the second part of this thesis involves two main steps. First, we employ data distillation to obtain a smaller synthetic dataset that captures the common trends of the private data. This synthetic dataset is carefully tailored for the target student model, ensuring it has enough discriminative properties to produce accurate predictions. We generate a smaller amount of synthetic distilled data with which a model with the same architecture as the target student model can be trained to have similar parameters to when trained with total private data. Secondly, we train an unprotected teacher model with a large capacity on private data directly, and it is used to generate soft labels for the distilled data. The target student model is then trained using knowledge distillation on the labeled distilled data. The lightweight student model can achieve comparable accuracy to the teacher model trained on extensive private data by training on this labeled distilled data. It is also robust to inference attacks since it is not directly exposed to private data during final model training.

The procedure is tailored for various datasets and can be employed directly in

a white box fashion without adding noise/modification. It is optimized for utility, whereas DP-based approaches are not directly optimized for utility and are not tailored for specific datasets; instead, they focus only on privacy where utility is ignored. Our empirical policy has a built-in defense mechanism against inference attacks with a satisfactory utility trade-off compared to existing methods where the model will be tailored for specific datasets. Direct data condensation [Caz+22; DZL22] does not focus on model privacy, and our data and model distillation synergy ensures tighter privacy bound of the model. Again, it helps to improve fairness [Far+20; YKF20] as class balance ensures fair training and privacy. Our empirical evidence showed that our model is almost free from the risk of MIA providing privacy leak of 50-53.5%, close to a random guess of 50%, nearly as precise or better than unprotected models ensuring 69.3%, 46.6%, and 97.3% test accuracy on Canadian Institute for Advanced Research (CIFAR)'s CIFAR-10, CIFAR-100, and Modified National Institute of Standards and Technology (MNIST) datasets, respectively.

1.3 Thesis statement

The objective of this thesis is to create a confidential data generator capable of producing high-quality private images for deep learning purposes. Additionally, a confidential and concise model was also developed in the second phase to train this data, ensuring a double-layered privacy guarantee for users.

1.3.1 Synthetic data generation

Medical image data contains private sensitive attributes. Therefore, training deep models on such data can leak private attributes or lead to a linking attack where such data can be re-identified. Anonymization is not a feasible solution, as standardizing such images globally is challenging. Thus, we need to come up with a generative approach that can generate medical data, (x, y), which is differentially private. We expect that model can ensure the same outcome irrespective of the presence of training data which is a challenge in high stake medical domain. We need a setup so that local hospitals can provide their data privately. We also want to develop reliable high-fidelity data to ensure proper decision-making in high-stakes diagnosis tasks. We will follow Chapter 3 to solve this problem.

1.3.2 Membership inference attack defense

This a popular attack where the adversary trains an inference model to identify whether a specific data record is trained using a specific model or not. So, this information breaches the privacy of the training member. Suppose we train a model in a supervised setting where (x, y) is the corresponding image and true label of the data, and we want to predict $\hat{y}$, then the adversary can access y and $\hat{y}$ both. An adversary may also have access to the prediction confidence and they can exploit such a relationship to identify whether a given data belongs to the training set or not. For example, if they can determine that data corresponds to a cancer hospital or rape victim database, it will implicitly expose some sensitive attributes about the identified individual. Training leads to unintended leakage (if data is private). So

following cahpter 4 solutions, we want to come up with a model where given the true label and calculated prediction confidence, the adversary cannot successfully identify whether data belongs to the corresponding training set or not. Why is it essential? Knowing that a particular patient's clinical record was used to train a model associated with a disease (e.g., predict cancer or specific medicine dose, etc.) can reveal that the patient has this disease. This a popular attack where the adversary trains an inference model to identify whether a specific data record is trained using a specific model or not. So, this information breaches the privacy of the training member.

1.4 Contributions

Our key contributions are summarized below:

- We designed a differentially private approach to generate both reliable and private high resolutions radiography images with selective noise addition (via W-GAN-based architecture) for the first time, ensuring close to real data accuracy, which is satisfactory.
  - Our approach can preserve higher utility by applying selective gradient sanitization. We apply sanitization only to the generator and not to the discriminator like previous approaches to ensure more stable training with reliable data.
  - We ensure implicit noise clipping and sensitivity bound of training using Wasserstein loss property of W-GAN [Gul+17; ACB17] that guarantee the

gradient is within a limit of 1 (due to 1-Lipschitz condition). It eliminates the need to search for a perfect clipping value that is sensitive and may cause bias.

  - We utilize a simple notion of privacy, ensuring that the deeper architecture can be trained with a feasible privacy budget. So, such notion will allow researchers to exploit deeper models for private data generation.

  - Our novel synthetic and private medical data generation method works both in the centralized and distributed setting under untrusted server assumptions. It ensures that we can also use such an approach if we do not trust a centralized server to store the client's private data and the client only receives the noisy gradient, so the dishonest client cannot access other clients' data via model weights.

- We expand our image classifier in addition to the differential privacy-based data generation method. We developed a classification model that is resistant to membership inference, nearly as precise as non-private models, and can be employed directly in a white box manner, where the procedure is customized for various datasets. For such optimization, we exploited a knowledge distillation-based approach where the resultant student model is trained on synthetic distilled data, and the model will not be exposed to private data.

  - This extended distillation-based approach ensures target model compactness and membership privacy together for the first time. The final model does not rely on any public data and effectively restricts repeated access

to private data. So it is more feasible in real-world settings.

- Distilled data facilitate class balance and capture general trends during knowledge transfer, promoting fair training and encouraging generalization. Furthermore, the generated data is tailored to suit the student model.
- Resultant student model is efficient as it requires less than half the model and data size to train and is robust to MIA risk. It provides superior utility improvements of 8%, 34%, and 6% in the CIFAR-10, CIFAR-100, and MNIST datasets with 50-53.5% privacy leak, similar to non-private models.

1.5 Thesis organization

I organize the remainder of this thesis as follows. First, in Chapter 2 we discussed some of the background definitions and in 2.2, we give an overview of the related works. More specifically, this chapter discusses different key components of our algorithm and various privacy-based algorithms. Chapter 3 outlines my Private synthetic image data generation approach. It shows how differential privacy can be combined with GANs to generate high-quality reliable image data. Chapter 4 proposes a Collaborative distillation approach for the image classification task. Unlike the previous supervised approaches, this approach overcame more public data requirements or reference data tuning steps. Finally, in Chapter 5, I conclude this thesis.

Chapter 2

Background and Related Works

2.1 Background

Let us review a few definitions: Generative adversarial networks (GANs), differential privacy (DP), Renyi DP, Gaussian noise, Membership Inference Attack in machine learning, data and knowledge distillation.

2.1.1 GAN

Generative adversarial networks (GANs) [Bea+19; Fri+19; TKP19; ZJW18; Li+20] are the approach to formulate generative task using deep-learning models. There will be an encoder-based generator, which will perform the generative tasks. The generative model will learn the image features from training data and generate realistic-looking synthesized data from random noise. There will be a discriminative model that will try to determine whether the data is fake or real. In this way, the criticism for the generated data will be back-propagated and used to update the

model. In this two-player game of generator and discriminator, the generator will improve over time to fool the discriminator and the discriminator will become more expert in classifying fake or real data and it will be rewarded or penalized based on its performance. This adversarial game like Eq. (2.1) will help us to learn a good mapping of the real data. It tries to minimize the loss of the generator G so that it generates real like image and at the same time tries to maximize the discriminator D's loss so that it cannot distinguish between real and fake data. In the beginning of the game, generator G is not that good, and it gradually improves over time while the discriminator D's parallel classification task's improvement forcefully lead to high-quality image generation incrementally:

$$\min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[1 - \log D(G(z))] \tag{2.1}$$

2.1.2 Differential privacy

For all data sets P and P', if they differ on at most one training example, any randomized algorithm K (for a set S of outcome where any $S \subseteq$ Range(K)) gives **ε-differential privacy (DP)** [DR14]. In practice, we add δ term as a failure probability to Eq. (2.2), which ensures (ε, δ)-privacy:

$$Pr[K(P) \in S] \leq e^{\varepsilon} \times Pr[K(P') \in S] + \delta \tag{2.2}$$

Here, DP algorithm considers epsilon ε, which indicates the upper bound of privacy loss. Particularly epsilon ε is the metric for privacy loss due to change in the data by one record. Lower epsilon value indicates better privacy budget but limited utility. We have to choose ε value wisely to maintain the utility-privacy trade-off. δ

is used to relax the notion. δ is the estimated probability of breaching the constraints of differential privacy [DR14]. Here, we used differential Privacy in the context of Machine Learning problem and K is the generative model. Machine learning models are data hungry, the more the data they use for training, the more accurately they perform. In the same time in spite the availability of data, it is also important to ensure privacy of the system against the leakage of sensitive information. It ensures that model's predictive behaviour does not differ when the model has to predict training data or test data.

2.1.3 Renyi differential privacy

Differential privacy has a tremendous contribution in current machine learning advancement preserving privacy for data usage. However, it also brings issues for system maintenance cost. Machine learning model training is an iterative process and it adds privacy cost sequentially. As a result privacy budget restriction is becoming the major issue for developing machine learning model. **Renyi Differential Privacy (RDP)** [MTZ19] solves this issue by bringing more relaxation in DP algorithm. It increases the accuracy of the algorithm and it also reduces the computation cost for calculating privacy loss:

$$D_\alpha(P||P') = \frac{1}{\alpha - 1} \log \left(E_{p'(x)} \left(\frac{P(x)}{P'(x)} \right)^{\alpha - 1} \right) \leq \varepsilon \tag{2.3}$$

This equation calculates Renyi divergence of order α of a distribution P from the distribution P'. Instead of using log likelihood to measure privacy loss, this method equips Renyi divergence to measure privacy loss. It will be described in more details in Section 2.1.3.

To make our generator deferentially private, we have to ensure that each example may not have any significant impact on the model's encoded weight. To limit the impact of each example on the back propagated gradient we need to add some noise to the gradient. If D and D' are two adjacent dataset, then we need to add some noise to the output of a mechanism M. If $f(D)$ is the query function, then it will add $\mathcal{N}$ noise which is parameterized by σ, C. The noise is added to modify the distribution in 0 with standard deviation σ following Eq. (2.4). In our case, we have to run the training for multiple iterations and **Gaussian noise** can be a good choice due to its additive property which will be efficient in our method:

$$M(D) \simeq f(D) + \mathcal{N}(0,\, \sigma^2 C^2 I) \tag{2.4}$$

2.1.4 Membership inference attack in machine learning

Suppose we train a model F in a supervised setting where (x, y) is the corresponding image and true label of the data, and we want to predict $\hat{y} = F(x)$. If an adversary model g can access (x, y) and prediction vector $F(x)$ with prediction confidence, then it can exploit such a set of features to infer whether a given labeled image (x, y) belongs to the F model's training set D. The accuracy of model g will be referred to as 'privacy accuracy'. In a black box setting, an adversary might train a binary attack model g, assuming they have access to some of the train data of a model F, with the prediction vector $F(x)$. In a white-box manner, an adversary may exploit different model features like loss of model F on (x, y) and exploit gradient information $\delta_F(x, y)$. If D^A and $D^{A'}$ are members and nonmembers of the portion of

the disjoint data accessible to the adversary, attack model g optimizes θ_g to maximize the following gain I in Eq. (2.5):

$$I^{\theta_g}_{D^A, D^{A'}}(g) = \sum_{(x,y)\in D^A} \frac{\log(g(x, y, F(x))}{|D^A|} + \sum_{(x,y)\in D^{A'}} \frac{\log(1 - g(x, y, F(x)))}{|D^{A'}|} \tag{2.5}$$

2.1.5 Dataset distillation

Dataset distillation [Wan+18] tries to keep the model fixed and to develop a smaller dataset that can preserve similar test performance on a reserved test set compared to the dataset trained on the original dataset. Like Eq. (2.6), this is a process that helps to squeeze the larger dataset D to a smaller sized dataset S, and the ability to retain the same performance $\mathcal{L}(F_{\theta^F_S}(;))$ on distilled data S as real data D's training performance $\mathcal{L}(F_{\theta^F_D}(;))$. Here, $\mathcal{L}$ is regular loss (e.g., cross entropy):

$$E_{x\sim P_D}[\mathcal{L}(F_{\theta^F_D}(x), y)] \simeq E_{s\sim P_S}[\mathcal{L}(F_{\theta^F_S}(s), y)] \tag{2.6}$$

Thus, it maintains a fixed model F parameterized by θ and scales down the amount of data to improve training effectiveness by reducing training complexity; Distilled data S can better capture the feature dynamics of actual data D.

2.1.6 Knowledge distillation

Hinton et al. [HVD15] improved the concept of Buciluǎ et al.'s model compression [BCN06] process to transfer the knowledge of an ensemble of cumbersome teacher models into a single compact student model. While teaching the student network, soft labels generated by teacher networks are used for supervision. This ensures the student network can distinguish between different class values, which helps preserve

similarity information between different classes. They used temperature τ to smooth the output logits' a_F, a_T value distribution to provide higher generalization capability. Some approaches also use intermediate values [Che+20a] than probability. Such generalization helps promotes regularization, which can help mitigate membership privacy information leaked via extremely confident logit values. If y is the ground truth label and $F(x)$, $T(x)$ are the softened label of student and teacher network, then KL divergence-based loss $\mathcal{L}_{KL}$ in Eq. (2.7) is used to compute knowledge distillation loss $\mathcal{L}_{KD}$:

$$\mathcal{L}_{KD} = \tau^2 \mathcal{L}_{KL}(F(x), T(x)); F(x) = \text{softmax}\left(\frac{a_F}{\tau}\right); T(x) = \text{softmax}\left(\frac{a_T}{\tau}\right) \quad (2.7)$$

2.2 Related works

2.2.1 Privacy preserving learning

Deep learning is gaining popularity in predictive tasks. But such models are data-hungry, and they use different types of data scraping to collect data from all possible sources. Data have also been collected from various hospitals. These models are fundamental in the medical sector because they can make the diagnostic more reliable, but they need a large amount of data to perform well. However, using such sensitive data from hospitals and health databases can easily cause alarming privacy breaches. Still, previous works [AC19] proved that it is possible to enforce privacy in deep neural networks with a limited privacy budget. They introduced a differential private variation of common stochastic gradient descent with the moment accountant technique [Aba+16], which helped to keep track of privacy using each of the moments other than mean, and variance and picks the tightest bounds. They clipped the gradient and added noise, limiting the information learned from any given an example. Clipping bound C is a hyperparameter that needs to be tuned, which is a complex process that can cause bias.

