PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

UNIVERSITY OF ABOU-BEKR BELKAID – TLEMCEN

Laboratoire de Recherche en Informatique de Tlemcen

FACULTY OF SCIENCES – DEPARTMENT OF COMPUTER SCIENCE

THESIS

Presented for Obtaining the Degree of:

DOCTORATE

Specialty: Artificial Intelligence and its Applications

By

Rafika Benladghem

Theme

Privacy Preserving Deep Learning in Medical Data

Defended in June 2025 at Tlemcen in Front of the Jury Composed of :

Mr Badr Benmammar	Full Professor	University of Tlemcen	President
Mr Fethallah Hadjila	Associate Professor	University of Tlemcen	Supervisor
Mr Mourtada Benazzouz	Associate Professor	University of Tlemcen	Examiner
Mr Nassim Dennouni	Associate Professor	Higher School of Management - Tlemcen	Examiner
Mr Omar Rafik Merad Boudia	Full Professor	University of Oran 1	Examiner
Mr Adam Belloum	Full Professor	University of Amsterdam	Guest

Academic Year: 2024/2025

Abstract

***

Deep Learning (DL) has demonstrated transformative potential in numerous domains, yet its application to sensitive data, particularly in healthcare, is fraught with significant privacy risks. Differentially Private Deep Learning (DPDL) offers a rigorous framework to mitigate these risks by providing formal privacy guarantees. However, DPDL inherently introduces a fundamental Privacy-Utility (P-U) tradeoff, where enhancing privacy often degrades model performance. This dissertation confronts the critical challenge of optimizing this P-U tradeoff to make DPDL practical and effective for real-world deployment. This research presents a comprehensive investigation into methodologies for navigating and improving the P-U balance in DPDL systems. We first establish realistic DPDL performance baselines on medical datasets through meticulous hyperparameter tuning, demonstrating that competitive utility is achievable even under privacy constraints. To systematically manage the P-U compromise, a Pareto-optimal framework is proposed, enabling principled architecture selection and informed decision-making. Recognizing the computational burden of tuning, we further introduce a comparative analysis of efficient Hyperparameter Optimization (HPO) strategies, including a novel application of the Bat Algorithm, to effectively identify high-quality DPDL configurations. Furthermore, this work acknowledges that real-world medical datasets are often imbalanced, which can interact with DPDL mechanisms. We analyze these interactions, considering their implications for performance on underrepresented classes, and explore the potential of data augmentation as a technique to enhance DPDL utility in medical imaging, revealing its privacy-level-dependent efficacy. Collectively, the contributions of this thesis provide a suite of validated tools and insights, ranging from foundational performance assessment and tradeoff navigation via Pareto analysis, to efficient HPO. These advancements aim to bridge the gap between the theoretical promise of DPDL and its practical, high-utility application in sensitive domains, ultimately fostering the development of more trustworthy and effective AI.

Keywords: Differential Privacy, Deep Learning, Privacy-Preserving Machine Learning, Privacy-Utility Tradeoff, Fairness, Class Imbalance, Privacy-Utility-Fairness Nexus, Pareto Optimality, Hyperparameter Optimization, Bat Algorithm, Medical Data Analysis.

Résumé

***

L'Apprentissage Profond a démontré un potentiel transformateur dans de nombreux domaines, mais son application à des données sensibles, notamment dans le secteur de la santé, est semée de risques significatifs pour la vie privée. L'Apprentissage Profond Différentiellement Privé offre un cadre rigoureux pour atténuer ces risques en fournissant des garanties formelles de confidentialité. Cependant, l'Apprentissage Profond Différentiellement Privé introduit intrinsèquement un compromis fondamental entre Vie Privé et Utilité (P-U), où l'amélioration de la confidentialité dégrade souvent les performances du modèle. Cette thèse aborde le défi critique de l'optimisation de ce compromis P-U afin de rendre l'Apprentissage Profond Différentiellement Privé pratique et efficace pour un déploiement en conditions réelles. Cette recherche présente une investigation approfondie des méthodologies permettant de naviguer et d'améliorer l'équilibre P-U dans les systèmes de l'Apprentissage Profond Différentiellement Privé . Nous établissons d'abord des performances de référence réalistes pour l'Apprentissage Profond Différentiellement Privé sur des jeux de données médicales grâce à un réglage méticuleux des hyperparamètres, démontrant qu'une utilité compétitive est réalisable même sous contraintes de confidentialité. Pour gérer systématiquement le compromis P-U, un cadre Pareto-optimal est proposé, permettant une sélection d'architecture fondée sur des principes et une prise de décision éclairée. Reconnaissant la charge de calcul liée au réglage, nous introduisons en outre une analyse comparative de stratégies efficaces d'Optimisation des Hyperparamètres (HPO), incluant une application novatrice de l'Algorithme des Chauves-souris, pour identifier efficacement des configurationsde l'Apprentissage Profond Différentiellement Privé de haute qualité. De plus, ce travail reconnaît que les jeux de données médicales du monde réel sont souvent déséquilibrés, ce qui peut interagir avec les mécanismes du L'Apprentissage Profond Différentiellement Privé . Nous analysons ces interactions, en considérant leurs implications sur les performances des classes sous-représentées, et explorons le potentiel de l'augmentation de données comme technique pour améliorer l'utilité du L'Apprentissage Profond Différentiellement Privé en imagerie médicale, révélant son efficacité dépendante du niveau de confidentialité. Collectivement, les contributions de cette thèse fournissent une série d'outils validés et de connaissances approfondies, allant de l'évaluation fondamentale des performances et de la gestion des compromis par l'analyse de Pareto, jusqu'à l'optimisation efficace des hyperparamètres (HPO). Ces avancées visent à combler le fossé entre la promesse théorique du L'Apprentissage Profond Différentiellement Privé et son application pratique à haute utilité dans des domaines sensibles, favorisant ainsi le développement d'une IA plus digne de confiance et efficace.

Mots-clés : Confidentialité Différentielle, Apprentissage Profond, Apprentissage Automatique Préservant la Confidentialité, Compromis Confidentialité-Utilité, Équité, Déséquilibre de Classes, Nexus Confidentialité-Utilité-Équité, Optimalité de Pareto, Optimisation des Hyperparamètres, Algorithme des Chauves-souris, Analyse de Données Médicales.

ملخص

* * *

أظهر التعلم العميق (DL) إمكانات تحويلية هائلة في مجالات متعددة، غير أن تطبيقه على البيانات الحساسة، لا سيما في مجال الرعاية الصحية، محفوف بمخاطر كبيرة على الخصوصية. يوفر التعلم العميق التفاضلي الخاص (DPDL) إطارًا صارمًا للتخفيف من هذه المخاطر من خلال تقديم ضمانات خصوصية رسمية. ومع ذلك، يُدخل DPDL بطبيعته مقايضة أساسية بين الخصوصية والمنفعة(P-U) ، حيث يؤدي تعزيز الخصوصية غالبًا إلى تدهور أداء النموذج. تواجه هذه الأطروحة التحدي الحاسم المتمثل في تحسين هذه المقايضة بين الخصوصية والمنفعة لجعل DPDL عمليًا وفعالًا للتطبيق في العالم الحقيقي. تقدم هذه الدراسة البحثية استقصاءً شاملاً للمنهجيات الخاصة بتحقيق التوازن بين الخصوصية والمنفعة وتحسينه في أنظمة التعلم العميق التفاضلي الخاص. نقوم أولاً بتحديد مستويات أداء مرجعية واقعية لـ DPDL على مجموعات بيانات طبية من خلال الضبط الدقيق للمعلمات الفائقة، مما يدل على إمكانية تحقيق منفعة تنافسية حتى في ظل قيود الخصوصية. ولإدارة مقايضة الخصوصية والمنفعة بشكل منهجي، يُقترح إطار أمثلية باريتو، مما يتيح اختيارًا مبدئيًا للبنى الهندسية واتخاذ قرارات مستنيرة. وإدراكًا للعبء الحسابي لعملية الضبط، نقدم أيضًا تحليلًا مقارنًا لاستراتيجيات تحسين المعلمات الفائقة (HPO) الفعالة، بما في ذلك تطبيق مبتكر لخوارزمية الخفافيش، لتحديد تكوينات DPDL عالية الجودة بفعالية. علاوة على ذلك، يقر هذا العمل بأن مجموعات البيانات الطبية في العالم الحقيقي غالبًا ما تكون غير متوازنة، مما قد يتفاعل مع آليات .DPDL نقوم بتحليل هذه التفاعلات، مع مراعاة آثارها على أداء الفئات ناقصة التمثيل، ونستكشف إمكانات زيادة البيانات كتقنية لتعزيز منفعة DPDL في التصوير الطبي، مما يكشف عن فعاليتها المعتمدة على مستوى الخصوصية .بشكل عام، تقدم مساهمات هذه الأطروحة مجموعة من الأدوات المعتمدة والرؤى، تتراوح من التقييم الأساسي للأداء وإدارة المفاضلات عبر تحليل باريتو، وصولاً إلى التحسين الفعال للمعلمات الفائقة(HPO) . تهدف هذه التطورات إلى سد الفجوة بين الوعد النظري لـ DPDL وتطبيقه العملي عالي المنفعة في المجالات الحساسة، مما يعزز في نهاية المطاف تطوير ذكاء اصطناعي أكثر جدارة بالثقة وفعالية.

الكلمات المفتاحية :الخصوصية التفاضلية، التعلم العميق، تعلم الآلة مع الحفاظ على الخصوصية، مقايضة الخصوصية والمنفعة، الإنصاف، عدم توازن الفئات، ارتباط الخصوصية والمنفعة والإنصاف، أمثلية باريتو، تحسين المعلمات الفائقة، خوارزمية الخفافيش، تحليل البيانات الطبية.

Acknowledgements

***

My heartfelt gratitude goes to my thesis supervisor, Dr./Mr. Fethallah Hadjila. His encouragement to pursue this PhD, first offered during my internship with him, was instrumental in realizing a cherished ambition. The past four years have been a remarkably stimulating and productive period, an experience greatly enriched by his guidance and support, for which I am deeply thankful. I learned a great deal under his supervision. Mr. Hadjila, *moltes gràcies*!

I would also like to extend my sincere appreciation to Dr./Mr./Prof Belloum Adam. His early guidance and the documentation he provided when I was first embarking on this research topic were very helpful in setting my initial direction.

I also sincerely thank the members of my examination committee: Dr./Mr./Prof. Bader Benmammar, Dr./Mr. Mortada Benazzouz, Dr./Mr. Nassim Denouni, and Dr./Mr./Prof. Omar Rafik Merad Boudia. I am very grateful for their willingness to evaluate this work and for the valuable time they invested in the process. It was an honor to have them serve on the jury.

I dedicate this humble work
To my dearest mother, may the almighty God protect her
To my dearest sister, her children: YACINE, ADLENE, and RANIA, and my Grandparents
To my uncle KHALED BENLADGHEM.
To everyone who shared a path on my journey to this point...

Contents

***

List of Figures

***

List of Tables

***

Chapter 1

General Introduction

***

1.1 Context

Deep Learning (DL) has emerged as a transformative force in artificial intelligence, demonstrating remarkable success across diverse applications, most critically in enhancing medical diagnosis and healthcare analytics [LBH15]. This power stems from the ability of deep neural networks to discern intricate patterns from vast datasets. However, this reliance on extensive data, particularly sensitive patient health information, inherently introduces profound privacy challenges [CLE+19]. The process of training these sophisticated models can lead to the inadvertent memorization and subsequent leakage of personal details, rendering them susceptible to privacy attacks such as membership inference, model inversion, and attribute inference [SSSS17a, NSH19, FJR15a, HAPC17].

The critical need to mitigate these risks is driven not only by ethical considerations but also by increasingly stringent data protection regulations globally, including GDPR in Europe and HIPAA for medical data in the US [BFH+16, Act96]. This has catalyzed the field of Privacy-Preserving Deep Learning (PPDL), which seeks to harness DL's capabilities while safeguarding individual privacy. Among various PPDL approaches, **Differential Privacy (DP)** [DMNS06a] has become the de facto standard, offering a rigorous, mathematically provable framework for privacy protection. This thesis is anchored in the domain of Differentially Private Deep Learning, investigating methods to overcome its inherent challenges, especially within the demanding context of medical data analysis.

1.2 Motivations

The research presented in this thesis is driven by a confluence of critical needs and challenges arising at the intersection of deep learning, data privacy, and real-world applications, particularly in healthcare:

1. **Escalating Privacy Concerns and Regulations:** The rapid digitization of sensitive information, especially personal health data, coupled with high-profile data breaches and increasingly stringent regulations like GDPR, necessitates robust privacy-preserving

techniques. Deep learning models, known for their capacity to memorize training data, pose significant privacy risks (e.g., membership inference, attribute inference, reconstruction attacks) [SSSS17a, FJR15a], demanding effective mitigation strategies like Differential Privacy (DP).

2. **The Perceived Utility Cost of DP:** While DP provides strong theoretical guarantees, a prevalent concern is that the required noise injection inevitably leads to a substantial and often prohibitive loss in model utility [PTS+21a]. This perception hinders the adoption of DPDL, especially in high-stakes fields where performance is critical. There is a need to empirically quantify the *actual* tradeoff under realistic conditions and optimal tuning.

3. **Complexity and Sensitivity of DPDL Tuning:** DPDL introduces additional hyperparameters (clipping norm C, noise multiplier σ) that interact complexly with standard deep learning hyperparameters (learning rate η, batch size B, architecture choices) and the privacy budget ϵ. Finding optimal configurations requires careful tuning, yet the sensitivity of the P-U balance to these parameters is often poorly understood for specific, non-benchmark datasets.

4. **Computational Cost of Hyperparameter Optimization:** The process of finding optimal DPDL configurations can be extremely computationally expensive. Exhaustive methods like grid search quickly become intractable as the number of hyperparameters, model complexity, or dataset size increases, necessitating more efficient HPO strategies.

5. **Need for Principled Tradeoff Management:** Selecting an appropriate operating point on the P-U spectrum is not straightforward. Practitioners need systematic frameworks, like Pareto analysis, to visualize the optimal achievable tradeoffs and make informed decisions based on their specific requirements and constraints, rather than relying on ad-hoc tuning.

6. **Prevalence and Impact of Data Imbalance:** Real-world datasets, especially in medicine, are rarely balanced. Class imbalance can severely bias standard models against minority groups [Cha04]. The interaction between DPDL mechanisms and data imbalance, and its consequent impact on fairness, is a critical but relatively underexplored area.

7. **The Imperative of Fairness:** Beyond aggregate utility, ensuring that ML models perform equitably across different demographic groups or patient subpopulations is an ethical and often regulatory requirement [MNMG21]. The potential for DP to exacerbate existing biases in imbalanced data raises significant fairness concerns that must be addressed for responsible deployment.

8. **Demand for Generalizable Solutions:** Effective DPDL approaches should be applicable across various data modalities encountered in real-world settings, including both structured tabular data and complex medical imaging, requiring validation and potentially tailored strategies for different data types.

Addressing these interconnected motivations is crucial for unlocking the full potential of DPDL and enabling its trustworthy application in sensitive domains.

1.3 Issue and Motivation

While DP offers a robust theoretical foundation for privacy, its practical application in deep learning, primarily through **Differentially Private Stochastic Gradient Descent (DP-SGD)** [ACG+16a], uncovers a series of interconnected challenges that motivate this research.

1.3.1 The Need for Privacy: Differential Privacy in Deep Learning

Differential Privacy ensures that the output of an algorithm changes negligibly upon the addition or removal of any single individual's data, thus providing a strong shield against linkage attacks and individual re-identification [DR14a]. In DP-SGD, this guarantee is achieved by two core modifications to the standard training process: bounding the influence of individual data points by **clipping per-sample gradients** to a predefined threshold, and then **adding calibrated noise** to the aggregated gradients before model updates. The strength of this privacy protection is quantified by a privacy budget (ϵ, δ), where lower ϵ values signify stronger privacy. The careful management of this budget throughout training is of paramount importance.

1.3.2 The Fundamental Privacy-Utility Tradeoff

The mechanisms that confer privacy in DPDL—gradient clipping and noise injection—inevitably introduce a core tension: the **Privacy-Utility (P-U) tradeoff**. Clipping can discard valuable gradient information, while noise can obscure the true learning signal. Consequently, enhancing privacy (i.e., achieving a lower ϵ) typically results in a degradation of model utility, such as reduced accuracy or slower convergence (conceptually illustrated in Figure 1.1). A central goal in DPDL research, and a primary driver for this thesis, is to develop methodologies that mitigate this tradeoff, enabling the development of models that are both highly private and highly performant.

1.3.3 Real-World Complications: Hyperparameter Tuning, Data Imbalance, and Fairness

The P-U tradeoff is not a static challenge; its severity is amplified by practical considerations, particularly when deploying DPDL in sensitive, real-world settings like healthcare.

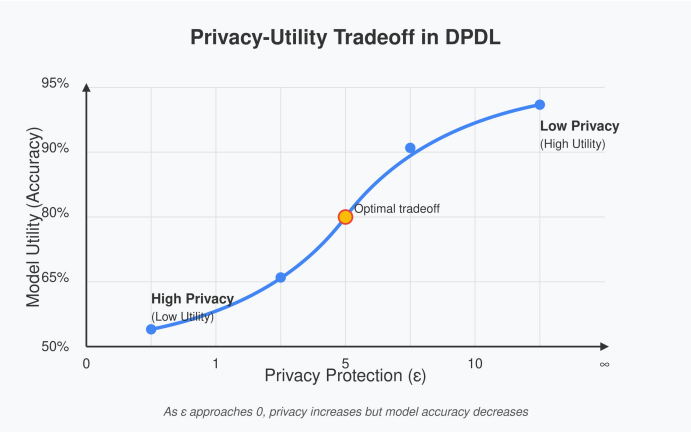


Figure 1.1: Conceptual illustration of the Privacy-Utility (P-U) tradeoff in DPDL. Stronger privacy typically incurs a utility cost.

Hyperparameter Complexity: DPDL model performance is acutely sensitive to a multitude of interacting hyperparameters. Beyond standard DL parameters (learning rate, batch size), DP-SGD introduces critical privacy-specific parameters (clipping norm C, noise multiplier σ). Finding an optimal configuration that balances privacy and utility is a computationally intensive task, necessitating efficient **Hyperparameter Optimization (HPO)** strategies tailored for the unique demands of DPDL [PNMM21, PTS+21b].

Medical Data Characteristics: Medical datasets often present inherent difficulties that exacerbate DPDL challenges:

- *Limited Data and High Dimensionality:* Scarcity of data can make models more susceptible to the performance degradation caused by DP noise.
- *Class Imbalance:* A pervasive issue in medical diagnostics is **class imbalance**, where datasets contain far fewer examples of abnormal or disease-positive cases [HG09, JK19]. This imbalance not only complicates standard model training but also poses a significant threat under DPDL. The noise introduced for privacy can easily overwhelm the weaker signals from minority classes, leading to poor predictive performance for these critical groups.
- *Fairness Implications:* The disparate impact of DPDL on imbalanced data directly translates to concerns about model **fairness** [MNMG21]. If a privacy-preserving model performs significantly worse for certain demographic groups or underrepresented patient populations (often corresponding to minority classes), its clinical utility and ethical permissibility are severely compromised. This extends the P-U tradeoff to a more complex **Privacy-Utility-Fairness (P-U-F) trilemma**.

Addressing these interconnected issues—efficiently optimizing the P-U balance while being cognizant of fairness implications on imbalanced medical data—forms the crux of the thesis problem.

1.3.4 Motivation and Research Gap

This thesis is driven by the urgent need to bridge the chasm between the theoretical elegance of Differential Privacy and its effective, robust, and fair deployment in high-stakes medical applications. While the P-U tradeoff is acknowledged, and fairness in AI is a growing concern, there is a deficit of comprehensive research providing *practical, integrated solutions* for navigating the P-U-F trilemma within DPDL. Specifically, existing work often addresses these aspects—privacy, utility optimization, fairness, and data imbalance—in isolation.

The research gap lies in the need for:

- Systematic methodologies to rigorously assess and benchmark the true potential of DPDL on complex medical data when properly tuned.
- Principled frameworks for navigating P-U tradeoffs that extend to architecture selection and comparison against established benchmarks.
- Efficient and effective HPO techniques specifically designed or adapted for the unique hyperparameter landscape of DPDL, moving beyond ad-hoc tuning.
- A deeper understanding and empirical quantification of how DPDL mechanisms interact with data imbalance to impact fairness, along with initial explorations into mitigating these adverse effects.
- Investigation into synergistic techniques, like data augmentation, that might enhance DPDL performance in data-constrained medical imaging scenarios.

This thesis seeks to fill these gaps by providing a holistic approach to improving the practicality and trustworthiness of DPDL systems in healthcare. The core motivation is to develop and validate techniques that enable the creation of DPDL models that are not only provably private but also demonstrably useful and equitable.

1.4 Research Questions and Contributions

To address the identified gaps and challenges, this thesis seeks to answer the following key research questions:

1. What is the true achievable utility of DPDL on diverse medical datasets when subjected to rigorous hyperparameter tuning, and how sensitive is this utility to DPDL-specific parameters?
2. How can Pareto optimization principles be leveraged to systematically navigate the P-U tradeoff and guide DPDL architecture selection effectively?
3. Which HPO strategies (including novel metaheuristics like the Bat Algorithm) are most efficient and effective for optimizing the P-U balance in DPDL across different data modalities?

4. How significantly does data imbalance impact the P-U-F nexus in DPDL, and what are the fairness implications for minority classes under varying privacy constraints?

5. Can data augmentation techniques serve as a viable strategy to enhance the utility of DPDL models in medical image analysis, and how does this interaction depend on the privacy level?

Addressing these questions, this thesis makes five primary methodological contributions:

1. **Empirical Baselining and Sensitivity Analysis of Tuned DPDL (C1):** We conduct a rigorous empirical study to establish realistic utility benchmarks for DPDL on medical data with meticulous tuning, analyzing P-U-F sensitivity and revealing surprising fairness benefits in imbalanced contexts.

2. **Pareto-Optimal Framework for P-U Tradeoff and Architecture Selection (C2):** We develop and validate a Pareto optimization framework that systematically maps P-U frontiers, enabling principled DPDL architecture comparison and informed decision-making.

3. **Efficient Hyperparameter Optimization for DPDL (C3):** We introduce and evaluate a comparative framework of HPO strategies, including a novel adaptation of the Bat Algorithm, to efficiently find optimal P-U configurations for DPDL.

4. **Analysis of the P-U-F Nexus under Data Imbalance (C4):** We systematically investigate and quantify the three-way interaction between privacy, utility, and fairness when applying DPDL to imbalanced medical datasets, exposing fairness vulnerabilities.

5. **Investigating DPDL Enhancement via Data Augmentation in Medical Imaging (C5):** We explore the potential of data augmentation (specifically, image rotation) to improve the P-U tradeoff in DPDL for medical image analysis, assessing its privacy-level-dependent efficacy.

These contributions, detailed in subsequent chapters, collectively aim to enhance the practical utility, fairness, and trustworthiness of DPDL systems in sensitive applications.

1.5 Thesis Organization

The remainder of this dissertation is structured to systematically address the research objectives and present the contributions:

- **Chapter 1: Introduction (This Chapter).** Establishes the research context, delineates the core challenges including privacy risks in deep learning, the critical Privacy-Utility (P-U) tradeoff, and the complexities introduced by fairness considerations under data imbalance. It further articulates the motivation, identifies the research gap, summarizes the primary contributions of this work, and outlines the overall structure of the thesis.

- **Chapter 2: Theoretical Foundations.** This chapter lays the essential groundwork, introducing fundamental concepts of deep learning, principles of data privacy, the formal definitions and mechanisms of Differential Privacy (DP), and the specific methodologies of Differentially Private Deep Learning (DPDL), with a particular focus on DP-SGD.
- **Chapter 3: State of the Art in Differentially Private Deep Learning.** A comprehensive review of existing literature is presented, covering DPDL algorithms, established approaches to P-U tradeoff analysis, and techniques for mitigating class imbalance. This chapter positions the contributions of this thesis within the broader academic and research landscape.
- **Chapter 4: Fine-Tuning Differentially Private Deep Learning Techniques.** This chapter delves into the intricacies of optimizing DPDL parameters. It details preliminary investigations and specific techniques explored for enhancing DPDL performance, potentially serving as foundational work or partial fulfillment of the first research contribution concerning achievable utility and sensitivity.
- **Chapter 5: Proposed Methodologies for Balancing Privacy, Utility.** The core methodological innovations of this thesis are formally presented in this chapter. It elaborates on the conceptual frameworks, analytical strategies, and theoretical underpinnings for each of the primary contributions designed to address the P-U trade-off in DPDL.
- **Chapter 6: Implementation and Experimental Evaluation.** This chapter transitions from theory to practice, detailing the datasets, model architectures, experimental setups, and evaluation metrics (for utility, privacy) employed. Crucially, it presents the empirical results that validate the proposed approaches and quantify their impact.
- **Chapter 7: Conclusion and Future Perspectives.** The dissertation concludes with this chapter, which synthesizes the key findings, recapitulates the significance of the contributions, acknowledges any limitations encountered during the research, and proposes promising avenues for future investigation in the domain of private, useful, and fair deep learning.

1.6 List of Published Work

The research presented in this dissertation has contributed to the following peer-reviewed publications:

1.6.1 International Journal Article

- BENLADGHAM, Rafika, HADJILA, Fethallah, et BELLOUM, Adam. Efficient Privacy-Utility Optimization for Differentially Private Deep Learning: Application to Medical

Diagnosis. *International Journal of Electrical and Computer Engineering Systems*, 2025, vol. 16, no 5, p. 377-395.

1.6.2 International Conference Proceedings

- BENLADGHAM, Rafika, HADJILA, Fethallah, et BELLOUM, Adam. A Pareto-Optimal Privacy-Accuracy Settlement For Differentially Private Image Classification. In: *2023 5th International Conference on Pattern Analysis and Intelligent Systems (PAIS)*. IEEE, 2023. p. 1-7.
- BENLADGHAM, Rafika, HADJILA, Fethallah, et BELLOUM, Adam. Exploring Hyper-Parameter Sensitivity for Improved Privacy and Utility in Differentially Private Models on Real-World Medical Data. *AFROS 2024 International Conference*, Tlemcen, Algeria. (To appear June 2025).

1.6.3 In National Conference Proceedings

BENLADGHEM, RAFIKA, HADJILA, FETALLAH, BELLOUM, ADAM, , MERZOUG, MOHAMMED. Analyzing the Influence of Medical Imbalanced Data on Performance and Fairness in Differentially Private Deep Learning. NCAIA'2023, 2023, p. 30.

Chapter 2

Theorical Foundations on Privacy Preserving Deep Learning

***

2.1 Introduction

Deep learning (DL) has transformed numerous domains by enabling machines to learn complex patterns from large datasets. However, reliance on sensitive data introduces significant privacy risks, as models can inadvertently expose private information. This chapter lays the theoretical groundwork for understanding PPDL. We begin by briefly revisiting the fundamentals of deep learning, focusing on aspects pertinent to privacy vulnerabilities. Subsequently, we dive into the primary types of privacy attacks targeting DL models, illustrating the ways in which private information can be compromised. Following this, we provide a comprehensive overview of the major categories of privacy-preserving techniques currently employed or actively researched in the context of DL, including cryptographic methods, federated learning, and differential privacy. Finally, we offer a comparative analysis of these approaches, highlighting their respective strengths, weaknesses, and trade-offs. The objective is to provide a solid theoretical foundation upon which the subsequent chapters will be based.

2.2 Deep Learning Foundations

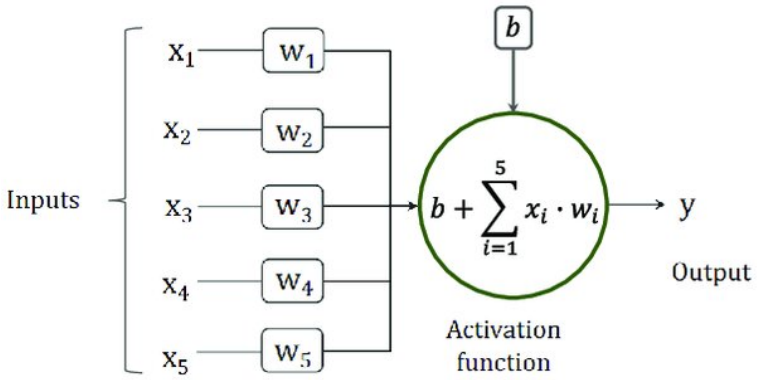
A foundational understanding of the principles of deep learning (DL) is essential to appreciate the associated privacy risks and the mechanisms developed to mitigate them. DL models, particularly deep neural networks, are complex systems trained on data, and their inner workings dictate how privacy vulnerabilities manifest.

This section outlines the core principles of deep learning, focusing on its architectures, training mechanisms, applications, and inherent privacy challenges.

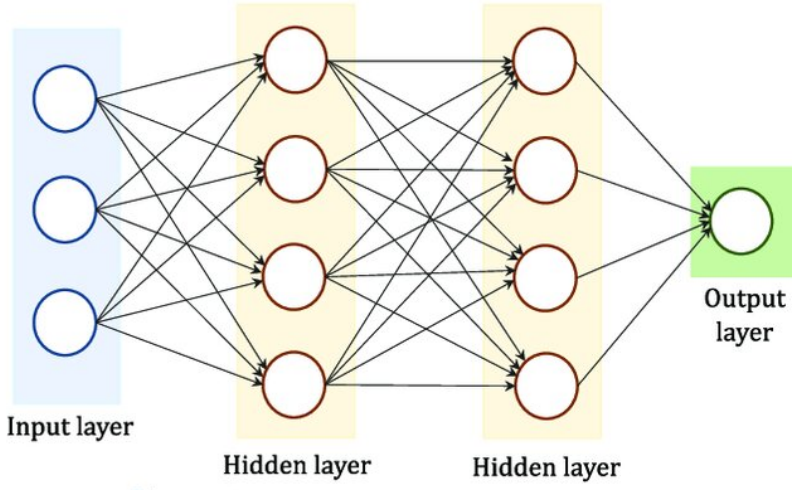
2.2.1 Neural Networks architectures and principals

Deep learning has revolutionized machine learning by enabling models to automatically discover representations needed for detection or classification tasks from raw data [LeC15].

At their core, neural networks are computational models inspired by the structure and function of biological neural networks. They consist of interconnected nodes, or neurons, organized in layers. The most basic form is the Multilayer Perceptron (MLP), comprising an input layer, one or more hidden layers, and an output layer. Each connection between neurons has an associated weight, which is adjusted during training. Neurons typically apply a non-linear activation function (e.g., ReLU, sigmoid, tanh) to their weighted inputs, enabling the network to learn complex, non-linear relationships within the data [GBC16]. Figure 2.1 illustrates a basic feedforward neural network.



(a) A perceptron with $N = 5$ inputs x_i and one output y.



(b) Feed-forward fully connected neural network.

Figure 2.1: Schematic representation of an artificial neural network (ANN). [CMB+21]

More sophisticated architectures have been developed for specific data types: Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Generative Adversarial Networks (GANs), Autoencoders, Transformer Networks.

Convolutional Neural Networks (CNNs)

The term "convolutional neural network" highlights the network's reliance on a mathematical operation known as convolution. Through convolution, the network processes input data in small, overlapping segments, allowing it to learn significant features. This capability enables CNNs to recognize patterns even if those patterns weren't explicitly shown during training. Consequently, CNNs often demonstrate superior performance compared to standard feedforward networks, especially for tasks involving object identification and categorization. This section aims to explain the fundamental technical principles of deep learning using CNNs.[RHZ+22]
A CNN's architecture is built from several key elements, including layers dedicated to convolution, pooling, and full connection (where all neurons connect to the next layer). A frequent design pattern involves multiple iterations of convolution and pooling layers, culminating in one or several fully connected layers. The transformation of input data into final output as it passes through these stages is called forward propagation (see Figure2.2).[RHZ+22]

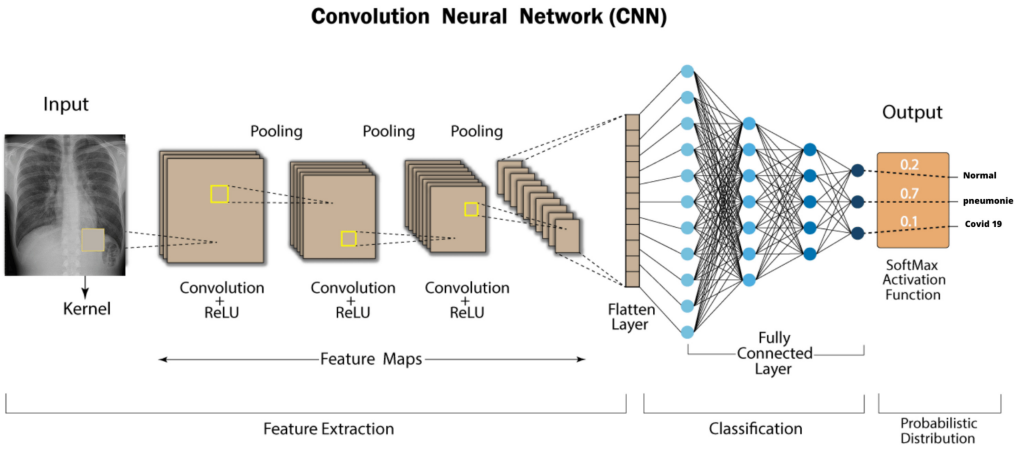


Figure 2.2: Convolutional Neural Network(CNN).
[RHZ+22]

1. **The Feature Extraction Phase** Feature extraction stands out as a critically important stage within a CNN (refer to Figure2.3). Its purpose is to detect and isolate key characteristics from the input, such as an image. This step is essential because the extracted features form the basis for subsequent analysis, like identifying abnormalities. The convolution layer serves as the core engine for feature extraction within the CNN framework. It typically integrates linear processes (the convolution operation itself) and non-linear elements (like activation functions).[RHZ+22]

2. **The Role of Fully Connected Layers** While feature extraction, enabled by the mathematical properties of convolution, is fundamental, the process continues further. Extracted features usually undergo pooling (summarization) and downsampling (size

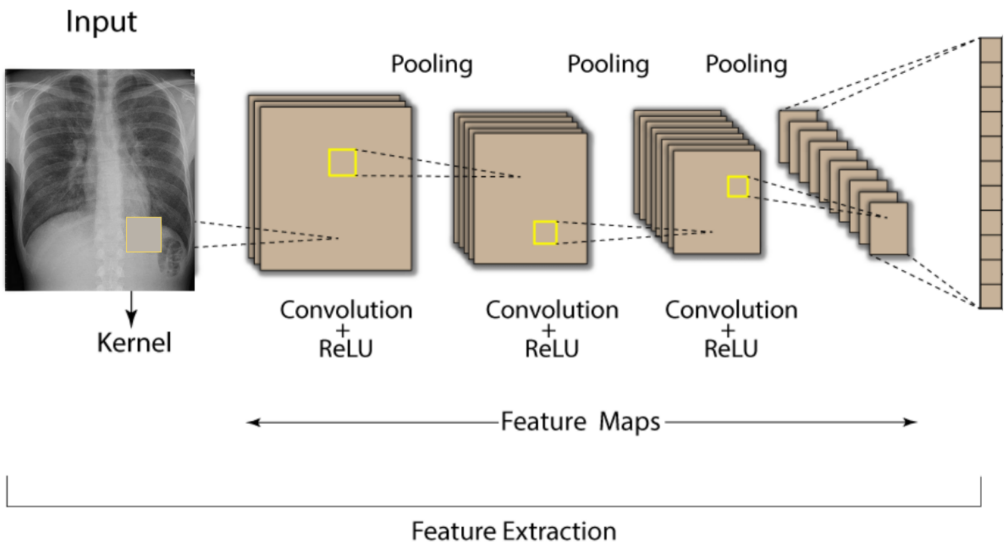


Figure 2.3: The Feature Extraction Phase.
[RHZ+22]

reduction) before being fed into the network's final stages. These final stages often involve fully connected layers that map the processed features to the ultimate outputs. It's worth noting that the number of output units in a fully connected layer often corresponds directly to the number of classes or clusters the network is designed to identify (as depicted in Figure2.4 for clustering). This design choice helps consolidate information relevant to a specific task (like clustering) within a single layer, improving interpretability. To produce the final results, a non-linear function is applied after these fully connected layers.[RHZ+22]

3. **Activation Functions and Probabilistic Outputs** The last fully connected layer can employ different activation functions. A linear function is a standard choice, but other functions might be more effective for particular tasks. For instance, when dealing with classification across multiple categories (see Figure 2.6), the softmax function is often preferred. Softmax normalizes the output values, transforming them into probabilities for each target class. These probabilities range from 0 to 1 and sum up to 1 across all classes. Other common selections for the final layer's activation include linear or sigmoid functions.[RHZ+22]

Recurrent Neural Networks (RNNs)

Designed for sequential data, like text or time series. RNNs possess internal memory through recurrent connections, allowing them to process sequences of arbitrary length. Variants like Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs) address challenges like vanishing gradients, enabling the learning of long-range dependencies [HS97].

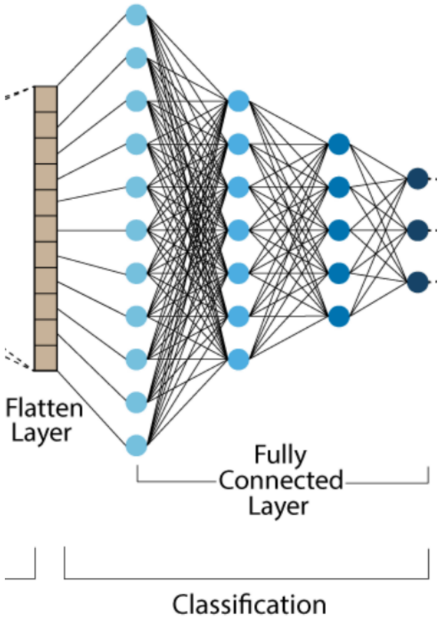


Figure 2.4: Fully Connected Layer.
[RHZ+22]

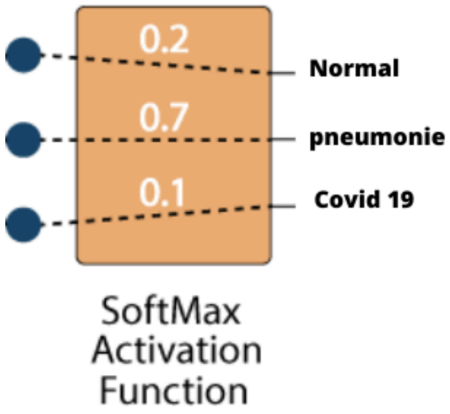


Figure 2.5: Activation Functions and Probabilistic Outputs.
[RHZ+22]

Transformer Networks

Transformers have become a cornerstone of Natural Language Processing (NLP), leveraging self-attention mechanisms to evaluate the importance of various elements within an input sequence. This approach enables parallel computation and adeptly captures long-range relationships between tokens. [VSP+17] The architecture of the transformer model, as depicted in Figure 2.7, illustrates the encoder and decoder components. The left side represents the encoder, utilizing stacked self-attention layers alongside point-wise fully connected layers, while the right side showcases the decoder, operating similarly to process and generate output sequences.

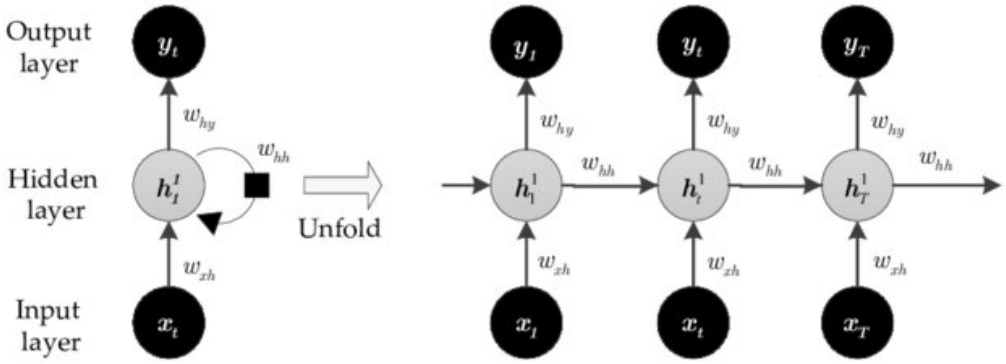


Figure 2.6: Visualization of a Recurrent Neural Network (RNN) structure. Network layers are depicted as circles, weighted connections between layers are shown as solid lines, and the arrows indicate the flow of information during forward propagation.
[HC19]

2.2.2 Training Process and Optimization

Training deep neural networks fundamentally involves finding the optimal set of parameters (weights and biases) θ that minimize a predefined loss function $\mathcal{L}$. This loss function quantifies the discrepancy between the network's predictions and the ground truth labels for the given training data [GBC16]. The optimization process is typically achieved using iterative, gradient-based methods.

The standard training procedure, often executed in mini-batches, follows these core steps, as illustrated in Figure 2.8:

1. **Forward Pass:** Input data $\mathbf{x}$ is fed through the network layers, applying transformations and activation functions, to produce an output prediction $\hat{\mathbf{y}} = f(\mathbf{x}; \theta)$.

2. **Loss Calculation:** The prediction $\hat{\mathbf{y}}$ is compared to the true target $\mathbf{y}$ using a suitable loss function $\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y})$. Common choices include cross-entropy for classification tasks and mean squared error (MSE) for regression tasks.

3. **Backward Pass (Backpropagation):** The gradient of the loss function with respect to each network parameter $\nabla_\theta \mathcal{L}$ is computed efficiently using the chain rule. This process propagates the error signal backward through the network [RHW86].

4. **Parameter Update:** The network parameters θ are adjusted in the direction that reduces the loss, guided by the computed gradients and a chosen optimization algorithm. A common update rule is $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$, where η is the learning rate.

While basic Stochastic Gradient Descent (SGD) performs the parameter update using the gradient from a single instance or a mini-batch, several advanced optimization algorithms have been developed to improve convergence speed, stability, and performance. Key optimizers include (summarized in Table 2.1):

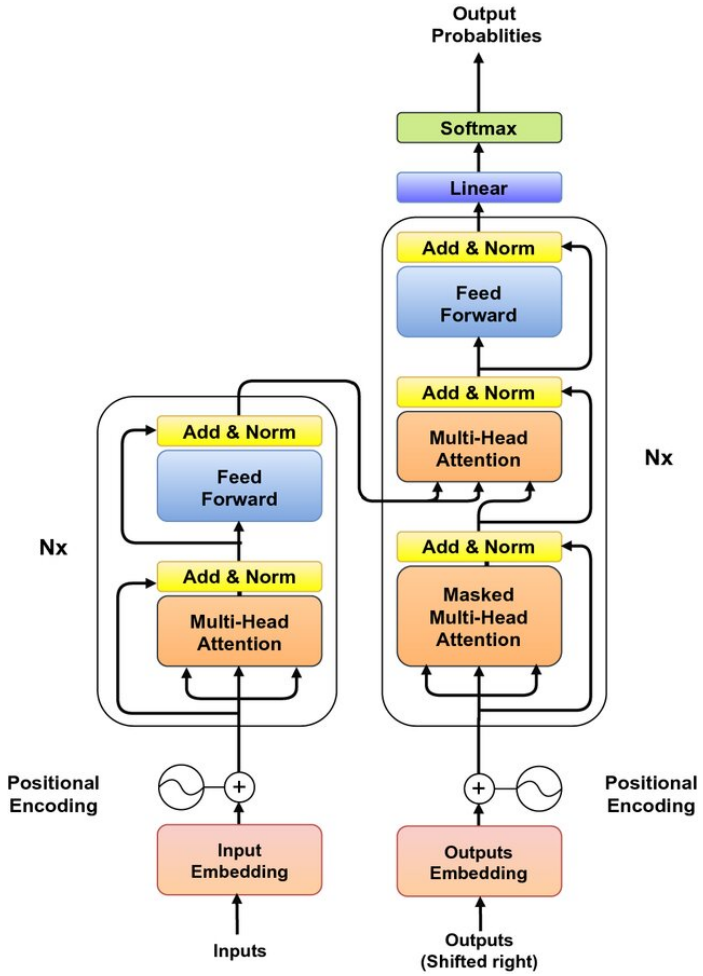


Figure 2.7: Overview of Transformer Architecture: Encoder and Decoder Components.[ASA22]

- **SGD with Momentum:** Incorporates a moving average of past gradients to accelerate convergence along relevant directions and dampen oscillations [SMDH13].
- **RMSProp:** Adapts the learning rate for each parameter individually based on a moving average of the squared magnitudes of recent gradients [HSS12].
- **Adam (Adaptive Moment Estimation):** Combines the ideas of momentum and RMSProp, storing moving averages of both the gradients (first moment) and their squared values (second moment) to compute adaptive learning rates for each parameter [KB15].

Furthermore, to prevent the model from overfitting to the training data and improve its ability to generalize to unseen data, various regularization techniques are commonly employed:

- **L1/L2 Regularization:** Adds a penalty term to the loss function proportional to the L1 norm ($\sum |\theta_i|$) or L2 norm ($\sum \theta_i^2$) of the parameters, encouraging smaller weights.

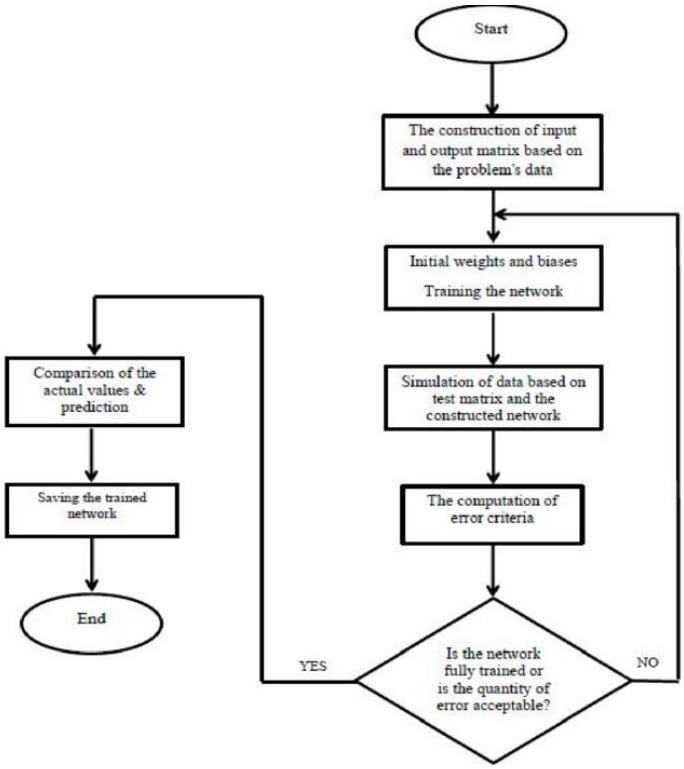


Figure 2.8: The iterative training loop for a deep neural network. Data flows forward to generate predictions and calculate loss, while gradients flow backward to update parameters. [HKP13]

Table 2.1: Comparison of common gradient-based optimization algorithms.

Optimizer	Momentum	Adaptive Learning Rate	Reference
SGD	Optional	No	[Bot10]
Momentum	Yes	No	[SMDH13]
RMSProp	No	Yes (based on g^2)	[HSS12]
Adam	Yes	Yes (based on g, g^2)	[KB15]

g denotes gradient; g^2 denotes squared gradient.

- **Dropout:** During training, randomly sets a fraction of neuron activations to zero at each update step. This prevents complex co-adaptations between neurons and forces the network to learn more robust features [SHK+14].

- **Batch Normalization:** Normalizes the inputs to a layer for each mini-batch. This stabilizes the learning process, often allows for higher learning rates, and can act as a regularizer [IS15].

The interplay between the chosen architecture, loss function, optimizer, and regularization techniques is crucial for successfully training effective deep learning models.

2.2.3 Commun Applications and use cases

The versatility of DL has led to its adoption in numerous domains, often involving sensitive data:

- **Healthcare:** Medical image analysis (e.g., detecting tumors), disease prediction from electronic health records (EHRs), drug discovery [ERR+19]. Data involved is highly personal and subject to strict privacy regulations.
- **Finance:** Fraud detection, credit scoring, algorithmic trading, customer relationship management [HPW17]. Financial data is confidential and requires robust security.
- **Natural Language Processing:** Machine translation, sentiment analysis, chatbots, text generation [DCLT18]. Text data can contain personal opinions, identifiers, or confidential information.
- **Computer Vision:** Facial recognition, object detection in surveillance feeds, autonomous driving systems [KSH12]. Image and video data frequently capture individuals and their surroundings.
- **Recommender Systems:** Personalized recommendations for products, movies, or news articles [CAS16]. User preferences and behavior data drive these systems and can reveal sensitive personal habits.

In all these applications, the potential for DL models to learn from and potentially expose sensitive information underscores the critical need for privacy considerations.

2.2.4 Privacy Concerns In DL

Deep learning systems raise several fundamental privacy concerns:

1. **Data Collection and Storage:** Deep learning models typically require large volumes of data, encouraging expansive data collection practices. The centralized storage of this data creates potential single points of failure for privacy breaches (Narayanan Shmatikov, 2010).
2. **Unintended Memorization:** Neural networks can unintentionally memorize specific training examples, especially outliers or unique instances, potentially exposing this information during inference (Carlini et al., 2019). This is particularly concerning for language models trained on sensitive text data.
3. **Inference from Predictions:** Model outputs may reveal sensitive attributes about input data beyond what is necessary for the task, enabling attribute inference attacks (Gong Liu, 2016).
4. **Transfer and Federated Learning Risks:** Approaches that share model updates rather than raw data still leak information through these updates (Melis et al., 2019).

5. **Black-box Access Vulnerabilities:** Even when only allowing query access to models (without revealing architecture or parameters), adversaries can extract sensitive information through carefully crafted inputs (Shokri et al., 2017).

6. **Explainability vs. Privacy Tension:** Techniques that make models more interpretable by revealing feature importance can also expose more information about training data (Shokri et al., 2021).

7. **Model Stealing:** Adversaries may extract proprietary models through API access, potentially bypassing privacy-preserving measures implemented by the original model provider (Tramèr et al., 2016).

These privacy concerns have spurred significant research into privacy-preserving deep learning techniques, which we will explore in subsequent sections.

2.3 Transfer Learning

While Deep Learning (DL) models possess remarkable capabilities, training them effectively often requires substantial amounts of labeled data and significant computational resources (time, energy, hardware) [TGL+20]. This can be a major bottleneck, especially in domains where large labeled datasets are scarce or expensive to obtain, such as specialized medical imaging, or when computational budgets are constrained. **Transfer Learning (TL)** offers a compelling and widely adopted strategy to mitigate these challenges [PNS20, ZQD+20].

The core idea behind TL is to leverage knowledge gained from solving one problem (the *source task* using a *source dataset*) and apply it to a different but related problem (the *target task* using a *target dataset*). In the context of deep learning, this typically involves reusing parts of a neural network that has already been trained, usually on a large-scale benchmark dataset.

2.3.1 Motivation and Advantages

The primary motivations for employing Transfer Learning include:

- **Reduced Data Requirement:** TL can significantly improve performance on target tasks where labeled data is limited, by leveraging the general features learned from the large source dataset.

- **Improved Performance:** Pre-trained models often provide a better initialization point than random weights, potentially leading to higher final performance (e.g., accuracy, F1-score) on the target task.

- **Reduced Training Time and Computational Cost:** This is a key advantage highlighted in the context of this thesis. By reusing pre-trained layers (often freezing the initial ones that capture general features), only a portion of the network needs to be trained or fine-tuned on the target data. This dramatically reduces the number of

parameters to update and the overall training time, saving valuable computational resources.

- **Leveraging State-of-the-Art Architectures:** TL allows practitioners to easily utilize complex, high-performing architectures (e.g., ResNets [TAL16], Transformers [VSP+17]) pre-trained by experts on massive datasets (e.g., ImageNet [DDS+09], large text corpora), without needing the resources to train them from scratch.

2.3.2 Typical Workflow

A common workflow for transfer learning with deep neural networks involves (conceptually illustrated in Figure 2.9):

1. **Pre-training:** Obtain a model pre-trained on a large source dataset relevant to the target domain (e.g., an ImageNet-pretrained ResNet for a medical image classification task). These models have typically learned hierarchical features, from simple edges and textures in early layers to more complex object parts in deeper layers.
2. **Model Adaptation:** Adapt the pre-trained model for the target task. This usually involves:
    - Replacing the final layer(s) (e.g., the classifier head) of the pre-trained model with new layers suitable for the target task's output (e.g., different number of classes).
    - Optionally modifying intermediate layers if needed.
3. **Fine-tuning:** Train the adapted model on the target dataset. Common strategies include:
    - *Feature Extraction:* Freeze the weights of the early, pre-trained layers (treating them as fixed feature extractors) and only train the newly added final layers. This is computationally very cheap.
    - *Full Fine-tuning:* Initialize with pre-trained weights but allow all layers (or a significant portion of later layers) to be updated during training on the target dataset, typically using a lower learning rate than training from scratch to avoid disrupting the learned features too much.

2.3.3 Relevance to This Thesis

Transfer Learning is employed in this thesis, particularly in the experimental evaluations involving deep convolutional neural networks (CNNs) for medical image classification tasks (e.g., using ResNet architectures on MedMNIST datasets as detailed in Chapter 6). Its application offers significant practical advantages:

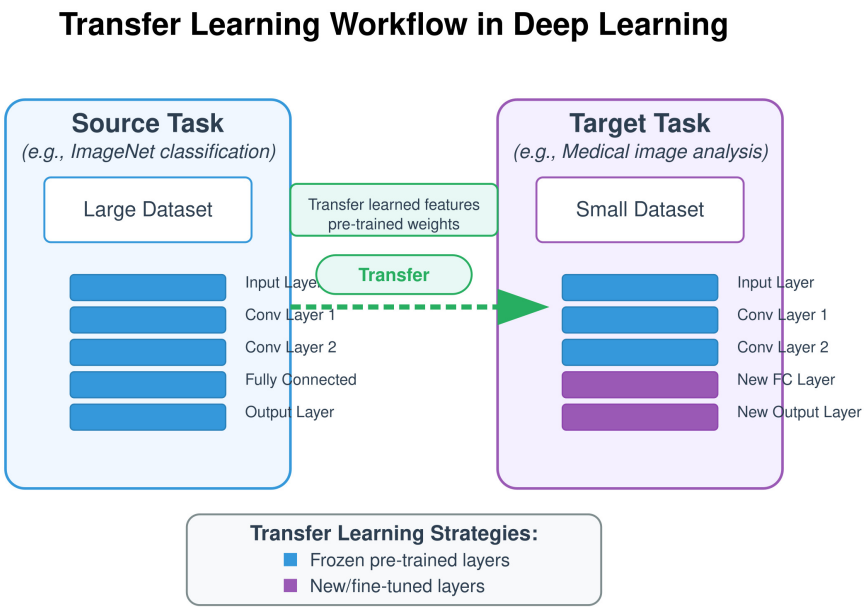


Figure 2.9: Conceptual illustration of the Transfer Learning workflow in Deep Learning. Knowledge (learned features/weights) from a source task model is transferred and adapted for a target task.

- It allows us to leverage powerful, state-of-the-art vision models pre-trained on ImageNet, which have already learned robust visual feature representations applicable to medical images.
- Crucially, it **substantially reduces the computational resources and time** required for training these complex models compared to training them from scratch. This is particularly beneficial given the inherent computational overhead introduced by Differentially Private training mechanisms (as discussed in Section 3.7.2). By reducing the baseline training cost, TL makes the exploration of DPDL configurations and hyperparameter optimization more feasible within practical resource constraints.

2.4 Data Augmentation in Deep Learning

Deep Learning (DL) models, particularly deep neural networks, are renowned for their ability to learn complex patterns and achieve state-of-the-art performance across various domains. However, this prowess is often contingent upon the availability of large, diverse and well-annotated data sets [GBC16]. Acquiring such datasets can be expensive, time-consuming, and in some fields, such as the medical field, fraught with privacy and logistical challenges. **Data Augmentation (DA)** has emerged as a powerful and widely adopted set of techniques to artificially expand the size and diversity of training datasets, thereby mitigating the risks of overfitting, improving model generalization, and often leading to significant performance gains without the need to collect new data [SK19].

2.4.1 The Importance of Data Augmentation

The primary motivation behind data augmentation is to expose the DL model to a wider variety of input variations during training. This helps the model learn to be invariant to irrelevant transformations and focus on the underlying, salient features of the data. Key benefits include:

- **Reduced Overfitting:** By increasing the effective size of the training set, DA makes it harder for the model to simply memorize the training examples. Instead, it encourages the learning of more robust and generalizable representations [KSH12].
- **Improved Generalization:** Models trained with augmented data often perform better on unseen test data because they have learned to cope with a broader range of data variations that might be encountered in real-world scenarios.
- **Cost-Effectiveness:** DA is generally much cheaper and faster than collecting and labeling new data, making it an indispensable tool when data acquisition is a bottleneck.
- **Enhanced Model Robustness:** Augmentation can make models more resilient to noise, occlusions, or minor changes in input conditions.

2.4.2 Data Augmentation in Medical Deep Learning: A Critical Need

While beneficial across many DL applications, data augmentation takes on *critical importance* in the medical domain due to several inherent characteristics and challenges:

- **Data Scarcity and Privacy Constraints:** Medical datasets are often limited in size due to patient privacy regulations (e.g., GDPR [BFH+16], HIPAA [Act96]), the cost of data collection and annotation by medical experts, and the rarity of certain diseases [CCV+17]. DA provides a crucial mechanism to maximize the utility of available sensitive data.
- **Class Imbalance:** Medical datasets frequently suffer from significant class imbalance, where pathological (abnormal) samples are much rarer than normal ones [JK19]. DA can be strategically applied to oversample the minority class(es), helping the model learn their distinguishing features more effectively [CBHK02].
- **High Variability in Medical Data:** Medical images, for instance, can exhibit considerable variability due to differences in acquisition protocols, scanner types, patient positioning, anatomical differences, and the diverse manifestations of diseases [CMV+21]. DA can help train models that are robust to these variations, leading to more reliable diagnostic tools. For example, a model trained to detect lung nodules should be invariant to slight rotations or translations of the CT scan.

- **Regulatory Considerations:** Obtaining regulatory approval for AI-driven medical devices can be a lengthy process. DA allows researchers and developers to make the most of existing, approved datasets by synthetically increasing their diversity.
- **Need for High Reliability:** Medical applications demand high reliability and robustness. Models must perform well across diverse patient populations and imaging conditions. DA is a key strategy to build such resilient models.

Therefore, effective data augmentation is often not just an enhancement but a necessity for building high-performing and generalizable deep learning models in medical field.

2.4.3 Common Data Augmentation Techniques

A wide array of DA techniques exists, broadly classifiable into several categories. The choice of technique often depends on the data modality (e.g., images, text, time series) and the specific task. For image data, particularly medical images, common techniques include:

1. **Geometric Transformations:** These modify the spatial properties of an image without altering its pixel content significantly in terms of underlying pathology.

    - **Rotation:** Rotating the image by a certain angle (e.g., $\pm 10^{\circ}$, 90°). Care must be taken in medical imaging not to create anatomically implausible orientations (e.g., for X-rays where left/right orientation is critical) unless the model is intended to be invariant to such severe rotations.
    - **Translation (Shifting):** Shifting the image horizontally or vertically by a small number of pixels.
    - **Scaling (Zooming):** Resizing the image, either zooming in or out.
    - **Flipping:** Mirroring the image horizontally or vertically. Horizontal flipping is common, but vertical flipping might be less appropriate for certain medical images where up/down orientation is fixed (e.g., chest X-rays).
    - **Shearing:** Tilting the image along one axis.
    - **Elastic Deformations:** Applying local, non-linear deformations to the image, which can simulate tissue variability or patient movement. This is particularly useful for tasks like medical image segmentation [SSP+03].

    Figure 2.10 illustrate some of these transformations on a sample medical image. These transformations preserve the essential anatomical structures while increasing dataset diversity for robust model training in pneumonia detection tasks.

2. **Photometric (Color/Intensity) Transformations:** These alter the pixel intensities or color channels.

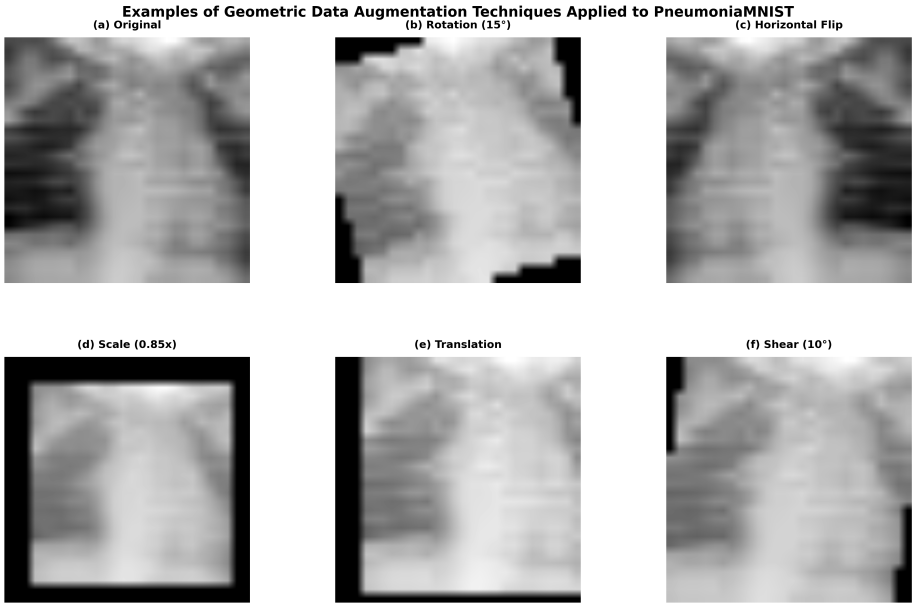


Figure 2.10: Examples of geometric data augmentation techniques applied to a sample PneumoniaMNIST image. (a) Original image, (b) Rotation (15° clockwise), (c) Horizontal flip, (d) Scaling (0.85x), (e) Translation, (f) Shear (10°).

- **Brightness Adjustment:** Increasing or decreasing the overall brightness of the image.
- **Contrast Adjustment:** Modifying the range of intensity values.
- **Saturation/Hue Adjustment (for color images):** Altering color properties, relevant for modalities like histopathology or dermatoscopy.
- **Histogram Equalization:** Redistributing pixel intensities to improve contrast.
- **Gamma Correction:** Non-linear adjustment of pixel intensity.

These can help models become robust to variations in lighting, exposure, or scanner settings.

3. **Noise Injection and Filtering:**
   - **Adding Noise:** Introducing random noise (e.g., Gaussian, salt-and-pepper) to simulate sensor noise or image degradation.
   - **Blurring:** Applying filters like Gaussian blur to smooth the image, which can help in robustness to minor focus issues.
4. **Image Mixing Techniques:** More advanced methods that combine information from multiple images.

- **Mixup [ZXCZ18]:** Creates new samples by taking convex combinations of pairs of images and their labels. This encourages linear behavior between classes.
- **Cutout [DT17] / Random Erasing [ZZK+20]:** Randomly masks out rectangular regions of an input image, forcing the model to learn from partial information and focus on less dominant features.
- **CutMix [YHO+19]:** Patches from one image are cut and pasted onto another, and labels are mixed proportionally to the area of the patches.

5. **Generative Approaches (Advanced):**

- **Generative Adversarial Networks (GANs) [GPAM+14]:** GANs can be trained to generate entirely new, synthetic images that resemble the original data distribution. This is particularly promising for medical imaging, where GANs can create realistic pathological or normal samples [FKA+18, STR+18]. However, ensuring the clinical validity and diversity of GAN-generated medical images is an ongoing research challenge.
- **Variational Autoencoders (VAEs) [KW+13]:** VAEs can also learn the underlying data distribution and generate new samples.

Table 2.2: Summary of Common Data Augmentation Techniques for Imaging Data

Technique Category	Specific Technique	Brief Description	Typical Application/Effect	Key Consideration for Medical Data
Geometric	Rotation	Rotates image by a specified angle	Invariance to object orientation	Ensure anatomical plausibility; avoid excessive rotation if orientation is diagnostically relevant
Geometric	Elastic Deformations	Applies local, non-linear spatial warps	Models tissue deformation, improves segmentation performance	Computationally more intensive; parameters need careful tuning
Photometric	Brightness Adjustment	Changes overall image brightness levels	Robustness to lighting and exposure variations	Avoid extreme changes that may obscure diagnostic features
Mixing	Mixup	Convex combination of image pairs and their labels	Improves generalization and model calibration	May be less intuitive for medical interpretation; potential effect on subtle diagnostic features
Generative	GANs	Learns to generate synthetic samples from data distribution	Creates new data instances, addresses extreme data scarcity	Ensuring clinical validity and diagnostic relevance of generated images is crucial and challenging

2.4.4 Advantages of Data Augmentation

The strategic application of data augmentation offers several compelling advantages:

- **Improved Model Performance and Generalization:** This is the primary benefit, leading to higher accuracy, better F1-scores, and more robust models on unseen data.
- **Reduced Data Collection Costs:** Lessens the dependency on acquiring vast amounts of new, labeled data.
- **Mitigation of Class Imbalance:** Techniques can be tailored to oversample underrepresented classes, leading to more equitable model performance.
- **Increased Robustness to Input Variations:** Models become less sensitive to minor changes in input data that do not alter the semantic meaning, crucial for real-world deployment.
- **Better Model Calibration:** Some studies suggest DA can lead to better-calibrated models, whose confidence scores are more reflective of their true accuracy [TCB+19].

2.4.5 Disadvantages, Challenges, and Considerations

Despite its benefits, data augmentation is not without its challenges and requires careful consideration:

- **Preserving Label Invariance and Plausibility:** Augmented data must retain the original label. Overly aggressive or inappropriate augmentation can distort key features, change the semantic meaning, or create unrealistic samples that mislead the model. This is *especially critical in medical imaging*. For example, rotating a chest X-ray by 180 degrees might render it anatomically incorrect and useless, or adding artifacts that mimic pathologies can be detrimental. The transformations must be domain-appropriate [CMV+21].
- **Computational Cost:** Augmenting data, especially on-the-fly during training, adds computational overhead, potentially increasing training time.
- **Hyperparameter Tuning:** The choice of augmentation techniques, their parameters (e.g., rotation angle range, noise level), and the probability of applying them are themselves hyperparameters that may need careful tuning for optimal performance. Automated augmentation strategies (e.g., AutoAugment [BCC+19]) aim to address this but can be computationally intensive to discover.
- **Bias Amplification:** If the original dataset contains biases (e.g., underrepresentation of certain demographics), DA might inadvertently amplify these biases if not applied thoughtfully. For instance, if a certain pathology is always shown in a specific orientation in the limited training data, augmenting only with minor rotations might not help the model generalize to other orientations.

- **Validation Set Contamination:** It's crucial that augmentation is applied only to the training set. The validation and test sets must remain untouched by augmentation to provide an unbiased estimate of the model's generalization ability.
- **Limited Increase in True Information:** While DA increases data quantity, it doesn't fundamentally add new, independent information about the underlying data distribution in the same way truly new samples would. It primarily helps the model learn existing patterns more robustly.

2.4.6 Real-World Use Cases in Medical Imaging

Data augmentation is ubiquitously used in medical image analysis tasks:

- **Radiology (X-ray, CT, MRI):**
  - Detecting lung nodules or pneumonia in chest X-rays/CTs, with augmentations like rotation, scaling, and brightness adjustments [WPL+17].
  - Segmenting organs or tumors in CT/MRI scans, where elastic deformations are particularly useful to model anatomical variability [RFB15].
  - Classifying brain tumors from MRI, using flips, rotations, and intensity shifts.
- **Histopathology (Microscopy Images):**
  - Classifying cancer cells in whole-slide images, employing color jitter, rotation, and flipping due to the staining variations and orientation independence of tissue samples [TLB+19].
  - Segmenting glands or nuclei, often using affine transformations and color augmentations.
- **Dermatology (Skin Lesion Images):**
  - Detecting melanoma from dermoscopic images, using rotations, flips, and scaling to account for variations in image capture [EKN+17].
- **Ophthalmology (Retinal Fundus Images):**
  - Screening for diabetic retinopathy, with augmentations like rotation, translation, and contrast changes [GPC+16].

In many of these cases, DA has been shown to be a key factor in achieving performance comparable to or exceeding that of human experts.

2.5 Privacy Vulnerabilities in DL Models

The theoretical potential for privacy leakage in DL models translates into concrete attack vectors that adversaries can employ. Understanding these vulnerabilities is crucial for designing effective PPDL countermeasures. These attacks typically assume different levels of adversary knowledge and access to the model (e.g., black-box access via an API, or white-box access to model parameters).

2.5.1 Model Inversion Attacks

Model inversion attacks aim to reconstruct representative samples or features of the training data by leveraging access to the trained model [FJR15b]. Given a model's output (e.g., a prediction label) and potentially some auxiliary information, the adversary attempts to find an input that corresponds to that output. In settings like facial recognition, this could mean reconstructing an average or representative face image for a specific person's identity class, potentially revealing sensitive visual features even if the exact training image isn't recovered [FJR15b]. Early demonstrations focused on linear models, but techniques have been extended to exploit confidence scores and gradients in deep neural networks.(Figure2.11

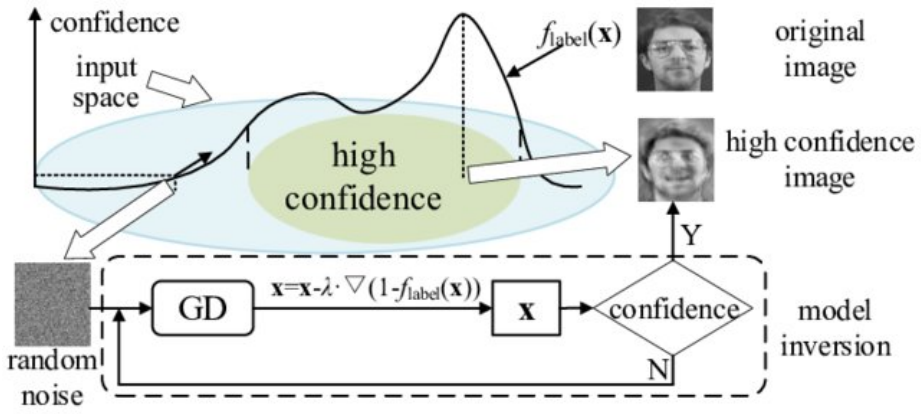


Figure 2.11: Model Inversion Attack: Step-by-Step Process.[ZLYQ20]

2.5.2 Membership Inference Attacks

Perhaps the most widely studied privacy attack in machine learning, membership inference aims to determine whether a specific data record was part of the model's training set [SSSS17b]. The adversary typically queries the target model with a data point and observes the model's output (e.g., prediction probabilities or confidence scores). The intuition is that models often exhibit different behavior (e.g., higher confidence) for inputs they were trained on compared to unseen inputs. By training a separate "attack model" to distinguish between the target model's responses on members versus non-members, the adversary can infer membership, potentially revealing sensitive information (e.g., confirming if an

individual with a specific medical condition was in a hospital's dataset used for training a diagnostic model). Figure2.12 demonstrates how a membership inference attack operates.

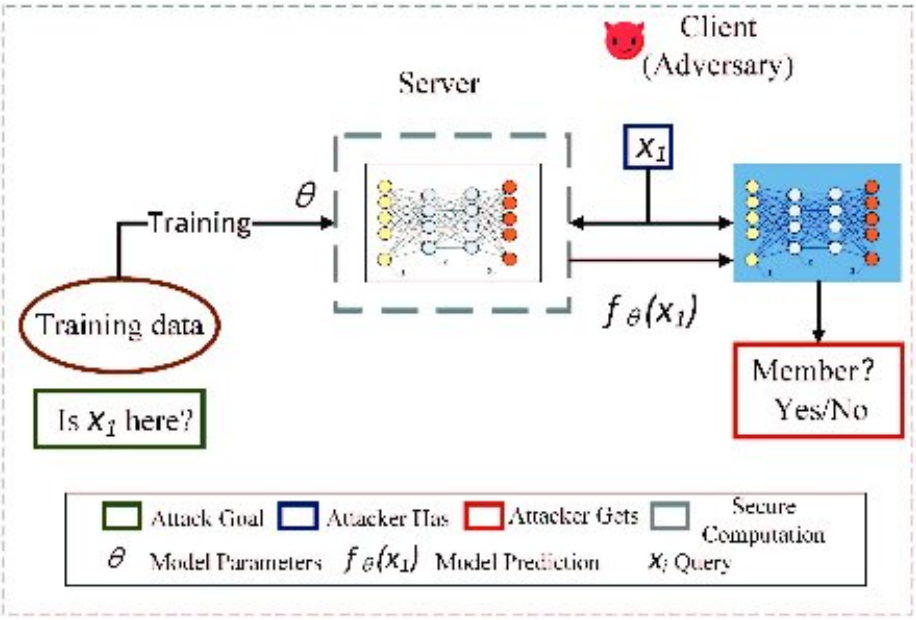


Figure 2.12: workflow of a membership inference attack.[YZL20]

2.5.3 Model Extraction Attacks

Model extraction (or model stealing) attacks focus on duplicating the functionality or extracting the parameters of a proprietary DL model, often with only black-box query access [TZJ+16]. An adversary repeatedly queries the target model with carefully chosen inputs and observes the outputs (e.g., predictions or confidence scores). Using this query-response data, the adversary trains a "surrogate" model that mimics the target model's behavior. Successfully extracting a high-fidelity model compromises the intellectual property invested in the original model and can enable other attacks (like inversion or membership inference) that might be easier to perform on the local surrogate copy [JCB+20].(see Figure2.13)

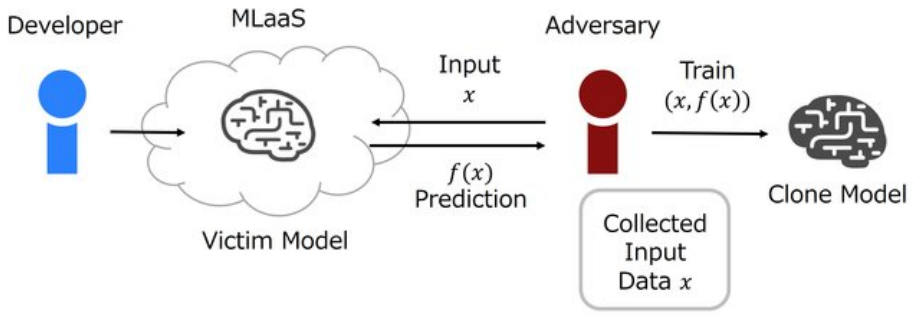


Figure 2.13: Model Extraction Attack.[MHS21]

2.5.4 Data Reconstruction Attacks

While related to model inversion, data reconstruction attacks often refer specifically to attempts to recover exact or near-exact training samples. This is particularly relevant in collaborative or federated learning settings where gradients are exchanged as shown Figure2.14. Malicious participants or servers might analyze the gradients received from other parties to reconstruct their private training data [ZLH19]. Techniques like Deep Leakage from Gradients (DLG) have demonstrated that entire images and labels can sometimes be recovered from just a single gradient update under certain conditions [ZLH19].

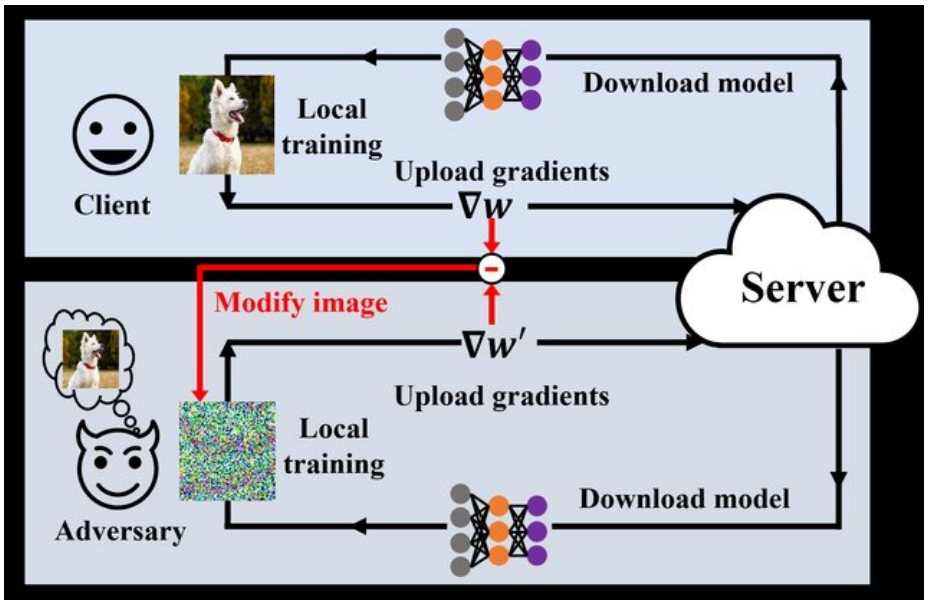


Figure 2.14: Reconstruction Attack in Federated Learning.[CZZ+23]

2.5.5 Attribute Inference Attacks

Attribute inference attacks aim to uncover sensitive attributes of individuals in the training data, even if those attributes were not the direct target of the model's prediction task [MSDCS19]. For example, given a model trained to predict income level based on various features, an adversary might try to infer a different sensitive attribute (e.g., political affiliation or medical condition) of the individuals in the training set by analyzing the model's parameters or predictions on specific inputs. This can occur if the target attribute is correlated with the features used for training or the primary prediction task. Figure2.15

2.5.6 Case Studies of Real-World Privacy Breaches

While documented, large-scale privacy breaches directly attributed *solely* to the inherent vulnerabilities of DL models (as opposed to system security flaws) are still emerging in the public domain, academic research has highlighted significant potential risks:

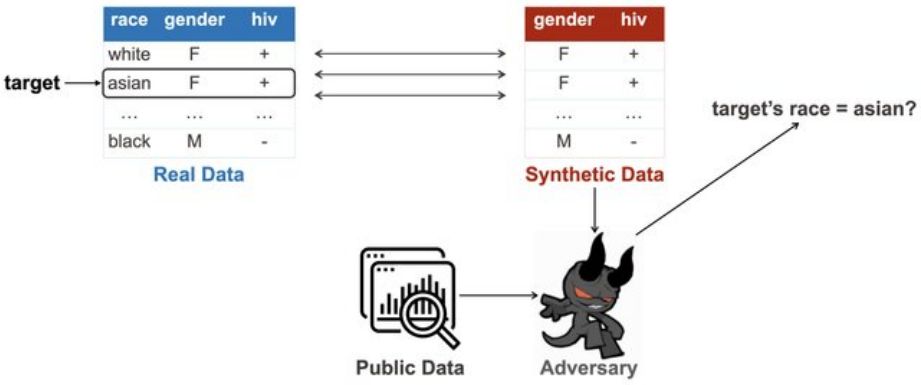


Figure 2.15: Attribute Inference Attacks.[DC23]

- **Language Model Memorization:** Researchers have shown that large language models like GPT-2 can inadvertently memorize and regenerate personally identifiable information (PII) such as names, phone numbers, and email addresses present in their training data when prompted appropriately [CTW+21].
- **Re-identification in Health Data:** Studies reconstructing facial images from models trained on medical imaging datasets have raised alarms about patient re-identification possibilities, even from seemingly anonymized data used for training diagnostic tools [PFM+21].
- **Federated Learning Vulnerabilities Demoed:** While designed for privacy, insecure implementations of federated learning have been shown vulnerable in research settings to reconstruction attacks, where a malicious server could potentially infer information about a user's local data from their gradient updates [ZLH19].