Pepernot et al. [Pap+17] introduced a teacher and student model concept in their PATE mechanism, which added noise to the outcome rather than during the training process, and it trained an ensemble of models based on multiple disjoint datasets. So, the privacy budget increased with iteration, and the model itself is not private. But, to make the model itself with encoded weight differentially private, To overcome the drawback of PATE, they proposed a new G-PATE mechanism [Pap+18] where they

used Gaussian distribution instead of Laplacian distribution using Renyi differential privacy. The student played the role of private discriminator so that the student could learn how to extract the feature of unlabeled public data through the adversarial battle with the pre-trained generator. Still, the gradient needs to be subdivided into bins manually to cope with the framework in such a method. Due to the higher dimensionality of gradients, the noise added to the gradient increases the privacy budget, which needs to be minimized using unsupervised dimensionality reduction. So, to solve those exponential privacy budget increment problems and lower quality noisy data generation problems, we came up with a new approach. Our approach can reduce the need to select a proper clipping parameter and the expense of unsupervised dimensionality reduction. DP-GAN [Xie+18] solved the problem of privacy leakage due to training via real data-based training, and here, this approach started to clip weight rather than gradients. Kunar et al. [Kun+21] proposed DT-GAN for generating tabular synthetic data with privacy analysis by differential privacy against membership and attribute inference attacks. Tantipongpipat et al. [Tan+19] utilized differential privacy, and it ensures a private synthetic data generation process that can generate both data and labels. Another approach utilized conditional GAN [TKP19], which provides partial privacy. We were inspired by [COF20] paper's W-GAN usage technique. But, most of such methods targeted MNIST datasets where the learning task is much easier than complex medical datasets. They still have the problem of coming up with excellent clipping value. We eliminated the need to search for an appropriate clipping value using W-GAN. We also utilized high-dimensional radiology images, and our model can generate high-quality synthetic medical data in both

centralized and distributed settings. DP-Fed AVG GAN [Aug+20; McM+18] works under the trusted server assumption. Still, it is difficult to assume that a centralized server is trusted because we also have to be prepared when the server becomes dishonest. Our approach ensures a federated system where the server only receives noisy gradients, so he cannot exploit the real data. So, it also works under the untrusted server assumption.

2.2.2 Generative models in the medical field

Most deep learning models are data-hungry, so they require a lot of data. Directly using those public medical data creates privacy issues. Most of those data contain a tag/header or identifier that includes the patient's sensitive information, diagnosis history, and hospital name. So, people are getting more into synthetic data because synthetic data does not have private information, and those data do not belong to any actual patient. GAN [Goo+14] has already performed significantly well in data generation tasks in different domains; author Skandarani et al. [SJL21] studied whether GANs can also work well in the medical data sector where the generated data should be reliable enough. Authors applied a range of generative architectures ranging from simpler DCGAN [RMC15; MO14; Gro16] to heavier style GANs [KLA19] on cine-MRI, liver CT scan, and retina images. The study indicated that good-performing models could develop realistic data with higher Frechet Inception Distance (FID) scores and satisfactory performance with U-net [RFB15]trained on generated data for segmentation. Bermudez et al. [Ber+18] used GAN to synthesize high-quality 2d axial slices of MRI in an unsupervised manner also supported by im-

age de-noising, which proved the power of deep learning in synthetic data generation. Dai et al. [Dai+20] developed a unified framework for generating synthetic images for multi-modal MRI. Motion in the images causes quality degradation because of image blurring or artifacts. Johnson et al. [JD19] proposed a GAN model that can predict quality brain images from corrupted data. Lei et al. [Lei+19] presented a method that can generate synthetic computed tomography (CT) images based on dense cycle-consistent generative adversarial networks (cycle GAN). In the case of a skin lesion for skin image analysis, a considerable amount of labeled and high-quality data for deep learning is lacking. Baur's [Kaz+20] framework using progressive, growing generative model was able to generate high-quality synthetic data compared to GAN, DCGAN [RMC15], and LAPGAN [Den+15]. Chuquicusma et al. [Chu+18] performed visual Turing test using radiologists to check the quality of their generated lung nodule samples. Their implicit assumption was that if they could learn to generate realistic data using DC-GAN and if it could fool the discriminator, then the model had known enough discriminative embedding. Some other works also exploited different generative methods to generate synthetic medical data of different type. [Bao+19; Wal+18; Gua+18; OOS17]Some previous works indicate that our radiology image generation approach is feasible and can lead to a satisfactory solution, but those works do not consider the privacy of the data. At the same time, our system can work with a differential privacy guarantee. Torfi et al. [TFR22] addressed medical data privacy problems by generating synthetic data with acceptable quality and standard. Their framework used convolutional autoencoders to encode the features and generative adversarial networks to preserve the semantic information in the

generated dataset. One positive side of their work is that, in the case of data generation, they followed robust method—Renyi differential privacy—to ensure and assess the privacy confidence of a system using such mathematical foundations, which also motivated us. Their model yielded better performance than state-of-the-art models based on publicly available benchmark data sets. Still, their model does not work well under higher noise for high-dimensional image type data. Choi et al. [Cho+17] handled binary and count feature-based electronic health record-based synthetic data generation using a specialized medGAN, which does not work for images. In their framework, they incorporated autoencoder and generative adversarial networks. One big problem in artificial data generation, mode collapse, is a common problem that this article successfully addressed using mini-batch averaging, and it was able to ensure limited privacy risk. But, such little privacy cannot provide patient's sensitive data protection properly. So, in our approach, we incorporate relaxed differential privacy that can still generate high fidelity image data (high-dimensional) despite a high noise multiplier, and our artificial data can ensure higher accuracy using simple Resnet18 model [He+16] also. Our approach solves the problem of mode collapse using Wasserstein loss, which works much better than regular binary cross-entropy loss, and it also ensures private data generation. Mode collapse indicate a situation where the generator can only generate a single or small set of output, which reduces diversity among generated images.

2.2.3 Membership privacy attack and defenses

Usually, there are two types of **membership inference attacks**: Black box and white box attacks. Nasr et al. [Sho+17] came up with the concept first. They trained shadow models on synthetic data (selected from distribution based on confidence score) and mathematically showed and quantified membership privacy leakage given black-box access only. Long et al.'s [Lon+18] findings demonstrated that models with good generalization capabilities can still be vulnerable to inference attacks, as they can be indirectly targeted by accessing associated data. Nasr et al. [NSH19] extended this attack by introducing a white box attack that exploits gradient information and showed how such gradient difference between member and non-member makes deep learning algorithms vulnerable in both central and federated settings. Atiqur et al. [Rah+18] also showed that guaranteed differential privacy could be susceptible if the trade-off expects acceptable utility. [SM20] improved the performance of MIA by using a class-dependant threshold based on prediction entropy and proposed a new risk score metric. Label-only attacks [Cho+21; LZ21] reduced dependency on prediction confidence and proved that the confidence masking defense is not very efficient. Certain studies [Ben+20] have indicated that to prevent attacks, it is necessary to address common enemies such as overfitting and generalization gap. Chen et al. [Che+20b] showed how attack models could leak the training data information of different generative models in different settings.

Some of the **defense** strategies are discussed here. For example, Nasr et al. [NSH18] maximized the classification performance of the model and, at the same time, minimize the most potent adversary's membership inference attack's gain based

on prediction confidence. But, adversarial regularization comes with a significant toll on utility. Other regularizations include L2 regularization, early stopping, Weight Decay, Weight Normalization, DropOut and label smoothing [KHD20] are also used to ensure privacy. Such regularizers deteriorate the utility due to a sub-optimal trade-off. Another approach is MemGuard which alters the output of the resultant model by adding noise to confuse the attack model, but it does not work for white-box attacks. DP-SGD [Aba+16] uses differential privacy-based optimization and adds noise to the gradient of micro-batches. Similarly, PATE [Pap+18; Pap+16], PATE with GN-MAX used an ensemble of teachers on different subsets of the data and started using knowledge distillation for privacy. DP-based GAN [Ho+21; Fai+22] tried to generate synthetic data with additional noise. Song et al. [SM20] developed a new privacy risk score and recognized both model sensitivity and generalization error pose a mutual threat. Caruana et al. [CLG00] showed how early stopping can help reduce overfitting problems and mitigate privacy risk.

DMP [SH21] used knowledge transfer initially, where the student model is trained on public data with private teacher-generated soft labels to achieve a superior utility trade-off. But, they require available reference data with desirable properties, like low entropy, to produce better results, and synthetic data-based DMP achieved moderate performance. Managing public data with such properties may be challenging in medical domains. KCD, SELENA [Cho+22; Tan+22] used data splitting and repeated private data usage. However, KCD's limitation is that it might not work if there is duplication in train data or class imbalance and outlier.

2.2.4 Dataset condensation and distillation

This is an emerging topic, and many works have been done in this domain, focusing on neural architecture search and training efficiency. The central concept behind this approach is maintaining the model while simultaneously creating a condensed dataset that allows the model to achieve comparable performance on the reduced data, as it does when trained on the complete dataset. The initial approach was introduced by Wang et al. [Wan+18] where they used an approach similar to meta-learning where a randomly initialized dataset is optimized in a few steps of gradient descent to come up with a smaller dataset that can ensure a similar performance of the model on the real data. To simplify the nested meta-training loop, Zhao et al. [ZMB21] concentrated on aligning the model's gradients based on generated and real data, which aims to ensure that the model follows the same path towards a solution. They [ZB21] also utilized differentiable Siamese augmentation where the actual images and synthetic images are transformed using the same augmentation before matching gradient, which helps to improve the performance significantly due to shared transformation. Some tried to use soft labels via label distillation [BYH20] and also considered text data [SS19]. Nguyen et al. [NCL20] used Kernel Ridge Regression and label solve technique via closed form solution, increasing efficiency. Lee et al. [Lee+22] emphasized class-wise differences in the loss and improved optimization stability using bi-level warm-up. Distribution Matching in embedded space [ZB22] helped to ensure faster computation, and Dong et al. [DZL22] demonstrated how data condensation is connected with privacy theory and training effectiveness. They provided theoretical justifications for the relationship between differential privacy and data condensation. However, their

assumption regarding adversary access to the compressed data becomes vulnerable in worst case, which occurs quite often practically. So, we combine model's knowledge transfer with privacy-oriented data distillation, restricting adversary access to distilled data to ensure better empirical privacy. Moreover, Cazenavette et al.'s [Caz+22] parameter-based imitation learning improved distillation efficiency. But, their work does not optimize data utilizing privacy, and their goal and deliverable are compressed data, not a private model. So, we embrace parameter-matching loss during privacy-oriented data distillation and sanitize the model using knowledge transfer to ensure compact private model delivery.

2.3 Summary

In this chapter, we reviewed a few definitions—namely, generative adversarial networks (GANs), differential privacy (DP), Renyi DP, Gaussian noise, membership inference attack in machine learning, data and knowledge distillation. We also discussed related works on privacy-preserving learning, generative models in the medical field, membership privacy attack and defenses, as well as dataset condensation and distillation.

As a preview, in Chapter 3, we develop a private data generator to utilize a noise-free discriminator, which helps to produce high-quality private radiology data whereas previous methods could not guarantee high-resolution radiology data with such a tight privacy budget. In Chapter 4, we come up with a private classifier model without directly accessing any private data directly, which ensures better privacy. Again, it can ensure a smaller model with high performance which requires half the

parameters compared to related works.

Chapter 3

Privacy-Preserving Synthetic Image Generation

Due to the recent development in the deep learning community and the availability of state-of-the-art models, medical practitioners are getting more interested in computer vision and deep learning for diagnosis tasks. Moreover, those medical diagnostic models can also increase the reliability of conventional findings. As radiology images can convey a lot of information for a patient's diagnosis task, the problem is that such medical data may contain sensitive private information in their content header. De-anonymization (i.e., removal of sensitive header information) does not work well due to the re-identification risk, which may link those images to essential details (e.g., birth date, SSN, institution name, etc.), and such an approach can also reduce utility. In the medical domain, utility is significant because a less accurate diagnosis may lead to the wrong course of treatment and/or loss of life. In this chapter, we develop a differentially private approach that can generate high-quality and

high dimensional synthetic medical image data with guaranteed differential privacy. It can be used to create sufficient quality data to train a deep model. Moreover, we use W-GAN for bounded gradient guarantee, which eliminates the need for an extensive clipping hyperparameter search. We also add noise selectively to the generator to maintain the privacy-utility trade-off. Due to a noise-free discriminator and such selective noise addition to the generator, high-quality and reliable generated radiology images can be utilized for diagnosis tasks. Moreover, our approach can work in a distributed system where different hospitals can contain their private images in the local server and use a central server to generate synthetic radiology images without storing patient data.

3.1 Method

We designed a privacy-preserving method to generate synthetic data. In our case, we have utilized Wasserstein GAN for a specific purpose. Some of the previous approaches [Cho+17; Gua+18; Xu+19; Wal+18] tried to generate medical data but without a privacy guarantee and yielded low-medium utility. Some methods are developed using DP-SGD using generative architecture. But, they used gradient clipping for both discriminator and generator. But We used a different approach to exploit the gradient in the generator to ensure privacy-preserving data. Instead of using a regular optimizer, We have used DP-SGD optimizer following previous techniques. We also used fully convolutional architecture instead of Multi Layer Perceptron to capture sensitive medical images' semantic and spatial information. We used W-GAN as it works slightly better to battle mode collapse. We utilized the

implicit 1-Lipschitz distance property of W-GAN to avoid the crucial hyperparameter tuning for gradient clipping. A proper hyperparameter C helps set the gradient clipping bound, but it sometimes causes bias and takes time to develop an optimal value. But, 1-Lipschitz continuity in our GAN helps keep the gradient norm within a range of 1, which implicitly ensures gradient clipping during the training process without explicitly setting a proper clipping the value. So with the synergy of Renyi differential privacy and such gradient penalty based on the unique property of WGAN, our GAN can generate high-quality synthetic medical data. Using fake and real image-based comparative loss instead of binary cross-entropy and other techniques also helped increase the variation of the trained data, which allowed the target classifier models to generalize well.

3.1.1 Renyi differential privacy implementation

In previous ε-DP approaches, the model creates some problems due to noise accumulation using strong composition [DR14]. As deep learning is an iterative process, noise upper bound gets multiplied with several training epochs. As we subsample images for micro-batch, subsampling also leads to high noise upper bound. Such loose upper bound increases the overall privacy cost. Generating data with privacy requires tracking the privacy budget and preserving the privacy of the generated data as each iteration requires adding noise. Hence, such an iterative learning process leads to a high privacy budget. But we need to minimize the privacy budget, and such an exponential increase in privacy budget may lead to a loose privacy upper bound. Such an upper bound with high noise deteriorate the quality of the image. So, we need

to use the Gaussian method to preserve privacy and keep the privacy bound more tightly under the composition mechanism. Such a Gaussian mechanism with a higher spread and lower peak helps maintain noise balance, but (ε, δ)-privacy does not allow usage of the Gaussian mechanism. To exploit the Gaussian mechanism and ensure a tighter privacy upper bound, we used a simple notion of differential privacy, which satisfies and provides a strict upper bound. Instead of looking at the log ratio of probabilities, this privacy mechanism looks at the distance. This privacy technique ensures a strong guarantee under composition, and it is well suited to the Gaussian mechanism. Gaussian distribution has a less sharp peak, and 95% of the data stays within two standard deviations of the distribution, ensuring the upper bound could be much more compact and tight. Such a strict upper bound reduces the exponential parameter growth problem under iterations. This also satisfies ε-DP privacy when $\lambda = \infty$. (λ, ε)-RDP ensures $(\varepsilon + \frac{\log(1/\delta)}{\lambda - 1})$-DP privacy. Using such relaxed privacy helped us avoid overestimating privacy loss during multiple iterations as Renyi differential privacy supports the composition of different mechanisms where the budget does not grow exponentially. We can consider D and D' as two distributions, and $Pr(M(D'))$ is the probability of D after applying the generative mechanism M. λ is a parameter of that equation. Here, different epoch's generation task is considered as different mechanisms:

$$\begin{aligned} & D_\lambda(M(D)||M(D'(x))) \\ = \; & \frac{1}{\lambda - 1} \log \mathbb{E}_{x \sim M(D)} \left(\frac{Pr(M(D))}{Pr(M(D'))} \right)^{\lambda - 1} \leq \varepsilon \end{aligned} \tag{3.1}$$