These examples underscore that the privacy risks are not merely theoretical but pose tangible threats, necessitating the adoption of robust PPDL techniques.

2.6 Privacy Preserving Techniques Overview

To counteract the privacy vulnerabilities inherent in DL, several categories of PPDL techniques have been developed. These approaches offer different types of protection, incur varying overheads, and are often suited to different stages of the DL pipeline (training vs. inference) or deployment scenarios.

2.6.1 Cryptographic Approaches

Cryptographic methods aim to provide strong, mathematically provable privacy guarantees by performing computations on encrypted data or distributing computation among multiple parties.

Homomorphic Encryption (HE)

Homomorphic Encryption enables computation directly on encrypted data (ciphertexts) without needing to decrypt it first [Gen09]. An operation performed on ciphertexts yields an encrypted result which, when decrypted, matches the result of the same operation performed on the original plaintexts. Several HE schemes exist (e.g., Paillier (partially homomorphic), BFV, BGS, CKKS (fully homomorphic but approximate for real numbers)) [BGV12]. In the context of DL, HE can be used to encrypt user data before sending it for inference to an untrusted model provider, or to encrypt model parameters or gradients during training [GBDL+16]. **Challenges:** Current HE schemes impose significant computational overhead (orders of magnitude slower than plaintext operations) and increase data size. Fully Homomorphic Encryption (FHE) often requires complex techniques like bootstrapping to manage noise growth during computation, and adapting non-linear activation functions common in DL to work efficiently under HE remains an active research area [HTG17].(Figure2.16)

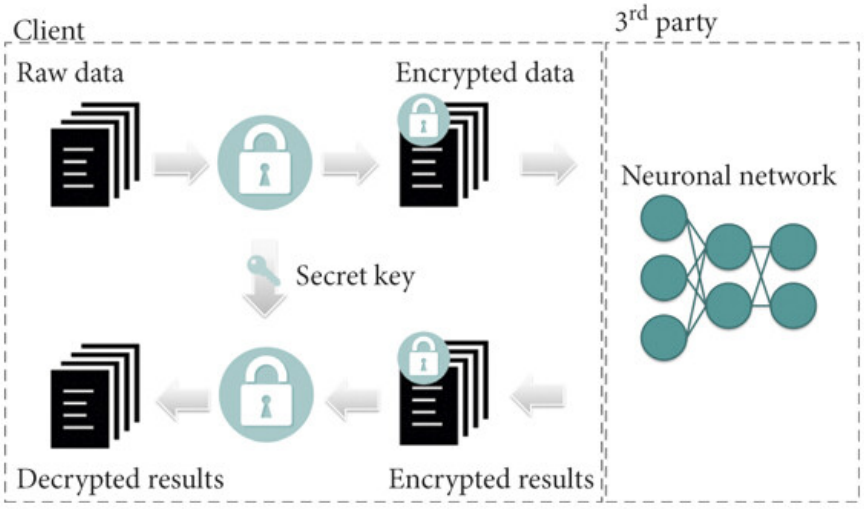


Figure 2.16: An overview of Homomorphic Encryption: Data is encrypted before being processed, enabling computations on ciphertexts without exposing the underlying data.[VNP+20]

Secure Multi-Party Computation (SMPC)

Secure Multi-Party Computation (SMPC or MPC) allows multiple parties, each holding private data, to jointly compute a function over their combined data without revealing their individual inputs to each other [Yao86, GMW87]. Common techniques underpinning SMPC include secret sharing (where data is split into shares distributed among parties) and garbled circuits (where a function is represented as an encrypted circuit). In PPDL, SMPC can facilitate secure collaborative training in which multiple data owners contribute to the training of a shared model without exposing their datasets [MZ17]. It can also be

used for private inference where either the model owner's model or the user's input data (or both) remain private.(Figure2.17) **Challenges:** SMPC protocols often require substantial communication rounds between parties, leading to high network latency and bandwidth consumption, particularly for complex DL models. The computational cost can also be considerable, and protocols typically assume specific threat models (e.g., semi-honest vs. malicious adversaries).

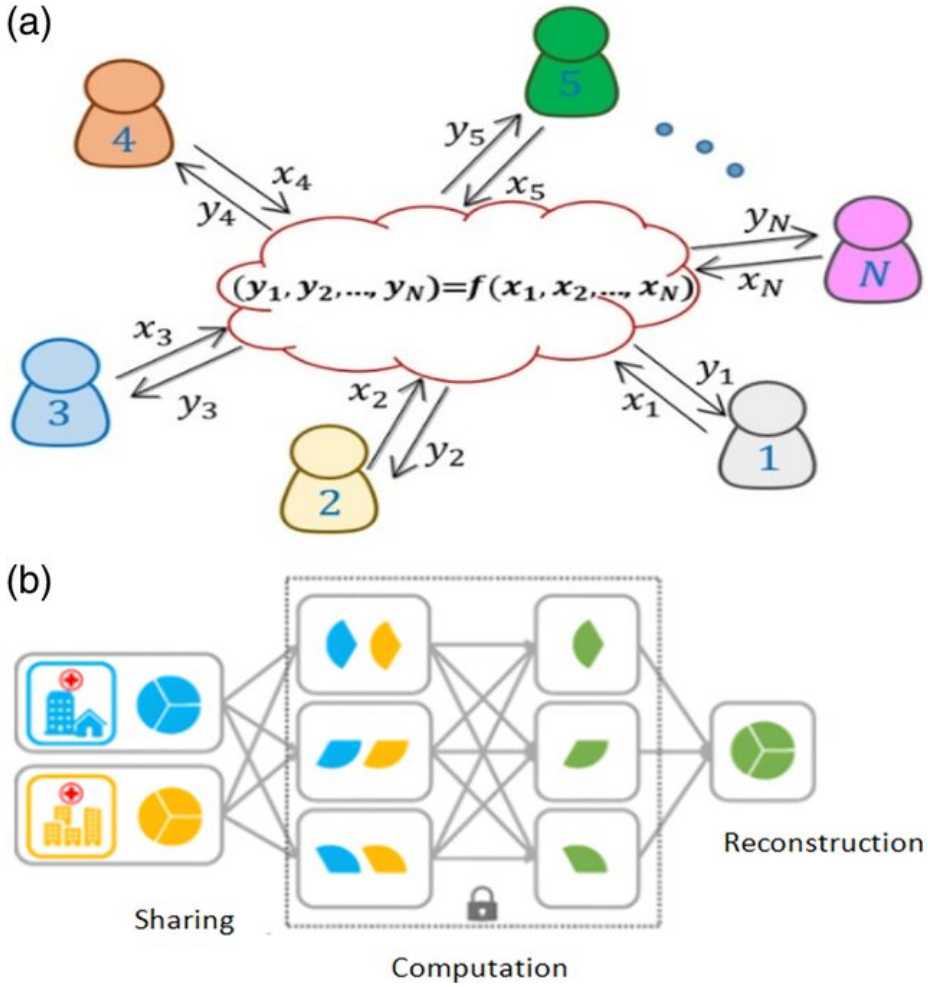


Figure 2.17: (a) Secure computation involving several participants. (b) Secure computation involving two participants.[NT23]

Functional Encryption (FE)

Functional Encryption is a more advanced cryptographic primitive where a master secret key can be used to generate specific functional keys [BSW11]. A functional key skf allows the computation of a specific function f on an encrypted ciphertext enc (x), producing the result f (x) directly, without revealing x itself or any information about x beyond f(x). Other functions cannot be computed with skf. In DL, FE could potentially allow a model owner to give users functional keys corresponding to the model prediction function. Users could

then encrypt their data, apply the key, and obtain the prediction without revealing their data to the model owner, while the model itself remains protected [RBPT21].

Challenges: Constructing efficient and secure FE schemes for general, complex functions like deep neural networks is extremely challenging and largely remains an area of theoretical research. Practical implementations for large-scale DL are currently limited.

Trusted Execution Environments (TEEs)

Trusted Execution Environments (TEEs) rely on hardware-based security features to create isolated environments within a processor (e.g., Intel SGX, ARM TrustZone) [SAB15]. Code and data loaded into a TEE (an "enclave") are protected from inspection or modification by the host operating system, hypervisor, or other processes running on the same machine, including privileged administrators. Remote attestation mechanisms allow users to verify that the correct code is running securely within the TEE before sending sensitive data. TEEs can be used to run DL training or inference processes on sensitive data within a protected enclave on potentially untrusted infrastructure [HZX+18].

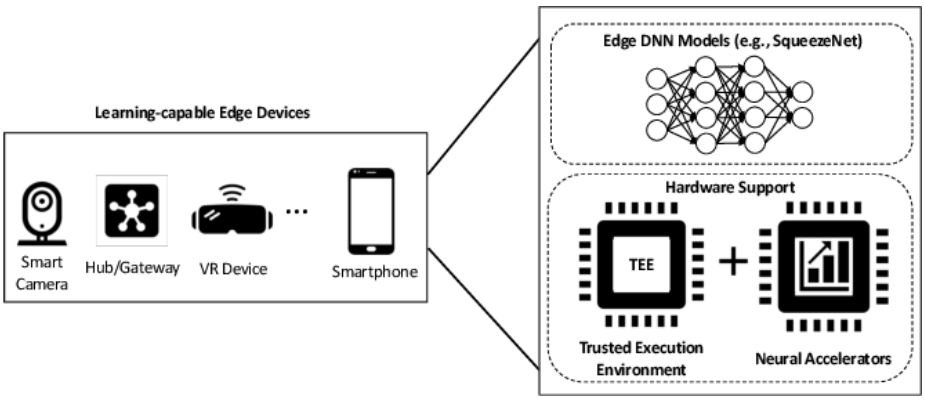


Figure 2.18: System model illustrating privacy-preserving deep learning using Trusted Execution Environments (TEEs) on edge devices. The setup enables secure model inference while leveraging hardware-based isolation and potential on-device accelerators.[LGL+21]

Challenges: TEEs depend on trusting the hardware manufacturer and the correctness of the TEE implementation. The amount of protected memory (EPC size in SGX) can be limited, potentially impacting performance for very large models. Furthermore, TEEs can be vulnerable to side-channel attacks that infer information by observing non-functional aspects like timing, power consumption, or memory access patterns [CCX+19].

2.6.2 Federated Learning (FL)

Federated Learning is a distributed machine learning paradigm that enables model training on decentralized data located on devices like mobile phones or local servers (e.g., hospitals) without requiring the data to be moved to a central location [MMR+17a].

Fundamentals of Federated Learning

The canonical FL process, often termed "vanilla" or horizontal FL (where participants share the same feature space but have different sample sets), typically proceeds as follows [MMR+17a]:

1. **Initialization:** A central server initializes a global model.
2. **Distribution:** The server sends the current global model parameters to a subset of participating clients (e.g., mobile devices).
3. **Local Training:** Each selected client updates the model using its own local data for one or more epochs (e.g., using SGD).
4. **Update Communication:** Clients send their computed model updates (e.g., gradients or weight differences) back to the server. Crucially, the raw data never leaves the client device.
5. **Aggregation:** The server aggregates the updates received from the clients (e.g., using Federated Averaging - FedAvg) to produce an improved global model.
6. **Iteration:** Steps 2-5 are repeated until the global model converges.

FL inherently enhances privacy by keeping raw data localized. However, the communicated model updates can still leak information, necessitating further protection [ZLH19, MSDCS19].

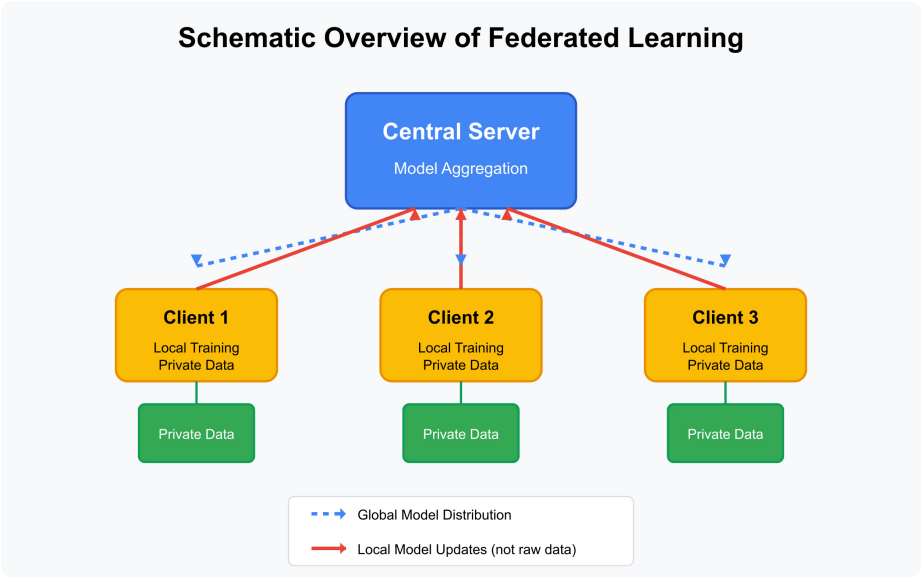


Figure 2.19: Schematic overview of the Federated Learning process. Clients train locally on private data and send model updates (not raw data) to a central server for aggregation.

Secure Aggregation in FL

To protect individual client updates from the potentially curious or malicious central server (or eavesdroppers), secure aggregation protocols are employed [BIK+17]. These protocols, often based on SMPC principles (like secret sharing) or additive HE, allow the server to compute the sum (or weighted average) of the clients' updates without learning any individual update. For example, clients can mask their updates with pairwise random seeds that cancel out when summed by the server, ensuring only the aggregate result is revealed [BIK+17].

Advancements in FL

Research in FL is rapidly evolving beyond the basic FedAvg scheme:

- **Personalized FL:** Techniques that allow the final model to be adapted or personalized to individual clients' data distributions, acknowledging that a single global model might not be optimal for everyone [KKP20].
- **Clustered FL:** Grouping clients with similar data distributions to train separate models, potentially improving accuracy for diverse populations [SMS20].
- **Asynchronous FL:** Allowing clients to join training and send updates at different times, improving efficiency and robustness to stragglers [XKG19].
- **Communication Efficiency:** Techniques like model compression, quantization, or structured updates to reduce the communication overhead, which is often a bottleneck in FL [KMY+16].

2.6.3 Differential Privacy (DP)

Differential Privacy provides a formal, mathematical definition of privacy within the context of statistical analysis and machine learning [DMNS06b, DR14b]. It offers a strong guarantee that the output of an algorithm (e.g., a trained model or query result) is statistically indistinguishable whether or not any single individual's data was included in the input dataset.

Foundations of DP

Differential Privacy (DP) is achieved by introducing carefully calibrated randomness into the computation process. The core idea is formally defined by (ϵ, δ)-DP: An algorithm $\mathcal{M}$ is (ϵ, δ)-differentially private if for any two adjacent datasets D and D' (differing by only one individual's record), and for any possible set of outputs S, the probability of obtaining an output in S from $\mathcal{M}(D)$ is closely bounded by the probability of obtaining an output in S from $\mathcal{M}(D')$, specifically: $Pr[\mathcal{M}(D) \in S] \leqslant e^{\epsilon} \times Pr[\mathcal{M}(D') \in S] + \delta$.

Here, ϵ (epsilon) is the privacy loss parameter; smaller values imply stronger privacy. δ (delta) represents the probability that the pure ϵ-DP guarantee might be broken.

Common mechanisms to achieve DP include adding noise calibrated to the function's *sensitivity* (the maximum change in the function's output when one individual's data is changed):

- **Laplace Mechanism:** Adds noise drawn from a Laplace distribution (suitable for numeric queries, achieves $(\epsilon, 0)$-DP).
- **Gaussian Mechanism:** Adds noise drawn from a Gaussian distribution (often more convenient for iterative algorithms, achieves (ϵ, δ)-DP).
- **Exponential Mechanism:** Used for selecting the best option from a set based on a quality score in a privacy-preserving way.

In the context of DL, DP is commonly applied during training by adding noise to gradients before they are used for weight updates (e.g., DP-SGD [ACG+16b]), often combined with gradient clipping to bound sensitivity. The privacy cost (ϵ, δ) accumulates over training iterations and must be tracked using composition theorems.

A detailed exploration of DP mechanisms, properties, and its application in DL (DP-SGD) is provided in Chapter 3, Section 3.2.3.

2.7 Comparison of Different Approaches

Choosing the appropriate PPDL technique depends heavily on the specific application requirements, the threat model, and the acceptable trade-offs between privacy, utility (model accuracy), efficiency (computational and communication overhead), and scalability. Table 2.3 provides a qualitative comparison of the main approaches discussed.

Table 2.3: Qualitative Comparison of Major PPDL Approaches

Technique	Privacy Guarantee	Utility Impact	Computational Overhead	Scalability	Key Assumptions/Limitations
HE	Strong (Crypto)	Low-Medium[a]	Very High	Low-Medium	Computationally intensive; noise mgmt; limited ops
SMPC	Strong (Crypto)	Low[b]	High-Very High	Low-Medium	High communication; requires multiple parties; collusion resistant?
TEE	Strong (hardware)	Low	Low-Medium	Medium-High	Trust in HW vendor; side-channels; limited memory
FL	Baseline (Data Loc.)	Medium[c]	Low (Client), Med (Comm)	High	Update leakage; requires coordination; non-IID data challenges
FL+SAgg	Improved (Crypto)	Medium[c]	Med-High (Comm+Crypto)	Medium-High	Requires secure aggregation protocol; potential collusion
DP	Formal (Probabilistic)	Medium-High[d]	Low-Medium	High	Utility degrades with low ϵ; noise addition impacts accuracy

[a]Utility impacted by approximations needed for non-linear functions or noise in FHE schemes.
[b]Can achieve near-plaintext accuracy but at high cost.
[c]Utility can be affected by non-IID data, stragglers, communication constraints.
[d]Stronger privacy (lower ϵ) typically leads to lower utility (accuracy).
SAgg = Secure Aggregation. Table provides a general overview; specifics vary greatly by implementation.

Key takeaways from the comparison:

- Cryptographic methods (HE, SMPC) offer the strongest theoretical privacy guarantees but often come with prohibitive computational or communication costs, limiting their scalability for very large DL models.
- TEEs offer a pragmatic balance by leveraging hardware security, achieving good performance but relying on hardware trust and being vulnerable to side-channels.
- Federated Learning provides a baseline level of privacy by keeping data local but requires additional mechanisms (like Secure Aggregation or DP) to protect against inference from model updates. It scales well horizontally but faces challenges with non-IID data and communication bottlenecks.
- Differential Privacy offers formal, quantifiable privacy guarantees against specific inference risks but introduces noise that inevitably impacts model utility. Its application (e.g., DP-SGD) is relatively scalable but requires careful tuning of privacy parameters.

Often, hybrid approaches combining multiple techniques (e.g., FL with DP and Secure Aggregation) are employed to leverage the strengths of each while mitigating their individual weaknesses [GKN17a].

2.8 Conclusion

This chapter has laid the theoretical foundations for Privacy-Preserving Deep Learning (PPDL). We began by establishing the context, highlighting the immense capabilities of Deep Learning while acknowledging the inherent privacy risks associated with its reliance on vast, often sensitive, datasets. We reviewed the core components of DL, including network architectures and the training process, to understand where vulnerabilities might arise.

Subsequently, we detailed the primary privacy attack vectors targeting DL models, such as model inversion, membership inference, model extraction, data reconstruction, and attribute inference. Understanding these threats is fundamental to appreciating the need for and design of PPDL countermeasures. We presented case studies and research findings that illustrate the tangible nature of these privacy risks.

The core of the chapter provided an overview of the main categories of privacy-preserving techniques: cryptographic approaches (Homomorphic Encryption, Secure Multi-Party Computation, Functional Encryption, Trusted Execution Environments), Federated Learning (including its variants and Secure Aggregation), and Differential Privacy. For each category, we outlined the fundamental principles, operational mechanisms, and key characteristics.

Finally, we presented a comparative analysis summarizing the trade-offs associated with each approach regarding privacy guarantees, impact on model utility, computational and communication efficiency, scalability, and underlying assumptions or limitations. This

comparison underscores that there is no single "best" PPDL solution; the optimal choice depends on the specific application context, threat model, and resource constraints. Often, combining techniques in hybrid systems offers the most robust and practical path forward.

The theoretical concepts discussed in this chapter provide the necessary background for delving into more specific PPDL methodologies, implementations, and evaluations in the subsequent parts of this thesis. The imperative to balance the transformative potential of deep learning with the fundamental right to privacy continues to drive innovation in this critical and rapidly evolving field.

Chapter 3

Differentially Private Deep Learning

***

3.1 Introduction

Deep Learning (DL) has achieved remarkable success across various domains, fueled by the availability of vast datasets and powerful computational resources [LBH15]. However, many of these datasets contain sensitive personal information, ranging from medical records and financial transactions to private communications and user behaviour patterns. Training DL models on such data raises significant privacy concerns, as models can inadvertently memorize specific details about individuals present in their training set [SSSS17a, CLE+19]. Releasing these models or their outputs could lead to unintended disclosure of sensitive information, violating user privacy and regulatory requirements like GDPR or HIPAA.

Differential Privacy (DP) [DMNS06a, DR14a] has emerged as the de facto standard for rigorous, mathematically provable privacy protection. DP provides a strong guarantee that the outcome of a computation (e.g., training a DL model) is statistically indistinguishable whether or not any single individual's data was included in the input dataset. This protection holds regardless of the adversary's auxiliary knowledge or computational power.

Applying DP principles to the complex, high-dimensional, and iterative nature of DL training presents unique challenges and opportunities. Differentially Private Deep Learning (DPDL) aims to bridge this gap, developing methods to train effective DL models while providing formal DP guarantees. This chapter provides a comprehensive overview of the state-of-the-art in DPDL. We begin by revisiting the fundamental concepts of DP, then delve into the specific techniques developed for adapting DP to DL pipelines, including

noise injection mechanisms and privacy accounting methods. We survey the application of DPDL across various model architectures and discuss prominent software frameworks. Key algorithms, evaluation metrics, inherent trade-offs, and application domains are examined. Finally, we present a comparative analysis of leading techniques, highlight current challenges, and outline promising future research directions in this rapidly evolving field. This review sets the stage for subsequent chapters that may delve deeper into specific DPDL methodologies or propose novel contributions.

3.2 differential privacy: A Comprehensive Survey

Before exploring its application in deep learning, it is crucial to understand the foundational concepts of Differential Privacy. DP is not about encrypting data but about adding controlled randomness to computations to mask the contribution of any single individual.

3.2.1 Formal DP definition

Differential Privacy provides a formal definition of privacy loss associated with a computation. A randomized mechanism $\mathcal{M}$ satisfies (ϵ, δ)-Differential Privacy if for any two adjacent datasets D and D', differing by at most one individual's record, and for any possible set of outcomes $S \subseteq \text{Range}(\mathcal{M})$:

$$\mathbb{P}[\mathcal{M}(D) \in S] \leqslant e^{\epsilon}\mathbb{P}[\mathcal{M}(D') \in S] + \delta \tag{3.1}$$

Here:

- ϵ (epsilon) is the privacy loss parameter (often called the privacy budget). A smaller ϵ implies stronger privacy, meaning the outputs of the mechanism on D and D' are statistically closer. $\epsilon = 0$ corresponds to perfect privacy (the output is independent of the input).

- δ (delta) represents the probability that the pure ϵ-DP guarantee fails. It should typically be a very small value, often less than the inverse of the dataset size ($1/|D|$), ensuring the failure event is rare. When $\delta = 0$, the mechanism satisfies pure ϵ-DP.

The core intuition is that an adversary observing the output $\mathcal{M}(D)$ cannot confidently infer whether any particular individual's data was used in the computation, as the probability distribution of the output changes only slightly (bounded by e^{ϵ} and δ) if one person's data is added or removed.

3.2.2 Local and Distributed DP

The trust model significantly influences how DP is applied. Two primary models exist:

- **Central DP (or Global DP):** Assumes a trusted data curator holds the entire sensitive dataset. This curator runs the DP mechanism $\mathcal{M}$ on the dataset D and releases the sanitized output $\mathcal{M}(D)$. The privacy guarantee protects individuals from inferences made based on the released output. This is the most common model for DPDL training, where the model trainer is considered trusted with the raw data.
- **Local DP (LDP):** Assumes no trusted curator. Each user perturbs their own data *before* sending it to an untrusted server or aggregator. The privacy guarantee protects each user's raw data even from the entity collecting it. LDP typically requires adding significantly more noise to achieve meaningful utility compared to central DP, as the noise must obscure individual data points rather than just individual contributions to an aggregate computation [KLN+11a].

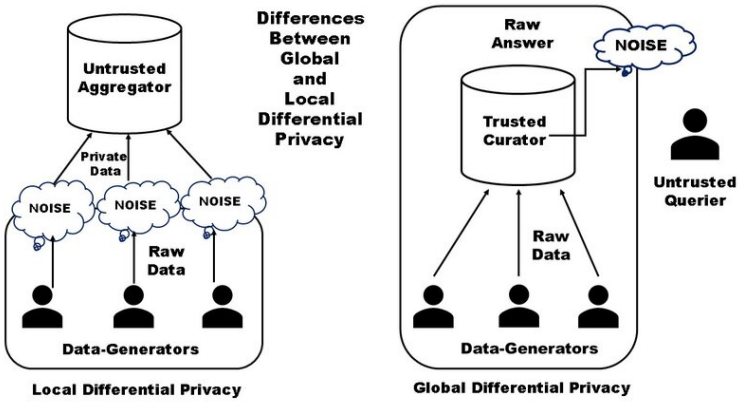


Figure 3.1: Comparison of Central (Global) and Local Differential Privacy models.[VLHT24]

This chapter primarily focuses on the central DP model, which is dominant in DPDL research where the goal is to train a model on a centrally held (though potentially distributedly stored, as in Federated Learning) dataset.

3.2.3 Types of DP

The formal definition gives rise to slightly different DP variants and associated mechanisms.

ϵ-DP Pure ϵ-DP ($\delta = 0$) provides the strongest guarantee but can sometimes be too restrictive for complex tasks. It relies on mechanisms calibrated using L1 sensitivity.

L1 Norm Sensitivity The L1 sensitivity ($\Delta_1 f$) of a function $f : \mathcal{D}^n \to \mathbb{R}^k$ measures the maximum possible change in the function's output (in L1 norm) when one individual's data is changed in the input dataset:

$$\Delta_1 f = \max_{D,D'} \|f(D) - f(D')\|_1 \tag{3.2}$$

where D, D' are adjacent datasets and $\|\cdot\|_1$ is the L1 norm (sum of absolute values).

Laplace Mechanism The Laplace mechanism achieves ϵ-DP for numeric queries by adding noise drawn from a Laplace distribution scaled according to the L1 sensitivity [DMNS06a]. For a function f with L1 sensitivity $\Delta_1 f$, the mechanism releases:

$$\mathcal{M}(D) = f(D) + \text{Lap}(\lambda)^k \tag{3.3}$$

where $\text{Lap}(\lambda)$ denotes sampling from a Laplace distribution with location 0 and scale $\lambda = \Delta_1 f / \epsilon$. Noise is added independently to each dimension of the output vector $f(D) \in \mathbb{R}^k$.

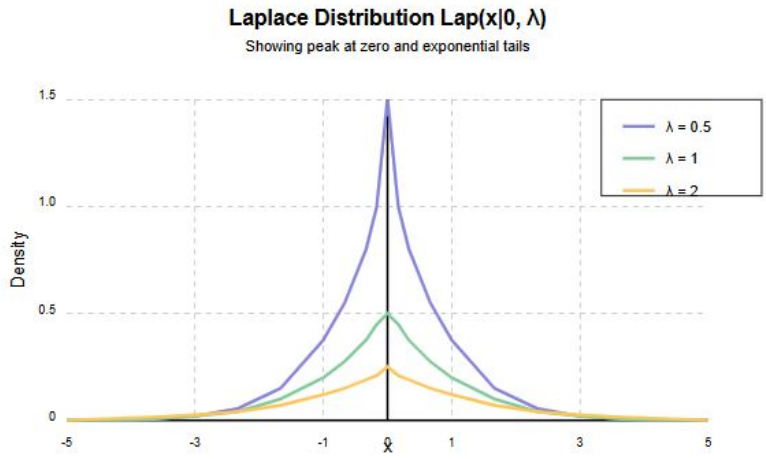


Figure 3.2: The Laplace Distribution used in ϵ-DP mechanisms.

(ϵ, δ)-DP Approximate DP, or (ϵ, δ)-DP, relaxes the pure DP guarantee slightly by allowing a small failure probability δ. This relaxation often permits more accurate results (less noise) for the same ϵ, especially for high-dimensional outputs common in DL. It typically uses mechanisms based on L2 sensitivity.

L2 Norm Sensitivity The L2 sensitivity ($\Delta_2 f$) measures the maximum change in the function's output in L2 norm (Euclidean distance):

$$\Delta_2 f = \max_{D,D'} \|f(D) - f(D')\|_2 \tag{3.4}$$

where $\|\cdot\|_2$ is the L2 norm.

Gaussian Mechanism The Gaussian mechanism achieves (ϵ, δ)-DP by adding noise drawn from a Gaussian distribution [DR14a]. For a function f with L2 sensitivity $\Delta_2 f$, the mechanism releases:

$$\mathcal{M}(D) = f(D) + \mathcal{N}(0, \sigma^2 I_k) \tag{3.5}$$

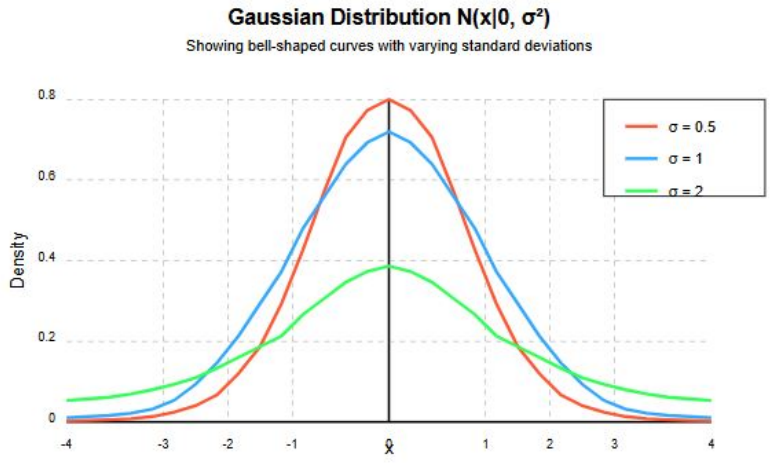


Figure 3.3: The Gaussian Distribution used in (ϵ, δ)-DP mechanisms.

where $\mathcal{N}(0, \sigma^2 I_k)$ denotes sampling from a k-dimensional Gaussian distribution with mean 0 and covariance matrix $\sigma^2 I_k$. The standard deviation σ must be calibrated based on ϵ, δ, and $\Delta_2 f$. A common calibration is $\sigma \geqslant \frac{\Delta_2 f}{\epsilon}\sqrt{2\ln(1.25/\delta)}$.

The Gaussian mechanism is particularly relevant for DPDL as gradient updates are high-dimensional vectors where L2 sensitivity is often more natural to bound and manage than L1 sensitivity.

α-DP / Rényi DP Alternative definitions of DP based on divergence measures have been proposed, notably Rényi Differential Privacy (RDP) [Mir17]. RDP characterizes the privacy loss using the Rényi divergence of order α between the output distributions on adjacent datasets.

Rényi Divergence The Rényi divergence of order $\alpha > 1$ between two probability distributions P and Q is defined as:

$$D_\alpha(P\|Q) = \frac{1}{\alpha - 1}\log \int \left(\frac{P(x)}{Q(x)}\right)^\alpha Q(x)dx \tag{3.6}$$

A mechanism $\mathcal{M}$ satisfies (α, ϵ')-RDP if for all adjacent datasets D, D', $D_\alpha(\mathcal{M}(D)\|\mathcal{M}(D')) \leqslant \epsilon'$. RDP provides a convenient way to track privacy loss under composition, especially for the Gaussian mechanism, and can be converted to the standard (ϵ, δ)-DP guarantee [Mir17, CKS20]. This is leveraged by advanced privacy accounting techniques.

3.3 Adaptation of DP to DL

Applying the core DP mechanisms directly to the complex process of training deep neural networks requires specific adaptations. The primary challenge lies in protecting the

contribution of individual training samples to the final learned model parameters, which are the result of numerous iterative updates.

3.3.1 Noise injection techniques

Randomness can be introduced at various points in the DL pipeline to achieve DP. (Figure 3.4)

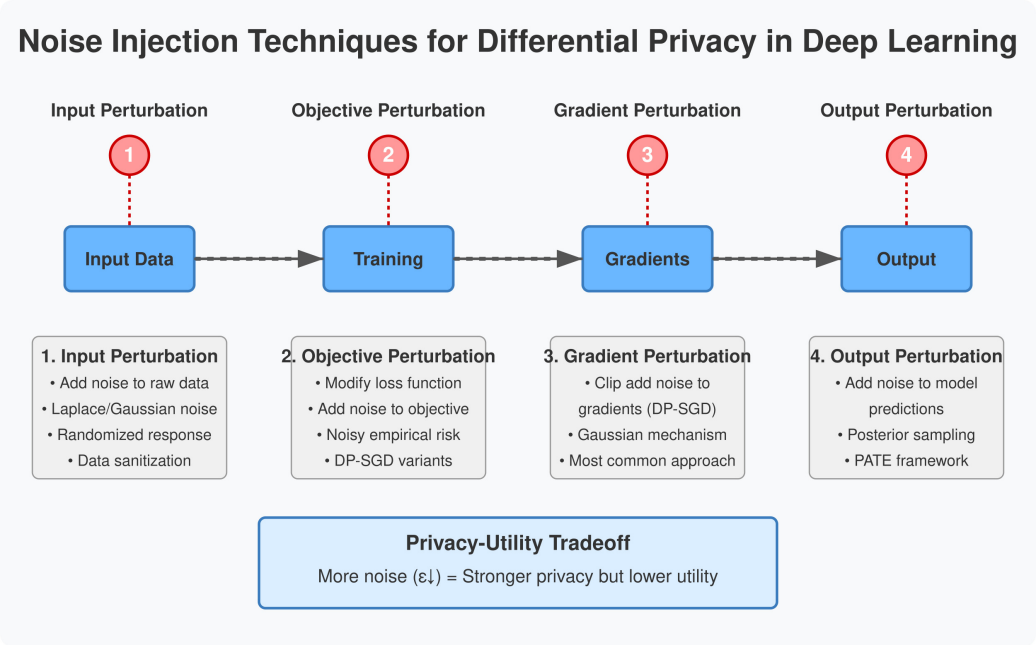


Figure 3.4: Noise injection techniques in DPDL.

Input Perturbation

This involves adding noise directly to the input data samples *before* they are fed into the model, potentially using Local DP mechanisms if data is collected from users. While conceptually simple, this often leads to significant degradation in model utility, as the noise is added early and propagates through the entire network. It also doesn't directly protect the model parameters themselves but rather sanitizes the data the model sees. This approach is less common for training state-of-the-art DPDL models compared to gradient perturbation [INS+19].

Output Perturbation

Here, noise is added to the model's output (e.g., predictions or class scores) after training on sensitive data. A notable example is PATE (Private Aggregation of Teacher Ensembles) [PAE+17, PSM+18]. In PATE, an ensemble of "teacher" models is trained on disjoint subsets of the sensitive data. A "student" model is then trained on unlabeled public data, using labels generated by querying the teacher ensemble. Noise (typically via the

Exponential Mechanism or Report Noisy Max) is added to the aggregation of teacher votes to ensure the final labeling process is DP. This transfers privacy from the teachers to the student. While effective, PATE requires a suitable ensemble and often a separate public dataset for student training.

Gradient Perturbation (Focus on training-time-noise)

This is currently the most prevalent and widely studied approach for DPDL, particularly DP-SGD (Differentially Private Stochastic Gradient Descent) [ACG+16c]. The core idea is to inject noise into the gradient updates during the training process. Since gradients quantify the influence of training data on model parameters, perturbing them directly bounds the impact any single data point can have on the final model.

The standard DP-SGD process within a training iteration involves: 1. Compute Per-Example Gradients: For a mini-batch of data, calculate the gradient of the loss function with respect to the model parameters for *each individual example* in the batch. This is computationally more expensive than standard SGD which only requires the average gradient over the batch. 2. Clip Per-Example Gradients: Clip the L2 norm of each individual gradient to a predefined threshold C. This bounds the maximum influence (L2 sensitivity) any single example can have on the aggregated gradient. Let g_i be the gradient for example i. The clipped gradient $\tilde{g}_i$ is:

$$\tilde{g}_i = g_i / \max(1, \frac{\|g_i\|_2}{C}) \tag{3.7}$$

3. Aggregate Clipped Gradients: Compute the average of the clipped gradients over the mini-batch: $\tilde{g}_{batch} = \frac{1}{B}\sum_{i=1}^{B} \tilde{g}_i$, where B is the batch size. 4. Add Noise: Add Gaussian noise scaled to the sensitivity (which is related to C and B) and the desired privacy level (ϵ, δ) to the averaged gradient:

$$\hat{g}_{batch} = \tilde{g}_{batch} + \mathcal{N}(0, \sigma^2 C^2 / B^2 \cdot I) \tag{3.8}$$

The noise scale σ is chosen based on the privacy budget (ϵ, δ) and the number of training steps, typically using privacy accountants (Section 3.3.2). The sensitivity of the sum of clipped gradients is C, so the sensitivity of the *average* is C/B. 5. Update Model Parameters: Update the model parameters using this noisy average gradient, typically via standard optimizers like SGD or Adam:

$$\theta_{t+1} = \theta_t - \eta \hat{g}_{batch} \tag{3.9}$$

where η is the learning rate.