3.1.2 GAN implementation

If G is the generator, it takes random noise z as input and generates an image $G(z)$ as output. In the usual case, we provide the features, and the classifier classifies whether it is fake or real. But, in generator G, we provided the label y information, and a random Bernoulli or Gaussian noise z to generate the features $\hat{x}$, which are pixel values of the X-ray image. To create variation in data, we can alter the noise z, which will generate different pixel intensity values leading to a slightly separate X-ray image. In the generation process, the discriminator plays a vital role, so we kept the gradient of the discriminator D, intact and noise-free. A reliable discriminator is necessary as it can provide information regarding how accurate the image is. The discriminator that takes generated image $G(z)$ as input and $D(G(z))$ produces 0 if it is fake and one if it is real, so D simply acts as a binary classifier. But, the confidence probability value of the generator $D(G(z))$ indicates how fake or real the data is so that such meaningful error can be corrected in the second iteration. In the case of D, we generally use binary cross-entropy to calculate the criticism feedback; then, the feedback is backpropagated through the generator so that generator can learn whether the generated image $\hat{x}$ is realistic or not. The generator and discriminator have been trained simultaneously so that both models become experts. But, the generator must not become superior to the discriminator. Because an overfitted discriminator becomes so accurate that it provides confidence value at the highest or lowest level, which cannot give any meaningful feedback to improve the generator. So, we updated the discriminator five times per generator iteration. If x is the input data, it tries to minimize the following loss in Eq. (3.2). So, the loss function in Eq. (3.2) consists of

θ_D and θ_G parameters for discriminator D and generator G and g^t from Eq. (3.4) is the loss for generator and discriminator:

$$\min_G \max_D \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[1 - \log D(G(z))] \tag{3.2}$$

$$\min_G \max_D \mathbb{E}_{x\sim p_{\text{data}}(x)}[D(x)] - \mathbb{E}_{z\sim p_z(z)}[D(G(z))] \tag{3.3}$$

So, we decided to use Wasserstein loss following Eq. (3.3) instead of binary cross-entropy loss. It approximates the earth mover distance between a real and fake distribution. So, it helps to remove the ceiling of 0 and 1 of loss, which helps to fight the vanishing gradient problem, and continuous feedback helps to keep the learning with feedback consistent:

$$g^t = \nabla_\theta \mathcal{L}(\theta_G, \theta_D) \tag{3.4}$$

We used Wasserstein loss with a clipping bound of 1. Usual approaches clip the gradient before updating parameters. So, if the gradient vector is g and the L2-norm of the gradient is $||g||_2$ then we do the clipping by following $g/g(\max(1, \frac{||g||_2}{C})$. This process helps to ensure that $||g||_2 \leq C$ where C is the clipping parameter. But we mentioned that we eliminated the need to set the C value as we are using the Wasserstein loss, which measures the statistical distance between fake and real image distribution. 1-L continuous condition ensures the norm of the gradient $||g||_2 \leq 1$. So we try to enforce such 1-L continuity during training. We can do it by using wight clipping by setting a maximum or minimum allowed weight range but enforcing clipping reduces the limited learning capability of the discriminator. So, in the case of a discriminator, we will use gradient penalty to keep the sensitivity bounded like Eq. (3.5). We will calculate the loss as the distance between the real image x from the P

distribution, and the fake image y from the Q distribution. In the loss term, we add a regularization term for calculating the loss for interpolated images from fake and real, multiplied with λ, a gradient penalty term. In such a way, we sample some points by interpolating between fake and real examples to get an interpolating image using a random number α. We deduct one from the gradient of discriminator's norm ∇D in Eq. (3.6), which ensures that the discriminator's gradient norm are bounded within a range of 1. This ensures clipping value as one without extensive hyperparameter tuning:

$$\begin{aligned} \mathcal{L}_D &= -\mathbb{E}_{x\sim P}[D(x)] + \mathbb{E}_{\hat{x}\sim Q}[D(\hat{x})] \\ &\quad +\lambda\mathbb{E}[(||\nabla D(\alpha x + (1-\alpha)\bar{x}|| - 1)^2] \qquad (3.5) \\ \mathcal{L}_G &= \mathbb{E}_{z\sim p_z(z)}[D(G(z))] \qquad (3.6) \end{aligned}$$

3.1.3 Privacy preserving training with santization

At this moment, by sanitization, we indicated refining the sensitive value by clipping and adding noise. The main learning mechanism of the machine learning model and deep learning depends on backpropagation. First, we provide a sample to the model, and it generates the output and calculates loss by comparing it with the real output. Then it uses loss for each sample to update the model per iteration. Our strategy is to add noise to the gradient so that updates regarding one single example cannot impact the overall learning. It follows the notion of differential privacy so that one individual sample cannot impact the overall dataset. Previous approaches applied sanitization on both the discriminator and generator. Still, following some recent works [COF20], we decided to add noise to the gradient of the generator G in

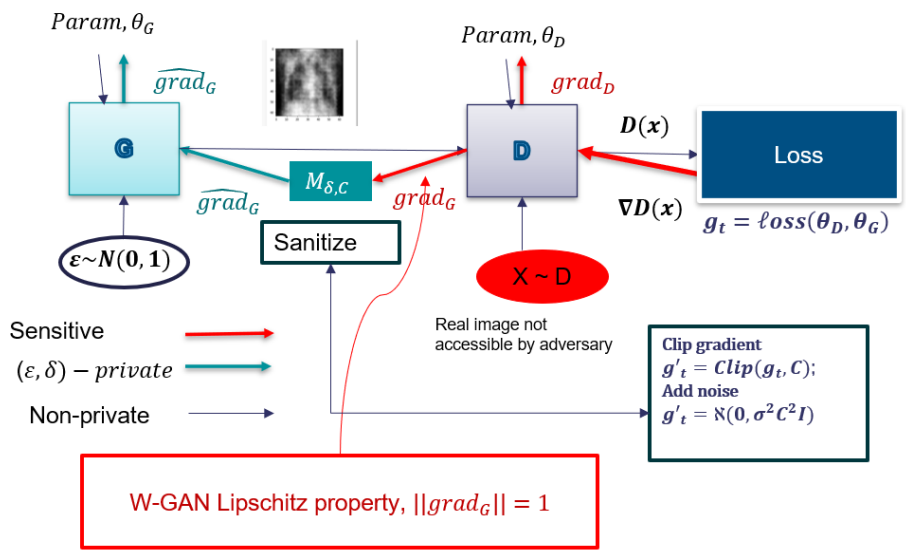


Figure 3.1: Data sanitization workflow

Eq. (3.10) only. We will not clip and add noise to the discriminator D in Eq. (3.9) because we are going to release the generator for data generation. If gr_G^t is the gradient of the generator G we apply gradient clipping and noise-adding mechanism $M_{\sigma,C}(gr_G^{(t)})$ to get that the modified gradient $\tilde{gr}_G^t$ so that each example cannot have a huge impact on the dataset as in Eq. (3.7). $M_{\sigma,C}(gr^{(t)})$ adds noise from Gaussian distribution with variance σ. We will not provide the discriminator to the client, and discriminator gradient $gr_D^{(t)}$ will remain unchanged, and it will be kept in a secure server. If we have to provide the discriminator, we will consider the federated learning scenario where we will have multiple discriminators for each client, which will be stored in client devices and will not breach privacy because each client will train their discriminator separately.

$$gr^t = M_{\sigma,C}(gr^{(t)}) \tag{3.7}$$

$$\theta^{(t+1)} = \theta^{(t)} - \eta.gr^{(t)} \tag{3.8}$$

$$\theta_D^{(t+1)} = \theta^{(t)_D} - \eta.gr_D^{(t)};$$

$$\{Discriminator : \tilde{gr}_D^{(t)} := gr_D^{(t)}\} \tag{3.9}$$

$$\theta_G^{(t+1)} = \theta^{(t)_G} - \eta.gr_G^{(t)};$$

$$\{Generator : \tilde{gr}_G^{(t)} := M_{\sigma,C}(gr_G^{(t)})\} \tag{3.10}$$

We applied a selective sanitization approach, which will clip the gradients of the initial layers of the generator and not apply it to the local layers because local layers are not getting exposed to private data. Our plan is that we will not add noise to the discriminator's gradient, but we will add noise to the generator's gradient. Our idea is that as the discriminator provides feedback on the X-ray image's quality, the discriminator's noisy update cannot identify the difference between fake and real data. But as we are not releasing the discriminator, a noise-free discriminator helps preserve more gradient information of the discriminator, leading to high-fidelity image data despite the noise multiplier's value. In the medical domain, image quality plays a crucial role because the semantic information of the image dictates a critical decision related to the disease. So, we tried to make a trade-off that can ensure both image quality and privacy, which is later proved by the satisfactory performance mentioned in our result section,

According to Figure 3.1, there are two parts to the generator's gradient. One part is local, which is going downwards, which comes back to the generator, and one part is coming upwards, which is not local because it comes back from the discriminator and is affected by real data. So, Instead of sanitizing the whole network's gradient, we decided to sanitize the gradient that is directly relevant to the noisy input. Following the chain rule, we can identify that upward gradient,gr_G is directly impacted by

real data, so we decided to sanitize this part of the gradient only so that the local gradient gr_G can preserve implicit gradient information, which is free from the impact of real data. The generator is updated twice during the training process. In GAN, when we update the generator, we keep the discriminator fixed and then update the discriminator and keep the generator fixed. According to the figure, the generator's updates back-propagated during discriminator evaluation are the upward gradient directly impacted by the real image. Hence, we decided to clip and add noise to the upward gradient. But during the downward gradient update, the gradient contains only relevant local information, which is not directly related to real data, so we do not sanitize the local gradient according to Figure 3.1. In such a way, applying such selective noise addition by breaking down the chain rule helps us preserve important gradient information. So, it leads to high-quality synthetic data where reliability is critical as the spatial features of the images will be used for medical diagnosis tasks. In Figure 3.1 red arrow indicates the sensitive gradient and the green arrow indicates sanitized gradient. The red $X \sim D$ indicates the real X-ray image data, which is sensitive so the gradient coming back from discriminator D's loss is indicated with a red arrow. The green arrow going out to generator G from Mechanism M is a green gradient because mechanism M is used to sanitize the gradients.

We use the WGAN, which has a special condition is that it should be 1-Lipschitz continuous that is the slope of the gradient of the discriminator should always be 1. According to the theory of 1-Lipschitz continuity it automatically bounds the value of gradient.

3.1.4 Federated approach

We also ensured a Federated learning approach where there will be a N_D number of discriminators that are trained in N client computers. Real data (x, y) will be exposed to the clients (hospitals) where they do not need to release sensitive data. Instead, they can train the lightweight discriminator on their personal computer. And the 1-Lipschitz property of Wasserstein helps to ensure implicit gradient clipping without performing sanitization. All of the discriminator's updates will be sent back to the central server's generator in Eq. (3.11) and the generator will be updated based on the accumulation of all gradient information. We need a reliable and accurate discriminator to stabilize the training and ensure high-fidelity synthetic data. We followed a pre-trained starting approach where the discriminators will be previously pre-trained in different client computers for such an approach. During training, the pre-trained discriminator will ensure that generators are updated from the start of the training so that we can generate data using fewer epochs. During the generator update, noise and gradient clipping are applied to the upward gradient, similar to the centralized approach. The iterative process increases the privacy budget, so pre-training will also help reduce the privacy budget. It will require fewer iterations to decrease the privacy budget with fewer iterations.

The main advantage of this approach is that if someone wants a private discriminator, this approach will also ensure it. Because the discriminator will be stored in the client's computer, it will only have access to that specific client's corresponding X-ray images. There are other risk factors that we did not ignore. For example, if the client cannot trust the server, the client needs privacy protection from the server.

But we tackled such a condition also because the client's gradient information that will be passed to the server will be sanitized, and so the encoded noisy weight cannot convey any information related to the client's real data to the server:

$$\theta^{(t+1)}_{D_{i=1\ldots N}} = \theta^{(t)_D} - \eta.g^{(t)}_D; \{Discriminator : \hat{g}^{(t)}_D := g^{(t)}_D\} \tag{3.11}$$

3.2 Experiment for medical data

For medical purposes, we considered Kaggle Chest X-ray Images (Pneumonia) [Ker+18] and also used MNIST dataset for qualitative and quantitative comparison purposes. Because most of the previous privacy-based data generation models used MNIST for study purposes to compare generated image-based data. This is the first time we have exploited a real high stake domain's x-ray image dataset to generate synthetic images. One problem with synthetic medical X-ray datasets is reliability. So to ensure reliability and to defend against mode collapse we used W-GAN, which is famous for its high-fidelity data synthesis performance. In each iteration, we generated different Bernoulli or Gaussian noise depending on user's choice to preserve the diversity of the dataset. Observed from Table 3.1, our approach's performance in terms of CNN is much closer to real data. In the experimental setting we trained our model on 24000 generated synthetic data and to avoid class imbalance we generated 12000 Normal patient data and 12000 Pneumonia patient's data. As the GAN training is computationally expensive, we resized the image to 64×64 size and with such a low resolution still, we were able to get upto 76% accuracy, which is within a satisfactory range. In MNIST, it also gained 77% accuracy, which is also good

Table 3.1: Experimental comparison (Accuracy %)

Data	Algorithm	CNN (0.07)	CNN (1.02)	MLP
MNIST	Real	99	97	98
	G-PATE	51	49	25
	DP-SGD GAN	63	60	52
	Our approach	78.2	76	77.2
X-ray	Real	71.56	74.78	76
	DP-SGD GAN	60	58	40
	Our approach	76.172	76.245	74.484

according to Table 3.1. As observed from the figure, despite of high noise multiplier of 0.07/1.02, our approach can generate quality images whereas previous models generate blurry and unclear images. In the case of X-ray images, it also gained really good results with MLP: 74.4% accuracy. We used a highly regularized CNN model to train using our synthetic data and such regularized model also will help to make it free from membership inference attacks. Using such a simple ResNet18 model for X-ray images, it gained an accuracy of 76.245% using CNN on the synthetic image, which is close to 74.78% accuracy for real image (according to Table 3.1). In the case of MLP, it also gained 74.484% accuracy based on artificial radiology data, which is really amazing and it is closer to the model's accuracy of 76% using real images. In our case, we used synthetic data in the training set and real data in the test set so I believe such a higher and comparable accuracy may validate that our model can generate reliable radiology images, which can be used for diagnostic modeling. In Table 3.1, the row for G-PATE is missing for the x-ray image because G-PATE is not applicable to our dataset type.

To analyze the impact of privacy parameters like noise multiplier we performed some experiments with varying noise level. In Figure 3.2, we showed our model's data

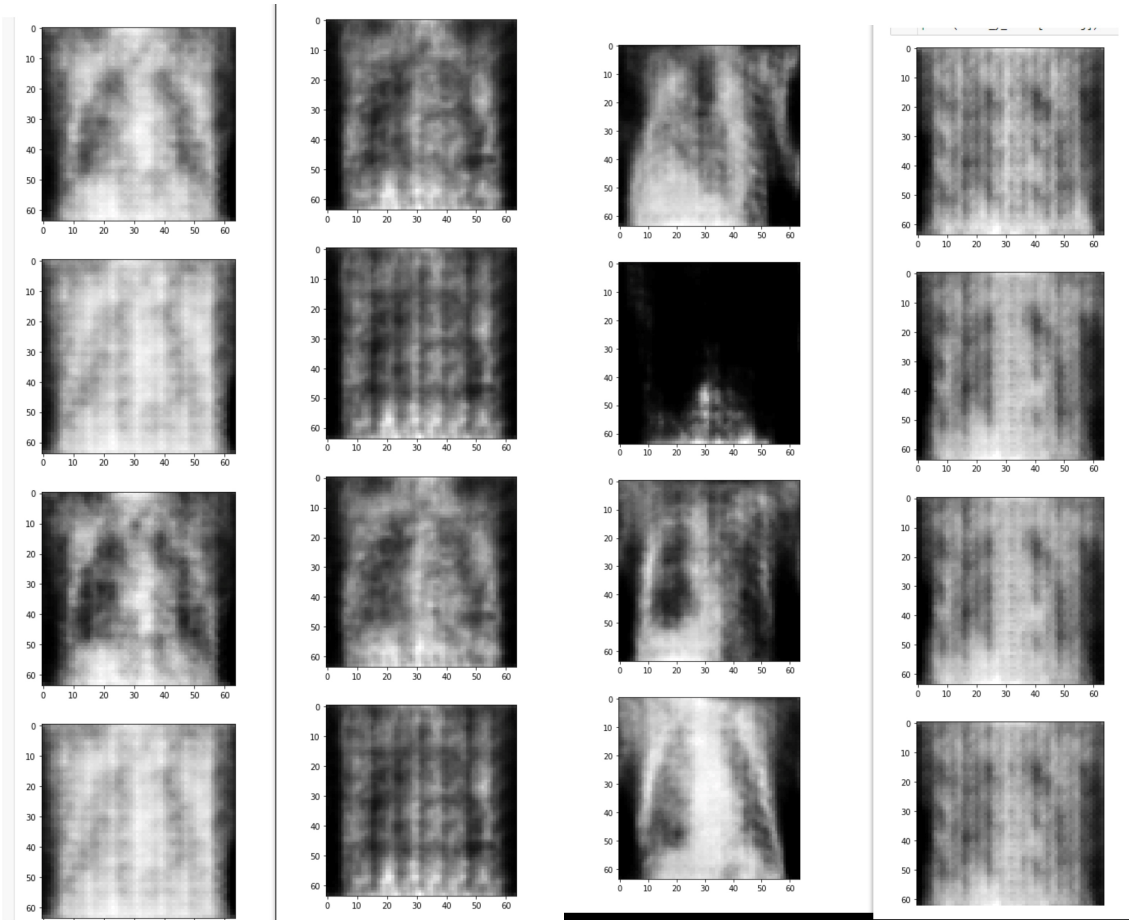


Figure 3.2: Normal patients (first 2 columns) & pneumonia patients (last 2 columns) with noise multiplier 0.07 (1st, 3rd column) vs. noise multiplier 1.02 (2nd, 4th columns)

quality concerning the noise multiplier. The first two columns indicate the standard patient images where the first column's data is generated with a noise multiplier of 0.07 and 1.02. Similarly, the third and fourth column shows the pneumonia patient's data. Here, the third column's pneumonia patient's data is generated using a noise multiplier of 0.07, and the fourth column's pneumonia patient's data is developed with a noise multiplier value of 1.02. From that part, we can observe that adding high noise of 1.02 still yields high-quality X-ray image data. In previous approaches, image

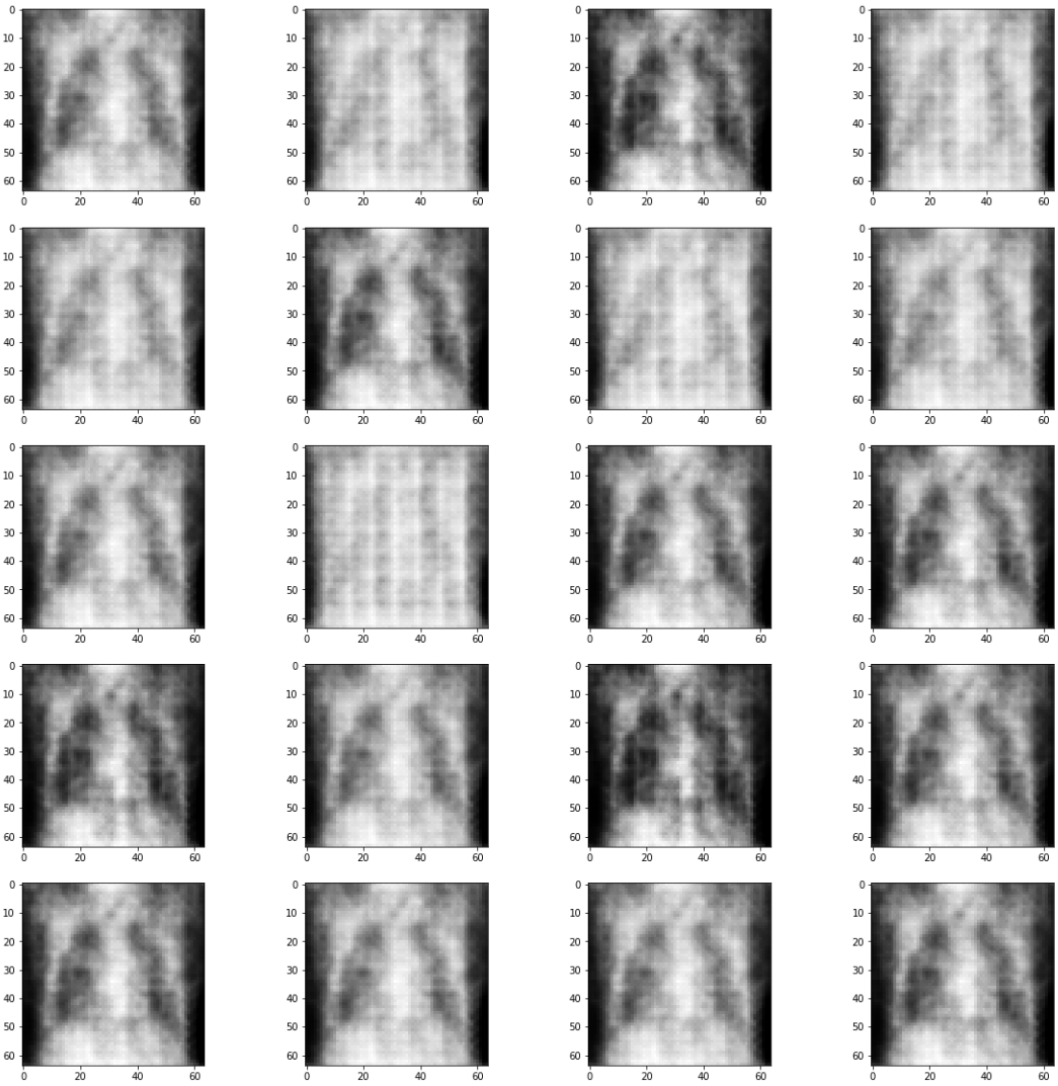


Figure 3.3: Generated normal patients data with noise multiplier 0.07

quality usually gets destroyed after the noise multiplier value of 0.1. But we are glad to mention that our approach yielded 76% accuracy with data generated via a 1.02 noise multiplier, which is satisfactory. For qualitative analysis of the result, we also showed the generated average patient's images in Figure 3.3, which is developed with ε value of 10 and noise multiplier 0.07. We also displayed the generated pneumonia images in

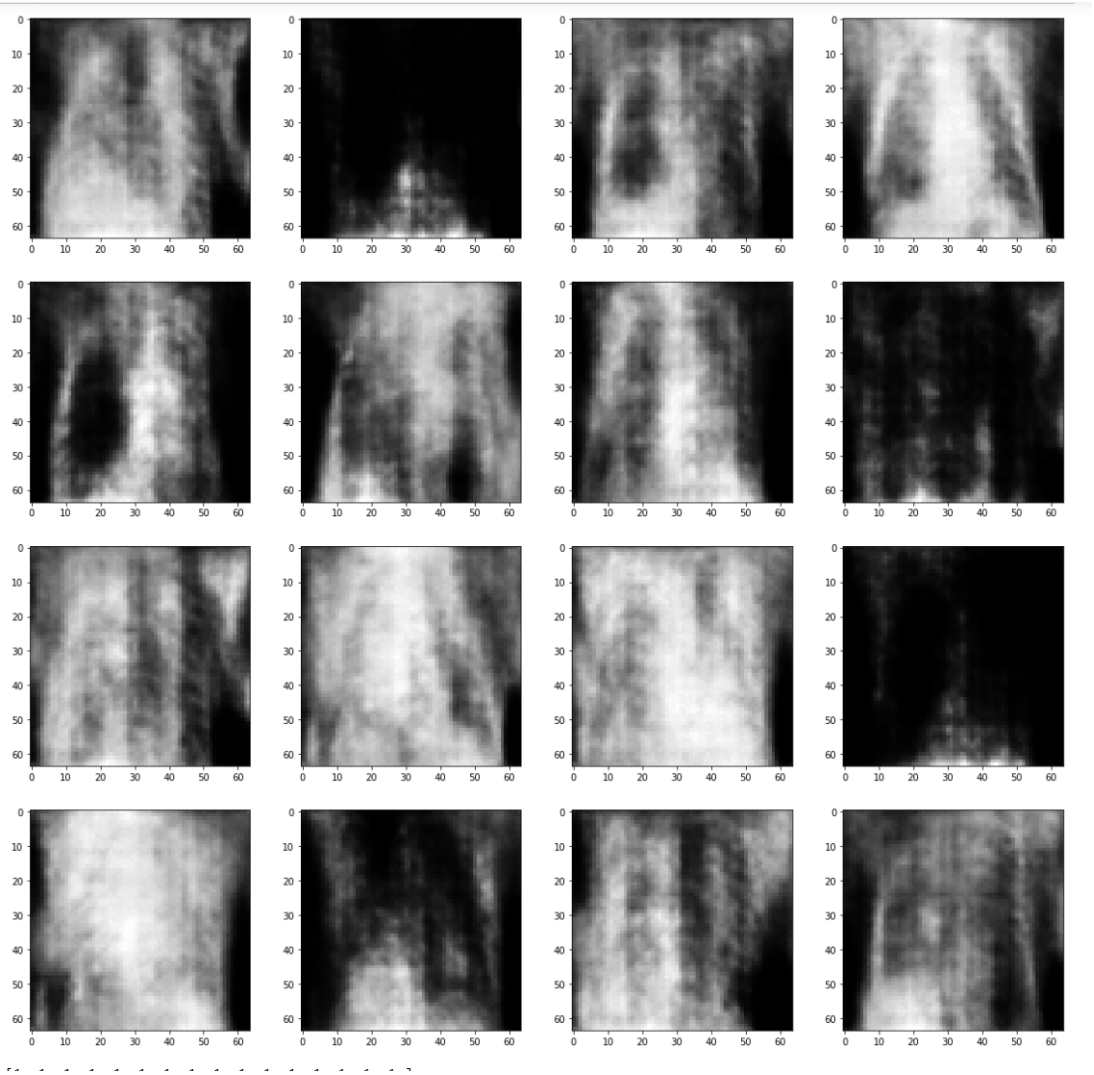


Figure 3.4: Generated pneumonia patients data with noise multiplier 0.07

Figure 3.4. Observed from those two grid views, our model can differentiate between normal and pneumonia patients based on semantic structure. We used a sampling rate of $\frac{1}{1000}$ and we considered a number of iterations to be 2000. According to the experiment in the worst case, our highest privacy budget for 24000 data and 2000 epochs is 3.194×10^4.

Table 3.2: Privacy communication

Method	epsilon ϵ	delta δ	CT bytes
FedAVGGan	9.99×10^6	1×10^{-5}	3.94×10^7
Ours	7.89×10^2	1×10^{-5}	1.95×10^5

Previous approaches did not use high-resolution images for high stake domains, like – the medical radiology image classification task, so we had some trouble comparing with baselines. We used the scaled-down images and modified the diffGAN [TFR22] architecture to generate X-ray images to compare with our model. We had to change the generator, encoder, and decoder architecture to support three-channel images with higher resolution. However, the generated image with a noise multiplier of 0.07 is blurry, and there is mode collapse occurring in the images. Most of the X-ray images look alike. In contrast, our generated image is much sharper and more precise compared to previously generated images from Figure 3.5.

3.2.1 Federated learning based experiment

In the federated approach, it ensured a much more efficient communication cost than previous approaches. Communication cost indicates how many bytes it consumes to perform one generator step by transferring the gradient to the server. It takes fewer bytes, as we followed a previous approach and decided to transfer only the gradient with respect to real samples and as the local discriminator models are contained within local clients only. Table 3.2 shows that Fed AVG GAN's total ε value was 9.99×10^6 with CT bytes 3.94×10^7. In contrast, in our approach, ε value was 7.89×10^2 and CT bytes 1.95×10^5. It has much higher gains in gradient communication in terms of CT bytes. Fed-AVG GAN cannot perform well with a noise multiplier value that is

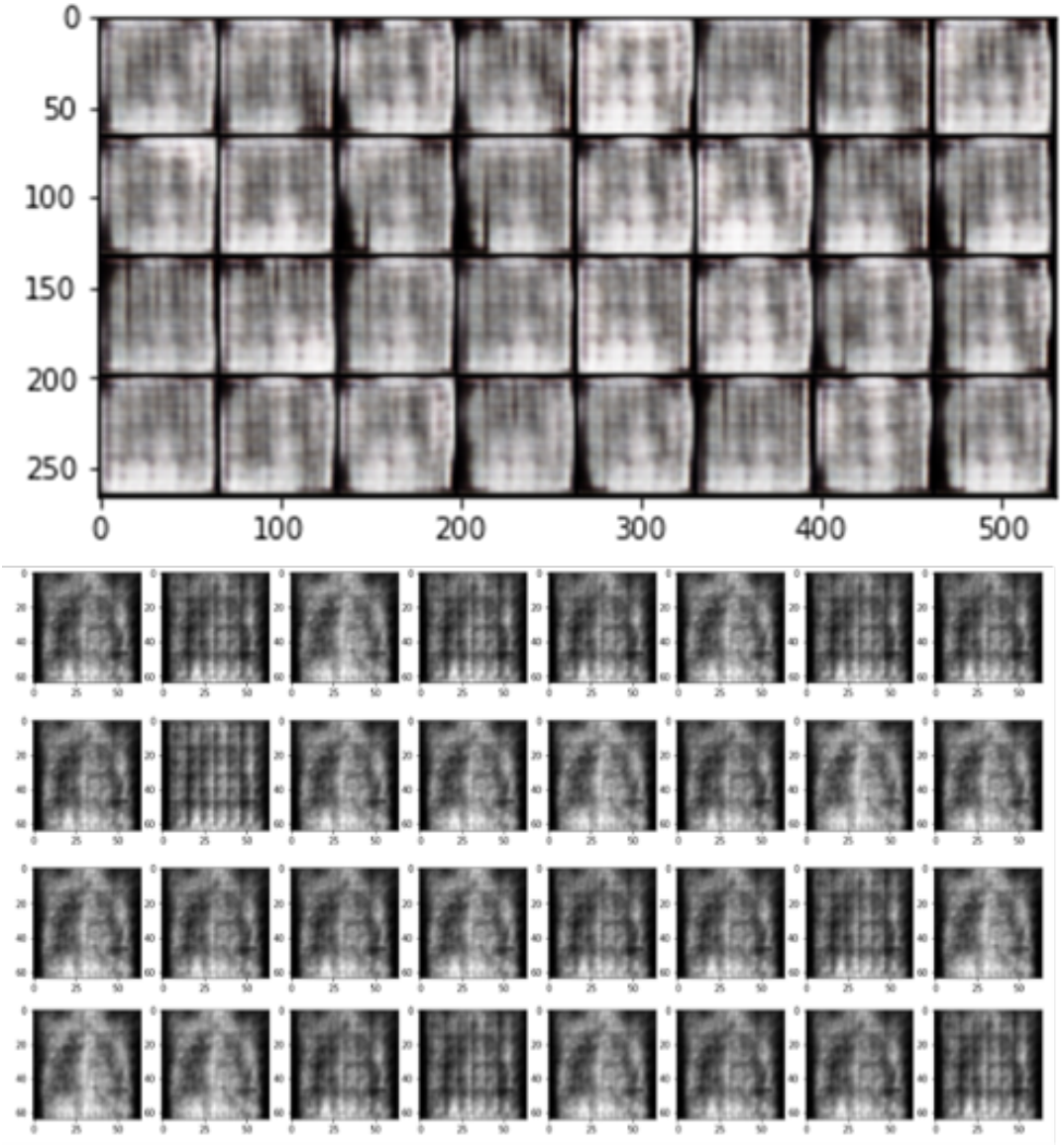


Figure 3.5: Modified diffGAN (top) vs. ours (bottom)

more than 0.1 whereas we have used 1.02 for the noise multiplier value and still our approach is able to generate quality data.