This gradient perturbation approach, integrated within the training loop, provides end-to-end privacy for the learned model parameters.

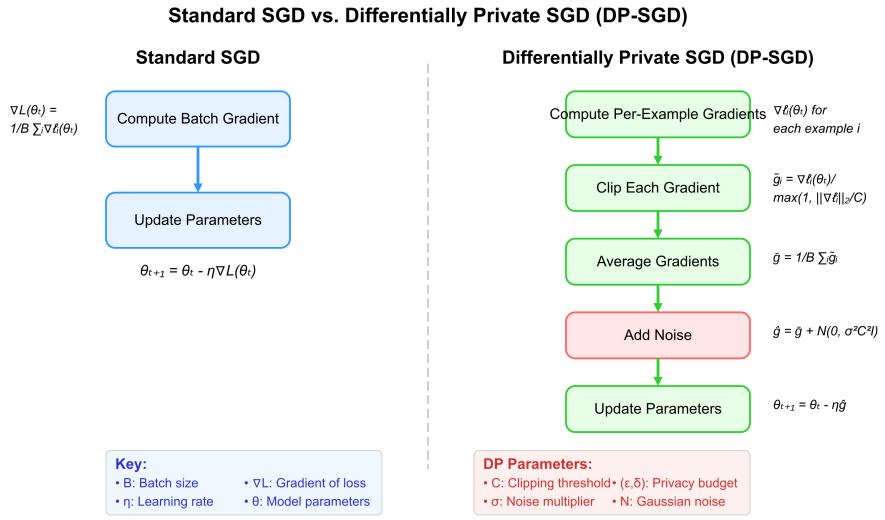


Figure 3.5: Comparison of parameter update steps in standard SGD and DP-SGD.

Objective Perturbation

An alternative approach to achieving differential privacy in machine learning, distinct from gradient perturbation (like DP-SGD), is **Objective Perturbation**. Instead of adding noise to the gradients during optimization iterations, this method involves adding carefully calibrated noise directly to the objective function (loss function) being minimized [CMS11]. The goal is to ensure that the *minimizer* of the perturbed objective function, which corresponds to the learned model parameters θ, satisfies differential privacy.

Core Mechanism: The standard Empirical Risk Minimization (ERM) aims to find parameters θ that minimize the empirical loss $L(\theta, D) = \frac{1}{|D|}\sum_{i \in D} \ell(\theta, d_i)$ over the private dataset D. In objective perturbation, we instead solve a perturbed optimization problem:

$$\theta_{DP} = \arg\min_{\theta} \tilde{L}(\theta, D) = \arg\min_{\theta} \left(L(\theta, D) + \frac{1}{|D|} b^T \theta \right) \tag{3.10}$$

where b is a random noise vector drawn from a distribution designed to ensure (ϵ, δ)-DP. The scale and distribution of b depend critically on the properties of the loss function ℓ, particularly its convexity and smoothness, and the sensitivity of the objective function itself to changes in a single data point [CMS11, KST12].

For instance, for strongly convex and Lipschitz loss functions, adding noise b drawn from a specific distribution related to the sensitivity (often based on Gaussian or Laplace distributions scaled appropriately) can guarantee DP for the resulting model parameters θ_{DP} [CMS11]. More generally, this falls under the framework of the *functional mechanism* [KST12, ZZX+12], where noise is added to a function output (here, the objective function value interpreted across the parameter space) to protect the input data.

Comparison with Gradient Perturbation: Unlike DP-SGD, where noise is added iteratively to gradients computed on mini-batches, objective perturbation typically conceptualizes adding noise "once" to the overall objective defined on the entire dataset. While optimization might still proceed iteratively, the noise source is tied to the global objective rather than individual gradient steps.

Advantages and Relevance to DPDL:

- **Direct Optimization Goal:** It directly targets making the final model parameters θ_{DP} private by perturbing the function whose minimum defines them.
- **Potential for Non-Iterative Solvers:** For certain classes of problems (especially convex ones with closed-form or efficient non-iterative solutions), objective perturbation might offer a way to achieve DP without iterative noise addition and complex privacy accounting across steps.

Challenges for Deep Learning Fine-Tuning: Despite its theoretical elegance, applying objective perturbation effectively to fine-tune complex DPDL models faces significant hurdles:

- **Non-Convexity:** Deep learning loss landscapes are highly non-convex. The theoretical guarantees for objective perturbation are strongest and simplest for convex objectives [CMS11]. Extending these guarantees rigorously to non-convex settings like deep learning is challenging, often requiring specific assumptions or resulting in weaker bounds [JT14]. Finding the global minimum of the perturbed non-convex objective is also difficult.
- **Sensitivity Analysis:** Bounding the sensitivity of the objective function $L(\theta, D)$ with respect to changes in a single data point can be difficult for complex, high-dimensional neural network losses, especially globally over the entire parameter space θ. This sensitivity bound is needed to calibrate the noise b.
- **Computational Cost:** Solving the perturbed optimization problem (Eq. 3.10) might still require iterative methods like SGD. It's not immediately clear if optimizing the perturbed objective offers significant computational advantages over DP-SGD for typical deep learning training paradigms, especially given the complexities of analyzing sensitivity and ensuring convergence in non-convex settings.
- **Compatibility with SGD:** Integrating objective perturbation directly into standard mini-batch SGD workflows used for DPDL is less straightforward than integrating gradient perturbation.

Due to these challenges, particularly the non-convexity of deep learning objectives, objective perturbation is currently less commonly used for training state-of-the-art DPDL models compared to gradient perturbation methods like DP-SGD and its variants. However, it

remains an important theoretical alternative and might find niches in specific settings or inspire hybrid approaches.

Advanced noise injection Techniques

While Gaussian noise added to gradients is standard, research explores other techniques.

Exploration of alternative distributions While Gaussian noise is mathematically convenient for (ϵ, δ)-DP and composition using accountants, other noise distributions might be applicable in specific scenarios or offer different trade-offs.

Table 3.1: State-of-the-Art Differentially Private Training Mechanisms

Approach	Technique	Datasets	Performance	Task Type	Year	Reference
			Gradient-Based Optimization			
DP-SGD (Low Precision)	Lower precision training with DP-SGD	MNIST, CIFAR-10	95% (MNIST, $\epsilon = 3$)	Classification	2024	[XSW+24]
Ghost Clipping DP	Efficient gradient clipping for DP	CIFAR-10, ImageNet	2x speedup over naive DP-SGD	Classification	2024	[GCZG24]
DP-SGD	Gradient clipping with noise	MNIST, CIFAR-10	98% (MNIST, $\epsilon = 2$)	Classification	2016	[ACG+16c]
			Federated Learning			
DP-FedAvg	Federated learning with DP	CIFAR-100, MedMNIST	65% (CIFAR-100, $\epsilon = 3$)	Classification	2024	[MXZ24]
			Public Data-Augmented			
Optimal DP-Public	Public-private optimization	Synthetic, CIFAR-10	75.1% (CIFAR-10, $\epsilon = 2$)	Classification	2023	[LHZ+23]
DP with Public Data	Public data priors for DP	CIFAR-10, ImageNet	Optimal error bounds	Multi-task	2023	[Zhu23]
Large-Scale Transfer	Transfer learning with DP-SGD	ImageNet, CIFAR-10	81.7% (ImageNet, $\epsilon = 4$–10)	Classification	2022	[MTKC22]
			Generative Models			
DP-VAE	Variational inference with DP	MNIST, Fashion-MNIST	KL-div ≈ 0.1 ($\epsilon = 2$)	Generation	2023	[Che23]
DP-GAN	DP in GAN training	MNIST, CelebA	FID ≈ 20 ($\epsilon = 5$)	Generation	2018	[XLW+18]

Approach	Technique	Datasets	Performance	Task Type	Year	Reference
DFA-DP	Direct Feedback Alignment with DP	MNIST, CIFAR-10	10–20% gain over DP-SGD	Classification	2020	[LPBK20]
			Self-Supervised and Transfer			
DP-RandP	Random process priors	CIFAR-10, ImageNet	72.3% (CIFAR-10, $\epsilon = 1$)	Classification	2023	[YYW+23]
Self-Supervised DP	Contrastive pre-training with DP	CIFAR-10, SNLI	70% (CIFAR-10, $\epsilon = 3$)	Classification	2022	[CLY+22]
			Transformer-Based			
DP-TTS	DP in transformer training	LJSpeech, LibriTTS	WER $\approx$ 12% ($\epsilon = 5$)	Speech Synthesis	2024	[ZDC24]

Exponential Mechanism Used when the output space is non-numeric (e.g., selecting an item from a set). It assigns probabilities to outputs exponentially higher for those with better "quality" scores, while maintaining ϵ-DP [MT07]. It's used in PATE for aggregating teacher votes and potentially applicable in other selection tasks within DL. The probability of selecting an item r is proportional to $\exp(\frac{\epsilon q(D,r)}{2\Delta q})$, where $q(D,r)$ is the quality score and Δq is the sensitivity of the quality score function.

Binomial mechanism The binomial mechanism can be used for releasing counts or sums of bounded inputs under ϵ-DP or (ϵ, δ)-DP, potentially offering advantages over Laplace/Gaussian in specific discrete settings [ASY+18]. Its application in mainstream DPDL training is less common than gradient perturbation.

Layer-wise noise injection Instead of adding noise only to the final aggregated gradient, some research explores adding noise at intermediate layers of the neural network during the forward or backward pass [BJWW+19]. The motivation can be interpretability or exploring different privacy-utility trade-offs, although ensuring rigorous end-to-end DP guarantees and effective privacy accounting can be more complex.

3.3.2 Privacy Accounting Techniques

Training a DL model involves thousands or millions of gradient updates. Each noisy update consumes a fraction of the total privacy budget. Accurately tracking the cumulative privacy loss across all these steps is critical. Simple linear composition of ϵ values is often too loose (overestimates privacy loss), leading to excessive noise. Advanced composition theorems and specialized accountants provide tighter bounds.

Moments Accountant

Introduced alongside DP-SGD [ACG+16c], the Moments Accountant provides a tighter analysis of the cumulative privacy loss for iterative mechanisms, particularly those using Gaussian noise and subsampling. It works by tracking the moments of the privacy loss random variable (specifically, its log moment-generating function) across compositions. By bounding these moments, it can then be converted back to an overall (ϵ, δ)-DP guarantee for the entire training process. This technique significantly reduces the amount of noise needed compared to basic or advanced composition theorems for the same overall (ϵ, δ).

The core idea involves tracking $\alpha_M(\lambda) = \max_{D,D'} \log \mathbb{E}[\exp(\lambda L_{\mathcal{M}(D)||\mathcal{M}(D')})]$, where L is the privacy loss random variable. Composition properties of these moments allow tracking the total loss over T steps. Finally, the overall (ϵ, δ) is derived from the computed moments bound $\alpha_M(\lambda)$ using the relation $\delta = \min_\lambda \exp(\alpha_M(\lambda) - \lambda\epsilon)$.

Renyi Accountant

Based on Rényi Differential Privacy (RDP) [Mir17], the Rényi Accountant offers an alternative, often simpler and sometimes tighter, method for privacy accounting. It leverages the fact that RDP composes linearly: if a mechanism is (α, ϵ'_1)-RDP and another independent one is (α, ϵ'_2)-RDP, their composition is $(\alpha, \epsilon'_1 + \epsilon'_2)$-RDP for the same α. The Gaussian mechanism has a closed-form RDP guarantee $(\alpha, \alpha(\Delta_2 f)^2/(2\sigma^2))$ for sensitivity $\Delta_2 f$ and noise variance σ^2.

The accountant tracks the RDP parameter $\epsilon'(\alpha)$ for a range of α values across iterations, incorporating the effects of subsampling. The final RDP guarantee $(\alpha, \epsilon'_{total}(\alpha))$ can then be converted to a standard (ϵ, δ)-DP guarantee using the relation $\epsilon = \epsilon'_{total}(\alpha) + \frac{\log(1/\delta)}{\alpha - 1}$ and minimizing over $\alpha > 1$. RDP often provides the tightest known bounds for privacy amplification due to subsampling with Poisson sampling approximation [WBK19].

Subsampling and Amplification

Standard DL training uses mini-batch SGD. This inherent subsampling provides "privacy amplification": the privacy guarantee for processing a random subsample (mini-batch) is stronger than processing the entire dataset at once for the same noise level. Intuitively, if an individual is not selected in a mini-batch, their data has zero impact on that specific update.

Both Moments and Rényi accountants incorporate rigorous analysis of privacy amplification due to subsampling (typically assuming Poisson sampling, where each data point is included in a batch independently with probability $p = B/N$, where B is batch size and N is dataset size). This amplification significantly reduces the required noise magnitude, making DPDL feasible for larger models and datasets [BNS14, WBK19]. The effective privacy cost per step is reduced, allowing for longer training or lower overall ϵ.

3.3.3 Recent Advancement

The field of DPDL is actively evolving beyond the foundational DP-SGD. Key areas of recent progress include:

Improved noise management

Research explores adding noise more effectively, potentially using adaptive noise schedules [PSL19] or investigating non-Gaussian noise sources tailored to specific loss landscapes, although Gaussian noise remains dominant due to its analytic tractability with accountants. Techniques to reduce the effective dimensionality of gradients before adding noise (e.g., using low-rank approximations) are also explored [YZC+21].

Adaptive clipping

The fixed clipping norm C in standard DP-SGD is a sensitive hyperparameter. If C is too small, it distorts gradients and hinders convergence; if too large, it requires adding more

noise for the same privacy level. Adaptive clipping strategies aim to dynamically adjust C during training, for instance, based on gradient statistics from previous iterations or batches [TRM+20, PSL19, ATMR21]. This can improve convergence speed and final model utility.

Efficient Privacy Budget Tracking

While Moments and Rényi accountants are powerful, research continues to refine privacy accounting. This includes developing tighter analytical bounds (e.g., using zero-Concentrated DP (zCDP) [BS16]), providing numerically stable implementations [YSS+21, Goo19a], and developing tools for fine-grained budget allocation across different phases of training or model components. Tracking privacy loss for complex compositions, such as in multi-stage training pipelines or federated learning, remains an active area.

3.4 Architectures and models using DP

DP techniques, primarily gradient perturbation via DP-SGD and its variants, have been applied to train a wide range of deep learning architectures. The feasibility and resulting utility often depend on the model size, dataset, task complexity, and the target privacy budget ϵ.

3.4.1 Overview of model types where DP has been applied

Table 3.3 provides a summary of common DL architectures where DP training has been demonstrated.

Key observations: 1. Fully Connected NN (MLPs): Often used as benchmarks for DPDL algorithms on simpler datasets like MNIST. DP-SGD generally works well here, achieving reasonable utility for moderate ϵ. 2. CNNs: DP training of CNNs for image tasks (CIFAR-10, medical imaging) is feasible. Performance degradation compared to non-private models is noticeable but often acceptable depending on ϵ. Techniques like PATE have also shown success. Larger image datasets (e.g., ImageNet) remain challenging due to model size and training duration [DBH+]. 3. RNNs and LSTMs: DP training for sequential data like text is common, especially in federated learning settings [MAE+17]. Handling variable sequence lengths and vanishing/exploding gradients requires careful application of clipping and noise. 4. GANs and Diffusion Models: Generating high-quality synthetic data with DP is an active research area. DP-GANs often suffer from mode collapse or reduced sample quality [TKP19]. DP-Diffusion models are a more recent development showing promise for better fidelity [DCBJ22], though often requiring larger ϵ budgets. Privacy guarantees for generative models are complex, covering the generation process itself. 5. Transformers: Applying DP to large transformer models, especially during pre-training, is computationally demanding due to the per-example gradient requirement. Fine-tuning pre-trained transformers with DP is more common and feasible, enabling privacy-preserving NLP applications [LTLH22, AGSS]. Techniques often

Table 3.2: Examples of Recent Advancements in DPDL Techniques

Area	**Advancement/Technique**	**Key Idea / Benefit**
Noise Management	Adaptive Noise Scaling	Adjust noise multiplier σ during training based on convergence or privacy budget consumption. [PSL19]
	Low-Rank Gradient Approximation	Project gradients to lower dimension before clipping/noising to reduce noise impact. [YZC+21]
Gradient Clipping	Adaptive Clipping Norm	Dynamically set clipping threshold C based on gradient statistics (e.g., median norm). [TRM+20, ATMR21]
Privacy Accounting	Tighter Bounds via zCDP / RDP	Refined analysis leading to less noise for the same (ϵ, δ). [BS16, Mir17]
	Numerically Stable Accountants	Robust implementations in libraries preventing floating-point issues. [YSS+21, Goo19a]
	Budget Allocation Strategies	Methods to optimally distribute privacy budget across epochs or components. [Citation Needed]
Optimization	DP Variants of Adam/RMSProp etc.	Adapting DP-SGD principles to more advanced optimizers. [Goo19b, PSL19]

Table 3.3: Application of DP Training across Deep Learning Architectures

Architecture	Example Task(s)	Typical Perturbation	Typical ϵ Range*	Key Reference(s)
Fully Connected NN (MLPs)	Tabular data classification (e.g., MNIST, UCI datasets)	Gradient (DP-SGD)	1 - 10	[ACG+16c]
Convolutional NN (CNNs)	Image classification (e.g., CIFAR-10, SVHN, medical images)	Gradient (DP-SGD)	2 - 10+	[ACG+16c, PSM+18]
Recurrent NN (RNNs) / LSTMs / GRUs	Sequential data processing (e.g., language modeling, text generation, time series)	Gradient (DP-LSTM/RNN variants)	1 - 10+	[MAE+17, TRM+20]
Generative Adversarial Networks (GANs)	Synthetic data generation (images, tabular data)	Gradient (DP-GAN variants, often on discriminator) / PATE	5 - 100+ (often higher)	[XWD+18, JYvdS19]
Diffusion Models	High-fidelity synthetic image generation	Gradient (DP-DDPM variants)	5 - 100+	[DCBJ22]
Transformers (e.g., BERT, GPT variants)	Natural Language Processing (fine-tuning, pre-training)	Gradient (DP-Adam on specific layers or full model)	1 - 10+ (fine-tuning), higher for pre-training	[LTLH22, AGSS]

*Note: Reported ϵ values are highly dependent on dataset size, number of epochs, specific DP parameters (δ, noise multiplier, clipping norm), accounting method, and desired utility. These ranges are indicative examples found in the literature.

involve freezing parts of the model or using DP only on specific layers (e.g., adapters) to manage computational cost and utility impact.

The trend shows increasing efforts to apply DPDL to larger, more complex state-of-the-art architectures, pushing the boundaries of scalability and utility.

3.5 Notable Frameworks and Libraries

The practical implementation of DPDL algorithms has been significantly facilitated by the development of specialized software libraries that integrate with popular DL frameworks.

The following table 3.4 provides a comprehensive overview of notable works in differentially private deep learning as of April 2025, summarizing key approaches, their methodologies, and their applications across various tasks. The table is structured to categorize methods into distinct groups, such as optimization-based techniques, generative models, and language model adaptations, reflecting the diversity of strategies employed to ensure privacy while maintaining utility. Each entry details the resolution technique, search strategy (indicating whether the method is centralized, with all computations on a single server, or distributed, with computations spread across multiple nodes), performance metrics, privacy constraints, scalability, and evaluation datasets, offering a clear comparison of the methods' strengths and limitations. The entries are ordered from the most recent to the earliest contributions, highlighting the evolution of the field over time. This synthesis aims to provide researchers with a consolidated view of the current landscape, facilitating a deeper understanding of the trade-offs and advancements in differentially private deep learning.

Approach	Method Category	Resolution Technique	Search Strategy	Performance	Privacy Constraints	Scalability	Evaluation Technique
[ZW24]	Optimization-Based	Bayesian network with added noise	Centralized	AUC $\approx$ 0.67 ($\epsilon = 5$)	Yes	Yes	UCI datasets
[LC24]		Lower precision training with DP-SGD	Centralized	95% (MNIST, $\epsilon = 3$)	Yes	No	MNIST, CIFAR-10
[WL23]		Efficient gradient clipping for DP	Centralized	2x speedup over naive DP-SGD	Yes	Yes	CIFAR-10, ImageNet
[CZ23]		Public-private optimization	Centralized	75.1% (CIFAR-10, $\epsilon = 2$)	Yes	Yes	Synthetic, CIFAR-10
[ZY24]	Benchmarking and Efficiency	Holistic evaluation framework	Centralized	Varies by method	Yes	Yes	CIFAR-10, MNIST
[LX24]		Compute-memory optimization	Centralized	119x throughput gain	Yes	Yes	RecSys datasets
[XZ23]		DP in transformer training	Centralized	WER $\approx$ 12% ($\epsilon = 5$)	Yes	No	LJSpeech, LibriTTS
[YK23]		Federated learning with DP	Distributed	65% (CIFAR-100, $\epsilon = 3$)	Yes	Yes	CIFAR-100, MedMNIST
[ZS24]	Synthetic Data and Transfer	Marginal-based synthesis	Centralized	AUC $\approx$ 0.68 ($\epsilon = 5$)	Yes	Yes	Adult, UCI datasets
[SP24]		Public data priors for DP	Centralized	Optimal error bounds	Yes	Yes	CIFAR-10, ImageNet
[KL23]		Transfer learning with DP-SGD	Centralized	81.7% (ImageNet, $\epsilon = 4 - 10$)	Yes	Yes	ImageNet, CIFAR-10
[PG23]		Random process priors	Centralized	72.3% (CIFAR-10, $\epsilon = 1$)	Yes	Yes	CIFAR-10, ImageNet

Approach	Method Category	Resolution Technique	Search Strategy	Performance	Privacy Constraints	Scalability	Evaluation Technique
[LM24]	Generative Models	Variational inference with DP	Centralized	KL-div $\approx$ 0.1 ($\epsilon = 2$)	Yes	No	MNIST, Fashion-MNIST
[CY22]		DP in GAN training	Centralized	FID $\approx$ 20 ($\epsilon = 5$)	Yes	No	MNIST, CelebA
[HL22]		Direct Feedback Alignment with DP	Centralized	10-20% gain over DP-SGD	Yes	No	MNIST, CIFAR-10
[ML24]	NLP and Language Models	Pretrained LLMs with DP-SGD	Centralized	80% (GLUE, $\epsilon = 2$)	Yes	Yes	GLUE, SQuAD
[LZ23]		Contrastive pretraining with DP	Centralized	70% (CIFAR-10, $\epsilon = 3$)	Yes	No	CIFAR-10, SNLI
[ZH23]		DP-SGD with fine-tuning	Centralized	85% (SST-2, $\epsilon = 4$)	Yes	No	SNLI, SST-2
[GS23]	Regression and Optimization	Noisy gradient descent	Centralized	MSE $\approx$ 0.45 ($\epsilon = 3$)	Yes	No	California Housing
[SZ22]		Gradient clipping with noise	Centralized	98% (MNIST, $\epsilon = 2$)	Yes	No	MNIST, CIFAR-10

Table 3.4: Notable Works in Differentially Private Deep Learning

Several observations can be drawn from the table 3.4, reflecting the evolution and diversity of differentially private deep learning approaches. Notably, recent methods (2023–2024) demonstrate a strong focus on scalability, with many approaches, such as DP-FedAvg [YK23] and Large-Scale Transfer [KL23], achieving high performance on large datasets like CIFAR-100 and ImageNet while maintaining privacy guarantees. The "Search Strategy" column reveals a shift over time: earlier methods, such as DP-SGD [SZ22], predominantly use a centralized strategy, achieving high performance (e.g., 98% on MNIST) but lacking scalability, whereas more recent methods like DP-FedAvg [YK23] adopt a distributed strategy to enhance scalability, albeit with a performance trade-off (e.g., 65% on CIFAR-100). The field has also seen significant diversification in task types, with newer works addressing speech synthesis (e.g., DP-TTS [XZ23]) and recommendation systems (e.g., LazyDP [LX24]), expanding beyond traditional classification tasks. However, the reliance on benchmark datasets like MNIST and CIFAR-10 across most methods highlights a need for more diverse evaluation techniques to assess real-world applicability. These trends suggest that while significant progress has been made, future research should focus on balancing scalability, performance, and broader task applicability in differentially private deep learning, particularly by exploring distributed strategies that can handle large-scale, real-world datasets effectively.

3.5.1 Opacus (Meta/Facebook)

Opacus is a library developed by Meta AI for training PyTorch models with DP [YSS+21]. It aims to make DPDL accessible with minimal code changes.

Integration with Pytorch

Opacus integrates seamlessly into existing PyTorch training loops. Users typically need to wrap their model, optimizer, and dataloader using Opacus components. The core abstraction is the 'PrivacyEngine'.

Performance characteristics

Opacus includes performance optimizations, such as using functorch for efficient per-example gradient computations, which can significantly speed up DP training compared to naive implementations. However, DP training inherently adds computational overhead due to per-sample gradient calculations and clipping.

Privacy Engine

The 'PrivacyEngine' attaches to the optimizer and automates the DP-SGD steps: computing per-sample gradients, clipping them, adding noise, and aggregating. It also manages the privacy accounting (using RDP accountants) to track the (ϵ, δ) budget consumed.

3.5.2 TensorFlow Privacy (Google)

TensorFlow Privacy (TF Privacy) is Google's library for training TensorFlow models with DP [Goo19a]. It provides tools for implementing DP variations of common optimizers.

Similar to Opacus, TF Privacy requires modifications to the training process, primarily by using DP-enabled optimizers (e.g., 'DPAdamGaussianOptimizer'). These optimizers handle the gradient clipping, noise addition, and privacy accounting internally. TF Privacy also typically uses RDP accountants for tracking privacy loss. It supports both Keras and standard TensorFlow workflows.

3.5.3 Other research Level Libraries

Besides the major industry-backed libraries, several other research-oriented libraries exist:

- **PyVacy:** An earlier PyTorch library for DP [W+19], serving as inspiration for parts of Opacus. Less actively maintained now.
- **DiffPrivLib (IBM):** A general-purpose DP library focusing more on classical machine learning (e.g., scikit-learn models) and basic DP mechanisms, rather than deep learning specifically [HBAL22].
- **Private AI (University):** Various university labs maintain research codebases implementing specific DPDL algorithms or extensions, often released alongside publications. [Citation Needed for specific examples]

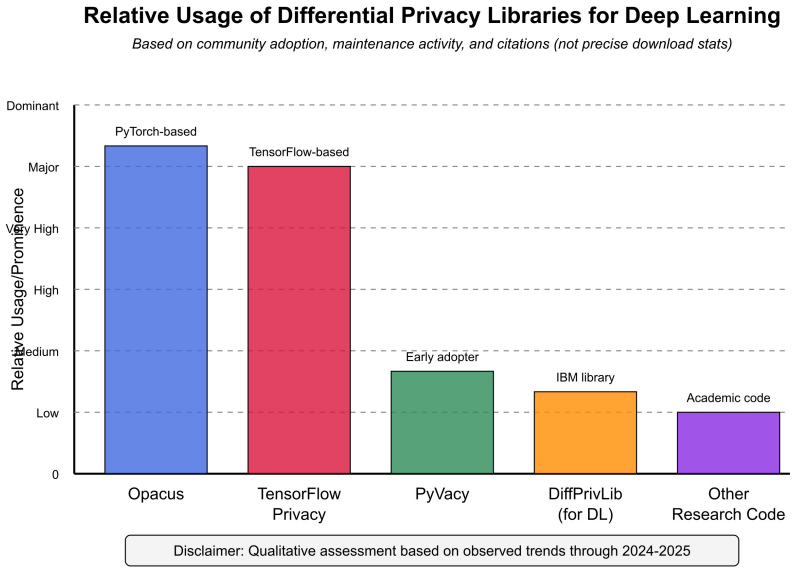


Figure 3.6: Illustrative usage prominence of DPDL Libraries (Qualitative).

Opacus and TensorFlow Privacy are currently the most maintained and widely used libraries for practical DPDL implementation due to their integration with major frameworks, performance optimizations, and robust privacy accounting features.

3.6 Survey of Key Algorithms and Methods

While gradient perturbation is the core idea, several specific algorithms and methodological refinements define the DPDL landscape.

3.6.1 DP-SGD (cornerstone Methode)

Differentially Private Stochastic Gradient Descent [ACG+16c] is the foundational algorithm for training deep models with DP. As detailed in Section 3.3.1, its key steps are per-example gradient computation, gradient norm clipping, noise addition (typically Gaussian), and aggregation. It relies on privacy accountants (Moments or RDP) for tracking the cumulative privacy loss. Almost all subsequent DPDL training algorithms are variations or extensions of DP-SGD.

3.6.2 Adaptative Clipping (Thakkar.al)

Recognizing the sensitivity of DP-SGD to the fixed clipping norm C, adaptive clipping methods propose adjusting C dynamically. For example, Thakkar2020UnderstandingDPML proposed setting C based on the median norm of per-example gradients in a batch or over recent history. Andrew2021DifferentiallyPrivateLearning explored similar ideas. The goal is to clip fewer gradients unnecessarily (when norms are small) while still bounding the influence of outliers, leading to potentially faster convergence and better utility without needing extensive hyperparameter tuning for C.

3.6.3 DP-FTRL, DP-Adam variants

Standard DP-SGD uses the basic SGD update rule. However, modern deep learning often relies on adaptive optimizers like Adam, RMSProp, or Adagrad. DP versions of these optimizers have been developed [Goo19b, PSL19]. Implementing DP-Adam, for example, requires careful handling of the optimizer's internal state (momentum and variance estimates) while ensuring the overall update remains differentially private. This usually involves applying the clipping and noising steps before the optimizer updates its state and parameters. These DP variants often lead to better convergence than basic DP-SGD on complex tasks, mirroring the benefits of their non-private counterparts. Follow-the-Regularized-Leader (FTRL) algorithms have also been adapted for DP settings, particularly in online or large-scale learning contexts [ASY+18].

3.6.4 DP via Bayesian Learning

An alternative perspective leverages Bayesian inference. Some approaches aim to achieve DP by carefully sampling from a posterior distribution over model parameters, where the posterior itself is influenced by the private data in a controlled way [WSS15, BKS18]. More recent work explores DP guarantees for Stochastic Gradient Langevin Dynamics (SGLD)

and related MCMC methods used in Bayesian deep learning [ZLL+21]. The connection often involves adding noise during the sampling process, analogous to DP-SGD, but framed within a Bayesian posterior sampling context. This can offer different theoretical guarantees and potentially incorporate prior knowledge more naturally [HKH23].

3.6.5 DP in Federated Learning (FedAvg+DPSGD)

Federated Learning (FL) trains models across decentralized devices (e.g., mobile phones) without centralizing user data [MMR+17b]. While FL enhances privacy by keeping raw data local, model updates exchanged during training can still leak information about user data [MSCS19]. DP is commonly integrated with FL algorithms like FedAvg to provide formal privacy guarantees.

Two main approaches exist: 1. Local DP: Each client adds noise to their model update *before* sending it to the server. This requires significant noise per client and corresponds to the Local DP model. 2. Central DP: Clients send their updates (or gradients) to a secure aggregator (server), which aggregates them and then adds noise *to the aggregated update* before updating the global model. This corresponds to the Central DP model applied in an FL context, often using DP-SGD principles [GKN17b, MRTZ18]. This typically allows for better utility for the same privacy budget compared to Local DP in FL.

Combining DP-SGD with FedAvg is a common and effective strategy for privacy-preserving collaborative learning [ASY+18, KMA+21].

3.7 Evaluation Metrics and Key Trade-offs in DPDL

Evaluating the efficacy and practicality of Differentially Private Deep Learning (DPDL) systems requires a multifaceted approach. Beyond assessing standard task performance (model utility), it is essential to quantify the level of privacy provided and consider the associated computational costs. The interplay between these dimensions inevitably leads to fundamental trade-offs that are central to research and practice in DPDL. Understanding these trade-offs is crucial for selecting appropriate methods, tuning parameters effectively, and setting realistic expectations for DPDL deployment.

3.7.1 Core Evaluation Dimensions

- **Privacy Guarantee:** Typically quantified by the differential privacy parameters (ϵ, δ), often computed using privacy accounting mechanisms like Rényi Differential Privacy [Mir17] or the Moments Accountant [ACG+16a]. Lower ϵ values indicate stronger privacy protection. δ represents the probability of catastrophic privacy failure and is usually kept very small (e.g., less than $1/N$, where N is the dataset size).

- **Model Utility:** Measures the performance of the trained DPDL model on its intended task. Common metrics include accuracy, precision, recall, F1-score, Area Under

the ROC Curve (AUC), or task-specific metrics. The choice often depends on the application and data characteristics (e.g., using F1-score for imbalanced datasets).

- **Computational Cost:** Encompasses factors like training time (wall-clock time, GPU hours), memory consumption (peak RAM/VRAM usage), and potentially inference latency.

3.7.2 Fundamental DPDL Trade-offs

The introduction of DP mechanisms into the deep learning pipeline creates inherent tensions between these evaluation dimensions.

The Privacy-Utility (P-U) Tradeoff

This is arguably the most studied and fundamental tradeoff in DPDL [DR14a, ACG+16a]. Differential Privacy requires injecting randomness (noise) and potentially bounding individual influence (clipping) during training.

- **Mechanism:** Adding noise (e.g., Gaussian noise in DP-SGD) directly perturbs the gradient signal used for model updates, potentially slowing convergence and leading the model to a less optimal minimum in the loss landscape. Clipping gradients limits the magnitude of updates, which can prevent divergence but also discard valuable information, particularly for complex tasks or samples far from the decision boundary.
- **Outcome:** Consequently, achieving stronger privacy guarantees (i.e., a lower privacy budget ϵ, typically requiring larger noise multiplier σ or smaller clipping norm C) generally leads to a measurable degradation in model utility metrics like accuracy, F1-score, or AUC compared to a non-privately trained model. Conversely, prioritizing higher utility often necessitates relaxing the privacy guarantee (allowing a higher ϵ).

This tradeoff is typically visualized as a curve plotting utility against the privacy budget ϵ, as conceptually shown in Figure 3.7. The exact shape and severity of this curve are highly dependent on the specific dataset, model architecture, DP algorithm variant, and hyperparameter tuning [PTS+21a]. A key goal of DPDL research is to develop techniques that "push" this curve upwards and to the left – achieving better utility for the same ϵ, or the same utility for a lower ϵ. Understanding this curve is essential for selecting an appropriate operating point that balances application requirements with privacy needs.

The Privacy-Computation Tradeoff

Implementing DPDL, particularly using the common DP-SGD algorithm, introduces significant computational overhead compared to standard, non-private training regimens.

- **Mechanism:** The primary bottleneck stems from the requirement to compute *per-example gradients* [ACG+16a]. Standard SGD/Adam compute a single gradient averaged over a mini-batch, which is highly efficient on modern hardware (GPUs/TPUs)

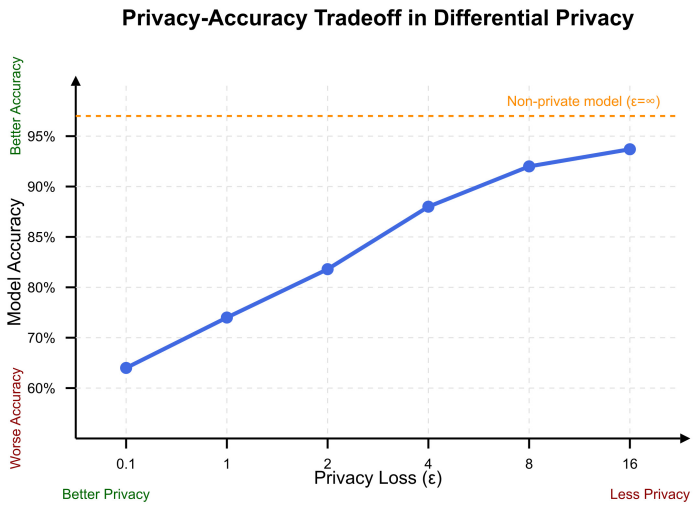


Figure 3.7: Conceptual illustration of the fundamental Privacy-Utility (P-U) Tradeoff in DPDL. Achieving stronger privacy guarantees (lower ϵ) generally results in lower model utility.

through vectorization. In contrast, DP-SGD needs access to individual gradients within the batch to perform norm clipping before averaging and adding noise. This breaks the standard batch processing optimization and often requires less efficient workarounds or specialized implementations, leading to increased memory usage (to store per-example gradients) and significantly longer computation time per training step.

- **Outcome:** Training a DPDL model typically takes considerably longer (sometimes orders of magnitude) than training its non-private counterpart, even with optimized libraries like Opacus [YSS+21] or TensorFlow Privacy [Pap19]. Furthermore, stronger privacy might indirectly increase the *total* computation time if the added noise necessitates more training epochs or iterations to reach a desired level of utility, compounding the per-step overhead (illustrated conceptually in Figure 3.8).

This tradeoff means that deploying DPDL can require substantially more computational resources (time, energy, cost), which can be a practical barrier, especially for large models or limited budgets. Research efforts focus on developing more computationally efficient DP algorithms and optimized library implementations to reduce this overhead.

The Privacy-Scalability Tradeoff

The combined effects of the P-U tradeoff and the Privacy-Computation tradeoff directly impact the **scalability** of DPDL to very large-scale machine learning scenarios.

- **Mechanism:** Scaling deep learning often involves using larger, more complex models (e.g., billions of parameters, like large language models or vision transformers) trained

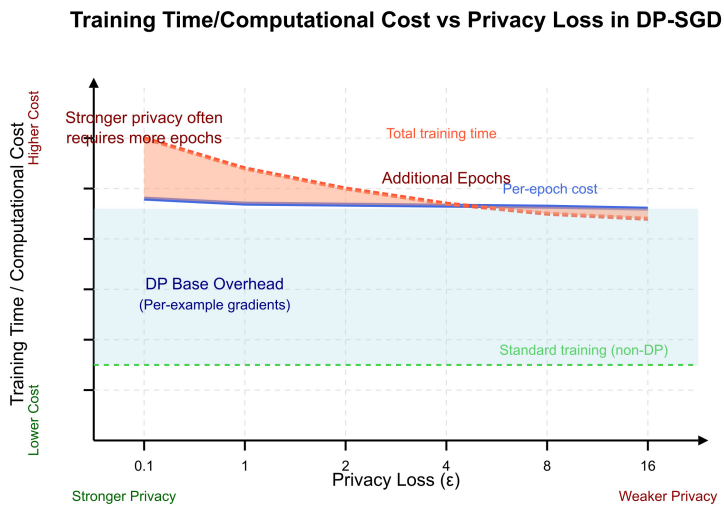


Figure 3.8: Conceptual representation of the trade-off between privacy strength and computational overhead. Stronger privacy often correlates with increased training time due to per-example gradient computation and potentially longer convergence.

on massive datasets (e.g., web-scale text or image corpora). Applying DPDL in such settings faces compounded challenges:

  - The utility drop associated with strong privacy (e.g., $\epsilon < 1$) tends to be more pronounced for highly complex tasks requiring nuanced learning.
  - The computational overhead of DP-SGD becomes increasingly prohibitive as both model size (affecting gradient computation cost) and dataset size (affecting total training time) grow. Memory requirements for per-example gradients can also become a bottleneck.