3.3 Summary

This chapter of the thesis presented a novel GAN structure that ensures differential privacy while generating high-fidelity and reliable artificial X-ray images. This architecture facilitates the generation of radiology data using both central and distributed processes. It' main feature is to introduce noise only to the generator part that uses real data while maintaining the integrity of the discriminator to produce high-quality images.

Chapter 4

Privacy-Preserving Learning via Data and Knowledge Distillation

In the current era of deep learning, computer vision, and image analysis have become ubiquitous across various sectors, ranging from government agencies and large corporations to small end devices, due to their ability to simplify people's lives. However, the widespread use of sensitive image data and the high memorization capacity of deep learning present significant privacy risks. Now, a simple Google search can yield numerous images of a person, and the knowledge that a specific patient's record was utilized for training a model associated with a disease may reveal the patient's ailment, potentially leading to advanced attacks in the future. Furthermore, these unprotected models may also suffer from poor generalization due to this overfitting to train data. Previous state-of-the-art methods like differential privacy (DP) and regularizer-based defenses compromised functionality, i.e., task accuracy, to preserve privacy. Such a trade-off raises concerns about the practicability of such defenses.

Other existing knowledge-transfer-based methods either reuse private data or require more public data, which could compromise privacy and may not be viable in certain domains. To address these challenges, where privacy is of utmost importance and utility cannot be compromised, we propose—in this chapter—a novel collaborative distillation approach that transfers the private model's knowledge based on a minimal amount of distilled synthetic data, leading to a compact private model in an end-to-end fashion. Empirically, our proposed method guarantees superior performance compared to most advanced models currently in use, increasing utility by almost 8%, 34%, and 6% for CIFAR-10, CIFAR-100, and MNIST, respectively. The utility resembles non-private counterparts almost closely while maintaining a respectable level of membership privacy leakage of 50-53.5%, despite employing a smaller model with 50% fewer parameters.

4.1 Method

Trained machine learning models demonstrate different behavior for data from other distributions. If the data is from a train set, they show a higher confidence score but offer lower confidence scores if it does not belong to the train set. The central concept of Membership privacy is to produce similar confidence scores and gradient values for members and non-members. To promote that indistinguishability, we can use the distilled data instead of the private training data while training the target student model so it can mitigate the spike in prediction's confidence value in the presence of seen private train data. Three types of networks are used in two phases: expert, teacher, and student. Experts and student models are used for data

distillation, and experts have the same architecture as the student model. Here, the teacher and reinitialized student are used for model distillation. Firstly, we use private data, a trained expert set of models trained on real data, and a student model trained on generated data to optimize the distilled data to develop a smaller amount of representative synthetic data that helps preserve the general pattern of the original dataset. It can generate similar performance while using real private data. Secondly, we train a large teacher model with private data and use it to generate soft labels of the distilled synthetic dataset. This data is used to train the target student model. As this distilled dataset size is smaller and balanced and the features are highly discriminative, this student training becomes more computationally efficient and fair regarding class balance. We also want our student model to be (almost half of the teacher model in terms of parameters) lightweight so that it does not overfit the train data; Instead, their limited capacity and soft label help promote generalizability. So, this collaboration of data distillation and knowledge transfer ensures that the resultant model behaves similarly for members and non-members due to sanitized knowledge of the private data. The high-level process is illustrated in detail in Figure 4.1.

The steps are as follows: (i) Generate a small number of synthetic data S with which the student model $\hat{\theta^F}$ can be trained to have similar performance and parameters to a set of expert networks θ^* of same architecture when trained on the full private data. (ii) Train unprotected large capacity teacher model T on private data (x, y). (iii) Use the trained teacher model T to compute soft labels $T(s)$ of distilled data S with higher temperatures τ to smooth out logit z_i's distribution. (iv) Then,

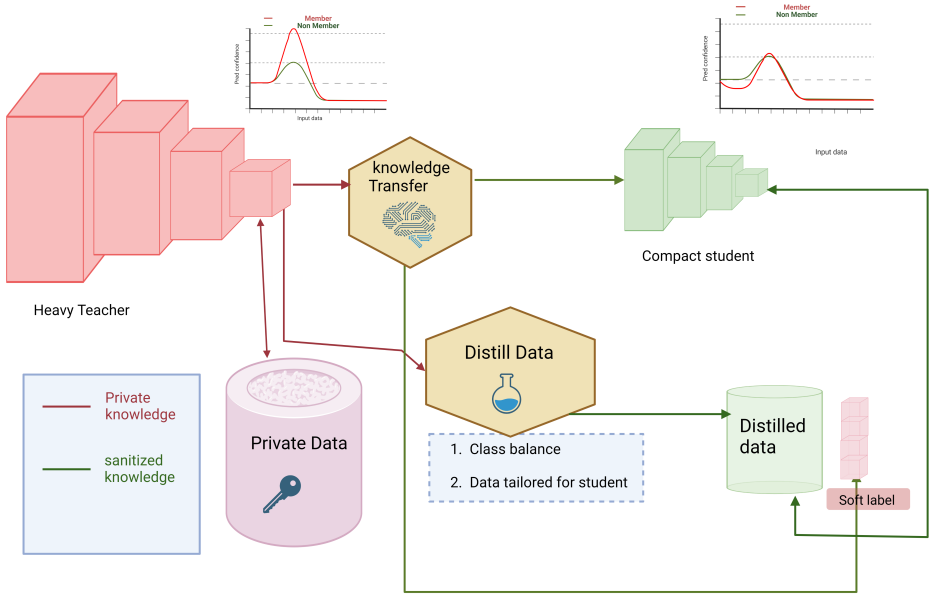


Figure 4.1: Transfer private teacher knowledge to a smaller student using distilled data

we train the reinitialized student model F using KL-divergence-based loss $\mathcal{L}_{KL}$ on distilled data and soft label $T(s)$ to predict $\hat{y}$ as $F(s)_{\theta_S^F}$ to transfer and sanitize the knowledge of the private model. We will talk about distilled data generation in Section 4.1.1 and how we transfer knowledge of private model via distilled synthetic data in Section 4.1.2.

4.1.1 Generating distilled data

We generate a smaller number of synthetic distilled data where the performance of the student model trained on such synthetic data will be similar to that of an ideal model trained on full private data. This data will be used later (in Section 4.1.2) for knowledge transfer to the student network. Initially, we train multiple expert networks that have similar architecture to our target student network on whole private

data using the regular cross-entropy loss like Eq. (4.1). We save the best expert network parameters for each time step interval during training and store these parameter snapshots as θ^*, indicating the upper bound of this k-class classification task on real data where c, p_c, y_c are class variable, predictions and true labels:

$$Loss_{experts_{\theta^*}} = -\sum_{c=1}^{k} y_c \log(p_c) \tag{4.1}$$

In that case, we utilized a set of models to generate expert snapshots, which increases the generalizability of the distilled data set by capturing all the discriminative features possible. The best among saved snapshots will guide the optimization of distilled data, where the student model trained on generated distilled data will be used to test the performance of distilled data. This student trained on synthetic data will be encouraged to have similar parameters to ideal experts when trained on private data.

We consider these saved parameters θ_t^* with time t as snapshots of the exemplary model behavior that the student model F should follow while optimizing generated distilled data S. From Eq. (4.2), we see that θ_t^* indicates the expert time sequence parameters or trajectories that are learned using real private data. After saving the parameters of experts, we fetch a random expert parameter θ_t^* from a random time step/epoch t from the saved list of expert parameters to initialize the student network F with that specific model weight named as $\hat{\theta}_t^F$ for time step t. After the student model is initialized at a randomly sampled time step t, it is trained for a defined no of iterations N on distilled data where the synthetic data S is updated/optimized based on the parameter matching loss $\mathcal{L}_{traj}$ shown in Eq. (4.2). This process is illustrated in Figure 4.2(a):

$$\mathcal{L}_{traj} = \frac{||\hat{\theta}_{t+N}^F - \theta_{t+M}^*||_2^2}{||\theta_t^* - \theta_{t+M}^*||_2^2} \tag{4.2}$$

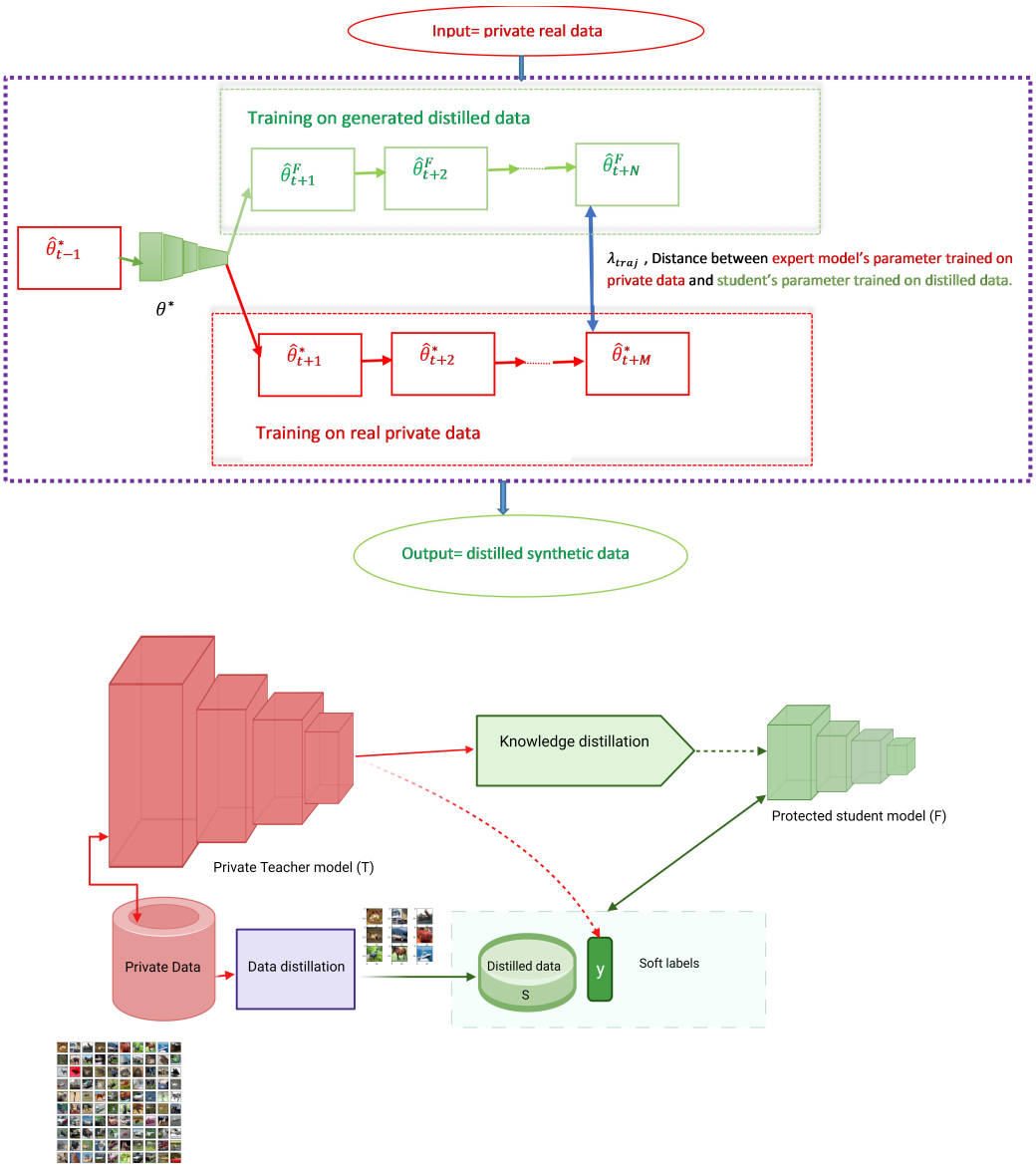


Figure 4.2: (a) Generating distilled data, (b) Distilling knowledge from teacher to student using distilled data

This loss calculates the relative error between learned student model parameter $\hat{\theta}^F_{t+N}$ after N distill data epoch update and known future expert parameters θ^*_{t+M} after designated M epochs to measure the similarity of student and ideal model's training

behavior. $\mathcal{L}_{traj}$ measures the normalized squared L2 error between the trained student network's parameter after the defined time step N and the expected future expert parameters after M epoch, which ensures that the student model's performance on the generated data will be identical to the expert's performance on real data. To normalize the distance between the parameters learned is divided by the total parameter distance covered by the experts θ_t^* in M epochs. Here, M is the number of updates for the expert, and N is the number of synthetic data updates of the student adjusted by the optimized trainable learning rate α in the Eq. (4.3).

We initialize synthetic data S with real images x and subsequently update the pixel of the synthetic dataset for N times using differentiable augmentation $A(S)$ to generated data S like Eq. (4.3) to minimize mentioned parameter matching loss. The data S is updated based on the back propagated L2-loss. This loss measures the similarity between the learned parameters of the ideal model trained on real data and the target model trained on updated synthetic data:

$$\hat{\theta}^F_{t+N+1} = \hat{\theta}^F_{t+N} - \alpha \nabla \mathcal{L}_{traj}(A(\text{minibatch}_{t+N}(S)); \hat{\theta}^F_{t+N}) \tag{4.3}$$

This penalization's gradient $\nabla \mathcal{L}_{traj}$ helps to develop a smaller dataset S, with all required semantic properties aggregated via different mini-batch used during distilled data S optimization. we choose different minibatch_{t+N} at each time step t during each gradient descent update step $N+1$ which helps to embrace generalizability in the dataset. $\hat{\theta}^F_{t+N+1}$ indicates the updated student parameter learned during N step gradient descent for distilled data optimization. This resultant optimized data S can ensure performance similar to real data.

4.1.2 MIA mitigation via knowledge transfer

We want to keep the private data untouched during final student training as models tend to generate high confidence scores for seen data prediction. So, we use the distilled data to train the student model, assuming that the adversary cannot access it. First, we directly train a teacher model T using labeled private training data (x, y) of dataset D. This model is considered as teacher model, which is comparatively a heavy model that has the capacity to learn the pattern from whole dataset. But, we want our target student model to be lightweight and private. We use our teacher model T to generate soft labels $T(s)$ of the synthetic dataset S with high temperature τ, which is used to transfer the unprotected teacher model's private data-based knowledge like Figure 4.2(b). Besides teacher, we use high SoftMax temperatures τ in Eq. (4.4) in the student's prediction also, which helps to maintain a balance in the confidence value of the prediction z_i among class labels. It helps to preserve the relativeness among classes. So, such high-temperature [Sho+17] works as a strong regularizer that can mitigate membership privacy and also increase the target model F's generalization capability:

$$q_{\theta_S^F} = \text{softmax}\left(\frac{a_F}{\tau}\right) = \frac{\exp(z_i/\tau)}{\sum_j (z_j/\tau)} \tag{4.4}$$

Using distilled data S instead of private data D, helps to restrict the F model's access to private data during training. It ensures that the student does not leak privacy. First, the student model F is reinitialized before the knowledge transfer task and then based on each datapoint s and soft labels $T(s)$, we will train a lightweight protected student model F using $(s, T(s))$. At the end of the process, the student model F parameterized by θ_S^F will be deployed directly for inference task $\hat{y}$. The KL-

divergence-based loss like Eq. (4.5) is utilized during the target model's distillation phase. Kullback-Leibler divergence helps measure the distance between prediction $F(s)$ and soft label $T(s)$ distributions:

$$\mathcal{L}_{KL}(F(s), T(s)) = \sum_{i}^{k-1} T(s)_i \log\left(\frac{T(s)_i}{F(s)_i}\right) \tag{4.5}$$

Optimal parameter θ_S^F for inference is obtained while optimizing using the following Eq. (4.6) based on the calculated $\mathcal{L}_{KL}(F(s), T(s))$ from the previous step.

$$\theta_S^F = \text{argmin}_\theta \frac{1}{|S|} \sum_{(s,T(s))\in(S,\theta_x^T)} \mathcal{L}_{KL}(F(s), T(s)) \tag{4.6}$$

To control the privacy trade-off and dependency on the true label y and soft label $T(s)$, we utilize an α hyperparameter which is implemented according to F's total loss function l_F in Eq. (4.7) that combines both $\mathcal{L}_{KL}(F(s), T(s))$ and cross-entropy $\mathcal{L}(F(s), y)$. We use the α to control the privacy trade-off.

$$l_F = \alpha \sum_{(s,T(s))\in S} \mathcal{L}_{KL}(F(s), T(s)) + (1-\alpha) \sum_{(s,y)\in S} \mathcal{L}(F(s), y) \tag{4.7}$$

In a nutshell, we utilize the synergy of collective distillation to sanitize private knowledge so that the resultant model is private, efficient, and well-generalized.

4.2 Results and experiment for PLDK

4.2.1 Datasets and model architecture

CIFAR-10 and CIFAR-100 datasets are commonly used for benchmark comparison in Computer Vision and privacy domain. CIFAR-10 consists of 60,000 32-sized color images with 10 classes where 50k and 10k are reserved for training and testing.

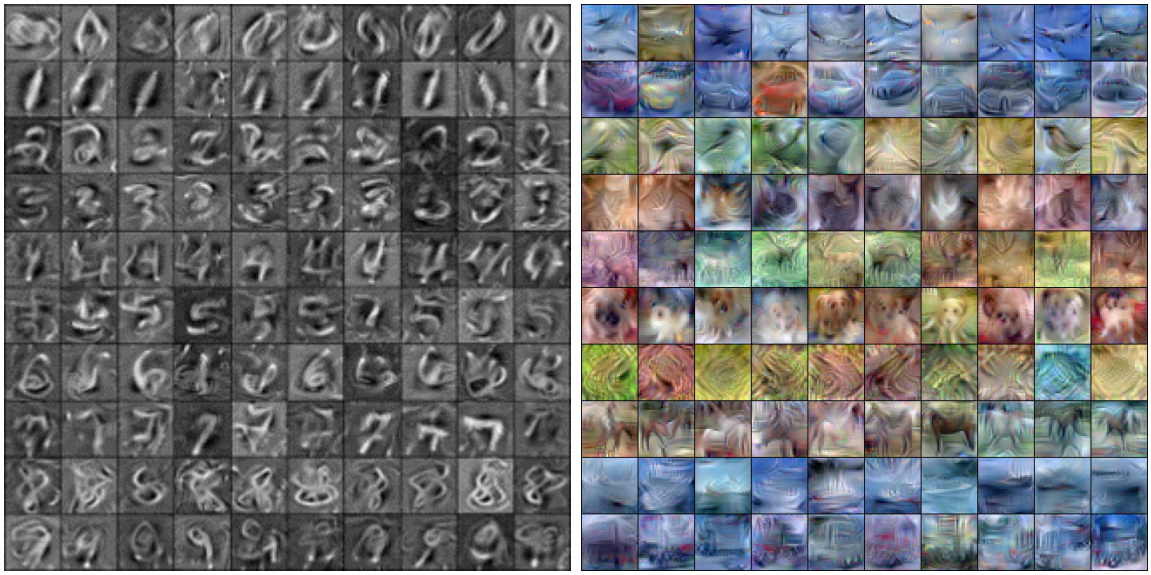


Figure 4.3: Synthetic distilled images: (a) MNIST, (b) CIFAR-10

CIFAR-100 has 100 classes, each containing 600 images, from which 500 for training and 100 for testing. The **MNIST** is a handwritten dataset of 28-sized images consisting of 60000 training and 10000 test data.

Previous approaches used heavier models like AlexNet/DenseNet/ResNet. But, we use lightweight student model as large ones are prone to higher privacy leakage and overfitting. So, we use 3 depth convNet (CNN) that has 320k parameters which are 7 times less than 2.4M parameters of AlexNet. We use small CNN to compare with algorithms' performance gained via larger models. Here, CNN's net width is 128, depth is 3, and InstanceNorm, ReLU activation, and average pooling are used after each block. The architecture has convolution blocks of size 3 with approximately 128 filters.

Table 4.1: Utility-privacy trade-off comparison (Accuracy %)

	Algorithm	Model	Test Acc	Priv Acc (A_b/A_w)
CIFAR-10	No defense	CNN;Alex	67.4; 68.7	76.8/77.2;77.5/77.9
	Regu (WD+LS)	AlexNet	53.2	53.0/53.8
	Adv Reg	AlexNet	53.4	51.2/51.9
	DMP	AlexNet	65	50.6/51.3
	DMP (Synthetic-37.5k)	AlexNet	57.5	51.9/52.1
	DMP (Synthetic-12.5k)	AlexNet	53	50.25/50.3
	PLDK (distilled s=500)	AlexNet	70.3	50.42/50.6
	PLDK (distilled s=500)	**CNN**	**69.3**	**50.21/50.28**
	PLDK (Pre-train with 10k Syn)	CNN	71.8	51.25/51.8
CIFAR-100	No defense	CNN;Alex	39.3; 44	87.8/88; 91.3/90.3
	Regu (WD+LS)	AlexNet	13	51/51.2
	Adv Reg	AlexNet	19.7	54/54.3
	DMP	AlexNet	35.7	55.6/55.7
	DMP (Synthetic-37.5k)	AlexNet	18.5	53.9/54.2
	PLDK (distilled s=500)	AlexNet	47.65	53.73/54.18
	PLDK(distilled s=500)	**CNN**	**46.6**	**53.55/54.1**
	PLDK (Pre-train with 10k Syn)	CNN	48.3	53.98/54.6
MNIST	No defense	CNN;R-18	96.7;98	92/91.4;92.5/91.3
	Regu (WD+LS)	R-18	92	59/59.4
	Adv Reg	R-18	90.48	58.83/58.75
	DMP	R-18	92.015	56.6/56.8
	DMP (Synthetic-37.5k)	R-18	82	54.5/54.6
	PLDK (distilled s=500)	R-18	97.9	53.01/53.67
	PLDK(distilled s=500)	**CNN**	**97.3**	**52.7/53.25**
	PLDK (Pre-train with 10k Syn)	CNN	98	52.9/53.1

4.2.2 Experimental setting

For CIFAR and MNIST we use a batch size of 128 while training the unprotected teacher model. We use ResNet50 for CIFAR-10, simpleNet/ResNet50 for CIFAR-100 and ResNet-18 as the teacher for MNIST. We use convNet as a student and use a few distilled data (50 per class) for CIFAR and MNIST to train student. The same ConvNet is used as experts for data distillation to match the result of convNet trained on optimized distilled data to stored expert ConvNets' best future checkpoints trained on regular data. We consider Blackbox A_b and Whitebox A_w attacks, and replicate

Table 4.2: Utility-privacy trade-off comparison (Accuracy %)

	Algorithm	Model	Test Acc	Priv Acc (A_b/A_w)
CIFAR-10	No defense	CNN;Alex	67.4; 68.7	76.8/77.2;77.5/77.9
	Regu (WD+LS)	AlexNet	53.2	53.0/53.8
	Adv Reg	AlexNet	53.4	51.2/51.9
	DMP	AlexNet	65	50.6/51.3
	DMP (Synthetic-37.5k)	AlexNet	57.5	51.9/52.1
	DMP (Synthetic-12.5k)	AlexNet	53	50.25/50.3
	PLDK (distilled s=500)	**CNN**	**69.3**	**50.21/50.28**
	PLDK (Pre-train with 10k Syn)	CNN	71.8	51.25/51.8
CIFAR-100	No defense	CNN;Alex	39.3; 44	87.8/88; 91.3/90.3
	Regu (WD+LS)	AlexNet	13	51/51.2
	Adv Reg	AlexNet	19.7	54/54.3
	DMP	AlexNet	35.7	55.6/55.7
	DMP (Synthetic-37.5k)	AlexNet	18.5	53.9/54.2
	PLDK(distilled s=500)	**CNN**	**46.6**	**53.55/54.1**
	PLDK (Pre-train with 10k Syn)	CNN	48.3	53.98/54.6
MNIST	No defense	CNN;R-18	96.7;98	92/91.4;92.5/91.3
	Regu (WD+LS)	R-18	92	59/59.4
	Adv Reg	R-18	90.48	58.83/58.75
	DMP	R-18	92.015	56.6/56.8
	DMP (Synthetic-37.5k)	R-18	82	54.5/54.6
	PLDK (distilled s=500)	**CNN**	**97.3**	**52.7/53.25**
	PLDK (Pre-train with 10k Syn)	CNN	98	52.9/53.1

the attack models from previous work [NSH19] where 25k data (12.5k member and 12.5k non-member) are reserved for MIA training. The rest of the data (25k in CIFAR and 35K in MNIST) is used for training teachers. We gain 69.3% utility in CIFAR-10 with 50.21% privacy, 46.6% utility in CIFAR-100 with 53.55% privacy and 97.3% utility in MNIST with 52.7% privacy accuracy. It gains higher accuracy by using a warm start, where we pre-trained students on 10k synthetic data (generated via TransGAN) before the transfer. Details on hyperparameter tuning and inference attack settings can be found in Sections 4.2.2 and 4.2.3.

Hyperparameter tuning

In the data distillation phase, we use simple convNet (like Gidaris and Komodakis [GK18]) commonly used in previous distillation works that are lightweight. We used the same architecture-based model as the student when we distilled the knowledge from the teacher. Before distillation, we train 100 convNets for CIFAR and 80 for MNIST with different seeds and initialization for 100 epochs on private data to generate expert parameters, which we will use as the ideal standard to compare parameters. Those parameters are stored in Hard Drive as buffers. Then, we compare those buffers' parameters trained on real data as upper bound with our student ConvNet's learned parameter trained on optimized distilled data to generate data using mentioned L2 loss. We perform a designated number of synthetic data update epochs to optimize 50 distilled images per class; Using an ensemble or set of convNets to formulate an ideal trajectory improves generalization and ensures that the expert parameters are optimal, which our student convNets should follow during distilled synthetic data generation. For CIFAR-10, the number of expert epochs is 30, and the max start epoch is 40. The dataset is updated 50 times as the synthetic step size is 50, the pixel learning rate is 10^3, and the learning rate for step size is 10^{-5}. For CIFAR-100, the learning rates are the same, and we settle at 40 again as the max starting epoch, and synthetic data update steps are 80 epochs. For MNIST, the learning rate is 10^2, the max start epoch is 20, and the learning rate for step size is 10^{-7}. Moreover, synthetic data update steps are 40 epochs for MNIST. During distilled dataset construction, We consider 50 images per class (ipc=50). So, for CIFAR-10 and MNIST, it is 500 images; for CIFAR-100, we use 5000 images in the distilled dataset.

During teachers training, we use an SGD optimizer to train teacher ResNet-50 and simple LDA with a learning rate of 0.04 for CIFAR-10 with learning rate decay in every 40 epochs, teacher ResNet-18 with a learning rate of 0.06 with decay for MNIST, and teacher ResNet-50 and simpleNet with a learning rate of 0.1 in first 60 epochs, 0.02 in next 40 and 0.004 for later epochs for CIFAR-100. We utilize 3-layer ConvNet (CNN) as a student for all of them. We utilize CNN as a student for CIFAR-10, using an SGD optimizer with a learning rate of 0.06 and decaying the learning rate by a gamma γ of 0.2 at each step size of 100 epochs. The aggregation mechanism of ResNet helps to better fuse different spatial information across deep layers. The ConvNet student gains 69.3 test accuracy with a privacy leakage of 50.21% in 80 epochs, which is almost 8% accuracy improvement for CIFAR-10 with such a smaller private model with almost half parameters and gains low membership inference accuracy close to 50%. For CIFAR-100, we train student CNN with the SGD optimizer maintaining a learning rate of 0.01 for the first 30 epochs and later 50 epochs with a learning rate of 0.001. After transferring the knowledge via distilled data, the student ConvNet gained 46.6% test accuracy and a lower privacy accuracy of 53.55%, which is almost a 34% accuracy gain than the previous best work-DMP. While training the student ConvNet for MNIST, we use a learning rate of 0.08 and decay the learning rate by a gamma γ of 0.2 at each step size of 30 epochs. For MNIST, we gain 97.3% accuracy (which is almost similar to the non-private counterpart), and the privacy gain is 52.7 (which is very close to random guess).

4.2.3 Inference attack

For CIFAR-10 and CIFAR-100 datasets, we use 25k; for MNIST, we use 35k private data for training the unprotected teacher model. For distillation, we consider 500 images for CIFAR-10, MNIST, and 5000 CIFAR-100 synthetic distilled data to train the final protected student model, where the soft labels are generated using the teacher model. We assume a stronger assumption for training the attack model like the previous knowledge transfer works (e.g., DMP), where we use 25k data to train the attack model. So, in this data, we use D^A of 12.5k as members and $D^{A\prime}$ consisting of 12.5k nonmembers to train the attack model. It uses part of samples from the training data distribution to train a model to distinguish between members and nonmembers, ensuring we use a highly potent adversary. For attack models, we utilize similar models implemented in the previous papers. We consider two types of attacks: Black box attack and white box attack where we replicate the same attack settings models from Nasr et al.'s [NSH19] work. Here, the white box attack A_w computes and exploits the gradient difference of member and nonmember's data besides label and prediction confidence like the NSH attack of Nasr et al. to take advantage of the gradient to leak private information. Here, the attacker has access to details, like architecture and other settings of the trained model. Generally, the member's gradient will be significantly different compared to the nonmember's gradient. In the case of the Black box attack A_b, we used an advanced and standard black box attack setting that utilizes prediction confidence and labels both for training. The attack model distinguishes members from nonmembers by exploiting the prediction confidence value's significant difference.

4.2.4 Comparison with regularization and knowledge transfer

We compared PLDK with previous empirical approaches like Adversarial Regularization. Trade-off due to minimizing MIA accuracy reduces utility. It works as a bottleneck in performance, and it uses private data directly to train the model, which leaks private information also. Our model outperforms AdvReg in CIFAR-10 by 30%, CIFAR-100 by 136%, and MNIST by 7.5%, which is very optimistic. In case of privacy, it ensured a privacy accuracy value close to 50.21-53.5%, which is a tighter bound because it does not provide the target model any access to private data. We compared our model with the regular **regularization technique** that helps reduce the target model's overfitting and memorization capability. We compared with common regularization-based approaches like weight decay (WD) and label smoothing (LS), which are very popular in reducing the generalization gap and privacy risk. We simultaneously compared our PLDK with a combination of weight decay and label smoothing. We observed that it gained better performance than the combination of both regularizers at a time. CIFAR-10, CIFAR-100 and MNIST achieved satisfactory performance improvement of 30%, 258% and 5.7% when compared with combined regularization.