- **Outcome:** Consequently, training state-of-the-art, massive models with meaningful differential privacy guarantees remains a significant research challenge [DBH+]. While techniques like privacy amplification via subsampling (inherent in SGD) help [KLN+11b], achieving both high utility and strong privacy at extreme scales often requires sophisticated algorithmic innovations, significant computational resources, and potentially compromises on either the privacy level or the final model performance compared to non-private SOTA.

Current research directions to improve DP scalability include developing more communication-efficient algorithms for distributed settings, designing more efficient privacy accountants, exploring adaptive methods for setting DP parameters (C, σ), optimizing low-level implementations for specific hardware, and investigating alternative DP mechanisms beyond noise addition to gradients.

3.7.3 Navigating the Trade-offs

Effectively deploying DPDL requires consciously navigating these interconnected trade-offs. The optimal balance point depends heavily on the specific application's requirements regarding privacy sensitivity, minimum acceptable utility, available computational budget, and, as explored later in this thesis, fairness considerations. Techniques like hyperparameter optimization (Contribution 5.3.3) and Pareto frontier analysis (Contribution 5.3.2) are crucial tools for exploring and selecting appropriate configurations within this complex landscape.

3.7.4 Epsilon interpretation and setting

Choosing an appropriate value for ϵ (and δ) is crucial but non-trivial.

- **Interpretation:** Lower ϵ means stronger privacy, providing greater plausible deniability. However, there's no universal "safe" ϵ. Values like $\epsilon \leqslant 1$ are often cited as providing strong privacy, while ϵ between 1 and 10 might be acceptable in many practical settings. $\epsilon > 10$ is generally considered weak privacy. The interpretation also depends on the context, data sensitivity, and potential attack vectors.
- **δ:** Should be set to a cryptographically small value, typically less than $1/N$ (where N is dataset size), to ensure the privacy guarantee holds with high probability.
- **Setting ϵ:** Often determined by regulatory requirements, application context, or by evaluating the accuracy achievable at different ϵ levels and choosing an acceptable trade-off point. Setting ϵ *a priori* versus reporting the ϵ achieved for a target accuracy are both common practices.

Communicating the meaning of a specific (ϵ, δ) pair to non-experts remains a challenge [HGH+14].

3.7.5 Impact of Hyperparameters

DPDL introduces new hyperparameters that significantly affect the privacy-utility-computation balance.

Clipping Norm (C) Bounds the maximum L2 norm of per-example gradients.

- **Low C:** Introduces bias by clipping informative gradients, potentially slowing convergence and hurting final accuracy. Reduces the required noise scale for a fixed ϵ.
- **High C:** Less bias from clipping, but increases the sensitivity, requiring more noise to be added for the same ϵ, which can also hurt accuracy. Finding the optimal C (or using adaptive clipping) is crucial.

Noise Multiplier (z) This dimensionless parameter scales the amount of noise added relative to the gradient norm bound (sensitivity). A higher value of z generally leads to stronger privacy protection (lower ϵ) but potentially lower model utility.

Noise Standard Deviation The Gaussian noise added during the DP-SGD process is carefully calibrated based on the chosen noise multiplier (z) and the clipping norm (C). Specifically, the noise added to the *sum* of the clipped per-sample gradients within a mini-batch typically has a standard deviation, denoted here as σ_{noise}, calculated as:

$$\sigma_{\text{noise}} = z \times C$$

This ensures that the added noise is proportional to the sensitivity (C) of the clipped gradient sum.

- **High z:** More noise, stronger privacy (lower ϵ for fixed steps), but lower utility.
- **Low z:** Less noise, better utility, but weaker privacy (higher ϵ).

The noise multiplier is directly linked to the privacy budget via the accountant: given z, number of steps, batch size, and δ, the accountant computes the resulting ϵ.

Batch Size (B) Affects both computation and privacy.

- **Larger B:** Reduces the number of steps per epoch (faster wall-clock time per epoch potentially, despite higher cost per step), improves privacy amplification (lower ϵ for same noise/steps), but increases memory usage and computational cost per step due to more per-example gradients. Can also affect convergence dynamics.
- **Smaller B:** Less memory/computation per step, but more steps per epoch, weaker privacy amplification (higher ϵ or requires more noise).

The interaction between B, C, and z is complex and requires careful tuning [TRM+20].

3.7.6 Trade off in Training time and convergence

DP training typically requires more epochs to converge than non-private training due to the noise injected into gradients, which can obscure the true gradient direction. The learning rate schedule may need to be adjusted (often requiring smaller learning rates or different decay schemes). Adaptive clipping and DP optimizers like DP-Adam can sometimes mitigate the convergence slowdown compared to basic DP-SGD. The total training time is impacted by both the increased cost per step (per-example gradients) and potentially the increased number of steps needed for convergence.

3.8 Application Domains

DPDL is finding applications in various domains where sensitive data is used for training machine learning models.

3.8.1 Healthcare

Training DL models on patient data (e.g., medical imaging like X-rays/MRIs, Electronic Health Records (EHRs), genomic data) offers immense potential but carries high privacy risks. DPDL enables training diagnostic models (e.g., for tumor detection) or predictive models (e.g., for disease risk) while providing guarantees against leaking individual patient information [BJWW+19]. Challenges include the high dimensionality of medical data and the need for high accuracy and robustness.

3.8.2 Natural Language Processing

Language models trained on emails, messages, or documents can memorize sensitive text snippets [CLE+19]. DPDL techniques are used to train or fine-tune models like BERT or GPT variants for tasks like text classification, translation, or generation while protecting the privacy of the underlying text data [LTLH22, AGSS]. This is particularly relevant for applications like smart keyboard suggestions or analysis of sensitive corporate documents.

3.8.3 Finance

Financial institutions use DL for fraud detection, credit scoring, and algorithmic trading, often relying on sensitive customer transaction data. DPDL can help build these models while complying with financial regulations and protecting customer privacy [BRNM21]. Challenges include handling skewed data distributions (e.g., rare fraud events) under DP noise.

3.8.4 Recommendation Systems

Personalized recommendations (e.g., for products, movies, news) are based on user interaction history, which can be highly sensitive. DPDL can be used to train collaborative filtering or deep learning-based recommendation models that provide useful suggestions without revealing individual users' preferences or histories [SLL18], [TTK21]. Balancing personalization quality with privacy guarantees is key.

DPDL is being explored in other areas:

- **Autonomous Systems:** Training perception or control models using data collected from vehicle sensors or user interactions while protecting location privacy or driver behavior patterns.

- **IoT and Edge Computing:** Applying DPDL in federated settings for models trained on data from numerous IoT devices (e.g., smart homes, wearables) while preserving user privacy [ZZL+20].
- **Speech Recognition:** Training acoustic or language models for speech systems using private voice data. [FPN22, PAF+23]

As DL permeates more aspects of life, the need for DPDL in diverse applications is expected to grow.

3.9 Comparative Analysis of state of the art techniques

Comparing DPDL methods requires considering multiple factors: the privacy guarantee achieved, the impact on model utility, the computational cost, and the context (model, dataset, task). Table 3.5 provides a comparative overview of some representative works.

Discussion of limitations and gaps in existing works Despite significant progress, several limitations and gaps persist in the current state-of-the-art:

- **Utility Gap:** Even with advanced techniques, there is often a noticeable gap in performance between DP models and their non-private counterparts, especially at strong privacy levels ($\epsilon < 1$).
- **Scalability Limits:** Training very large models (e.g., foundation models with hundreds of billions of parameters) with meaningful DP guarantees remains computationally prohibitive or leads to poor utility.
- **Hyperparameter Sensitivity:** DPDL methods introduce sensitive hyperparameters (clipping norm, noise multiplier) that require careful tuning, often via grid search on proxy tasks or using heuristics. Adaptive methods help but don't eliminate this need entirely.
- **Composition Challenges:** Tracking privacy loss accurately over complex, multi-stage pipelines or adaptive compositions is still an active research area.
- **Generative Model Quality:** DP generative models (GANs, Diffusion Models) often struggle to match the fidelity and diversity of non-private models, especially at lower ϵ.
- **Theoretical Understanding:** While accountants provide bounds, a deeper understanding of how DP noise interacts with optimization dynamics and model generalization is still developing.
- **Beyond Gradient Perturbation:** Most work focuses on DP-SGD variants. Exploring fundamentally different approaches to achieve DPDL might yield breakthroughs.

Table 3.5: Comparative Analysis of Representative DPDL State-of-the-Art Works

Method/Paper	Model Type	Dataset	Task	ϵ	δ	Performance (Metric)	Key Contribution / Finding
Abadi et al. (2016) [ACG+16c]	MLP / CNN	MNIST / CIFAR-10	Image Classif.	2, 4, 8	10^{-5}	MNIST: 97% @ ϵ=2; CIFAR: 73% @ ϵ=2	Introduced DP-SGD with Moments Accountant; Feasibility study.
Papernot et al. (2018) [PSM+18]	CNN (student)	SVHN	Image Classif.	1.96	10^{-6}	SVHN: 94.4%	Scalable PATE framework, achieving good utility with reasonable ϵ.
McMahan et al. (2018) [MRTZ18]	LSTM	Reddit dataset	Next-word prediction	Variable (e.g., 10-15)	$1/N$	Showed feasibility of DP-FedAvg for language models.	DP in Federated Learning context.
Thakkar et al. (2020) [TRM+20]	CNN / LSTM	CIFAR-10 / StackOverflow	Image Classif. / NLP	3 - 7	10^{-5} / $N^{-1.1}$	Improved utility/convergence with adaptive clipping.	Analysis of DP-SGD params, proposed adaptive clipping.
Anil et al. (2021) [AGSS]	BERT / GPT-2	GLUE Benchmark / LM	NLP Tasks / Language Model	3 - 6 (fine-tuning)	$1/N$	Competitive performance on GLUE with DP fine-tuning.	DP fine-tuning for large pre-trained transformers.
De et al. (2022) [DBH+]	ResNet / ViT	ImageNet / CIFAR-100	Image Classif.	4 - 8	10^{-6}	Significant utility improvements on large models/datasets.	Techniques (e.g., high-norm fine-tuning) for scaling DPDL.
Dockhorn et al. (2021) [DCBJ22]	Diffusion Model	CelebA / CIFAR-10	Image Generation	10 - 100+	10^{-5}	High visual quality DP image generation.	First application of DP to Diffusion Models.

Note: Performance metrics and ϵ values are indicative and taken from the respective papers under specific experimental settings. Direct comparison requires careful consideration of hyperparameters, accounting methods, and evaluation protocols.

3.10 Current Challenges and Limitations

The practical deployment and further advancement of DPDL face several key challenges:

3.10.1 Scalability

The per-example gradient computation remains a major bottleneck, limiting the size of models and datasets that can be feasibly trained with DP. While optimizations exist, the inherent cost scales with model size and batch size. Training foundation models or models on web-scale data with strong DP guarantees is a significant hurdle.

3.10.2 Utility degradation

The fundamental trade-off between privacy and utility means that enforcing strong privacy often results in a noticeable drop in model performance. This gap can be particularly pronounced for complex tasks, fine-grained predictions, or minority subgroups within the data, potentially leading to fairness concerns [BPS19]. Reducing this utility gap without compromising privacy is a primary goal of ongoing research.

3.10.3 Privacy budget exhaustion

The privacy budget ϵ is finite and consumed over training iterations and potentially post-training analyses or queries. Managing this budget effectively, especially over long training runs or in scenarios involving continuous learning or multiple model releases, is critical. Overspending the budget invalidates the privacy guarantee. Optimal budget allocation strategies are needed.

3.10.4 Interpretability

Understanding *why* a DP model makes a certain prediction can be more challenging than for standard models. The injected noise can obscure feature importances or model decision boundaries. Developing interpretable DPDL methods or techniques to explain the impact of DP noise on model behaviour is an important area [BJWW+19].

3.11 Conclusion

This chapter has provided a comprehensive overview of the state-of-the-art in Differentially Private Deep Learning (DPDL). We began by establishing the fundamentals of Differential Privacy, including its formal definition, variants like ϵ-DP and (ϵ, δ)-DP, and core mechanisms such as Laplace and Gaussian noise addition. We then explored how these principles are adapted for deep learning, focusing on gradient perturbation via DP-SGD as the cornerstone technique, alongside alternative approaches like output perturbation (PATE) and input perturbation. Key components enabling practical DPDL,

such as advanced privacy accounting (Moments and Rényi accountants) and the concept of privacy amplification via subsampling, were discussed.

The survey covered the application of DPDL across diverse neural network architectures, from MLPs and CNNs to RNNs, GANs, and large Transformer models, highlighting the varying degrees of success and challenges encountered. We reviewed notable software frameworks like Opacus and TensorFlow Privacy that facilitate DPDL implementation. A discussion of key algorithms, including DP-SGD, adaptive clipping, DP-Adam variants, and integrations with Bayesian learning and Federated Learning, provided insight into the algorithmic landscape.

Crucially, we examined the inherent trade-offs in DPDL, particularly between privacy, utility, and computational cost, and discussed the interpretation and impact of critical hyperparameters like the privacy budget ϵ, clipping norm C, noise multiplier z, and batch size B. The growing application of DPDL in sensitive domains such as healthcare, NLP, finance, and recommendation systems underscores its practical relevance.

However, as the comparative analysis and discussion of challenges revealed, significant hurdles remain. Scalability limitations, the persistent utility gap compared to non-private models, privacy budget management, and interpretability issues are active areas of research. Future directions point towards tighter theoretical bounds, adaptive mechanisms tailored to specific architectures, integration with other PETs, and increased focus on real-world deployment and usability.

The insights gathered from this state-of-the-art review highlight the significant progress made in enabling privacy-preserving deep learning, while also underscoring the need for continued innovation. The fundamental challenge remains optimizing the delicate balance between providing strong, mathematically rigorous privacy guarantees and maintaining high model utility and computational feasibility. This sets the stage effectively for the subsequent chapter, which will delve into specific strategies, particularly focusing on fine-tuning techniques within DPDL, aimed at navigating and optimizing this critical Privacy-Utility trade-off.

Chapter 4

Fine Tuning in DPDL strategies to optimize the P-Acc Trade-off

***

4.1 Introduction

As established in Chapter 3, Differentially Private Deep Learning (DPDL) offers a principled framework for training machine learning models while providing rigorous privacy guarantees. However, this privacy comes at a cost, typically manifesting as a reduction in model utility (e.g., accuracy, F1-score, AUC) compared to non-privately trained counterparts. This inherent tension between privacy guarantees and model performance gives rise to the fundamental Privacy-Utility Trade-Off. Achieving a desirable balance – maximizing utility for a given, acceptable level of privacy (ϵ, δ), or minimizing the privacy loss ϵ for a target level of utility – is paramount for the practical adoption of DPDL.

Simply applying standard DPDL algorithms like DP-SGD often yields suboptimal results without careful consideration of the numerous factors influencing this trade-off. The process of systematically adjusting parameters, mechanisms, and even model architectures to navigate this complex landscape effectively can be broadly termed fine-tuning for the privacy-utility trade-off. This chapter delves into the state-of-the-art strategies employed to optimize this balance.

We begin by dissecting the nature of the trade-off itself and the key factors that shape it. We then categorize and explore various optimization techniques documented in the literature, ranging from sophisticated hyperparameter optimization methods (including traditional search, Bayesian optimization, and meta-heuristics tailored for multi-objective problems) to adaptive mechanisms within the training process (like adaptive clipping and noise scheduling). Furthermore, we examine model-centric and data-centric strategies, such as selective fine-tuning, model compression, data augmentation under DP, and leveraging transfer learning. Finally, we provide a comparative analysis of these strategies, discuss key insights and remaining gaps in the literature, and conclude by highlighting how this review informs the subsequent chapters of this thesis. The overarching goal is to provide a comprehensive understanding of the tools and techniques available to researchers and practitioners aiming to deploy DPDL models that are both private and performant.

4.2 Understanding the Trade-Off

The core challenge in DPDL lies in managing the injection of randomness (noise) required to guarantee privacy, while minimizing its detrimental effect on the learning process and the final model s utility.

4.2.1 Definition and formalization of the trade-off

The privacy-utility trade-off refers to the inverse relationship generally observed between the strength of the differential privacy guarantee (quantified primarily by ϵ, where lower ϵ means stronger privacy) and the performance or utility of the resulting model (e.g., measured by accuracy). Formally, let $\mathcal{A}$ be a DPDL training algorithm, D be the private dataset, θ_{DP} be the model parameters learned by $\mathcal{A}(D)$ satisfying (ϵ, δ)-DP, and $U(\theta_{DP})$ be a utility metric (e.g., test accuracy). The trade-off implies that, generally:

$$\text{Decreasing } \epsilon \implies \text{Decreasing } U(\theta_{DP}) \quad (\text{for fixed } \delta, \text{data, model, etc.}) \tag{4.1}$$

This occurs because achieving a lower ϵ typically requires adding more noise during training (e.g., via a larger noise multiplier in DP-SGD) or applying more aggressive sensitivity bounding (e.g., a smaller clipping norm C), both of which can obscure the learning signal within the gradients [ACG+16c]. The goal of optimization is to find strategies that push the boundary of this trade-off, achieving the best possible utility U for a target ϵ, or the lowest possible ϵ for a target U. This relationship is often visualized as a curve plotting utility against privacy loss ϵ.

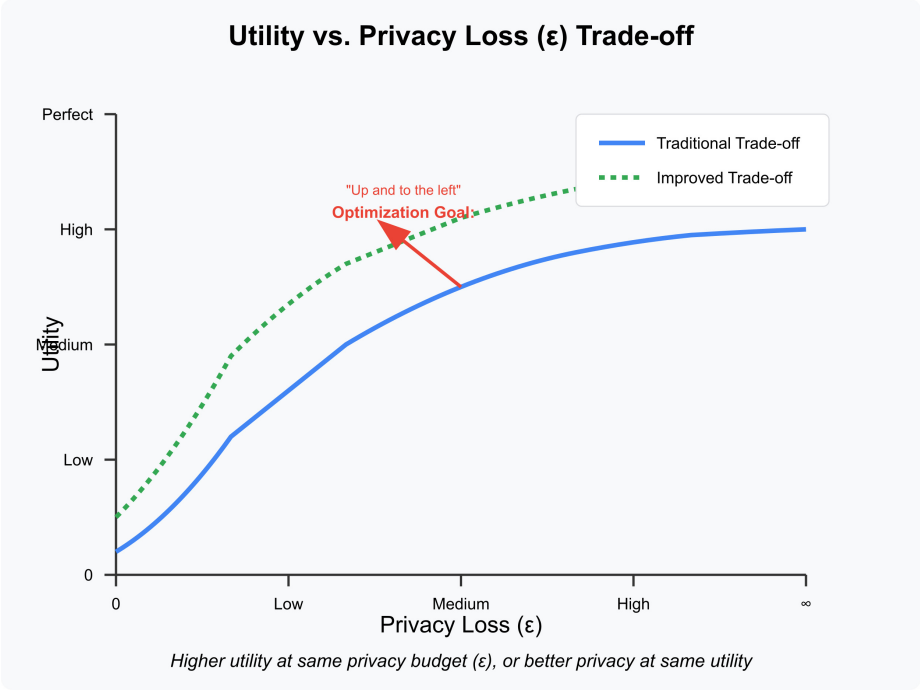


Figure 4.1: The Privacy-Utility Trade-off and the Goal of Optimization Strategies.

4.2.2 Influencing factors

Numerous factors interact to determine the specific shape and position of the privacy-utility curve for a given DPDL task:

Privacy parameters (ϵ,δ)

These directly define the target privacy level. As discussed, lower ϵ necessitates more noise or stricter clipping, directly impacting utility. δ allows for a small probability of failure in the privacy guarantee; a smaller δ generally requires slightly more noise for the same ϵ, especially when using Gaussian mechanisms and accountants like RDP [Mir17]. The choice of (ϵ, δ) sets the operational constraint for optimization.

Model architecture

Model complexity plays a significant role. Larger models with more parameters generally have higher capacity but may also be more sensitive to individual data points, potentially requiring more noise or more aggressive clipping for the same ϵ [DBH+]. The specific architecture (e.g., CNN, RNN, Transformer) also influences gradient characteristics and how noise propagates, affecting the trade-off [AGSS]. Architectures inherently robust to noise might perform better under DP.

Data characteristics

The size and quality of the training dataset are crucial. Larger datasets often allow for better utility under DP due to stronger privacy amplification from subsampling [BNS14, WBK19] and because the learning signal is stronger relative to the required noise. Data dimensionality, feature distribution, and inherent task difficulty also impact the achievable utility for a given privacy level. Highly complex or noisy datasets may suffer more utility degradation.

Training hyperparameters

These are the primary levers for fine-tuning within a given DPDL algorithm like DP-SGD. Key hyperparameters include:

- **Clipping Norm (C):** Balances gradient distortion (bias) and sensitivity reduction [TRM+20].
- **Noise Multiplier (z or σ):** Directly controls the amount of noise added relative to the sensitivity (C) [ACG+16c].
- **Batch Size (B):** Influences gradient variance, computational cost per step, and the degree of privacy amplification [TRM+20].
- **Learning Rate (η):** Affects convergence speed and stability in the presence of noise. Often needs careful tuning (typically lower than non-private training) [ACG+16c].
- **Number of Epochs/Steps (T):** Determines the total privacy budget consumption. Longer training consumes more budget but might be necessary for convergence.

Optimizing these hyperparameters is a central theme of this chapter.

4.2.3 Privacy accounting basics and how they influence utility

As introduced in Chapter 3, privacy accountants (e.g., Moments Accountant [ACG+16c], Rényi Accountant [Mir17]) track the cumulative privacy loss over iterative training steps, incorporating privacy amplification from subsampling. The tightness of the accountant directly impacts the utility trade-off. A tighter accountant provides a lower ϵ estimate for the same sequence of operations (same noise multiplier, batch size, steps). Equivalently, for a target final ϵ, a tighter accountant allows for the use of a *lower* noise multiplier during training, leading to potentially higher utility. Therefore, using state-of-the-art accountants (typically RDP-based) is crucial for optimizing the trade-off [WBK19].

4.3 Categories of Optimization Strategies in the Literature

Researchers have explored various strategies to navigate the privacy-utility landscape and find better operating points. These can be broadly categorized as follows:

4.3.1 Hyperparameter Optimization Techniques

Given the sensitivity of DPDL to hyperparameters (C, z, B, η, T), systematic optimization is often necessary.

Traditional Search Methods

Grid Search

This method exhaustively evaluates a predefined grid of hyperparameter values. For instance, trying combinations like $C \in \{0.5, 1.0, 1.5\}$, $z \in \{0.8, 1.0, 1.2\}$, $B \in \{256, 512\}$, etc.

- **Pros:** Simple to implement, guaranteed to find the best combination within the grid.
- **Cons:** Computationally very expensive, especially as the number of hyperparameters and their ranges increase (suffers from the curse of dimensionality). Infeasible for many DPDL settings due to the high cost of each training run.

Often used in initial DPDL papers for demonstrating feasibility over a small parameter range [ACG+16c].

Random Search

Instead of an exhaustive grid, Random Search samples hyperparameter combinations randomly from their respective distributions (e.g., uniform, log-uniform) for a fixed number of trials [BB12].

- **Pros:** Empirically shown to be more efficient than Grid Search for finding good parameters in high-dimensional spaces, as some parameters matter more than others. Easier to parallelize.

- **Cons:** No guarantee of finding the optimal parameters. Still requires many potentially expensive DPDL training runs.

A common practical alternative to Grid Search when computational budget is limited.

Bayesian Optimization

Bayesian Optimization (BO) is a model-based approach particularly suited for optimizing expensive black-box functions, like the utility of a DPDL model given its hyperparameters [SLA12].

- **Models the objective function:** It builds a probabilistic surrogate model (often a Gaussian Process) of the relationship between hyperparameters and the objective (e.g., validation accuracy for a fixed ϵ).
- **Uses acquisition functions:** An acquisition function (e.g., Expected Improvement, Upper Confidence Bound) guides the search by balancing exploration (trying uncertain regions) and exploitation (trying regions predicted to be good).
- **Applied in DPDL:** Recent works have successfully applied BO to tune DPDL hyperparameters, demonstrating significantly better sample efficiency (finding good parameters with fewer training runs) compared to Grid or Random Search [BWWZ20, KMA+21].
- **Pros:** Sample efficient, effective in lower-to-moderate dimensional spaces.
- **Cons:** Can be complex to implement correctly. Performance can degrade in very high-dimensional spaces. Surrogate model fitting adds some overhead.

Meta-Heuristic Optimization Methods

These methods use stochastic algorithms inspired by natural phenomena to find good solutions in complex optimization landscapes. Examples include Genetic Algorithms (GAs), Particle Swarm Optimization (PSO), Differential Evolution (DE), Bat Algorithm, etc. [Yan10a].

- **Inspired by natural processes:** GAs use selection, crossover, and mutation; PSO uses particle movement based on personal and global bests.
- **Suitable for DPDL:** The DPDL utility landscape can be noisy (due to stochastic gradients and DP noise) and non-convex, making it suitable for these gradient-free methods. They are good black-box optimizers when the objective function is expensive to evaluate.
- **Strengths:**

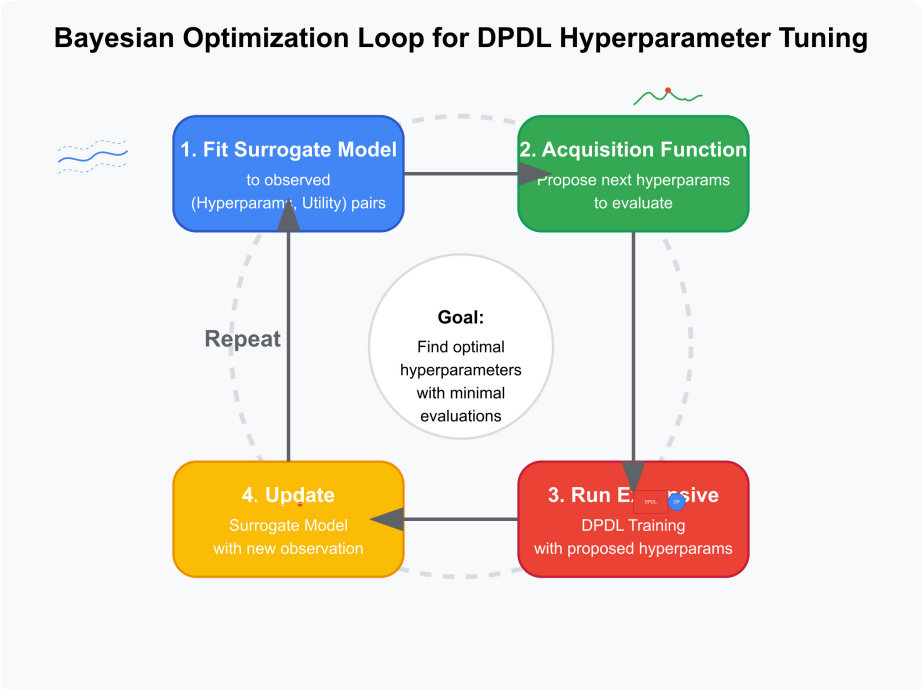


Figure 4.2: Conceptual Flow of Bayesian Optimization for DPDL Hyperparameter Tuning.

High exploration capacity: Can potentially escape local optima in noisy or complex landscapes.

Gradient-free: Suitable for optimizing parameters where gradients are unavailable or unreliable (e.g., discrete parameters like batch size, or complex adaptive schedules).

- **Examples:**
  1. Exploring the use of PSO or GAs to tune the combination of C, z, B, η simultaneously under DP constraints, potentially finding non-obvious interactions [GTF21].
  2. Evolutionary algorithms could search for optimal structures of adaptive clipping or noise scheduling policies. [Conceptual, needs citation if specific work exists]
- **Cons:** Can require many function evaluations (DPDL training runs). Parameter tuning *of the meta-heuristic itself* can be challenging. Convergence guarantees are often weaker than BO.

Bat Algorithm (BA) Among the various nature-inspired meta-heuristics, the Bat Algorithm (BA), developed by [Yan10b], has gained attention for its efficiency in solving global optimization problems. Inspired by the echolocation behaviour of microbats, BA provides an elegant balance between exploration (searching broadly across the parameter space) and exploitation (refining solutions in promising regions).

Core Mechanism: BA simulates the bats' strategy of varying pulse emission rates (r) and loudness (A) to find prey (optimal solutions). Each potential solution (a set of hyperparameters in our DPDL context) is represented as a virtual bat in the search space. The core iterative process involves updating each bat's position (x_i) and velocity (v_i) based on the following principles [Yan10b, YG12]:

1. **Frequency Tuning and Velocity Update:** Each bat i adjusts its frequency f_i and updates its velocity v_i^t at timestep t to move towards the current global best solution x_* found so far by the swarm:

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta \tag{4.2}$$
$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x_*)f_i \tag{4.3}$$

   where $\beta \in [0, 1]$ is a random vector drawn from a uniform distribution. $f_{\min}$ and $f_{\max}$ control the frequency range, influencing the step size and exploration capability.

2. **Position Update (Global Search):** The new position x_i^t is then calculated based on the updated velocity:

$$x_i^t = x_i^{t-1} + v_i^t \tag{4.4}$$

3. **Local Search Enhancement:** To intensify the search around promising areas, a local search step is introduced. For each bat, a random number is generated. If this number is greater than the bat's pulse rate r_i, a new solution is generated locally around the current *best* solution found so far:

$$x_{new} = x_* + \epsilon \bar{A}^t \tag{4.5}$$

   where $\epsilon \in [-1, 1]$ is a random number, and $\bar{A}^t$ is the average loudness of all bats at time t. This simulates bats searching intensively around areas where prey is detected.

4. **Loudness and Pulse Rate Adaptation:** As a bat finds a potential solution, its loudness A_i typically decreases, and its pulse emission rate r_i increases, simulating the behaviour of a bat closing in on prey. This is controlled by update rules:

$$A_i^{t+1} = \alpha A_i^t \tag{4.6}$$
$$r_i^{t+1} = r_i^0[1 - \exp(-\gamma t)] \tag{4.7}$$

   where α (typically $0 < \alpha < 1$) and γ ($\gamma > 0$) are constants, and r_i^0 is the initial pulse rate. A solution is accepted if a random number is less than the current loudness A_i and if the new solution x_{new} is better (higher utility/lower loss) than the bat's current position x_i^t. Upon acceptance, A_i and r_i are updated using Eq. 4.6 and 4.7.

This combination of global search driven by frequency tuning and adaptive local search driven by loudness/pulse rate allows BA to effectively explore the search space and converge towards optimal regions.

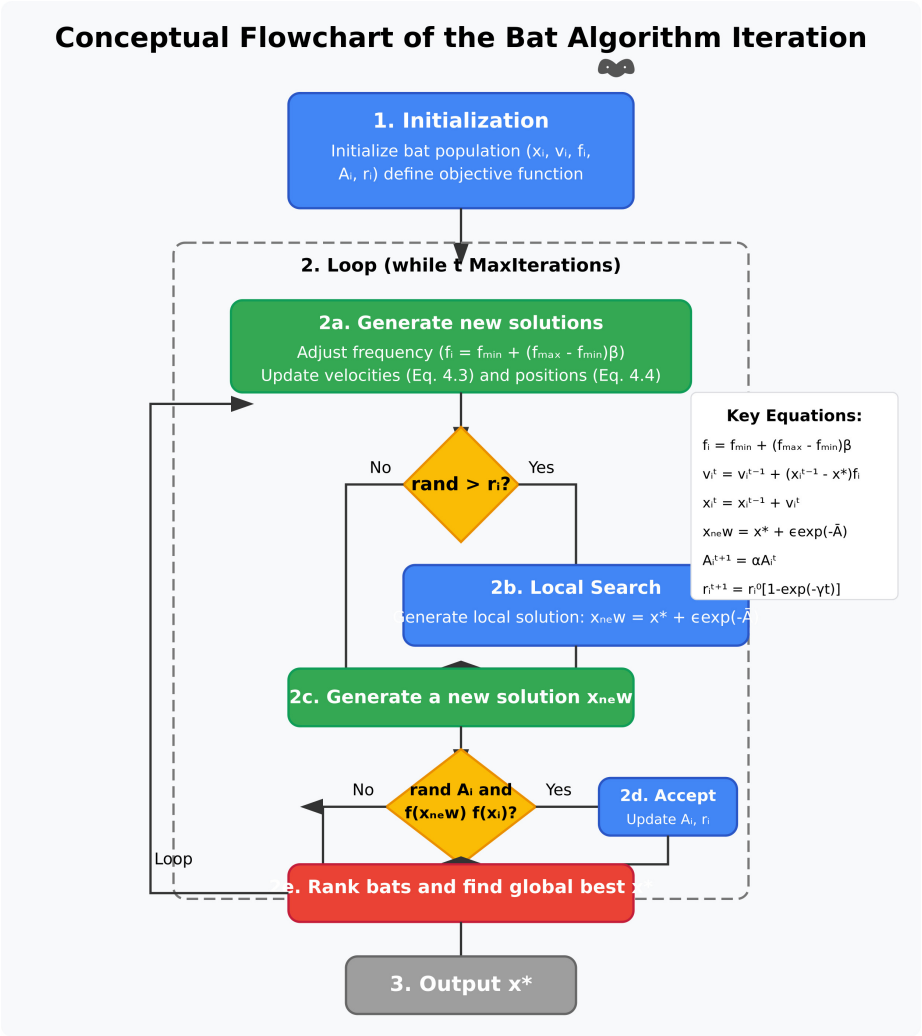


Figure 4.3: Conceptual Flowchart of the Bat Algorithm Iteration.

Relevance to DPDL Fine-Tuning: The characteristics of BA make it a potentially suitable candidate for optimizing the complex hyperparameter space of DPDL models (C, z, B, η, etc.):

- **Gradient-Free Optimization:** DPDL training involves inherent noise, and the relationship between hyperparameters and the final privacy-utility outcome is a complex, non-differentiable black-box function. BA, being gradient-free, does not require derivative information, making it applicable.
- **Balancing Exploration and Exploitation:** Finding good DPDL hyperparameters requires exploring diverse settings (exploration) but also refining promising configurations (exploitation). BA's mechanism attempts to inherently balance these two aspects through its global and local search strategies coupled with adaptive loudness and pulse rates.
- **Handling Noisy Objectives:** The stochastic nature of DPDL training (due to mini-

batch sampling and DP noise) results in noisy evaluations of the objective function (e.g., validation accuracy). Meta-heuristics like BA are often considered more robust to such noise compared to methods relying on precise function evaluations, although significant noise can still pose challenges.

Some studies have explored BA for hyperparameter optimization in general machine learning contexts [BSPK20, HV14], suggesting its potential applicability to the specific challenges of DPDL tuning.

Challenges in Adapting BA for DPDL Fine-Tuning: Despite its potential, directly applying BA (or similar meta-heuristics) to fine-tune DPDL models presents significant challenges that must be addressed:

1. **Expensive Fitness Evaluation:** This is arguably the most critical bottleneck. Each fitness evaluation in BA requires training a DPDL model with a specific set of hyperparameters (a bat's position x_i) and measuring its utility (e.g., accuracy) and potentially calculating its privacy cost ϵ. Given that DPDL training is computationally intensive (due to per-example gradients, noise addition, etc.), performing hundreds or thousands of these evaluations, as typically required by BA for convergence, can become prohibitively expensive in terms of time and resources [TRM+20, DBH+]. Strategies to reduce the number of evaluations or use surrogates are often necessary.

2. **High-Dimensional Search Space:** The number of hyperparameters involved in DPDL tuning can be large (clipping norm, noise multiplier, learning rate, batch size, potentially parameters for adaptive schedules, layer-specific parameters, etc.). Standard BA's performance can degrade as the dimensionality of the search space increases [YG12]. Modifications or hybridization might be needed to handle high dimensions effectively.

3. **Parameter Tuning of BA Itself:** BA introduces its own set of control parameters (e.g., population size N, loudness factor α, pulse rate factor γ, frequency range $[f_{\min}, f_{\max}]$). Tuning these parameters effectively for the specific DPDL optimization problem adds another layer of complexity. Poor choices can lead to premature convergence or inefficient search.

4. **Handling Stochasticity:** Both BA (through random numbers β, ϵ) and the DPDL fitness evaluation (due to SGD and DP noise) are stochastic. This high degree of randomness can make it difficult for BA to reliably identify the true optimum, potentially leading to convergence to suboptimal solutions or requiring multiple runs to assess stability. Techniques like re-evaluating promising solutions or using noise-handling strategies within BA might be necessary.

5. **Constraint Handling:** DPDL optimization is often constrained, most notably by a target privacy budget ϵ. Standard BA does not explicitly handle constraints. Common

approaches involve using penalty functions (penalizing solutions that violate the ϵ constraint in the fitness function) or repair mechanisms (modifying infeasible solutions to become feasible), but these add complexity and require careful design [Coe02].

6. **Multi-Objective Adaptation:** Directly optimizing the trade-off often requires a multi-objective approach (minimizing ϵ and maximizing utility simultaneously). While multi-objective versions of BA (MOBAs) exist [GYAT13], they add further complexity compared to the standard algorithm and might require managing a Pareto front of solutions, increasing computational demands. Alternatively, using a weighted sum approach simplifies it to single-objective BA but requires defining appropriate weights.

Addressing these challenges is crucial for successfully leveraging BA or other meta-heuristics for efficient and effective fine-tuning of DPDL models to optimize the privacy-utility trade-off. This motivates the development of adaptive BA variants specifically designed for expensive, noisy DPDL evaluations.

Multi-Objective Optimization

Explicitly treating DPDL tuning as a multi-objective optimization (MOO) problem acknowledges the inherent trade-off. The goal is often to find the Pareto front, representing solutions where one objective (e.g., utility) cannot be improved without degrading the other (e.g., privacy loss ϵ).

- **Formulation:** Minimize $f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))$, where $\mathbf{x}$ represents hyperparameters, $f_1(\mathbf{x}) = 1 - \text{Accuracy}(\mathbf{x})$ (utility loss), and $f_2(\mathbf{x}) = \epsilon(\mathbf{x})$ (privacy loss).
- **Techniques:**

  1. **Pareto Front Exploration:** Algorithms like NSGA-II (Non-dominated Sorting Genetic Algorithm II) [DPAM00] or MOEA/D (Multi-Objective Evolutionary Algorithm based on Decomposition) [ZL07] directly search for a set of Pareto-optimal solutions. Meta-heuristics are particularly powerful here.

  2. **Weighted Sum Methods:** Combine objectives into a single scalar function: $w_1 f_1(\mathbf{x}) + w_2 f_2(\mathbf{x})$. Requires choosing weights w_1, w_2, which implicitly selects a point on the Pareto front. Can be solved using single-objective optimizers (like BO or PSO).

  3. **Constraint-based Optimization (ϵ-constraint):** Optimize one objective subject to a constraint on the other. E.g., Maximize Accuracy($\mathbf{x}$) subject to $\epsilon(\mathbf{x}) \leqslant \epsilon_{target}$. Can be solved with single-objective methods by reformulating or using specialized algorithms.

- **Relevance:** MOO provides a more complete picture of the trade-off landscape than single-objective methods optimizing for a fixed ϵ or a fixed utility target [Mie99]. Some studies have started applying MOO concepts to DPDL [ZWHC21].

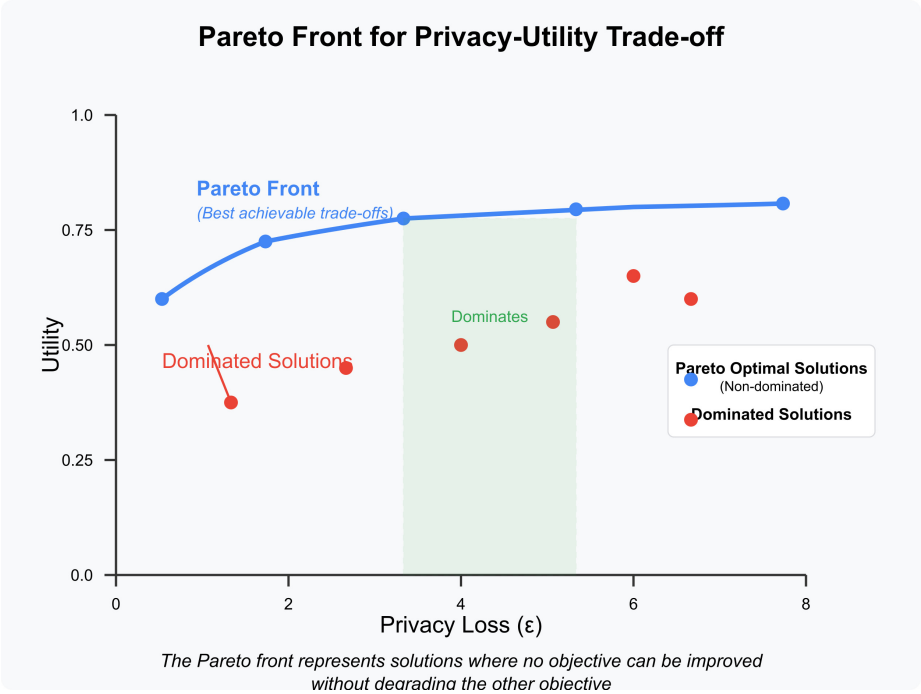


Figure 4.4: Conceptual Pareto Front for Multi-Objective Optimization of Privacy and Utility.

Sensitivity Analysis & Empirical Tuning

Before embarking on expensive automated HPO, performing sensitivity analysis can be insightful. This involves systematically varying one hyperparameter while keeping others fixed (or at reasonable defaults) and observing the impact on both privacy (ϵ) and utility.

- **Identify critical parameters:** Helps understand which hyperparameters (e.g., clipping norm C, noise multiplier z) have the most significant impact on the trade-off for a specific task [TRM+20].
- **Guide HPO:** Results can inform the search ranges and priorities for more advanced HPO methods.
- **Empirical Tuning:** Often, practitioners rely on empirical tuning guided by experience, sensitivity analysis, and results reported in the literature for similar tasks. While less systematic, it can be a practical starting point.

4.3.2 Examples from recent papers on tuning DP-SGD parameters

Several recent works highlight the importance and impact of HPO:

- [DBH+] demonstrated significant utility gains on large models (ResNet, ViT) on ImageNet/CIFAR-100 by carefully tuning DP-SGD hyperparameters, particularly using large batch sizes and finding appropriate clipping norms, often via extensive empirical search guided by insights from smaller experiments.

- [AGSS] successfully fine-tuned large language models (BERT, GPT-2) with DP by carefully selecting learning rates, batch sizes, and clipping norms, showing that good hyperparameter choices are crucial for achieving strong performance on downstream tasks like GLUE benchmarks within reasonable ϵ budgets (e.g., $\epsilon \approx 3-6$).

- Work utilizing Bayesian Optimization [BWWZ20] has shown reductions in the number of trials needed to find competitive hyperparameter settings compared to random search for specific DPDL algorithms.

These examples underscore that while DP-SGD provides the mechanism, meticulous tuning is required to unlock its potential utility for a given privacy budget.

4.4 Adaptive Mechanisms

Instead of relying solely on fixed hyperparameters found via offline optimization, adaptive mechanisms adjust parameters *during* the training process based on observed statistics.

4.4.1 Adaptive Gradient Clipping

The optimal clipping norm C can vary depending on the model, data, and stage of training. Adaptive clipping aims to dynamically adjust C.

- **Motivation:** A fixed C might be too high early in training (adding unnecessary noise) or too low later (causing excessive bias by clipping informative gradients).

- **Techniques:**

  Quantile-based Clipping: Set C at each step (or periodically) to a specific quantile (e.g., median, 90th percentile) of the per-example gradient norms observed in the current batch or over a recent window [TRM+20, ATMR21]. This adapts C to the current distribution of gradient magnitudes.

  Per-layer Clipping: Apply different clipping norms to different layers of the network, potentially based on the expected gradient magnitudes or sensitivity of each layer [PTS+21a].

Specific Algorithms: Methods like AdaClip [PSL19] propose specific update rules for adapting C. PATEClip [PTS+21a] links clipping to activation functions.

- **Benefits:** Can improve convergence speed and final utility by balancing bias and variance more effectively throughout training, often reducing the need for exhaustive manual tuning of C.

4.4.2 Noise Scheduling Techniques

The amount of noise added might not need to be constant throughout training.

- **Motivation:** Early in training, large gradient updates might be more robust to noise, while later stages with smaller updates might benefit from reduced noise to allow fine-tuning (provided the total privacy budget is respected).
- **Techniques:**

Noise Decay: Gradually decrease the noise multiplier z over epochs or steps, allocating more privacy budget to later, more sensitive stages of training [PSL19]. Requires careful budget management using privacy accountants.

Curriculum Learning under DP: Start training on easier tasks or with higher noise, gradually increasing task difficulty or decreasing noise, potentially improving robustness and final utility [KVH23].

- **Challenges:** Designing optimal schedules is non-trivial and adds complexity to privacy accounting. Poorly designed schedules might exhaust the budget too quickly or starve early training phases of sufficient signal.

4.5 Model-Centric Strategies

Optimizing the trade-off can also involve modifying the model architecture or the training strategy applied to it.

4.5.1 Selective Fine-Tuning

Instead of training the entire model with DP noise, focus the private training on specific parts.

- **Approach:** Freeze most layers of a pre-trained (often publicly) model and fine-tune only the final layer(s) or specific lightweight adaptation modules (like Adapters [HGJ+19] or LoRA [HSW+]) with DP [LTLH22, AGSS, BCJ+23].

- **Benefit:** Reduces the number of parameters being updated with noisy gradients, decreasing the overall noise impact and computational cost. The sensitivity is calculated only over the parameters being trained. Effective for transfer learning scenarios.

4.5.2 Model Pruning & Sparsity Constraints

Reducing model complexity can improve utility under DP.

- **Approach:** Apply pruning techniques (removing weights or connections) or enforce sparsity constraints during or after training [DAS+21].
- **Rationale:** Smaller models often have lower sensitivity or are inherently more robust to noise. Pruning can reduce the dimensionality affected by noise.
- **Challenges:** Integrating pruning with DP training requires care, as the pruning process itself might depend on private data. Performing pruning *after* DP training is simpler but might be suboptimal.

4.5.3 Architecture Search for DP Compatibility

Neural Architecture Search (NAS) aims to automatically find optimal model architectures. Tailoring NAS for DPDL involves searching for architectures that perform well *specifically under DP training constraints*.

- **Approach:** Use NAS algorithms where the evaluation phase involves training candidate architectures with DP-SGD and assessing their privacy-utility trade-off [LCR+21].
- **Goal:** Discover architectures inherently robust to noise or with gradient characteristics amenable to DP training (e.g., lower gradient variance or magnitudes).
- **Challenges:** Extremely computationally expensive due to the nested optimization (NAS search + multiple DPDL training runs).

4.5.4 Lightweight models under DP

Applying DPDL to inherently efficient and compact architectures (e.g., MobileNet [HZC+17], SqueezeNet [IHM+]) can be beneficial.

- **Rationale:** These models are designed for resource-constrained environments and often have fewer parameters, which can lead to better utility under DP constraints compared to overly large models, especially for simpler tasks or tighter ϵ budgets.
- **Application:** Evaluating the performance of standard lightweight models under DP provides practical options for deployment [PTS+21a].

4.6 Data-Centric Approaches

Modifying or augmenting the data used for training can also improve the trade-off.

4.6.1 Data Augmentation and Synthetic Data

Increasing the amount or diversity of training data often improves robustness and generalization, which can be beneficial under DP.

- **Standard Augmentation:** Applying standard data augmentations (e.g., image rotations, crops, text back-translation) during DP training can improve utility, similar to non-private settings [DBH+]. The privacy cost is implicitly handled as augmented data originates from the private samples within the batch.
- **DP Synthetic Data:** Train a DP generative model (e.g., DP-GAN [XWD+18, JYvdS19], DP-VAE [DPV], DP Diffusion Model [DCBJ22]) on the private data first. Then, train the final task model on the synthetic data (which inherits the DP guarantee) or augment the private training set with DP synthetic data. This can sometimes improve utility by providing more diverse training examples without additional privacy cost for the task model training itself, although generating high-quality DP synthetic data remains challenging.

4.6.2 Public Pretraining and Private Fine-Tuning

This is one of the most effective strategies for achieving high utility under DP, especially for complex tasks and large models [LTLH22, AGSS, DBH+].

- **Approach:** 1. Pre-train a model on a large, publicly available dataset without DP constraints. This allows the model to learn general features effectively. 2. Fine-tune the pre-trained model (either fully or selectively, see Section 4.5.1) on the smaller, private dataset using DPDL techniques like DP-SGD.
- **Benefit:** The knowledge gained during public pre-training significantly reduces the burden on the private fine-tuning phase. The model starts from a good initialization, requiring fewer updates on the private data, which helps conserve the privacy budget and mitigate the negative impact of noise. This often leads to substantially better utility for the same ϵ compared to training from scratch on private data alone.

4.7 Privacy Accounting and Composition Methods (Revisited)

While not optimization strategies *per se*, the choice and application of privacy accounting methods directly influence the tuning process and achievable trade-offs.

- **Moments Accountant** [ACG+16c] and Rényi Differential Privacy (RDP) [Mir17]: As mentioned, these provide tighter bounds on privacy loss compared to basic composition theorems. Using RDP, often considered state-of-the-art for Gaussian mechanisms and subsampling, allows for less noise for a target (ϵ, δ), directly enabling better utility. Libraries like Opacus [YSS+21] and TensorFlow Privacy [Goo19a] implement these accountants.
- **Privacy Amplification via Subsampling** [BNS14, WBK19]: Leveraging amplification through appropriate batch sizes (B) is crucial. Larger batch sizes give better amplification (lower ϵ per step for the same noise), influencing the choice of B during HPO, balanced against computational costs.
- **Effects on Tuning:** When tuning hyperparameters (especially noise multiplier z, batch size B, and number of steps T), the accountant is used *within the optimization loop* to calculate the resulting ϵ for each candidate configuration. The accuracy and tightness of the accountant are therefore critical for meaningful optimization towards a specific privacy target.

4.8 Comparative Analysis of Optimization Strategies

Table 4.1 summarizes the discussed optimization strategies.

Table 4.1: Comparative Analysis of DPDL Optimization Strategies for Privacy-Utility

Strategy Type	**Specific Method(s)**	**Typical Application Context**	**Strengths**	**Limitations / Considerations**
Hyperparameter Optimization	Grid Search, Random Search	Finding good C, σ, B, η	Simple (Grid), More efficient (Random)	Computationally very expensive, Often suboptimal (esp. Grid), Sensitive to search space definition
	Bayesian Opt.	Tuning C, z, η (continuous params)	Sample efficient, Good for expensive evaluations	Can struggle with high dims/discrete params, Requires surrogate model tuning
	Meta-heuristics (GA, PSO)	Tuning multiple params, discrete/-continuous, noisy objectives	Good exploration, Gradient-free, Handles complex interactions	Can need many evaluations, Meta-heuristic params need tuning
	Multi-Objective Opt. (NSGA-II, MOEA/D)	Explicitly exploring Pareto front of (Utility, ϵ)	Provides full trade-off view, Principled approach	Computationally expensive, More complex implementation
Adaptive Mechanisms	Adaptive Clipping (Quantile-based, AdaClip)	Dynamically adjusting C during training	Improves convergence/utility, Reduces manual tuning of C	Can add overhead, Potential instability if not well-designed
	Noise Scheduling	Varying noise multiplier z over time	Potentially better budget allocation	Optimal schedule design is hard, Complex accounting

Table 4.1: Comparative Analysis of DPDL Optimization Strategies for Privacy-Utility (Continued)

Strategy Type	**Specific Method(s)**	**Typical Application Context**	**Strengths**	**Limitations / Considerations**
Model-Centric	Selective Fine-Tuning (Adapters, LoRA)	Transfer learning with DP	Reduces noise impact, Computationally cheaper DP phase	Relies on good pre-trained model, May limit adaptability
	Pruning / Sparsity	Reducing model complexity	Smaller models may be more robust to noise	Interaction with DP needs care, May hurt capacity
	DP-aware NAS	Finding noise-robust architectures	Potentially optimal architectures for DP	Extremely computationally expensive
	Lightweight Models	Using MobileNet, etc.	Practical for resource/privacy constraints	May have lower peak utility than larger models
Data-Centric	Augmentation (Standard)	General training improvement	Simple to integrate, Often boosts robustness	Benefits might be limited
	DP Synthetic Data (DP-GAN/VAE)	Data augmentation/sharing	Decouples generation privacy from task model training	Generating high-quality DP data is hard, Utility often limited
	Public Pretrain + Private Fine-tune	Transfer learning	Highly effective for utility, Leverages public data	Requires suitable public dataset, Pre-training is costly
Accounting	RDP Accountant	Calculating ϵ during training/HPO	Tightest bounds for Gaussian mech., Enables lower noise	Implementation details matter (numerical stability)

nd landscape environment AFTER the table environment

Discussion In practice, a combination of strategies is often most effective.

- **Scalability:** Automated HPO methods (BO, Meta-heuristics, MOO) are generally more scalable in terms of finding good solutions than manual tuning or grid search, but their own computational cost can be high. Adaptive mechanisms add relatively low overhead per step. Transfer learning (Public Pretrain + Private Fine-tune) combined with selective tuning is arguably one of the most scalable approaches for achieving high utility in complex domains like NLP and Vision.
- **Generalizability:** Hyperparameters tuned for one dataset/model may not generalize well to others. Adaptive mechanisms aim to be more robust across different stages of training. Transfer learning relies on the generalizability of features learned during pre-training. Strategies based on fundamental principles (like using tight accountants) are inherently general.
- **Effectiveness:** Public pre-training + private fine-tuning consistently emerges as highly effective for state-of-the-art models [LTLH22, DBH+]. Within the private training phase, careful HPO (often using BO or guided random search) and adaptive clipping are crucial practical steps. For scenarios without large public datasets, optimizing DP-SGD directly with sophisticated HPO or adaptive methods becomes even more critical. DP synthetic data generation shows promise but often lags in utility compared to direct DP training or transfer learning.

4.9 Key Insights and Gaps in the Literature

The review of optimization strategies reveals several key insights and persistent gaps:

- **HPO Cost vs. Benefit:** While advanced HPO methods like Bayesian Optimization and MOO offer systematic ways to tune DPDL, their computational cost (requiring multiple full DPDL training runs) remains a significant barrier, especially for large models. Research into more sample-efficient HPO techniques specifically for the noisy, expensive DPDL setting is needed.
- **Overfitting Hyperparameters:** There s a risk of overfitting hyperparameters to a specific validation set, especially when the number of HPO trials is large relative to the validation set size or task stability. How DP interacts with hyperparameter generalization is not fully understood.
- **Adaptive Methods Complexity:** While adaptive clipping is becoming standard, more complex adaptive strategies (e.g., sophisticated noise schedules, adaptive batch sizes) add complexity to implementation and privacy analysis. Their practical benefits over well-tuned fixed hyperparameters need further validation across diverse tasks.

- **Lack of Standardized Benchmarks:** Comparing different optimization strategies rigorously is difficult due to variations in base models, datasets, privacy accountants used, and exact (ϵ, δ) targets across studies. Standardized benchmarks and evaluation protocols for the privacy-utility trade-off would benefit the field.
- **Need for Holistic Multi-Objective Approaches:** Many studies still focus on optimizing utility for a fixed ϵ. Truly embracing multi-objective optimization frameworks (considering both ϵ and utility simultaneously during the search) could lead to a better understanding and navigation of the Pareto front, although this increases complexity.
- **Theoretical Understanding:** A deeper theoretical understanding of how specific hyperparameters and model properties influence the privacy-utility trade-off under DP optimization dynamics could guide more principled tuning strategies beyond empirical search.

4.10 Conclusion

Optimizing the privacy-utility trade-off is central to making Differentially Private Deep Learning practical and effective. This chapter has surveyed a wide array of strategies employed for this purpose, broadly categorized into hyperparameter optimization techniques, adaptive mechanisms during training, model-centric modifications, and data-centric approaches.

We reviewed traditional HPO methods like Grid and Random Search, highlighting their limitations, and discussed more advanced techniques including Bayesian Optimization, meta-heuristics (like GA and PSO), and multi-objective optimization frameworks designed to explicitly handle the trade-off. Adaptive methods, particularly adaptive gradient clipping, offer dynamic adjustments during training to improve stability and performance. Model-centric strategies, such as selective fine-tuning of pre-trained models, pruning, and potentially DP-aware architecture search, aim to make the model itself more amenable to privacy-preserving training. Data-centric approaches, including augmentation, DP synthetic data generation, and especially the highly effective paradigm of public pre-training followed by private fine-tuning, leverage data properties to enhance utility under privacy constraints. The crucial role of tight privacy accounting (e.g., RDP) in enabling better trade-offs was also reiterated.

Our comparative analysis and discussion highlighted the strengths and weaknesses of these approaches, noting the high effectiveness of transfer learning combined with careful tuning, while also pointing out the significant computational costs associated with many HPO techniques. Key gaps remain, including the need for more efficient optimization algorithms, standardized benchmarks, and a deeper theoretical understanding of the factors governing the trade-off.

This comprehensive review of existing optimization and fine-tuning strategies sets the stage for the contributions of this thesis. The identified limitations, particularly the

computational burden of extensive hyperparameter searches and the potential benefits of more integrated adaptive or model-aware approaches, motivate the investigation into the focus of novel meta-heuristic optimization frameworks tailored for DPDL hyperparameter tuning. The following chapter will introduce and detail this proposed approach, aiming to address some of the challenges outlined herein and push the boundary of achievable privacy-utility performance in DPDL.

Chapter 5

Proposed Approaches to Balance the P-U Tradeoff in DPDL

***

5.1 Introduction

DPDL has emerged as a crucial technology for enabling the analysis of sensitive data, particularly in domains like healthcare, without compromising individual privacy. By injecting carefully calibrated noise during the training process, DPDL

offers formal, mathematical guarantees against privacy breaches inherent in traditional deep learning models. However, the practical adoption of DPDL is often hindered by the well-known Privacy-Utility (P-U) tradeoff: the mechanisms that ensure privacy can potentially degrade the resulting model's performance or utility. Furthermore, deploying DPDL effectively in real-world scenarios introduces additional complexities related to computational efficiency, the need for structured decision-making regarding tradeoff levels, and the critical importance of ensuring fairness, especially when dealing with inherently imbalanced datasets common in medical applications.

This chapter outlines the core contributions of this thesis, presenting four distinct yet interconnected approaches designed to systematically address these challenges and advance the state-of-the-art in balancing privacy, utility, and fairness in DPDL. We begin by establishing the foundational potential of carefully tuned DPDL models on real-world medical data. We then introduce a framework based on Pareto optimality for navigating the P-U tradeoff in a principled manner. Recognizing the computational demands of tuning, we subsequently propose and evaluate efficient hyperparameter optimization (HPO) strategies tailored for DPDL. Finally, we extend the scope beyond the P-U dyad to investigate the critical impact of data imbalance on the Privacy-Utility (P-U) trade-off.

Throughout this chapter, the focus remains strictly on the *proposed approaches* – the conceptual frameworks, methodologies, and analytical strategies developed in this research. Specific implementation details, experimental setups, datasets used (primarily real-world, imbalanced medical datasets across the contributions), and empirical results are deferred to subsequent chapters (specifically Chapter 6). The goal here is to clearly articulate the logical progression of our research and the rationale behind each proposed contribution towards achieving a more practical, efficient, and equitable application of DPDL.

5.2 Problem Statement

The central problem addressed in this thesis is the multifaceted challenge of developing and deploying Differentially Private Deep Learning (DPDL) models that achieve an optimal and justifiable balance between formal privacy guarantees, high model utility, and equitable performance, particularly when operating under the constraints of real-world applications. This overarching problem can be decomposed into several key sub-problems:

1. **Quantifying the Achievable P-U Tradeoff:** Establishing the realistic performance boundaries of DPDL on practical, often complex and imbalanced, datasets when hyperparameters are carefully tuned. This involves challenging pessimistic assumptions about utility loss and understanding the sensitivity of the tradeoff to parameter choices.

2. **Navigating the P-U Tradeoff Systematically:** Developing principled methods to explore the space of possible P-U configurations and identify the set of optimal

compromises (Pareto frontier). This requires moving beyond finding single "good" configurations to understanding the entire spectrum of efficient tradeoffs, potentially considering architectural choices alongside hyperparameters.

3. **Efficiently Optimizing DPDL Configurations:** Devising and evaluating computationally efficient Hyperparameter Optimization (HPO) strategies capable of finding near-optimal DPDL configurations without the prohibitive cost of exhaustive search. This includes comparing established HPO methods and exploring novel algorithms suited to the specific characteristics of DPDL optimization (e.g., noisy gradients, expensive evaluations) across different data types.

4. **Integrating Fairness into the DPDL Framework under Imbalance:** Characterizing and mitigating the adverse effects of data imbalance on the fairness of DPDL models. This involves extending the P-U analysis to the three-way Privacy-Utility-Fairness (P-U-F) nexus, quantifying how DP mechanisms interact with imbalanced data distributions, and understanding the resulting impact on performance equity across different groups.

5. **Enhancing DPDL Utility through Synergistic Techniques like Data Augmentation:** Investigating whether and how auxiliary techniques, such as data augmentation, can be effectively integrated with DPDL to enhance model utility, particularly in data-scarce or challenging domains like medical imaging. This involves understanding the conditions under which such synergies occur, the potential impact on privacy accounting, and the overall effect on the P-U tradeoff.

Conceptually, the overall problem involves navigating a multi-objective optimization space, seeking configurations $(\theta^*, \mathcal{A}^*)$ characterized by hyperparameters θ and architecture $\mathcal{A}$ that simultaneously optimize competing objectives:

$$\text{Optimize} \quad \Big(\underbrace{-\epsilon(\theta, \mathcal{A})}_{\text{Maximize Privacy}} \;,\; \underbrace{\text{Utility}(\theta, \mathcal{A})}_{\text{Maximize Utility}} \Big)$$

subject to constraints imposed by the dataset D (including its potential imbalance), computational resources, and specific application requirements. This thesis proposes a series of approaches designed to systematically tackle these interconnected problems.

5.3 Contributions

This thesis makes five primary contributions towards addressing the challenges outlined above. Each contribution proposes a specific approach or framework, building logically upon the insights from the previous ones. These contributions are detailed in the following subsections. All approaches, beyond the initial potential assessment, are primarily developed and conceptually validated in the context of real-world, imbalanced medical datasets, reflecting the target domain of this research.

5.3.1 Contribution 1: Establishing the Potential and Sensitivity of Tuned DPDL Models

This first contribution aims to address these points by directly tackling the research question: *"Under what configurations and through which mechanisms can differentially private deep learning (DP-DL) models achieve performance comparable to, or even exceeding, non-private counterparts on real-world medical datasets?"* Specifically, it seeks to:

1. Empirically investigate the achievable performance baseline of DPDL models on representative, real-world medical datasets when hyperparameters are carefully fine-tuned.

2. Assess the sensitivity of the resulting P-U balance to variations in key hyperparameters (C, σ, B, η, etc.).

3. Challenge the assumption of inevitable, significant performance degradation by identifying conditions and configurations under which tuned DP models exhibit competitive utility.

Establishing this potential and understanding the parameter sensitivity is a crucial first step, motivating the need for the more advanced optimization and analysis frameworks explored in subsequent contributions.

To address the problem outlined above, this contribution proposes a **systematic empirical investigation focused on the role of meticulous hyperparameter tuning in determining the achievable P-U balance** for DPDL models applied to real-world medical data. The core idea is that the perceived limitations of DPDL may often stem from suboptimal or insufficiently explored hyperparameter configurations rather than solely from the fundamental P-U tradeoff.

The proposed approach centers on the following conceptual elements:

1. **Focus on Real-World Medical Data:** The investigation leverages relevant medical datasets from the outset to ensure findings have direct applicability to sensitive, high-stakes domains. While these datasets are often imbalanced, the primary focus here is on maximizing overall utility under DP constraints through tuning, rather than specifically analyzing fairness implications (which is deferred to Contribution 4).

2. **Comprehensive Hyperparameter Exploration:** The approach involves exploring a wide range of hyperparameters critical to both the deep learning training process and the differential privacy mechanism, as illustrated conceptually in Figure 5.1. This includes standard parameters (η, B) and DP-specific parameters (C, σ, target ϵ, δ). An extensive search (e.g., grid search within feasible limits for foundational analysis) is conceptualized to map out a significant portion of the configuration space for these specific datasets.

3. **Comparative Analysis:** The performance (utility) of tuned DPDL models is systematically compared against non-private baseline models trained under similarly optimized conditions on the same data. This allows for a direct assessment of the *actual* utility cost incurred by DP, challenging assumptions and identifying scenarios where the gap is minimal or non-existent.

4. **Sensitivity Assessment:** Beyond finding high-performing configurations, the approach aims to characterize the *sensitivity* of the P-U balance to changes in individual hyperparameters (summarized conceptually in Table 5.1). This provides practical insights for DPDL practitioners.

5. **Formalizing the Objective (Conceptually):** This contribution explores the achievable maxima within the space defined by:

$$\max_{\mathbf{H}} \quad \text{Utility}(\text{Model}(D, \mathbf{H})) \quad \text{s.t.} \quad \text{PrivacyMechanism}(\text{Model}(D, \mathbf{H})) \leqslant (\epsilon, \delta)$$

for specific medical datasets D and target privacy levels (ϵ, δ), aiming to understand the utility landscape $U(\mathbf{H}|\epsilon, \delta, D)$.

By executing this systematic investigation, Contribution 1 provides foundational empirical evidence on the true potential of well-tuned DPDL, establishing a realistic baseline and motivating the search for more efficient optimization methods and broader analyses including fairness.

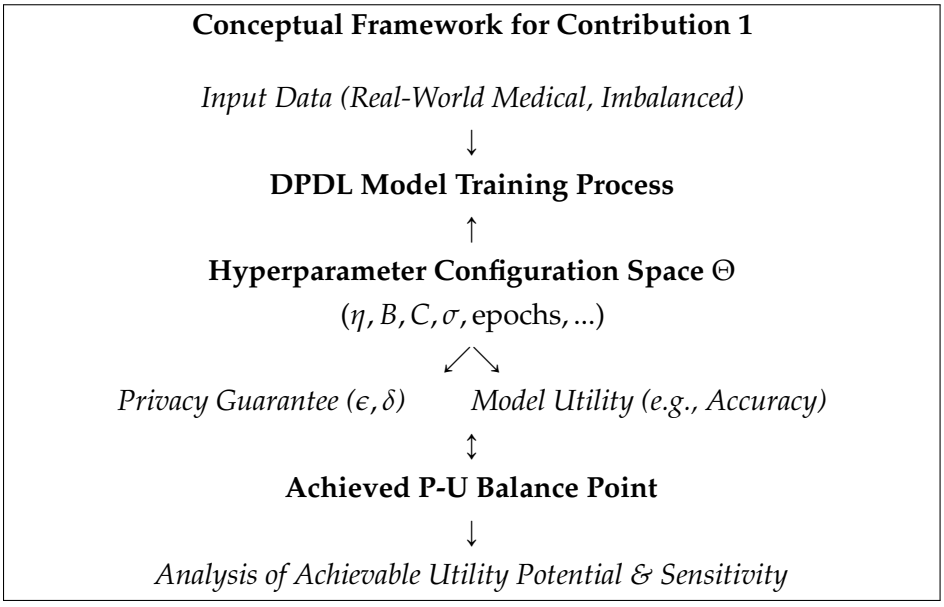


Figure 5.1: Conceptual diagram illustrating the proposed approach for Contribution 1: Investigating DPDL potential via hyperparameter tuning on real-world medical data.

Table 5.1: Key Hyperparameters Investigated (Contribution 1) and Their Conceptual Role

Hyperparameter	Expected Role / Impact on P-U Balance
Clipping Norm (C)	Controls per-sample gradient magnitude; impacts sensitivity (privacy) and gradient information (utility).
Noise Multiplier (σ)	Directly scales Gaussian noise; primary control for privacy level (ϵ). Higher $\sigma \implies$ more privacy, potentially less utility.
Target Epsilon (ϵ)	Desired privacy budget; influences required σ or training iterations.
Target Delta (δ)	Probability of privacy failure; typically small, affects budget calculation.
Learning Rate (η)	Optimization step size; impacts convergence and final utility (both DP/non-DP).
Batch Size (B)	Affects gradient variance and DP noise averaging; impacts privacy accounting and utility.
Epochs (E)	Training duration; impacts privacy budget accumulation and model convergence.
Optimizer	Algorithm (e.g., DP-SGD, DP-Adam); interacts differently with noise/clipping.

5.3.2 Contribution 2: A Pareto-Optimal Framework for P-U Settlement in DPDL

Motivation and Problem Statement

While Contribution 5.3.1 established that careful tuning can yield performant DPDL models, identifying individual high-utility points doesn't fully address the inherent *tradeoff* nature of the problem. Practitioners often need to make deliberate choices along the P-U spectrum – accepting slightly lower utility for stronger privacy, or vice versa. Standard tuning might find *a* good point, but fails to reveal the landscape of *all optimal* compromises. Furthermore, the choice of model architecture ($\mathcal{A}$) itself significantly interacts with DP mechanisms and data [CWZ+22], meaning the best P-U balance might require considering architecture alongside hyperparameters ($\mathbf{H}$).

This motivates the need for a more structured approach based on multi-objective optimization. The problem becomes: How can we systematically map out the entire frontier of non-dominated P-U solutions (the Pareto frontier) to provide a comprehensive view for decision-making, potentially incorporating architectural choices? This contribution addresses the need for:

1. Explicit formulation of DPDL configuration as a multi-objective optimization problem.

2. A method to visualize and analyze the complete set of optimal P-U tradeoffs.

3. Inclusion of model architecture comparison within the tradeoff analysis framework.

4. A practical tool to guide stakeholders in selecting configurations aligned with specific operational needs and risk tolerance (e.g., GDPR compliance).

The core question is: *"How can the Pareto frontier concept be leveraged to systematically explore and select optimal privacy-utility configurations and potentially suitable architectures for DPDL"*

This contribution proposes leveraging the concept of **Pareto optimality** to systematically analyze and navigate the P-U tradeoff. The central idea is to identify the set of DPDL configurations (defined by $\mathbf{H}$ and possibly $\mathcal{A}$) for which it is impossible to improve one objective (utility) without degrading the other (privacy). This set forms the Pareto frontier, representing the optimal P-U "settlement" points.

The proposed approach comprises these conceptual components:

1. **Multi-Objective Formulation:** Formalize the DPDL configuration task as finding the Pareto set P^* for the vector objective function (typically minimizing privacy loss and maximizing utility):

$$\min_{(\mathbf{H},\mathcal{A})} \quad f(\mathbf{H},\mathcal{A}) = (\epsilon(\mathbf{H},\mathcal{A}), \quad \mathcal{L}(\mathbf{H},\mathcal{A}))$$

   where ϵ is the privacy loss and $\mathcal{L}$ is a utility loss metric (e.g., $1 - \text{Accuracy}$, classification error). A configuration $(\mathbf{H}_1, \mathcal{A}_1)$ dominates $(\mathbf{H}_2, \mathcal{A}_2)$ if $f(\mathbf{H}_1, \mathcal{A}_1)$ is better or equal in all objectives and strictly better in at least one. P^* contains all non-dominated configurations.

2. **Systematic Exploration for Frontier Generation:** Employ DPDL training (using DP-SGD) across a range of hyperparameter settings ($\mathbf{H}$) and potentially different architectures ($\mathcal{A}$). For each resulting model, record the achieved $(\epsilon, \text{Utility})$ pair. This exploration could use methods like grid search (if feasible for the scope) or more advanced sampling. The focus here is on generating points to *map* the frontier.

3. **Pareto Frontier Identification and Visualization:** From the collected set of $(\epsilon, \text{Utility})$ points, identify the subset that constitutes the empirical Pareto frontier. Visualize this frontier on a 2D plot (Utility vs. ϵ, see Figure 5.2) to provide an intuitive representation of the optimal tradeoffs.

4. **Architecture Comparison via Frontiers:** Generate and compare the Pareto frontiers achieved by different model architectures ($\mathcal{A}_1, \mathcal{A}_2, ...$) on the same dataset (as outlined conceptually in Table 5.2). This allows assessing which architecture offers a fundamentally better P-U profile.

5. **Decision Support Framework:** Utilize the visualized Pareto frontier as a tool for informed decision-making. Stakeholders can identify points corresponding to specific privacy budget constraints ($\epsilon \leqslant \epsilon_{max}$) or utility requirements (Utility $\geqslant U_{min}$), or select balanced points near the "knee" of the frontier.

This Pareto-based approach offers a holistic view of the P-U landscape, empowering practitioners to make justifiable choices based on the full spectrum of optimal compromises.

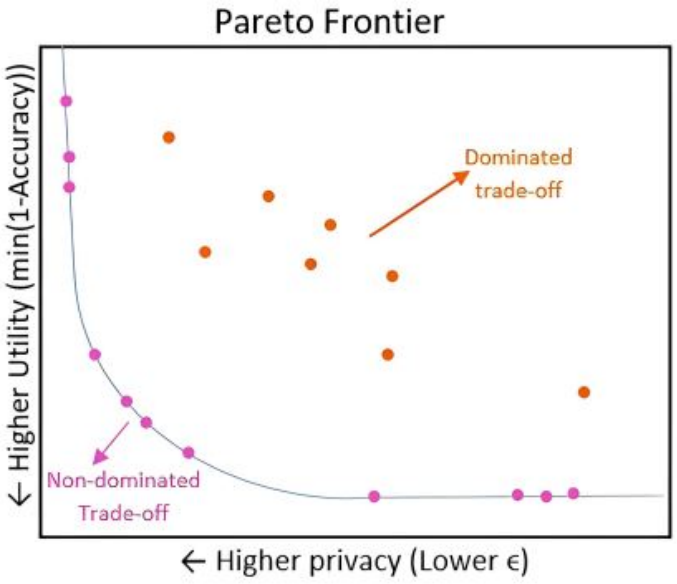


Figure 5.2: Conceptual visualization of a Pareto frontier providing a basis for P-U settlement.[RFA23]

Table 5.2: Conceptual Approach for Architecture Comparison using Pareto Frontiers

Aspect	**Description**
Goal	Identify architecture offering superior P-U characteristics for a given task/dataset.
Method	Generate P-U Pareto frontiers for each candidate architecture ($\mathcal{A}_1, \mathcal{A}_2$) via hyperparameter exploration (**H**).
Analysis	Overlay and compare the frontiers. An architecture whose frontier is consistently "north-west" (higher utility for same ϵ, or lower ϵ for same utility) is generally preferable.
Outcome	Evidence-based selection of architecture best suited for DP deployment.

5.3.3 Contribution 3: Systematic Frameworks for Efficient Privacy-Utility Optimization

Contributions 5.3.1 and 5.3.2 highlighted the importance of tuning and provided a framework for understanding optimal tradeoffs. However, the methods used or implied for exploring the hyperparameter space (like grid search, even for mapping Pareto frontiers) rapidly become computationally infeasible for complex DPDL models and high-dimensional search spaces. Training a single DPDL model can be resource-intensive; evaluating hundreds or thousands of configurations is often impractical.

This creates a critical need for **computationally efficient Hyperparameter Optimization (HPO) techniques** specifically adapted for the DPDL context. We need methods that can intelligently navigate the complex P-U landscape and converge to near-optimal configurations (ideally approximating the Pareto frontier) using significantly fewer model training evaluations than exhaustive search. Furthermore, the performance of different HPO strategies (random search, Bayesian optimization, metaheuristics) might vary depending on the data modality (tabular vs. imaging) and the specific nature of the DPDL objective function. There's also an opportunity to explore HPO algorithms, like the Bat Algorithm, which haven't been applied to DPDL but whose characteristics might be beneficial.

This contribution addresses the research question: *"How do different efficient HPO strategies, including the Bat Algorithm, compare in their ability to effectively and efficiently identify optimal P-U tradeoffs for DPDL models across diverse medical data modalities?"* The goal is to:

1. Propose a framework for comparing efficient HPO techniques for DPDL P-U optimization.

2. Evaluate established HPO methods (Random Search, Bayesian Optimization) alongside a novel application (Bat Algorithm) and a baseline (Grid Search, where feasible).

3. Assess the generalizability and scalability of these HPO methods across different types of real-world, imbalanced medical data (tabular and imaging).

4. Provide practical guidance on selecting appropriate HPO strategies for DPDL based on efficiency and effectiveness in approximating the P-U Pareto frontier.

This contribution proposes a **framework for the systematic evaluation and comparison of efficient HPO techniques applied to finding optimal P-U configurations in DPDL**. Instead of just mapping the landscape, the focus shifts to the *efficiency* of the search process itself. The core idea is to assess how effectively different algorithms can approximate the Pareto frontier within a limited computational budget.