We also compared our model with a similar **knowledge transfer** approach of DMP. We used the direct implementation of DMP and utilized the values from their experiment Table. In that case, our method gains increment in utility using a lightweight model with 1/10th parameters. It has more than 8% and 34% gain in utility in CIFAR-10 and CIFAR-100. It also ensured privacy accuracy of 50.21,

Table 4.3: Comparison with KCD using ResNet-18 architecure

Algo	Model (s)	Test Acc	Priv Acc(A_w)
No Defense	Wide Res-28	82.1	66.2
KCD	Wide Res-28	82.2	56.2
MemGuard	Wide Res-28	82	66
PLDK (not pre-trained)	**Res-18**	**82.85**	**52.7**
PLDK (200, (pre-train with 10k Syn)	Res-18	84.3	54.63
PLDK (200, (pre-train with 25k Syn)	Res-18	85.4	54.92

which is very close to 50% in CIFAR-10 and 53.55% privacy in CIFAR-100. It also gained 5.3% accuracy increase in MNIST dataset, with better privacy of 52.7%. We also compared our performance with the most recent approach, KCD in the case of CIFAR-10 only. We do not have access to the code of the KCD paper, and it used more advanced Wide ResNet-28 for CIFAR-10 only. Thus, we utilize slightly larger ResNet-18 as our student to compare with them. It gains higher performance than KCD and MemGuard using a similar but smaller model and ensures much better empirical performance,82.85% utility with 52.7% privacy leakage. It provides a better trade-off than all the state-of-the-art models displayed in Table 4.3 and performance also improves with pre-training. Though, it does not require direct access to private data like KCD. This result shows that using a heavier architecture also provides better performance, giving users the freedom to choose a heavy model if they are less concerned about efficiency.

4.2.5 Comparison with differential privacy

According to theory, differential privacy is a strong tool that can be utilized during optimization to ensure guaranteed privacy statistically. So, we tried to make some

Table 4.4: Empirical comparison with differential privacy

	Algorithm	Model	Accuracy	Privacy Acc (A_w)
CIFAR-10	No Defense	(Alex)	68.7	77.9
	DP-SGD	(Alex, $\epsilon = 198.5$)	55.2	51.7
	DP-SGD	(Alex, $\epsilon = 50.2$)	37.9	50.9
	DP-SGD	(CNN, $\epsilon = 198.5$)	52.9	51.43
	PATE (t=100)	CNN	45.4	49.9
	Ours (PLDK)	**CNN**	**69.3**	**50.28**
CIFAR-100	No Defense	(Alex)	44	90.3
	DP-SGD	(Alex, $\epsilon = 198.5$)	15.77	53.96
	DP-SGD	(Alex,$\epsilon = 50.2$)	13	52.81
	DP-SGD	(CNN, $\epsilon = 198.5$)	15.62	53.81
	PATE (t=100)	CNN	9.9	53.56
	Ours (PLDK)	**CNN**	**46.6**	**54.1**
MNIST	No Defense	(R-18)	98	91.3
	DP-SGD	(R-18, $\epsilon = 198.5$)	95.63	55.92
	DP-SGD	(CNN, $\epsilon = 198.5$)	95	55.62
	DP-SGD	(CNN,$\epsilon = 50.2$)	93.7	54.37
	PATE (t=100)	CNN	94.8	54.49
	Ours (PLDK)	**CNN**	**97.3**	**53.25**

empirical comparisons with DP-based methods like DP-SGD and PATE to ensure that our empirically optimal solution can also be comparable to the contemporary approaches. We show the outcome in Table 4.4. Here, we can observe that DP-SGD with lower privacy bound of $\epsilon = 198.5$ gave a low test accuracy of 55.2, 15.77 and 95.63 for CIFAR-10, CIFAR-100 and MNIST. PATE [Pap+16; Pap+18] with 100 teachers give low test accuracy, whereas our PLDK ensures higher test accuracy and almost comparable privacy accuracy. Moreover, our lightweight architecture CNN can outperform the heavier AlexNet/ResNet. In the case of DP-SGD, we follow the attack model's similar architecture and design choices from Abadi et al.'s [Aba+16] paper using $\sigma = 0.3108$ and C=1. For PATE, we also follow the previous implementation approach from the PATE paper; In PATE, the data-dependent ϵ value is 707.47,1440.8

and 152.5 for CIFAR-10, CIFAR-100 and MNIST and data-independent ϵ values are around 1451.53.

4.2.6 Ablation study

We conducted an ablation study to identify which components play a significant role in our framework. We show these results in Table 4.5. Firstly, we **remove our knowledge transfer** component from this framework and try to check whether only distilled data can provide similar performance. For CIFAR-10, only distilled data without knowledge transfer gives 65.51% utility. For CIFAR-100, it gives 42.5% utility; in MNIST, it gives only 95% utility which is less than the total PLDK using KD with up to 1.7-2% higher privacy risk. So, this performance drop indicates that knowledge transfer is a significant component of our approach. Moreover, we compare with Dong et al.'s [DZL22] privacy analysis of state-of-the-art condensation methods DSA, DM, and KIP and show that our method achieves better utility trade-off. Using Cazenavette et al.'s [Caz+22] distillation (MTT) method reduces privacy compared to our combined model and data distillation method.

Then we use direct synthetic data instead of distillation and **remove data distillation part**. We use one of the most successful state of the art method-transGAN [JCW21] to generate high-fidelity synthetic data, and using it with knowledge transfer degrades utility. We use it because Transformers are highly encouraged for their superior performance. Using high-quality synthetic data (e.g., 9.26 FID score for CIFAR-10) during knowledge transfer results in a degraded utility value of 62%, 36% and 94% in CIFAR-10,100 and MNIST. Performance is not as good as using distilled

Table 4.5: Ablation study using CIFAR-10

Algorithm	Model (s)	Test Acc	Priv Acc
PLDK (DD+KD)	CNN	69.3	50.21
pre-trained (25k synthetic data)	CNN	70	54.8
Synthetic data (No Data Distillation)	CNN	62	52.8
distill data (No Knowledge Transfer)	CNN	65.51	52.1
More data (image per class=200)	CNN	70.54	51.1
Large model	R-18	82.65	53.63
lower alpha (α=0.6)	CNN	70.5	52.92
distill data only (DC method-DM)	CNN	63	61.2
distill data only (DC method-DSA)	CNN	60.6	57.46
distill data only (DC method-KIP)	CNN	64.7	56.32
distill data only (DC method-MTT)	CNN	65.3	54.25

data. DP-GAN [Ho+21] generated private synthetic data considerably deteriorates utility. Moreover, trying with public reference data gives a similar result to DMP, as this method uses additional public data during knowledge transfer. Lower α, pre-training, more data and a heavy model slightly improve the score but reduce privacy. The following contain additional ablation study and analysis on the impact of the model and distill data size on performance.

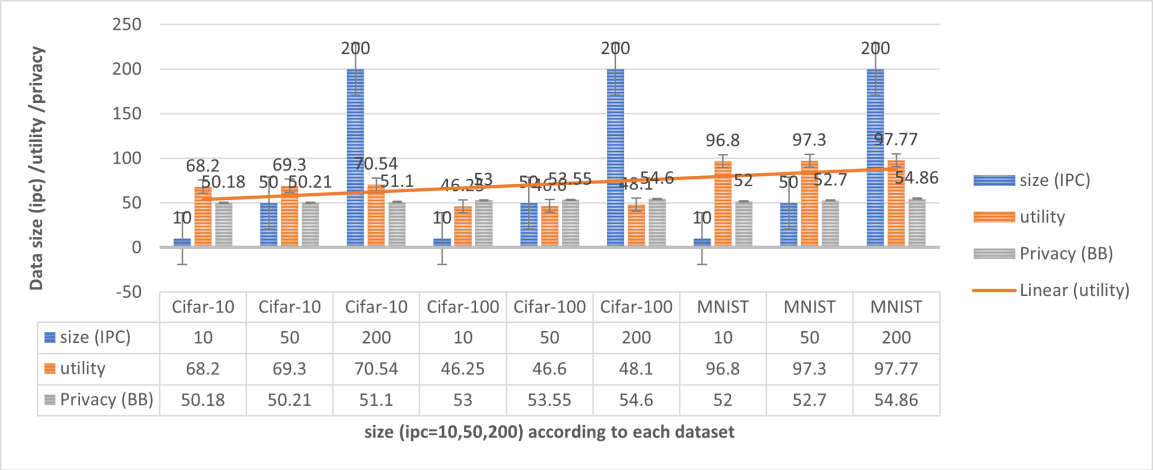
Larger architecture-based model

We tried with larger architecture ResNet-18 on our dataset, and the resultant accuracy is better than the lightweight CNN model, but it comes with a slight membership accuracy increase of 2%, and training heavier models with more distilled data leads to 3% privacy leakage. So, it indicates heavier model helps to improve the utility, but it comes with a small cost of privacy. It also gives 97.98% utility with a 1.75% increase in privacy leakage for MNIST. So, in both cases, privacy deteriorates slightly, supporting our intuition of using a smaller model. So, this experiment aligns

with our goal that utilizing a lightweight model with appropriate capacity helps ensure less overfitting, higher privacy, and more productivity. Again, using a smaller student model during data generation ensures that it not only (a) requires less computational expenses to calculate the expert parameter's buffer during the data distillation phase but also (b) improves training efficiency. Thus, after the efficient and private final model production, it can be directly deployed for practical usage in limited-capacity end devices. The user is given a choice to choose the size of the resulting model architecture since they are the ones who know best which privacy bound and utility trade-off would best serve their needs so they can tune this framework according to their need.

Increase data point via distillation (ipc)

When we increase data size in CIFAR-10 from 50 to 200 per image, it gives slightly better utility which is 70.54, with increased privacy accuracy of 51.1%. Here we use Blackbox (indicated as BB in Figure 4.4) privacy only to compare privacy performance (inference accuracy). In the case of CIFAR-100, it provides 48.1 test accuracy with an MIA risk of 54.6%, and in the case of MNIST, it gives 97.7 with a risk accuracy of 54.86%. This is an average 1-1.5% increment in utility but almost a 1-2% downgrade in privacy. So, according to our experiment, IPC (image per class) of value 50 gives the optimal trade-off compared to the previous state-of-the-art methods. In a nutshell, increasing data up to 200 images per class ensures a 0.7% increase in test accuracy with 1% increased privacy leak in CIFAR-10, almost 1.85% increase in utility and 1.06% degradation in privacy in CIFAR-100 and 0.4% utility



	Cifar-10	Cifar-10	Cifar-10	Cifar-100	Cifar-100	Cifar-100	MNIST	MNIST	MNIST
size (IPC)	10	50	200	10	50	200	10	50	200
utility	68.2	69.3	70.54	46.25	46.6	48.1	96.8	97.3	97.77
Privacy (BB)	50.18	50.21	51.1	53	53.55	54.6	52	52.7	54.86

Figure 4.4: Distilled data size vs. utility/privacy

increase with almost 2.16% privacy reduction in MNIST. Pre-training or warming up the student model with 10-25k synthetic data before training on distilled data improves utility by a smaller percentage, showing that data distillation can be used in conjunction with synthetic data, and this combination can make such a method more robust.

Tuning privacy

The hyperparameter α manages dependency on the true and soft labels. In the loss, α in the first part is for soft labels, and the second part is for ground truth labels. So, we usually use α of 0.9 but decreasing α to 0.6 ensures less dependence on the soft label and more focus on the hard label. Minimum α reduces protection. It slightly improves accuracy up to 70.5% with degraded privacy in CIFAR-10. So, based on trade-off preference, users can also exploit this α parameter to ensure optimal performance in their respective application's domain.

4.3 Summary

This chapter of the thesis presented a privacy-preserving learning framework. It offers a solution to prevent membership inference attacks without the need for extra public data, data adjustment, or repetitive access to confidential information. It involves combining data distillation and knowledge transfer to create a model that performs similarly to a non-private model while ensuring privacy accuracy comparable to random guesses.

Chapter 5

Conclusions & Future work

5.1 Conclusions

In the current era of deep learning, maintaining data privacy is crucial since deep learning models can potentially reveal private information. Users are willing to share their data for model training only if data privacy is guaranteed. Private learning can increase data availability and improve model performance. In my research, I have developed a framework that combines data privacy and model privacy to ensure a secure training process. To achieve this, Chapter 3 introduces a private data generator that creates synthetic private data not linked to any particular user. Then, Chapter 4 introduces a private target model that ensures both model compactness and privacy. These two steps provide a double-layered privacy mechanism, preventing data breaches and misuse of private data.

In this thesis, in the first part, we designed a differentially private GAN architecture to generate synthetic X-ray images, which supports both central and distributed

radiology data generation processes for the first time. Our main goal is to only add noise to the generator part exposed to real data. We will keep the discriminator intact, ensuring high-quality image generation. We need to release the generator only, and the generator part of our final model works as a private black-box model hence that it will be differentially private. Our approach guarantees the user data privacy also if we want to release the discriminator, however, in that case, each client has to use and store their discriminator model locally, and there will be a generator in the central server; it will also ensure that any third party will not be able to reconstruct source data exploiting already learned weights because the encoded weights are noisy. We will also use a highly regularized model to test the generated data's utility to fight against inference attacks. Our evaluation results demonstrate that our approach ensured higher-quality private X-ray images, ensuring a feasible privacy budget with more profound architecture. Selective noise addition and W-GAN's implicitly clipping property will help to make it possible. But for confidential model training, which is resistant to inference attack, our model-tailored data and knowledge distillation framework will help to come up with compact and efficient private models that can ensure a satisfactory utility trade-off over state-of-the-art Membership inference-proof models (a smaller model with less than one-tenth of the parameters of comparable models).

In the second part, we proposed a framework that can mitigate membership inference attacks that do not require additional public data, data tuning, or repetitive access to private data. This process utilizes the collaboration of data distillation and knowledge transfer to develop a model that ensures similar performance to a non-

private model, and the privacy accuracy of this model is close to random guesses. To the best of our knowledge, this is the *first work to investigate the synergy of distilled data to transfer the knowledge of the private model to produce an MIA-resistant model that is also efficient*, ensuring superior performance than existing approaches. This has a competitive advantage over prior works where the final model is as lightweight as half in size (has 320k parameters which are 7 times less than previous 2.4M) in terms of parameters than previous models so that it can be directly deployed in any remote edge device. Our extensive experimentation has confirmed that our strategy can provide a better trade-off and is suitable for real-world activities where utility and privacy are equally crucial. As this framework provides an end-to-end solution to produce a scalable, protected model given only private data, future research will support and analyze such an approach. Moreover, this work primarily focused on privacy in computer vision tasks, as there are fewer defensive training strategies in this domain.

5.2 Future work

As *future work*, I want to explore a better generative model, like de-noising diffusion or transformers for a higher resolution image that can ensure higher utility. In my opinion, utilizing an advanced generative approach can lead to the production of high-quality data at scale. By incorporating this data into the classification model, there is potential to enhance both its utility and precision in high stake domains. Moreover, I plan to extend the data distillation approach to textual and tabular data in the future so that the private classifier model also works for tabular data.

Bibliography

[Aba+16] Martín Abadi et al. "Deep Learning with Differential Privacy". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (2016).

[AC19] Mohammad Al-Rubaie and J. Morris Chang. "Privacy-Preserving Machine Learning: Threats and Solutions". In: *IEEE Security & Privacy* 17.2 (2019), pp. 49–58. DOI: 10.1109/MSEC.2018.2888775.

[ACB17] Martín Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein Generative Adversarial Networks". In: *ICML*. 2017.