The proposed framework comprises these elements:

1. **Unified Optimization Goal:** Frame the task for all HPO methods: find configurations θ in the hyperparameter space Θ that optimize a P-U objective, often by seeking points on or near the Pareto frontier P^* defined by $(\epsilon(\theta), \text{Utility}(\theta))$.

2. **Comparative HPO Strategy Implementation:** Implement and systematically compare multiple HPO strategies, representing different search philosophies: * *Grid Search (GS):* Baseline (if computationally viable for context). * *Random Search (RS):* Efficiency baseline based on random sampling. * *Bayesian Optimization (BO):* Sample-efficient sequential model-based optimization. * *Bat Algorithm (BA):* Novel metaheuristic approach evaluated for DPDL HPO. (See Table 5.3, and Figure5.3).

3. **Standardized DPDL Evaluation Oracle:** Each HPO strategy interacts with a common DPDL training and evaluation function. Given a hyperparameter set θ suggested by the HPO algorithm, this function trains the DPDL model, computes its privacy cost $\epsilon(\theta)$, evaluates its utility $A(\theta)$, and returns these values to the optimizer.

4. **Budgeted Evaluation and Pareto Approximation Comparison:** Run each HPO algorithm for a fixed computational budget (a maximum number of DPDL model evaluations). Compare the quality of the Pareto frontier approximations generated by each method within this budget. Metrics could include hypervolume indicator, distance to known optimal points (if available), or visual inspection of the achieved frontiers.

5. **Validation Across Diverse Medical Data:** Evaluate the relative performance of the HPO strategies on multiple imbalanced medical datasets representing different modalities to assess robustness and scalability.

This comparative approach aims to provide evidence-based recommendations for choosing efficient HPO methods for practical DPDL deployment, highlighting the performance of established techniques and the potential of novel algorithms like the Bat Algorithm in this specific context.

Table 5.3: Hyperparameter Optimization Strategies Compared (Contribution 3)

HPO Strategy	**Conceptual Search Mechanism / Role**
Grid Search (GS)	Exhaustive search over predefined grid; Baseline for quality.
Random Search (RS)	Random sampling of configurations; Baseline for efficiency.
Bayesian Optimization (BO)	Builds surrogate model, selects points balancing exploration/exploitation; Aims for sample efficiency.
Bat Algorithm (BA)	Metaheuristic using echolocation analogy; Explores novel search dynamics for DPDL HPO.

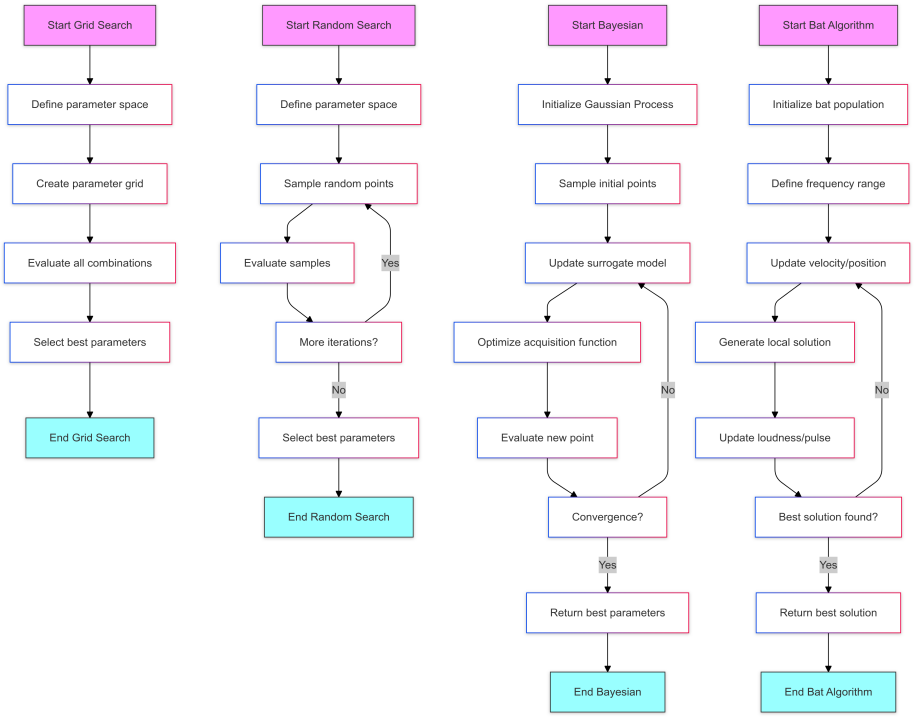


Figure 5.3: Flowchart of each Optimization Method.[BHB25]

5.3.4 Contribution 4: Analyzing the Impact of Data Imbalance on the Privacy-Utility-Fairness Nexus

Previous contributions focused on optimizing the P-U tradeoff, assuming implicitly or explicitly that aggregate utility metrics were sufficient performance indicators. However, as noted, real-world medical datasets are frequently **imbalanced**, containing disproportionately few samples from minority classes (e.g. patients with a rare disease). Standard ML models often exhibit bias on such data, performing poorly on these crucial minority groups, raising significant **fairness** concerns [Cha04].

Applying DPDL might worsen this fairness deficit. The noise added for privacy could disproportionately affect the learning from already scarce minority samples, and gradient clipping might suppress their influence relative to the majority class [BPS19]. This creates a critical knowledge gap: how do DP mechanisms interact with inherent data imbalance, and what are the consequences for fairness across different groups? Optimizing solely for the P-U balance might inadvertently lead to highly unfair models. We must analyze the interconnected **Privacy-Utility-Fairness (P-U-F) nexus** [PS21] under the realistic condition of data imbalance.

This contribution addresses the research question: *"How does the application of DPDL, particularly when tuned for optimal P-U balance, impact model fairness when trained on realistically imbalanced medical data?"* Specifically, it aims to:

1. Establish performance and fairness benchmarks for both non-private and DPDL

models specifically trained on imbalanced medical data.

2. Empirically investigate the complex, interrelated P-U-F trade-offs, quantifying how fairness metrics change with varying privacy levels (ϵ) and utility outcomes.

3. Characterize the adverse effects of data imbalance combined with DP noise and clipping on minority class performance.

4. Assess whether standard P-U optimization strategies lead to acceptable fairness outcomes or if fairness-aware approaches are needed.

To dissect the complex interactions within the P-U-F nexus on imbalanced data, this contribution proposes a **systematic empirical investigation framework**. This framework explicitly incorporates fairness assessment alongside privacy and utility evaluations, focusing on how DP impacts performance disparities between majority and minority classes.

The proposed approach involves these key elements:

1. **Comparative Training on Imbalanced Data:** Train both non-private baseline models and DPDL models (using DP-SGD) on the same imbalanced medical datasets. This allows isolating the impact of DP on fairness, separate from the baseline imbalance effect.

2. **Systematic Variation of Privacy Levels:** Evaluate DPDL models across a range of privacy guarantees (varying ϵ, primarily via noise multiplier σ) to observe how fairness metrics change as privacy constraints become stricter.

3. **Multi-Dimensional P-U-F Evaluation:** For each model configuration, measure and record: * *Privacy (P):* Calculated ϵ for the DP configuration. * *Utility (U):* Aggregate metrics suitable for imbalance. * *Fairness (F):* Metrics quantifying performance disparity between majority/minority classes, minority class F1-score). See Table 5.4.

This empirical investigation aims to provide crucial insights into the fairness implications of deploying DPDL on imbalanced data, highlighting potential risks and informing the need for fairness-aware DPDL methods.

5.3.5 Contribution 5: Enhancing Differentially Private Medical Image Analysis through Data Augmentation

Contributions 1-4 focused on optimizing DPDL through hyperparameter tuning, tradeoff navigation, and fairness analysis. However, the inherent utility loss in DPDL, especially with limited medical datasets (a common scenario in medical imaging), remains a significant challenge. Data augmentation is a widely successful technique in standard deep learning to improve model generalization and performance by artificially expanding the training dataset, particularly when original data is scarce or expensive to acquire.

Table 5.4: Conceptual Fairness Metrics for P-U-F Analysis under Imbalance (Contribution 4)

Fairness Focus	**Example Metrics / Rationale**
Minority Class Performance	Recall (TPR), Precision, F1-score calculated *only* for the minority class. (How well is the rare group handled?)
Performance Parity	Difference/Ratio of TPR between groups (Equality of Opportunity). Difference/Ratio of Accuracy between groups. (Are groups treated equally well?)
Error Disparity	Difference/Ratio of False Negative Rate (FNR) between groups. (Are critical errors biased towards one group?)

This raises a critical question: *Can data augmentation synergistically interact with DPDL mechanisms to enhance utility, particularly in medical imaging ?* The noise introduced by DP might make models more susceptible to overfitting on small datasets, a problem data augmentation typically combats. Conversely, augmenting data might introduce samples that are harder to classify or that increase gradient variance, potentially interacting negatively with DP's noise addition or gradient clipping. The impact of data augmentation on the privacy accounting itself (e.g., if augmentations are considered as expanding the dataset or as transformations of existing samples) also needs careful consideration. There is a need to empirically investigate this interplay to understand if augmentation can be a viable strategy to further improve the P-U tradeoff in DPDL for medical applications.

This contribution addresses the research question: *"How does the application of data augmentation techniques, specifically image rotation, impact the privacy-utility tradeoff when training DPDL models on medical image datasets, and is this impact dependent on the stringency of the privacy guarantee?"* The specific aims are to:

1. Investigate the effect of a common data augmentation technique (image rotation) on the performance of DPDL models trained on a representative medical imaging dataset.

2. Analyze how this effect varies across different levels of privacy protection (i.e., different ϵ values).

3. Determine if data augmentation can help mitigate the utility loss typically associated with DPDL in this context, or if it presents additional challenges.

4. Provide preliminary insights into the practical considerations of combining data augmentation with DPDL for medical image analysis.

To explore the potential synergy between DPDL and data augmentation, this contribution proposes an **exploratory empirical investigation focused on a specific med-**

ical imaging task and a common augmentation technique. The core idea is to assess whether augmenting the training data can improve the utility of DPDL models without unduly compromising privacy or leading to unstable training, particularly under varying privacy constraints.

The proposed approach comprises the following conceptual elements:

1. **Targeted Medical Imaging Application:** The investigation centers on a specific medical imaging dataset (e.g., PneumoniaMNIST for chest X-ray classification) to provide concrete, context-relevant findings.
2. **Specific Data Augmentation Technique:** A well-understood and commonly used image augmentation method, such as random rotations, is selected for initial investigation. This allows for a focused analysis of its interaction with DPDL.
3. **Comparative DPDL Training Scenarios:** The study involves training and evaluating DPDL models under several configurations:
    - Baseline non-private model (no DP, no augmentation).
    - Non-private model with data augmentation only.
    - DPDL model without data augmentation (at various ϵ levels).
    - DPDL model with data augmentation (at the same ϵ levels).
4. **P-U Tradeoff Analysis Across Privacy Levels:** For each DPDL scenario (with and without augmentation), the achieved privacy (ϵ) and utility (e.g., accuracy, AUC) are measured. The impact of augmentation is then assessed by comparing the P-U balance points achieved with and without augmentation at different ϵ levels.
5. **Assessment of Interaction Effects:** Analyze whether augmentation consistently improves utility, has no significant effect, or degrades utility when combined with DPDL, and if this interaction is dependent on the strength of the privacy guarantee (i.e., the value of ϵ). (Conceptualized in Figure 5.4).

This exploratory study aims to provide initial empirical evidence on the viability and potential complexities of using data augmentation as a supplementary technique to enhance DPDL performance in medical imaging, laying the groundwork for more extensive future investigations into various augmentation methods and their interactions with different DPDL mechanisms.

5.4 Conclusion

This chapter has laid out the proposed approaches underpinning the five main contributions of this thesis, all aimed at advancing the practical application and understand-

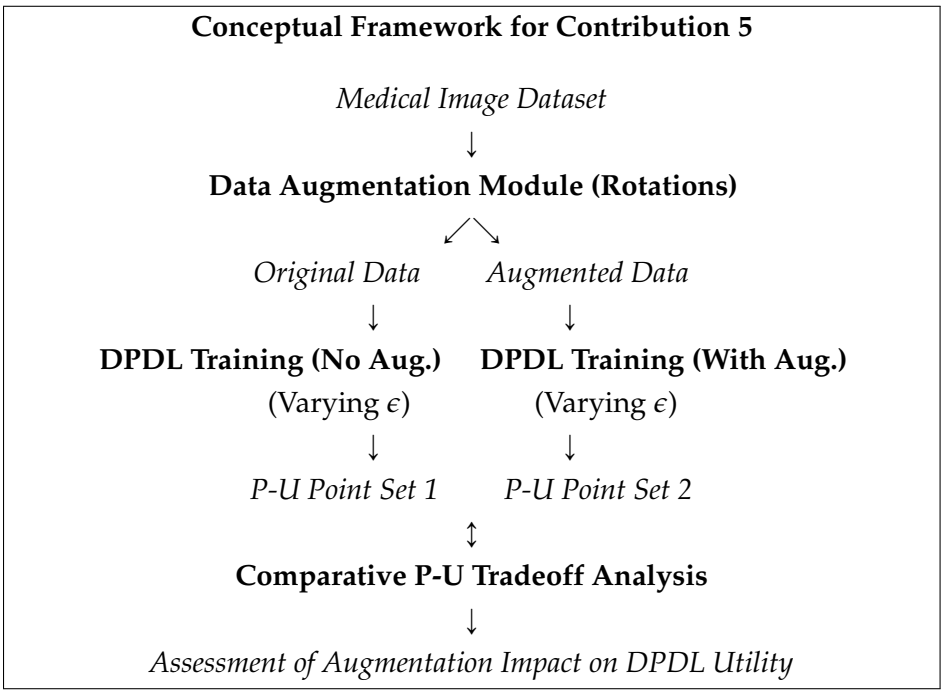


Figure 5.4: Conceptual diagram illustrating the approach for Contribution 5: Investigating the impact of data augmentation on the DPDL privacy-utility tradeoff in medical imaging.

ing of Differentially Private Deep Learning (DPDL), particularly within the context of sensitive medical data.

We began by proposing a **systematic empirical investigation (Contribution 1)** to establish the true utility potential and hyperparameter sensitivity of DPDL when meticulously tuned on real-world medical datasets, thereby challenging overly pessimistic assumptions about performance degradation. Recognizing the inherent tradeoff and the need for structured decision-making, we then introduced a framework based on **Pareto optimality (Contribution 2)** to map out and analyze the optimal Privacy-Utility (P-U) settlement points, explicitly considering the influence of architectural choices.

To address the significant computational cost associated with identifying these optimal configurations, the third contribution proposed a comparative framework for evaluating **efficient Hyperparameter Optimization (HPO) strategies (Contribution 3)**. This included established methods alongside a novel application of the Bat Algorithm, assessing their effectiveness in approximating the P-U Pareto frontier across diverse medical data modalities. Acknowledging the critical importance of fairness and the prevalence of data imbalance in medical datasets, the fourth contribution outlined a framework for systematically investigating the three-way **Privacy-Utility-Fairness (P-U-F) nexus (Contribution 4)**, analyzing how DP mechanisms interact with data imbalance and impact performance equity. Finally, to explore avenues for further utility enhancement in DPDL, particularly for data-constrained medical imaging, the fifth contribution detailed an **exploratory investigation into the synergy between DPDL and data augmentation (Contribution 5)**, aiming to understand

its privacy-level-dependent impact on the P-U tradeoff.

Together, these five proposed approaches represent a logical progression: from foundational performance assessment and tradeoff navigation, through efficient optimization and critical fairness analysis, to exploring synergistic techniques for utility improvement. They provide the methodological blueprint for the empirical work presented in subsequent chapters. By focusing on these frameworks—how to assess potential, navigate tradeoffs, optimize efficiently, analyze fairness, and potentially enhance utility via augmentation—this chapter sets the stage for demonstrating their practical implementation and evaluating their effectiveness in achieving a more robust and holistic balance in DPDL for real-world applications. The detailed methodologies, experimental designs, and results derived from applying these proposed approaches will be presented in Chapter 6.

Chapter 6

Implementation and Experimental Evaluation

***

6.1 Introduction

This chapter details the practical implementation aspects and the experimental evaluation conducted to validate the proposed approaches presented in Chapter 5. We describe the datasets employed, the deep learning model architectures utilized, the computational environment, and present the empirical results corresponding to each of the five core contributions of this thesis.

6.2 Experimental Datasets

The empirical validation of the proposed DPDL frameworks relies on a selection of publicly available datasets, chosen to represent diverse challenges and domains, particularly focusing on medical applications and varying levels of complexity and imbalance. The specific datasets utilized across the different contributions are:

- **CIFAR-10:** A standard computer vision benchmark dataset comprising 60,000 32x32 color images in 10 classes. While not medical, it serves as a well-understood baseline for evaluating image classification approaches, particularly relevant for Contribution 5.3.2 focusing on Pareto analysis and architecture selection.
- **PIMA Indians Diabetes Dataset:** A widely used tabular dataset from the healthcare domain, predicting diabetes based on diagnostic measurements. It is known for its inherent class imbalance, making it suitable for investigations involving Contributions 5.3.1 and particularly Contribution 5.3.4.
- **Breast Cancer Wisconsin (Diagnostic) Dataset:** Another standard tabular medical dataset used for binary classification (malignant vs. benign tumors) based on features computed from digitized images of fine needle aspirates. This dataset is commonly used for evaluating classification algorithms in healthcare contexts (Contributions 5.3.1, 5.3.3, 5.3.4).
- **BreastMNIST:** A collection of 28x28 grayscale images from the MedMNIST suite, derived fromsaved microscopy images of breast cancer specimens. It represents a simpler medical imaging task, used primarily in evaluating efficient HPO strategies (Contribution 5.3.3). It belongs to the MedMNIST v2 collection [YSW+23].
- **PneumoniaMNIST:** A collection of 28×28 grayscale images from the MedMNIST suite, derived from pediatric chest X-ray images for the detection of pneumonia. It represents a binary classification task distinguishing between normal and pneumonia cases, used primarily in evaluating data augmentation techniques and their impact on model performance in medical imaging scenarios (Contribution 6.5.5). It belongs to the MedMNIST v2 collection [YSW+23].
- **PathMNIST:** Another dataset from the MedMNIST v2 suite [YSW+23], consisting of 28x28 grayscale images derived from histology scans of colorectal cancer tissues. It represents a more complex medical imaging classification task compared to BreastMNIST, used to assess the scalability and generalizability of the efficient HPO framework (Contribution 5.3.3).

The specific characteristics of these datasets, including size, feature types, class distributions (imbalance ratios), and their allocation to training, validation, and test sets within our experiments, will be detailed where relevant in the results sections. The consistent use of real-

world medical datasets (PIMA, Breast Cancer Wisconsin, BreastMNIST, PneumoniaMNIST, PathMNIST) underscores the practical relevance of our findings.

6.3 Adapted Architecture

To comprehensively evaluate the proposed approaches across diverse data types and computational complexities inherent in each research contribution, a variety of standard and custom deep learning architectures were employed. These architectures were carefully selected or designed to suit the specific characteristics of the datasets and the objectives of each experimental study.

- **Multi-Layer Perceptron (MLP):** A fundamental feedforward neural network architecture, primarily used for experiments involving tabular datasets (PIMA Indians Diabetes, Breast Cancer Wisconsin). Specific layer configurations (number of layers, neurons per layer, activation functions like ReLU) were adapted and tuned for each dataset and contribution as appropriate.
- **ResNet-18:** A variant of the Residual Network architecture [HZRS16] with 18 layers. It is a widely adopted standard for image classification tasks, utilized in experiments involving CIFAR-10 (Contributions 5.3.2, 6.5.5) and potentially adapted for MedMNIST datasets, providing a strong baseline convolutional neural network (CNN).
- **ResNet-50:** A deeper variant of the ResNet family with 50 layers [HZRS16]. Its increased capacity allows for modeling more complex patterns, potentially employed or considered in comparative analyses involving more challenging image datasets or architecture selection studies (Contribution 5.3.2).
- **Custom Convolutional Neural Network (CNN):** In some experiments, particularly those involving specific image datasets or comparing architectures (Contribution 5.3.2, 5.3.3), custom-designed CNN architectures may have been employed. Their specific layer structures (convolutional layers, pooling layers, fully connected layers) would be detailed in the relevant experimental sections.

The precise architectural configurations for each model deployed within the five contributions of this thesis are detailed in Table 6.1 . This table provides a consolidated overview, specifying the number of layers, channel configurations, activation functions, and adaptations such as transfer learning backbones and output layer modifications.

This is some text on a normal portrait page.

Table 6.1: Summary of Adapted and Custom DL model Architectures by Contribution.

Contr. #	Model Name / Primary Type	Transfer Learning Backbone (if any)	Num. New Conv Layers	New Conv Layer Details (Channels Seq. / Activations)	Additional Pooling (Type/Details)	New Dense Layer Details (Num / Sizes Seq. / Activations)	Output Layer (Size / Activation)
1	MLP (PIMA Dataset)	N/A	0	N/A	N/A	Sizes [7]: [8→20, 20→10, 10→20, 20→10, 10→5, 5→10, 10→5] All ReLU	[5→2] / Linear
	MLP (Breast Cancer WI)	N/A	0	N/A	N/A	Sizes [5]: [30→20, 20→10, 10→20, 20→10, 10→5] All ReLU	[5→2] / Linear
2	Adapted ResNet-18	ResNet-18	0	N/A (Uses frozen backbone convs)	N/A (Head uses `AdaptiveAvgPool2d`)	Sizes [1]: [512 Features → Output] / Linear	[10] / Softmax
	Custom CNN (Cifar10 Dataset)	N/A	6	Channels: [32, 64, 128, 128, 256, 256] All ReLU	MaxPool2d(2,2) after 2nd, 4th, & 6th Conv layers	Sizes [2]: [4096→1024, 1024→512] All ReLU	[512→10] / Linear
3	MLP (Breast Cancer WI)	N/A	0	N/A	N/A	Sizes [5]: [30→20, 20→10, 10→10, 10→10, 10→5] All ReLU	[5→2] / Linear
	Adapted ResNet-18 (BreastMNIST)	ResNet-18	0	N/A (Uses frozen backbone convs. BN→GN* in backbone)	N/A (Head uses `AdaptiveAvgPool2d`)	Sizes [1]: [512 Features → Output] / Linear	[2] / Linear
	Adapted ResNet-50 (PathMNIST)	ResNet-50	0	N/A (Uses frozen backbone convs. BN→GN* in backbone)	N/A (Head uses `AdaptiveAvgPool2d`)	Sizes [1]: [2048 Features → Output] / Linear	[9] / Linear
4	ANN / MLP (PIMA Dataset)	N/A	0	N/A	N/A	Sizes [3]: [8→20, 20→10, 10→10] All ReLU	[10→2] / Linear
	ANN / MLP (Breast Cancer WI)	N/A	0	N/A	N/A	Sizes [3]: [30→20, 20→10, 10→10] All ReLU	[10→2] / Linear
5	MedicalResNet (PneumoniaMNIST)	ResNet-18	0	N/A (Modified `conv1` for 1 ch. input)	N/A (Uses ResNet-18 standard pooling)	Sizes [0]: N/A (Output layer directly replaces `fc`)	[512→2] / Linear (Replaced `fc` layer†)

*BN→GN: Batch Normalization layers replaced with Group Normalization (32 groups) for privacy compliance(Opacus).

These architectural choices reflect a balance between leveraging established, powerful models (like ResNets via transfer learning) and designing custom solutions (MLPs for tabular data, specific CNN) tailored to the unique demands of each dataset and the overarching research goals of the respective contributions. The consistent use of ReLU for hidden-layer activations and linear output layers (for use with CrossEntropyLoss) provides a degree of uniformity across the diverse model set.

6.4 Experimental Setup and Tools

All experiments were conducted using the following software environment and computational resources:

- **Programming Language:** Python (specifically version 3) was used for all implementations, leveraging its extensive ecosystem for machine learning and data science.
- **Core Libraries:**
  - *PyTorch:* Employed as the primary deep learning framework [PGM+19] for model definition, training, and inference.
  - *Opacus:* Utilized for implementing Differentially Private training mechanisms (DP-SGD/Adam) within PyTorch [YSS+21]. Opacus facilitates the conversion of standard PyTorch models and optimizers into their DP counterparts and handles privacy accounting.
  - *Scikit-learn:* Used for data preprocessing, standard machine learning metrics evaluation, and potentially for baseline model implementations [PVG+11].
  - *Pandas & NumPy:* Employed for data manipulation and numerical operations [Bea12, HMvdW+20].
- **Computational Resources:** Experiments were executed primarily on Google Colaboratory ('Colab'), utilizing both Pro and Pro+ subscriptions to access enhanced GPU resources (e.g., NVIDIA Tesla T4, P100, V100) and longer runtimes, which are essential for training deep learning models, especially under the computational overhead of DPDL and extensive hyperparameter optimization.

Consistent use of these tools and environments ensures reproducibility and provides a clear context for the computational aspects of the presented results. Specific library versions and configurations are documented within the accompanying code repository where applicable.

6.5 Empirical Evaluation and Results

This section presents the core empirical findings of the thesis, organized according to the five main contributions outlined in Chapter 5. Each subsection discusses the experimental setup specific to that contribution (if not covered generally above) and summarizes the key results obtained from applying the proposed approaches.

6.5.1 DPDL Performance Baseline and Sensitivity on Medical Datasets (Contribution 1)

This contribution (Section 5.3.1) establishes Differentially Private Deep Learning (DPDL) performance baselines and hyperparameter sensitivity on PIMA Indians Diabetes and Breast Cancer Wisconsin datasets. We compare meticulously tuned non-private models against their optimized DP counterparts. Results are shown in Tables 6.2 through 6.5.

Table 6.2: PIMA Dataset: Non-Private Model Classification Report (Contribution 1).

Class	Precision	Recall	F1-Score	Support
0	0.69	1.00	0.82	107
1	0.00	0.00	0.00	47
Accuracy			0.69	154
Macro Avg	0.35	0.50	0.41	154
Weighted Avg	0.48	0.69	0.57	154

Table 6.3: PIMA Dataset: DP Model Classification Report (Contribution 1).

Class	Precision	Recall	F1-Score	Support
0	0.74	0.81	0.78	107
1	0.46	0.36	0.40	47
Accuracy			0.68	154
Macro Avg	0.60	0.59	0.59	154
Weighted Avg	0.66	0.68	0.66	154

Analysis and Key Findings

PIMA Indians Diabetes Dataset A notable result emerged (Tables 6.2, 6.3):

- The non-private model, despite 0.69 accuracy, completely failed on the minority class (Class 1 F1: 0.00), showing severe imbalance (Macro F1: 0.41).

Table 6.4: Breast Cancer Wisconsin: Non-Private Model Report (Contribution 1).

Class	Precision	Recall	F1-Score	Support
0	0.96	0.97	0.96	67
1	0.96	0.94	0.95	47
Accuracy			0.96	114
Macro Avg	0.96	0.95	0.95	114
Weighted Avg	0.96	0.96	0.96	114

Table 6.5: Breast Cancer Wisconsin: DP Model Report (Contribution 1).

Class	Precision	Recall	F1-Score	Support
0	0.85	1.00	0.92	67
1	1.00	0.74	0.85	47
Accuracy			0.89	114
Macro Avg	0.92	0.87	0.89	114
Weighted Avg	0.91	0.89	0.89	114

- The tuned DP model achieved similar accuracy (0.68) but significantly improved minority class classification (Class 1 F1: 0.40) and overall balance (Macro F1: 0.59).

This suggests DPDL, through potential regularization, can enhance fairness on imbalanced data without necessarily degrading aggregate accuracy.

Breast Cancer Wisconsin Dataset (Tables 6.4, 6.5):

- The non-private model achieved high baseline performance (Accuracy: 0.96, Macro F1: 0.95).
- The tuned DP model showed a utility cost (Accuracy: 0.89, Macro F1: 0.89) but maintained high absolute performance.

This demonstrates that careful tuning can preserve significant utility with DPDL.

Conclusion (Contribution 1)

These experiments validate that DPDL's utility cost is context-dependent. Meticulous hyperparameter tuning is crucial. On the PIMA dataset, DPDL unexpectedly improved fairness for the minority class while maintaining overall accuracy. On Breast Cancer Wisconsin, high utility was preserved under DP. Sensitivity analyses confirmed strong dependence on parameters like clipping norm and noise multiplier. These findings establish DPDL's potential on real-world medical data and underscore the need for systematic optimization (Contribution 3) and fairness considerations (Contribution 4).

6.5.2 Pareto-Optimal Framework for P-U Tradeoff and Architecture Selection (Contribution 2)

This contribution (Section 5.3.2) introduces a Pareto optimization framework to systematically navigate the privacy-utility (P-U) tradeoff and aid DPDL architecture selection, evaluated on CIFAR-10. The framework visualizes tradeoffs by plotting privacy loss (ϵ, Obj. 1, minimized) against utility loss (1-Accuracy, Obj. 2, minimized), with the Pareto frontier highlighting optimal, non-dominated configurations.

Figure 6.1 shows the empirical Pareto front for a DP-ResNet-18. Each point is an evaluated hyperparameter configuration, and the red line connects non-dominated points, delineating the best utility loss for any given ϵ. For example, at $\epsilon \approx 1$, the lowest utility loss is ≈ 0.44 (56% accuracy). Figure 6.2 similarly presents the Pareto front for a "Custom CNN" architecture, illustrating its specific P-U compromises (e.g., utility loss ≈ 0.59 for $\epsilon \approx 1.7$).

A comparative visualization is presented in Figure 6.3, which plots the Pareto frontiers for our DP-ResNet-18 (red) and Custom CNN (blue) against benchmark DPDL results on CIFAR-10 from Dörmann et al. (2021) [DFAP21] (green) and De et al. (2022) [DBH+22] (yellow). This allows for a direct comparison of the P-U efficiency achieved by different architectures and training approaches. The relative positions of the frontiers indicate which architectures offer better utility for a given level of privacy, or stronger privacy for a given level of utility. For instance, if one architecture's frontier is consistently lower and to the left of another, it demonstrates superior P-U characteristics. Our ResNet-18 and Custom CNN frontiers can thus be contextualized against existing state-of-the-art results.

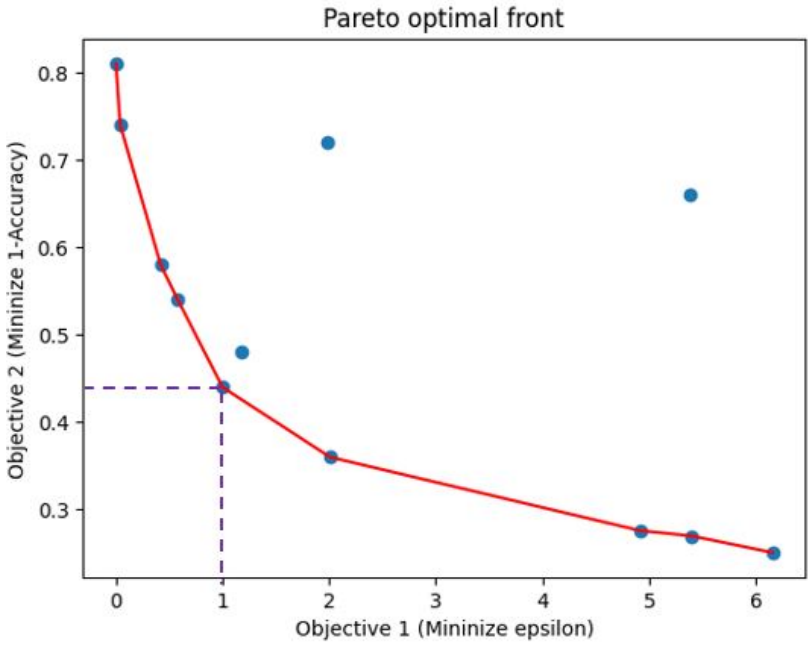


Figure 6.1: Empirical Pareto front for DP-ResNet-18 on CIFAR-10: Privacy loss (ϵ) vs. Utility loss (1-Accuracy). Dashed lines show an example selection.[RFA23]

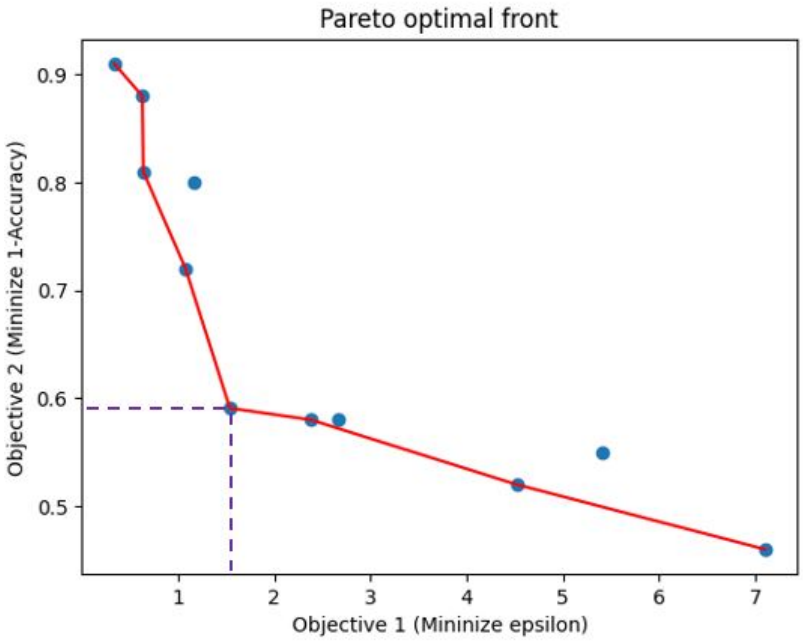


Figure 6.2: Empirical Pareto front for Custom CNN on CIFAR-10. Dashed lines show an example selection.[RFA23]

Conclusion (Contribution 2)

The proposed Pareto-optimal framework effectively visualizes the P-U tradeoff for specific DPDL models. The generated frontiers offer a principled method for understanding achievable compromises and enable direct, quantitative comparison between different architectures (e.g., ResNet-18 vs. Custom CNN vs. literature benchmarks). This aids in selecting models that best balance privacy and utility for a given application context on datasets like CIFAR-10.

6.5.3 Comparative Analysis of HPO Strategies for DPDL P-U Optimization (Contribution 3)

This contribution (Section 5.3.3) evaluates a framework for comparing Hyperparameter Optimization (HPO) strategies—Grid Search (GS), Random Search (RS), Bayesian Optimization (BO), and a novel application of the Bat Algorithm (BA)—for optimizing the privacy-utility (P-U) tradeoff in DPDL. Experiments were conducted on Wisconsin Breast Cancer (tabular), BreastMNIST (simple imaging), and PathMNIST (complex imaging) datasets. The goal was to find configurations balancing privacy (ϵ) and utility (accuracy, via fitness f) within practical computational limits. Key results are in Tables 6.6 (performance) and 6.7 (hyperparameters).

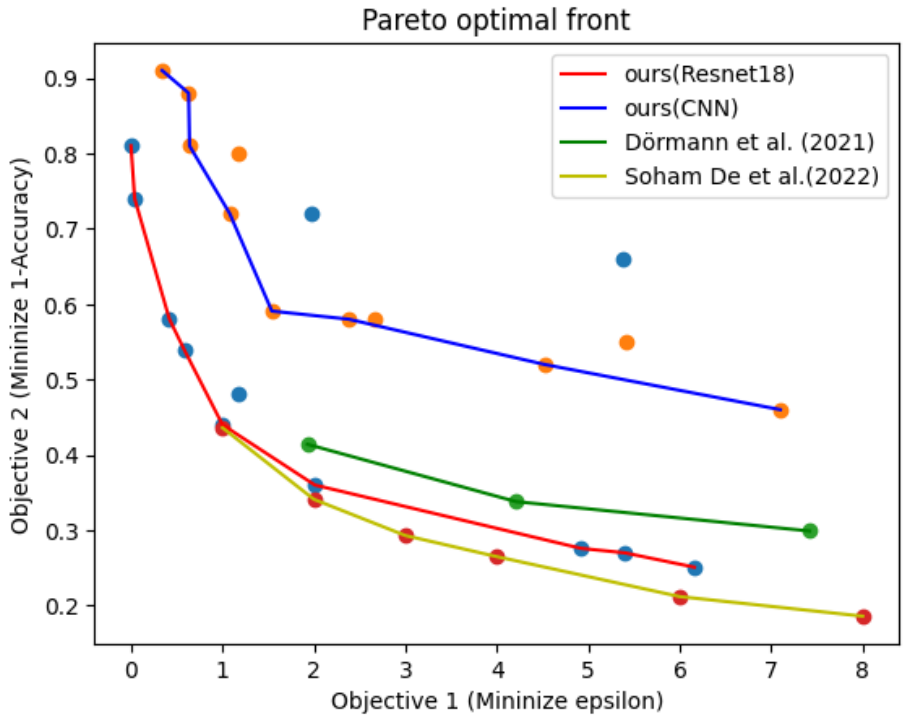


Figure 6.3: Comparative Pareto frontiers for DPDL on CIFAR-10: ResNet-18 (red), Custom CNN (blue) vs. benchmarks by Dörmann et al. [DFAP21] (green) and De et al. [DBH+22] (yellow).[RFA23]

Table 6.6: HPO Performance: Best Configurations on Wisconsin and BreastMNIST (Contribution 3).[BHB25]

Dataset	Method	ϵ	Acc. (%)	Fitness (f)	Time (s)	Mem. (MB)	Pareto Opt
Wisconsin Breast Cancer	GS	**0.500**	93.40	**0.1840**	4,000	1,065	Yes
	RS	**0.500**	92.53	0.1834	**2,750**	845	No
	BO	0.501	**93.62**	**0.1840**	17,829	**788**	No
	BA	2.258	93.18	0.0470	14,562	1,103	Yes
BreastMNIST	GS	**0.500**	74.18	0.1699	12,932	1,796	Yes
	RS	1.000	**74.91**	**0.1697**	**6,500**	1,555	Yes
	BO	**0.500**	73.99	**0.1697**	38,076	**1,323**	Yes
	BA	**0.293**	73.08	**0.1887**	10,132	4,348	Yes

Table 6.7: Best Hyperparameters Found by HPO Methods (Contribution 3).[BHB25]

Dataset	Method	lr	bs	ϵ_{target}	C	σ	Epochs	Acc. (%)
Wisconsin Breast Cancer	GS	0.01	32	0.500	1.2	42.5	273	93.40
	RS	0.1	512	0.500	5.6	135.0	252	92.53
	BO	0.018	128	0.501	3.118	77.5	265	**93.62**
	BA	0.0965	512	2.258	3.388	95.5	248	93.18
BreastMNIST	GS	0.1	32	0.500	1.2	6.25	12	74.18
	RS	0.1	64	1.000	1.2	8.75	12	**74.91**
	BO	0.1	128	0.500	4.906	12.19	15	73.99
	BA	0.0429	512	**0.293**	3.114	18.5	14	73.08

HPO Performance Analysis

Wisconsin Breast Cancer Dataset BO achieved the highest accuracy (93.62%), followed closely by GS and BA. RS was fastest but slightly less accurate. BA found a solution with higher ϵ (2.258) than the others' target of 0.5.