[Ali+20] Sheraz Ali et al. "Towards Privacy-Preserving Deep Learning: Opportunities and Challenges". In: *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)* (2020), pp. 673–682.

[Aug+20] Sean Augenstein et al. "Generative Models for Effective ML on Private, Decentralized Datasets". In: *ArXiv* abs/1911.06679 (2020).

[Bao+19] Mrinal Kanti Baowaly et al. "Synthesizing electronic health records using improved generative adversarial networks". In: *Journal of the American Medical Informatics Association* 26 (2019), pp. 228–241.

[BAZ20] Abubakar Bomai, Mohammed Shujaa Aldeen, and Chuan Zhao. "Privacy-Preserving GWAS Computation on Outsourced Data Encrypted under Multiple Keys Through Hybrid System". In: *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)* (2020), pp. 683–691.

[BCN06] Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. "Model compression". In: *Knowledge Discovery and Data Mining*. 2006.

[Bea+19] Brett K. Beaulieu-Jones et al. "Privacy-Preserving Generative Deep Neural Networks Support Clinical Data Sharing". In: *Circulation. Cardiovascular Quality and Outcomes* 12 (2019), e005122–e005122.

[Ben+20] Jason W. Bentley et al. *Quantifying Membership Inference Vulnerability via Generalization Gap and Other Model Metrics*. 2020. DOI: 10.48550/ARXIV.2009.05669. URL: https://arxiv.org/abs/2009.05669.

[Ber+18] Camilo Bermudez et al. "Learning implicit brain MRI manifolds with deep learning". In: *Medical Imaging 2018: Image Processing*. Vol. 10574. International Society for Optics and Photonics. 2018, p. 105741L.

[BYH20] Ondrej Bohdal, Yongxin Yang, and Timothy M. Hospedales. "Flexible Dataset Distillation: Learn Labels Instead of Images". In: *CoRR* abs/2006.08572 (2020). arXiv: 2006.08572. URL: https://arxiv.org/abs/2006.08572.

[Caz+22] George Cazenavette et al. "Dataset Distillation by Matching Training Trajectories". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2022), pp. 4749–4758.

[Che+20a] Defang Chen et al. "Cross-Layer Distillation with Semantic Calibration". In: *AAAI Conference on Artificial Intelligence*. 2020.

[Che+20b] Dingfan Chen et al. "GAN-Leaks: A Taxonomy of Membership Inference Attacks against Generative Models". In: *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. CCS '20. Virtual Event, USA: Association for Computing Machinery, 2020, pp. 343–362. ISBN: 9781450370899. DOI: 10.1145/3372297.3417238. URL: https://doi.org/10.1145/3372297.3417238.

[Cho+17] E. Choi et al. "Generating Multi-label Discrete Patient Records using Generative Adversarial Networks". In: *MLHC*. 2017.

[Cho+21] Christopher A. Choquette-Choo et al. "Label-Only Membership Inference Attacks". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, July 2021, pp. 1964–1974. URL: https://proceedings.mlr.press/v139/choquette-choo21a.html.

[Cho+22] Rishav Chourasia et al. "Knowledge Cross-Distillation for Membership Privacy". In: *Proceedings on Privacy Enhancing Technologies* 2022.2 (Mar. 2022), pp. 362–377. DOI: 10.2478/popets-2022-0050. URL: https://doi.org/10.2478%5C%2Fpopets-2022-0050.

[Chu+18] Maria JM Chuquicusma et al. "How to fool radiologists with generative adversarial networks? A visual turing test for lung cancer diagnosis". In: *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*. IEEE. 2018, pp. 240–244.

[CLG00] Rich Caruana, Steve Lawrence, and Lee Giles. "Overfitting in Neural Nets: Backpropagation, Conjugate Gradient, and Early Stopping". In: *Proceedings of the 13th International Conference on Neural Information Processing Systems*. NIPS'00. Denver, CO: MIT Press, 2000, pp. 381–387.

[COF20] Dingfan Chen, Tribhuvanesh Orekondy, and Mario Fritz. "GS-WGAN: A Gradient-Sanitized Approach for Learning Differentially Private Generators". In: *NIPS'20* 10.5555/3495724.3496787 (2020).

[Dai+20] Xianjin Dai et al. "Multimodal MRI synthesis using unified generative adversarial networks". In: *Medical physics* 47.12 (2020), pp. 6343–6354.

[Den+15] Emily L. Denton et al. "Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks". In: *NIPS*. 2015.

[DR14] Cynthia Dwork and Aaron Roth. "The Algorithmic Foundations of Differential Privacy". In: *Found. Trends Theor. Comput. Sci.* 9 (2014), pp. 211–407.

[DZL22] Tian Dong, Bo Zhao, and Lingjuan Lyu. "Privacy for Free: How does Dataset Condensation Help Privacy?" In: *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland,*

USA. Ed. by Kamalika Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, 2022, pp. 5378–5396. URL: https://proceedings.mlr.press/v162/dong22c.html.

[Fai+22] Fahim Faisal et al. "Generating Privacy Preserving Synthetic Medical Data". In: *2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA)*. 2022, pp. 1–10. DOI: 10.1109/DSAA54385.2022.10032429.

[Far+20] Tom Farrand et al. "Neither Private Nor Fair: Impact of Data Imbalance on Utility and Fairness in Differential Privacy". In: *Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice*. PPMLP'20. Virtual Event, USA: Association for Computing Machinery, 2020, pp. 15–19. ISBN: 9781450380881. DOI: 10.1145/3411501.3419419. URL: https://doi-org.uml.idm.oclc.org/10.1145/3411501.3419419.

[Fri+19] Lorenzo Frigerio et al. "Differentially Private Generative Adversarial Networks for Time Series, Continuous, and Discrete Open Data". In: *ArXiv* abs/1901.02477 (2019).

[GK18] Spyros Gidaris and Nikos Komodakis. "Dynamic Few-Shot Visual Learning Without Forgetting". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), pp. 4367–4375.

[Goo+14] Ian Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems* 27 (2014).

[Gro16] Sam Gross. "CONTEXT-CONDITIONAL GENERATIVE ADVERSARIAL NETWORKS". In: 2016.

[Gua+18] Jiaqi Guan et al. "Generation of Synthetic Electronic Medical Record Text". In: *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (2018), pp. 374–380.

[Gul+17] Ishaan Gulrajani et al. "Improved Training of Wasserstein GANs". In: *NIPS*. 2017.

[He+16] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 770–778.

[Ho+21] Stella Ho et al. "DP-GAN: Differentially private consecutive data publishing using generative adversarial nets". In: *Journal of Network and Computer Applications* 185 (2021), p. 103066. ISSN: 1084-8045. DOI: https://doi.org/10.1016/j.jnca.2021.103066. URL: https://www.sciencedirect.com/science/article/pii/S1084804521000904.

[HVD15] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. "Distilling the Knowledge in a Neural Network". In: *ArXiv* abs/1503.02531 (2015).

[JCW21] Yifan Jiang, Shiyu Chang, and Zhangyang Wang. *TransGAN: Two Pure Transformers Can Make One Strong GAN, and That Can Scale Up*. 2021. DOI: 10.48550/ARXIV.2102.07074. URL: https://arxiv.org/abs/2102.07074.

[JD19] Patricia M Johnson and Maria Drangova. "Conditional generative adversarial network for 3D rigid-body motion correction in MRI". In: *Magnetic Resonance in Medicine* 82.3 (2019), pp. 901–910.

[Kaz+20] S Kazeminia et al. "GANs for medical image analysis. Artificial Intelligence in Medicine". In: (2020).

[Ker+18] Daniel S. Kermany et al. "Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning". In: *Cell* 172 (2018), 1122–1131.e9.

[KHD20] Yigitcan Kaya, Sanghyun Hong, and Tudor Dumitras. *On the Effectiveness of Regularization Against Membership Inference Attacks*. 2020. DOI: 10.48550/ARXIV.2006.05336. URL: https://arxiv.org/abs/2006.05336.

[KLA19] Tero Karras, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 4401–4410.

[Kun+21] Aditya Kunar et al. "DTGAN: Differential Private Training for Tabular GANs". In: *arXiv preprint arXiv:2107.02521* (2021).

[Lee+22] Saehyung Lee et al. "Dataset Condensation with Contrastive Signals". In: *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*. Ed. by Kamalika Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, 2022,

pp. 12352–12364. URL: https://proceedings.mlr.press/v162/lee22b.html.

[Lei+19] Yang Lei et al. "MRI-only based synthetic CT generation using dense cycle consistent generative adversarial networks". In: *Medical physics* 46.8 (2019), pp. 3565–3581.

[Li+20] Jeffrey Li et al. "Differentially Private Meta-Learning". In: *ArXiv* abs/1909.05830 (2020).

[Lon+18] Yunhui Long et al. "Understanding Membership Inferences on Well-Generalized Learning Models". In: *CoRR* abs/1802.04889 (2018). arXiv: 1802.04889. URL: http://arxiv.org/abs/1802.04889.

[LZ21] Zheng Li and Yang Zhang. "Membership Leakage in Label-Only Exposures". In: *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. CCS '21. Virtual Event, Republic of Korea: Association for Computing Machinery, 2021, pp. 880–895. ISBN: 9781450384544. DOI: 10.1145/3460120.3484575. URL: https://doi-org.uml.idm.oclc.org/10.1145/3460120.3484575.

[McM+18] H. B. McMahan et al. "Learning Differentially Private Recurrent Language Models". In: *ICLR*. 2018.

[MO14] Mehdi Mirza and Simon Osindero. "Conditional Generative Adversarial Nets". In: *ArXiv* abs/1411.1784 (2014).

[MTZ19] Ilya Mironov, Kunal Talwar, and Li Zhang. "Rényi Differential Privacy of the Sampled Gaussian Mechanism". In: *ArXiv* abs/1908.10530 (2019).

[NCL20] Timothy Nguyen, Zhourung Chen, and Jaehoon Lee. "Dataset Meta-Learning from Kernel Ridge-Regression". In: *ArXiv* abs/2011.00050 (2020).

[NSH18] Milad Nasr, Reza Shokri, and Amir Houmansadr. "Machine Learning with Membership Privacy Using Adversarial Regularization". In: *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. CCS '18. Toronto, Canada: Association for Computing Machinery, 2018, pp. 634–646. ISBN: 9781450356930. DOI: 10.1145/3243734.3243855. URL: https://doi-org.uml.idm.oclc.org/10.1145/3243734.3243855.

[NSH19] Milad Nasr, R. Shokri, and Amir Houmansadr. "Comprehensive Privacy Analysis of Deep Learning: Stand-alone and Federated Learning under Passive and Active White-box Inference Attacks". In: *2019 IEEE Symposium on Security & Privacy,San Francisco, CA* abs/1812.00910 (2019).

[OOS17] Augustus Odena, Christopher Olah, and Jonathon Shlens. "Conditional Image Synthesis with Auxiliary Classifier GANs". In: *ICML*. 2017.

[Pap+16] Nicolas Papernot et al. "Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data". In: *ArXiv* abs/1610.05755 (2016).

[Pap+17] Nicolas Papernot et al. "Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data". In: *ArXiv* abs/1610.05755 (2017).

[Pap+18] Nicolas Papernot et al. "Scalable Private Learning with PATE". In: *ArXiv* abs/1802.08908 (2018).

[Rah+18] Md.Atiqur Rahman et al. "Membership Inference Attack against Differentially Private Deep Learning Model". In: *Trans. Data Priv.* 11 (2018), pp. 61–79.

[RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *MICCAI*. 2015.

[RMC15] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks". In: *arXiv preprint arXiv:1511.06434* (2015).

[SH21] Virat Shejwalkar and Amir Houmansadr. "Membership Privacy for Machine Learning Models Through Knowledge Transfer". In: *AAAI Conference on Artificial Intelligence*. 2021.

[Sho+17] Reza Shokri et al. "Membership Inference Attacks Against Machine Learning Models". In: *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*. IEEE Computer Society, 2017, pp. 3–18. DOI: 10.1109/SP.2017.41. URL: https://doi.org/10.1109/SP.2017.41.

[SJL21] Youssef Skandarani, Pierre-Marc Jodoin, and Alain Lalande. "Gans for medical image synthesis: An empirical study". In: *arXiv preprint arXiv:2105.05318* (2021).

[SM20] Liwei Song and Prateek Mittal. "Systematic Evaluation of Privacy Risks of Machine Learning Models". In: *USENIX Security Symposium*. 2020.

[SS19] Ilia Sucholutsky and Matthias Schonlau. "Soft-Label Dataset Distillation and Text Dataset Distillation". In: *2021 International Joint Conference on Neural Networks (IJCNN)* (2019), pp. 1–8.

[Tan+19] Uthaipon Tao Tantipongpipat et al. "Differentially Private Mixed-Type Data Generation For Unsupervised Learning". In: *ArXiv* abs/1912.03250 (2019).

[Tan+22] Xinyu Tang et al. "Mitigating Membership Inference Attacks by Self-Distillation Through a Novel Ensemble Architecture". In: *31st USENIX Security Symposium (USENIX Security 22)*. Boston, MA: USENIX Association, Aug. 2022, pp. 1433–1450. ISBN: 978-1-939133-31-1. URL: `https://www.usenix.org/conference/usenixsecurity22/presentation/tang`.

[TFR22] Amirsina Torfi, Edward A Fox, and Chandan K Reddy. "Differentially private synthetic medical data generation using convolutional gans". In: *Information Sciences* 586 (2022), pp. 485–500.

[TKP19] Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict J. Paten. "DP-CGAN: Differentially Private Synthetic Data and Label Generation". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2019), pp. 98–104.

[Wal+18] Jason A. Walonoski et al. "Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record". In: *Journal of the American Medical Informatics Association : JAMIA* 25 (2018), pp. 230–238.

[Wan+18] Tongzhou Wang et al. "Dataset Distillation". In: *ArXiv* abs/1811.10959 (2018).

[Xie+18] Liyang Xie et al. "Differentially Private Generative Adversarial Network". In: *ArXiv* abs/1802.06739 (2018).

[Xu+19] Lei Xu et al. "Modeling Tabular data using Conditional GAN". In: *NeurIPS*. 2019.

[YAC20] Emre Yilmaz, Mohammad Al-Rubaie, and J. Morris Chang. "Naive Bayes Classification under Local Differential Privacy". In: *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)* (2020), pp. 709–718.

[YKF20] Shen Yan, Hsien-te Kao, and Emilio Ferrara. "Fair Class Balancing: Enhancing Model Fairness without Observing Sensitive Attributes". In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. CIKM '20. Virtual Event, Ireland: Association for Computing Machinery, 2020, pp. 1715–1724. ISBN: 9781450368599. DOI: 10.1145/3340531.3411980. URL: https://doi-org.uml.idm.oclc.org/10.1145/3340531.3411980.

[ZB21] Bo Zhao and Hakan Bilen. "Dataset Condensation with Differentiable Siamese Augmentation". In: *ICML* (2021).

[ZB22] Bo Zhao and Hakan Bilen. "Dataset Condensation with Distribution Matching". English. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision 2023 (WACV)*. IEEE/CVF

Winter Conference on Applications of Computer Vision, 2023, WACV 2023 ; Conference date: 03-01-2023 Through 07-01-2023. IEEE, Oct. 2022. URL: https://wacv2023.thecvf.com/.

[ZJW18] Xinyang Zhang, Shouling Ji, and Ting Wang. "Differentially Private Releasing via Deep Generative Model". In: *ArXiv* abs/1801.01594 (2018).

[ZMB21] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. "Dataset Condensation with Gradient Matching". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL: https://openreview.net/forum?id=mSAKhLYLSsl.