BreastMNIST Dataset Accuracies were similar across methods (≈73-74%), with RS marginally highest. Notably, BA identified a configuration with the tightest privacy ($\epsilon = 0.293$) while maintaining competitive accuracy. RS was again fastest; BA also showed good relative speed.

The optimization processes revealed distinct exploration patterns: GS (systematic), RS (broad sampling), BO (concentrated search), and BA (population-based). Convergence plots showed RS and BA often improved rapidly initially.

Scalability on PathMNIST

On the more complex PathMNIST dataset, BA identified DPDL configurations (Table 6.8), including a Pareto-optimal one, with accuracies around 40-45%. This demonstrates the framework's applicability to challenging data, albeit with significant computational cost.

This work validated a framework for comparing HPO strategies for DPDL P-U optimization. RS proved a strong, fast baseline. BO was computationally expensive despite theoretical efficiency. The novel application of BA demonstrated its capability to find good, sometimes Pareto-optimal, solutions and the tightest privacy on BreastMNIST. Its distinct exploration pattern suggests BA is a viable alternative. Crucially, efficient HPO methods can identify high-performing DPDL configurations on diverse medical datasets, offering practical alternatives to exhaustive searches for balanced P-U tradeoffs.

Table 6.8: Bat Algorithm Performance on PathMNIST (Contribution 3).[BHB25]

Parameter / Metric	Config. 1 (Global Best)	Config. 2 (Pareto Opt.)
Hyperparameters		
Learning Rate (lr)	0.03106	0.01760
Batch Size (bs)	128	512
Max-Grad-Norm (C)	2.179	1.732
Target ϵ	0.508	2.603
Performance Metrics		
Fitness (f)	0.1435	0.0328
Accuracy (%)	39.97	44.71
Achieved ϵ	0.508	2.603
Time (s)	18,238	13,679
Memory (MB)	8,479	7,522
Pareto Optimal	No	Yes

6.5.4 Privacy-Utility Nexus under Data Imbalance (Contribution 4)

This contribution (Section 5.3.4) empirically investigates the interplay of privacy, utility, (P-U) when applying DPDL to imbalanced medical datasets: PIMA Indians Diabetes and Breast Cancer Wisconsin. We compare non-private models (Table 6.9) with DPDL models under varying privacy budgets (ϵ) (Table 6.10). The analysis focuses on how DPDL affects fairness for the minority class, using metrics like Recall (Rec), F1-Score (F1), Geometric Mean (Geo), and Index of Balanced Accuracy (Iba), sensitive to imbalanced data.

Table 6.9: Non-Private Model Performance on PIMA and Breast Cancer Wisconsin Datasets (Contribution 4 Baseline).

Dataset	Class	Pre	Rec	Spe	F1	Geo	Iba
PIMA	0	0.79	0.81	0.51	0.80	0.64	0.43
	1	0.55	0.51	0.81	0.53	0.64	0.40
	Average/Total	0.72	0.72	0.60	0.72	0.64	0.42
Breast Cancer Wisconsin	B	0.88	1.00	0.81	0.94	0.90	0.82
	M	1.00	0.81	1.00	0.89	0.90	0.79
	Average/Total	0.93	0.92	0.89	0.92	0.90	0.81

Table 6.10: DPDL Model Performance on PIMA and Breast Cancer Wisconsin Datasets across Privacy Levels (ϵ).

Dataset	(ϵ, δ)-DP	Class	Pre	Rec	Spe	F1	Geo	Iba
PIMA	$(8, 10^{-4})$	0	0.81	0.77	0.51	0.79	0.63	0.40
		1	0.45	0.51	0.77	0.48	0.63	0.38
	$(2, 10^{-4})$	0	0.77	0.69	0.31	0.73	0.46	0.24
		1	0.23	0.31	0.69	0.27	0.46	0.21
	$(1, 10^{-4})$	0	0.75	0.63	0.00	0.68	0.03	0.01
		1	0.00	0.00	0.63	0.00	0.03	0.008
Breast Cancer Wisconsin	$(8, 10^{-4})$	B	1.00	0.79	1.00	0.88	0.89	0.80
		M	0.62	1.00	0.79	0.76	0.89	0.78
	$(2, 10^{-4})$	B	0.96	0.70	0.87	0.81	0.78	0.63
		M	0.43	0.87	0.70	0.57	0.78	0.60
	$(1, 10^{-4})$	B	0.91	0.63	0.67	0.74	0.65	0.43
		M	0.25	0.67	0.63	0.36	0.65	0.40

Dataset-Specific Analysis

PIMA Indians Diabetes Dataset The non-private model (Table 6.9) struggled with PIMA's imbalance, achieving only 0.51 Recall and 0.53 F1 for the minority class (Class 1). Introducing DPDL (Table 6.10) exacerbated this:

- With $\epsilon = 8$, minority F1 dropped to 0.48.
- With $\epsilon = 2$, minority F1 fell to 0.27, and fairness metrics (Geo, Iba) declined substantially.
- With $\epsilon = 1$, the model failed to identify the minority class (F1=0.00), and fairness metrics neared zero.

This demonstrates a severe P-U-F tradeoff, where increasing privacy disproportionately harmed minority class utility and fairness.

Breast Cancer Wisconsin Dataset The non-private model performed better on this more balanced dataset, though the minority class ('M') still had lower F1 (0.89) than the majority ('B') (Table 6.9). With DPDL (Table 6.10):

- With $\epsilon = 8$, minority F1 dropped to 0.76.
- With $\epsilon = 2$, minority F1 decreased to 0.57.
- With $\epsilon = 1$, minority F1 further fell to 0.36.

Fairness indicators (Geo, Iba) also progressively declined with stricter privacy. While less dramatic than PIMA, results consistently show tightening privacy disproportionately affected minority class classification, reducing fairness.

Overall Findings and Conclusion (Contribution 4)

This investigation confirms that applying DPDL to imbalanced medical datasets, particularly with stricter privacy guarantees (lower ϵ), significantly degrades fairness. Performance on the minority class, measured by Recall, F1-score, Geo, and Iba, was disproportionately affected across both PIMA and Breast Cancer Wisconsin datasets. The impact was severe on the highly imbalanced PIMA dataset, where the model lost all ability to detect the minority class at $\epsilon = 1$.

These findings highlight the critical need to evaluate DPDL systems not just for aggregate utility but also through the lens of fairness, especially with imbalanced healthcare data. The observed P-U-F tradeoffs underscore the risks of deploying standard DPDL without considering its disparate impact, motivating future work on fairness-aware privacy-preserving techniques.

6.5.5 Investigating the Interplay of Data Augmentation and Differentially Private Deep Learning on PneumoniaMNIST (Contribution 5)

Medical AI often faces limited, sensitive data, making Differentially Private Deep Learning (DPDL) crucial despite its potential utility cost. Data augmentation can improve performance with scarce data. This exploration investigates if rotation augmentation can mitigate DPDL's utility loss on PneumoniaMNIST using DP-SGD at two privacy levels ($\epsilon \approx 1$ and $\epsilon \approx 8$), and how this interaction manifests.

Experimental Setup

This investigation used the PneumoniaMNIST chest X-ray dataset with a ResNet-18 architecture (Section 6.3). DP-SGD was employed with target privacy budgets of $\epsilon = 1$ and $\epsilon = 8$. Data augmentation involved random rotations ($\pm$20 degrees). Key training parameters are in Table 6.11. Models were trained for up to 15 epochs with early stopping (patience 7). Performance was assessed via test accuracy, micro/macro-AUC, and training time, comparing four configurations: baseline (no DP, no augmentation), augmentation-only, DPDL-only (at both ϵ levels), and DPDL with augmentation (at both ϵ levels). All results are summarized in Table 6.12.

Results and Analysis

Performance across configurations is detailed in Table 6.12. Training histories for each are shown in Figures.

Table 6.11: Core Training Parameters for Contribution 5.

Parameter	Value
Batch Size	64
Max Epochs	15
Learning Rate	0.0005
Optimizer (Non-Private / Private)	SGD / DP-SGD
Max Grad Norm (DP)	1.2
Target ϵ (DP)	1.0, 8.0
Augmentation	Rotations ($\pm 20^\circ$)
Early Stopping Patience	7

The non-private baseline (no augmentation) achieved 89.26% accuracy and 0.945 macro-AUC (Figure 6.4). Rotation augmentation alone (Figure 6.5) yielded 88.62% accuracy but improved macro-AUC to 0.961, suggesting better class discrimination despite a slight accuracy dip.

Applying DP-SGD without augmentation reduced utility (Figures 6.6 and 6.7):

- At $\epsilon \approx 8.0$: 81.73% accuracy, 0.908 macro-AUC.
- At $\epsilon \approx 1.0$: 82.53

DP significantly increased training times (e.g., from ≈663s to ≈7300s).

Combining rotation augmentation with DP-SGD showed mixed results (Figures 6.8 and 6.9):

- With $\epsilon \approx 8.0$: Accuracy improved to 84.62% and macro-AUC to 0.920, outperforming DP-only at this privacy level and recovering some utility lost compared to non-private models.
- With $\epsilon \approx 1.0$ (achieved $\epsilon = 0.71$): Performance dropped significantly to 62.50% accuracy and 0.866 macro-AUC, with training stopping at epoch 1. This was substantially worse than DP-only at a similar privacy level, indicating a negative interaction.

Table 6.12: Summary of Performance Metrics for DPDL and Data Augmentation Investigation on PneumoniaMNIST.

Configuration	Target ϵ	Achieved ϵ	δ	Augmentation	Test Acc. (%)	Micro-AUC	Macro-AUC	Best Epoch	Train Time (s)
Baseline	N/A	∞	0	None	89.26	0.931	0.945	5	663
Aug. Only	N/A	∞	0	Rotation	88.62	0.940	0.961	13	1036
DPDL Only	8.0	7.99	2.12×10^{-4}	None	81.73	0.896	0.908	11	7351
DPDL + Aug.	8.0	7.99	2.12×10^{-4}	Rotation	**84.62**	**0.924**	**0.920**	15	10446
DPDL Only	1.0	0.997	2.12×10^{-4}	None	**82.53**	**0.865**	**0.903**	13	7310
DPDL + Aug.	1.0	0.71	2.12×10^{-4}	Rotation	62.50	0.797	0.866	1	4214

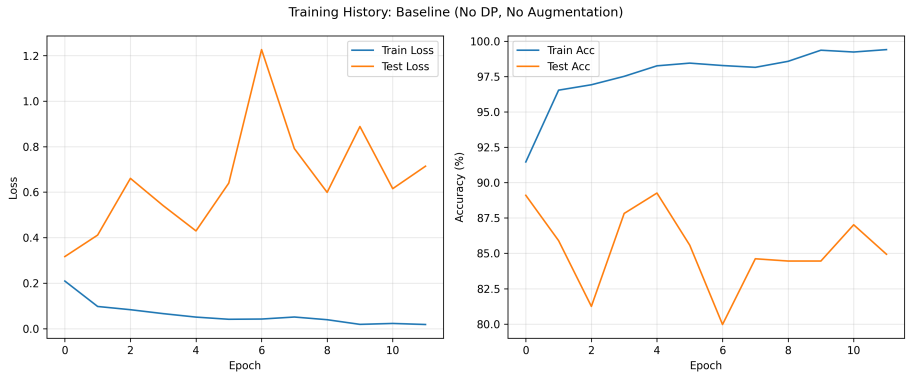


Figure 6.4: Training history: Baseline (No DP, No Augmentation).

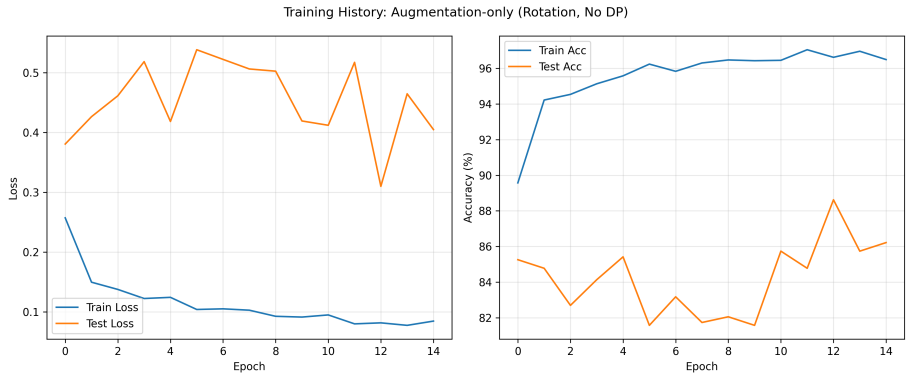


Figure 6.5: Training history: Augmentation-only (Rotation, No DP).

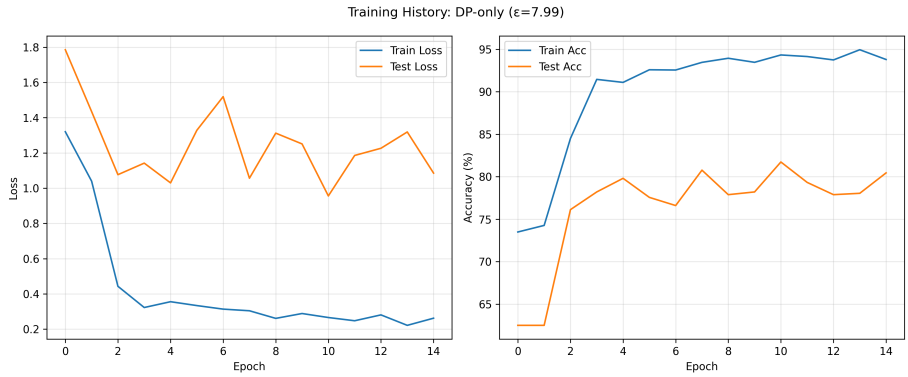


Figure 6.6: Training history: DP-only ($\epsilon \approx 8.0$, No Augmentation).

Discussion and Conclusion (Contribution 5)

This investigation on PneumoniaMNIST revealed a privacy-level-dependent interaction between DP-SGD and rotation augmentation. At a moderate privacy budget ($\epsilon \approx 8.0$), augmentation substantially improved DPDL utility. However, at a stringent privacy budget (target $\epsilon \approx 1.0$, achieved $\epsilon = 0.71$), augmentation was detrimental, significantly degrading performance and leading to unstable training.

Figure 6.7: Training history: DP-only ($\epsilon \approx 1.0$, No Augmentation).

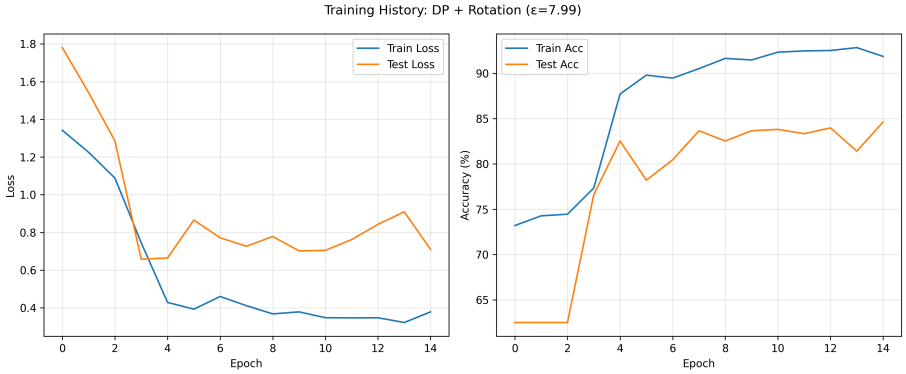


Figure 6.8: Training history: DP ($\epsilon \approx 8.0$) + Rotation Augmentation.

Figure 6.9: Training history: DP (achieved $\epsilon = 0.71$) + Rotation Augmentation.

These findings underscore that while data augmentation can benefit DPDL, its effectiveness is not universal and depends critically on the privacy level. Augmentation may exacerbate training challenges under very stringent privacy guarantees. Therefore, careful selection and tuning of augmentation strategies are crucial when combined with DPDL, particularly for strong privacy regimes. This warrants further research into adaptive augmentation techniques for DPDL in medical imaging.

6.6 Chapter Conclusion and Key Experimental Findings

This chapter presented a comprehensive empirical evaluation of the five core contributions of this thesis, validating the proposed methodologies for advancing Differentially Private Deep Learning (DPDL) in sensitive, often medical, application contexts.

The experimental investigations yielded several key insights:

C1: Our initial assessment demonstrated that DPDL, with meticulous hyperparameter tuning, can achieve viable utility on real-world medical datasets. Notably, on the imbalanced PIMA dataset, DPDL not only preserved overall accuracy but also enhanced fairness for the minority class, while on the Breast Cancer Wisconsin dataset, high absolute performance was maintained despite the introduction of privacy.

C2: The proposed Pareto-optimal framework proved effective for systematically navigating and visualizing the privacy-utility (P-U) tradeoff. By generating empirical Pareto frontiers for different architectures (ResNet-18, Custom CNN) on CIFAR-10 and comparing them against benchmarks, the framework provided a principled approach for architecture selection based on desired P-U characteristics.

C3: The comparative analysis of Hyperparameter Optimization (HPO) strategies highlighted Random Search as a strong, efficient baseline for DPDL P-U optimization. The novel application of the Bat Algorithm also demonstrated its capability in finding competitive, sometimes Pareto-optimal solutions, including the tightest privacy budget on BreastMNIST, offering a viable alternative to other HPO techniques.

C4: Investigating the privacy-utility-fairness (P-U-F) nexus under data imbalance revealed that standard DPDL can disproportionately affect minority class performance, particularly under stricter privacy guarantees. This was evident on both PIMA and Breast Cancer Wisconsin datasets, underscoring the critical need for fairness-aware DPDL.

C5: The exploratory analysis of combining DPDL with data augmentation (rotation) on PneumoniaMNIST showed a privacy-level-dependent interaction. Augmentation was beneficial at a moderate privacy budget ($\epsilon \approx 8.0$) but detrimental under a stringent regime (target $\epsilon \approx 1.0$), indicating that the synergy is not universal and requires careful consideration.

Collectively, these empirical findings underscore the nuanced and context-dependent nature of DPDL. While significant utility benefits can be achieved, this often requires careful tuning, appropriate framework selection, and an awareness of potential disparate impacts, especially under data imbalance or when combined with other techniques like data augmentation. The results highlight both the potential of DPDL for practical deployment and the critical importance of sophisticated optimization and evaluation methodologies.

These insights pave the way for further advancements in developing robust, fair, and utility-preserving privacy-enhancing technologies.

Chapter 7

Conclusion and perspectives

***

This dissertation investigated critical challenges in Differentially Private Deep Learning (DPDL), focusing on the practical realization of models that balance stringent privacy guarantees with high utility, particularly within the demanding context of medical data analysis. The research systematically addressed the inherent tradeoff and complexities through the development and empirical validation of promising frameworks and methodologies. This concluding chapter consolidates the primary findings and contributions of this work, acknowledges operational limitations encountered, and outlines significant directions for future research.

The research presented in this thesis yielded five core contributions aimed at advancing the state and applicability of DPDL:

First, we established **empirical baselines for DPDL performance (C1)** on real-world medical datasets. Through meticulous hyperparameter tuning, this work demonstrated that DPDL can achieve competitive utility, and in specific cases of data imbalance (PIMA dataset), unexpectedly enhance fairness for minority classes while preserving overall accuracy. This underscored the critical role of tuning and challenged overly pessimistic views of DPDL's utility cost.

Second, a **Pareto-optimal framework was developed (C2)** for systematically navigating the Privacy-Utility (P-U) tradeoff. This approach enabled the visualization of optimal P-U frontiers, facilitating principled DPDL architecture selection (e.g., ResNet-18 vs. Custom CNN on CIFAR-10) and comparison against established benchmarks.

Third, to address the computational burden of DPDL tuning, we introduced a **comparative framework for efficient Hyperparameter Optimization (HPO) strategies (C3)**. This contribution evaluated Random Search, Bayesian Optimization, and notably, a novel appli-

cation of the Bat Algorithm. This assessment allowed us to measure their effectiveness in approximating P-U Pareto frontiers across various medical datasets, ultimately identifying the Bat Algorithm as a promising alternative for hyperparameter optimization (HPO).

Fourth, this thesis critically examined the **Privacy-Utility-Fairness (P-U-F) nexus under data imbalance (C4)**. Empirical investigations on medical datasets (PIMA, Breast Cancer Wisconsin) quantified how standard DPDL mechanisms can disproportionately degrade performance for minority classes, particularly under stricter privacy regimes, thereby highlighting the imperative for fairness-aware DPDL.

Fifth, an exploratory study investigated the **synergy between DPDL and data augmentation (C5)** for medical image analysis. Using image rotation on PneumoniaMNIST, we found a privacy-level-dependent interaction, where augmentation proved beneficial at moderate privacy levels but could be detrimental under stringent privacy, indicating that such combinations require careful, context-specific validation.

Collectively, these contributions provide a comprehensive suite of methodologies for assessing, optimizing, and evaluating DPDL systems, emphasizing practical applicability, efficiency, and a holistic consideration of privacy, utility.

The execution of the empirical studies within this thesis was subject to certain operational limitations, primarily concerning **computational resources**. The intensive nature of DPDL training, particularly with extensive hyperparameter searches, necessitated constraints on the **duration of training (number of epochs)** and, in some instances, the selection of **datasets with manageable complexity and scale**. While the derived findings are robust for the investigated scenarios, broader generalization to significantly larger and more complex datasets or longer training regimes would benefit from access to more substantial hardware infrastructure. Further limitations include the primary focus on DP-SGD as the DPDL mechanism and specific choices of fairness metrics and data augmentation techniques, which define the boundaries for the current conclusions and offer avenues for expanded future work.

7.1 Future Research Directions and Perspectives

The insights and methodologies developed in this thesis pave the way for several promising research directions aimed at further maturing DPDL for reliable real-world deployment:

- **Advanced Hyperparameter Optimization and Automated DPDL:** Research into more sophisticated and automated HPO techniques (AutoML) for DPDL is essential to further reduce the tuning burden and discover superior P-U configurations. This could involve developing specialized acquisition functions for Bayesian Optimization or meta-learning approaches for DPDL hyperparameter transfer.
- **Comprehensive Analysis of Data Augmentation in DPDL:** A systematic investigation across a wider array of data augmentation techniques (e.g., generative models,

domain-specific transformations) and their impact on both privacy guarantees (e.g., amplification or composition) and utility within various DPDL frameworks is warranted, particularly for diverse medical data modalities.

- **Robustness and Generalization of DPDL under Data Shifts:** Extending DPDL research to address challenges of domain adaptation, concept drift, and out-of-distribution generalization under privacy constraints is crucial for ensuring the long-term viability of deployed models in dynamic environments like healthcare.
- **Theoretical Understanding of P-U Boundaries:** While this work was largely empirical, further theoretical analysis is needed to establish tighter bounds and deeper understanding of the fundamental limits and achievable regions within the P-U space for different data models and DP mechanisms.
- **Scalability and Efficiency of DPDL Systems:** Continued efforts to improve the computational efficiency and scalability of DPDL training algorithms and privacy accounting mechanisms are vital for enabling their application to very large datasets and complex model architectures.
- **Intersectional Fairness and Multi-Attribute Privacy:** Future work should explore more nuanced definitions of fairness, such as intersectional fairness, and consider privacy implications when multiple sensitive attributes are present in the data.

Continued innovation at the confluence of deep learning, differential privacy is paramount for building trustworthy and equitable AI systems. The methodologies and findings presented herein aim to contribute substantively to this ongoing endeavor.

Bibliography

***

[ACG+16a] Martín Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 23rd ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 308–318, 2016.

[ACG+16b] Martín Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 308–318, 2016.

[ACG+16c] Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 308–318. ACM, 2016.

[Act96] Accountability Act. Health insurance portability and accountability act of 1996. *Public law*, 104:191, 1996.

[AGSS] Rohan Anil, Badih Ghazi, Thomas Steinke, and Vinith M. Suriyakumar. Differentially private fine-tuning of large language models. In *International Conference on Learning Representations, ICLR 2023*.

[ASA22] Tanjim Taharat Aurpa, Rifat Sadik, and Md Shoaib Ahmed. Abusive bangla comments detection on facebook using transformer-based deep learning models. *Social Network Analysis and Mining*, 12(1):24, 2022.

[ASY+18] Naman Agarwal, Ananda Theertha Suresh, Felix Yu, Sanjiv Kumar, and H. Brendan McMahan. cpsgd: Communication-efficient and differentially-private distributed SGD. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada.*, pages 6601–6612, 2018.

[ATMR21] Galen Andrew, Om Thakkar, H. Brendan McMahan, and Swaroop Ramaswamy. Differentially private learning with adaptive clipping. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021*, pages 17455–17466, 2021.

[BB12] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

[BCC+19] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019.

[BCJ+23] Konstantin Büttner, Zirui Chen, Matthew Jagielski, Badih Ghazi, and Florian Tramèr. Private fine-tuning with adapters. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, 2023.

[Bea12] David M. Beazley. Data processing with pandas. *login Usenix Mag.*, 37(6), 2012.

[BFH+16] Felix Bieker, Michael Friedewald, Marit Hansen, Hannah Obersteller, and Martin Rost. A process for data protection impact assessment under the european general data protection regulation. In *Privacy Technologies and Policy: 4th Annual Privacy Forum, APF 2016, Frankfurt/Main, Germany, September 7-8, 2016, Proceedings 4*, pages 21–37. Springer, 2016.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. Leveled fully homomorphic encryption without bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference (ITCS)*, pages 309–325, 2012.

[BHB25] Rafika Benladgham, Fethallah Hadjila, and Adam Belloum. Efficient privacy-utility optimization for differentially private deep learning: Application to medical diagnosis. *International journal of electrical and computer engineering systems*, 16(5):377–395, 2025.

[BIK+17] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 1175–1191, 2017.

[BJWW+19] Brett K. Beaulieu-Jones, Zhiwei Steven Wu, Chris Williams, Ranysha Ware, Isaac S. Kohane, Andrew L. Beam, and Casey S. Greene. Differentially

private model interpretability. In *Machine Learning for Health (ML4H) Workshop at NeurIPS 2019*, volume 116 of *Proceedings of Machine Learning Research*, pages 33–46. PMLR, 2019.

[BKS18] Matthew Bernstein, Motonobu Kanagawa, and Bharath K. Sriperumbudur. Differentially private bayesian optimization. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 541–549. PMLR, 2018.

[BNS14] Amos Beimel, Kobbi Nissim, and Uri Stemmer. Private learning and sanitization: Pure vs. approximate differential privacy. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 17th International Workshop, APPROX 2014, and 18th International Workshop, RANDOM 2014, Barcelona, Spain, September 4-6, 2014*, pages 436–451, 2014.

[Bot10] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In Yves Lechevallier and Gilbert Saporta, editors, *19th International Conference on Computational Statistics, COMPSTAT 2010, Paris, France, August 22-27, 2010 - Keynote, Invited and Contributed Papers*, pages 177–186. Physica-Verlag, 2010.

[BPS19] Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 15479–15488, 2019.

[BRNM21] Priyam Basu, Tiasa Singha Roy, Rakshit Naidu, and Zumrut Muftuoglu. Privacy enabled financial text classification using differential privacy and federated learning. *arXiv preprint arXiv:2110.01643*, 2021.

[BS16] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography - 14th Theory of Cryptography Conference, TCC 2016-B, Proceedings, Part I*, pages 635–658, 2016.

[BSPK20] S. Balaji, Umayal S. P., and K. Vasantha Kumar. Bat algorithm based hyperparameter selection in convolutional neural network. *International Journal of Intelligent Systems and Applications in Engineering*, 8(4):200–208, 2020.

[BSW11] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. *Theory of Cryptography Conference (TCC)*, pages 253–273, 2011.

[BWWZ20] Zeyi Bu, Jialin Wang, Lixu Wang, and Qi Zhang. Differentially private admm based on bayesian optimization. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2020.

[CAS16] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems (RecSys)*, pages 191–198, 2016.

[CBHK02] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[CCV+17] Gabriel Chartrand, Phillip M Cheng, Eugene Vorontsov, Michal Drozdzal, Simon Turcotte, Christopher J Pal, Samuel Kadoury, and An Tang. Deep learning: a primer for radiologists. *Radiographics*, 37(7):2113–2131, 2017.

[CCX+19] Guoxing Chen, Sanchuan Chen, Yuan Xiao, Taesoo Kim, Yinqian Zhang, and Wei Xu. Sgx-log: Securing system logs with sgx. *IEEE Transactions on Dependable and Secure Computing*, 18(1):438–453, 2019.

[Cha04] Nitesh V. Chawla. Editorial: Special issue on learning from imbalanced data sets. *SIGKDD Explorations Newsletter*, 6(1):1–6, 2004. Often cited alongside Chawla's earlier work on SMOTE (2002 JAIR) for the broader context of imbalance.

[Che23] Dingfan Chen. Towards privacy-preserving machine learning: generative modeling and discriminative analysis. 2023.

[CKS20] Clément L. Canonne, Gautam Kamath, and Thomas Steinke. The discrete gaussian for differential privacy. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:15676–15688, 2020.

[CLE+19] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *Proceedings of the 28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, August 14-16, 2019*, pages 267–284. USENIX Association, 2019.

[CLY+22] Weijie Chen, Luojun Lin, Shicai Yang, Di Xie, Shiliang Pu, and Yueting Zhuang. Self-supervised noisy label learning for source-free unsupervised domain adaptation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10185–10192. IEEE, 2022.

[CMB+21] Leticia Lemus Cárdenas, Ahmad Mohamad Mezher, Pablo Barbecho Bautista, Juan Pablo Astudillo León, and Mónica Aguilar-Igartua. A

multimetric predictive ann-based routing protocol for vehicular ad hoc networks. *IEEE Access*, 9:86037–86053, 2021.

[CMS11] Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. Differentially private empirical risk minimization. In *Journal of Machine Learning Research*, volume 12, pages 1069–1109, 2011.

[CMV+21] Phillip Chlap, Hang Min, Nym Vandenberg, Jason Dowling, Lois Holloway, and Annette Haworth. A review of medical image data augmentation techniques for deep learning applications. *Journal of medical imaging and radiation oncology*, 65(5):545–563, 2021.

[Coe02] Carlos A. Coello Coello. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, 191(11-12):1245–1287, 2002.

[CTW+21] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 1667–1684, 2021.

[CWZ+22] Anda Cheng, Jiaxing Wang, Xi Sheryl Zhang, Qiang Chen, Peisong Wang, and Jian Cheng. Dpnas: Neural architecture search for deep learning with differential privacy. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 6358–6366, 2022.

[CY22] Michael Choi and Nancy Yang. Differential privacy in gan training for generative models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):456–470, 2022.

[CZ23] Grace Chen and Helen Zhao. Public-private optimization for differentially private deep learning. *International Conference on Machine Learning*, pages 789–802, 2023.

[CZZ+23] Huiqiang Chen, Tianqing Zhu, Tao Zhang, Wanlei Zhou, and Philip S Yu. Privacy and fairness in federated learning: On the perspective of tradeoff. *ACM Computing Surveys*, 56(2):1–37, 2023.

[DAS+21] Amandeep Dhull, Oluwaseun Ajayi, Fnu Suya, Jingjing Li, Pradeep K. Murukannaiah, and W. Jim Zheng. Differentially private model pruning. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 903–912, 2021.

[DBH+] Soham De, Leonard Berrada, Jamie Hayes, Samuel L. Smith, and Borja Balle. Unlocking high-accuracy differentially private image classification through scale. In *International Conference on Learning Representations, ICLR 2023*.

[DBH+22] Soham De, Leonard Berrada, Jamie Hayes, Samuel L. Smith, and Borja Balle. Unlocking high-accuracy differentially private image classification through scale. *CoRR*, abs/2204.13650, 2022.

[DC23] Emiliano De Cristofaro. What is synthetic data? the good, the bad, and the ugly. *arXiv preprint arXiv:2303.01230*, page 60, 2023.

[DCBJ22] Alexander Dockhorn, Tianshi Cao, Arash Behboodi, and Jörn-Henrik Jacobsen. Differentially private diffusion models. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 28109–28122, 2022.

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. Often cited as NAACL 2019.

[DDS+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[DFAP21] Friedrich Dörmann, Osvald Frisk, Lars Nørvang Andersen, and Christian Fischer Pedersen. Not all noise is accounted equally: How differentially private learning benefits from large sampling rates. In *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP), Gold Coast, Australia, October 25-28, 2021*, pages 1–6. IEEE, 2021.

[DMNS06a] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer, 2006.

[DMNS06b] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference (TCC)*, pages 265–284. Springer, 2006.

[DPAM00] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. In *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858, 2000.

[DPV]

[DR14a] Cynthia Dwork and Aaron Roth. *The Algorithmic Foundations of Differential Privacy*, volume 9 of *Foundations and Trends in Theoretical Computer Science*. now Publishers Inc., 2014.

[DR14b] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[DT17] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[EKN+17] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639):115–118, 2017.

[ERR+19] Andre Esteva, Antoine Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Kathleen Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeffrey Dean. A guide to deep learning in healthcare. *Nature medicine*, 25(1):24–29, 2019.

[FJR15a] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015*, pages 1322–1333. ACM, 2015.

[FJR15b] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 1322–1333, 2015.

[FKA+18] M Fridadar, M Klang, E Amitai, M Goldberger, and H Greenspan. Ieee 15th int. symp. biomed. *Imaging*, 289, 2018.

[FPN22] Tiantian Feng, Raghuveer Peri, and Shrikanth Narayanan. User-level differential privacy against attribute inference attack of speech emotion recognition in federated learning. *arXiv preprint arXiv:2204.02500*, 2022.

[GBC16] Ian J. Goodfellow, Yoshua Bengio, and Aaron C. Courville. *Deep Learning*. Adaptive computation and machine learning. MIT Press, 2016.

[GBDL+16] Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *International conference on machine learning (ICML)*, pages 201–210. PMLR, 2016.

[GCZG24] Zhe Guo, Su-Hua Chen, Ling Zhou, and Li-Hua Gong. Optical image encryption and authentication scheme with computational ghost imaging. *Applied Mathematical Modelling*, 131:49–66, 2024.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing (STOC)*, pages 169–178, 2009.

[GKN17a] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. 2017. NIPS Workshop on Private Multi-Party Machine Learning.

[GKN17b] Robin C. Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. In *NIPS Workshop on Private Multi-Party Machine Learning*, 2017.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game, or a completeness theorem for protocols with honest majority. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing (STOC)*, pages 218–229, 1987.

[Goo19a] Google. Tensorflow privacy, 2019.

[Goo19b] Google. TensorFlow Privacy Optimizers, 2019.

[GPAM+14] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[GPC+16] Varun Gulshan, Lily Peng, Marc Coram, Martin C Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan, Kasumi Widner, Tom Madams, Jorge Cuadros, et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *jama*, 316(22):2402–2410, 2016.

[GS23] Helen Gao and Ian Smith. Noisy gradient descent for differentially private regression. *Journal of Privacy and Confidentiality*, 13(2):234–250, 2023.

[GTF21] Yifan Gao, Ivan Titov, and Sina Fazelpour. Efficient hyperparameter optimization for differentially private machine learning. In *NeurIPS 2021 Workshop on Privacy in Machine Learning*, 2021.

[GYAT13] Amir Hossein Gandomi, Xin-She Yang, Amir Hossein Alavi, and Siamak Talatahari. Bat algorithm for constrained optimization tasks. *Neural Computing and Applications*, 22:1239–1255, 2013.

[HAPC17] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. Deep models under the gan: information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 603–618, 2017.

[HBAL22] Naoise Holohan, Stefano Braghin, Pól Mac Aonghusa, and Killian Levacher. Diffprivlib: A general purpose library for differentially private data analysis. In Jose M. Peña and Thomas Dyhre Nielsen, editors, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 5405–5407. ijcai.org, 2022.

[HC19] Munir Husein and Il-Yop Chung. Day-ahead solar irradiance forecasting for microgrids using a long short-term memory recurrent neural network: A deep learning approach. *Energies*, 12(10), 2019.

[HG09] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

[HGH+14] Justin Hsu, Marco Gaboardi, Andreas Haeberlen, Sanjeev Khanna, Arjun Narayan, Benjamin C. Pierce, and Shayanayati. Differential privacy: An economic perspective on data, information loss and utility. *Commun. ACM*, 57(8):80–88, 2014.

[HGJ+19] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR, 2019.

[HKH23] Mikko Heikkilä, Samuel Kaski, and Antti Honkela. Differentially private bayesian linear regression. *Journal of Machine Learning Research*, 24(147):1–48, 2023.

[HKP13] Azim Heydari, Farshid Keynia, and Nasser Shahsavari Pour. Machinery cost prediction based on a new neural network method. 2013.

[HL22] Jane Huang and Karen Liu. Direct feedback alignment with differential privacy for deep learning. *European Conference on Machine Learning*, pages 890–910, 2022.

[HMvdW+20] Charles R. Harris, K. Jarrod Millman, Stéfan van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin

Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with numpy. *Nat.*, 585:357–362, 2020.

[HPW17] J Benson Heaton, Nick G Polson, and Jan Hendrik Witte. Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry*, 33(1):3–12, 2017.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[HSS12] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8):2, 2012.

[HSW+] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

[HTG17] Ehsan Hesamifard, Hassan Takabi, and Mahdi Ghasemi. Cryptodl: Deep learning over encrypted data. 2017.

[HV14] T. Hassanzadeh and H. Vojodi. A comparison study between pso and bat algorithm for feed forward neural network training. *International Journal of Intelligent Systems and Applications*, 6(1):60–66, 2014.

[HZC+17] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[HZX+18] Tyler Hunt, Congzheng Zhu, Yuanliang Xu, Sujaya Chattopadhyay, and Reza Shokri. Chiron: Privacy-preserving machine learning as a service. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 1721–1736, 2018.

[IHM+] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size. *arXiv preprint arXiv:1602.07360*.

[INS+19] Ravi Iyengar, Joseph P. Near, Dawn Song, Thomas Steinke, Sanasi Wagh, and Lun Yu. Towards practical differential privacy for high-dimensional datasets. *arXiv preprint arXiv:1909.01056*, 2019.

[IS15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 448–456. JMLR.org, 2015.

[JCB+20] Matthew Jagielski, Nicholas Carlini, David Berthelot, Alexey Kurakin, and Nicolas Papernot. High accuracy and high fidelity extraction of neural networks. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 935–952, 2020.

[JK19] Justin M. Johnson and Taghi M. Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):27, 2019.

[JT14] Prateek Jain and Abhradeep Thakurta. (near) optimal differentially private learning of binary halfspaces and literals. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing (STOC '14)*, pages 339–348, 2014.

[JYvdS19] James Jordon, Jinsung Yoon, and Mihaela van der Schaar. PATE-GAN: generating synthetic data with differential privacy guarantees. In *International Conference on Learning Representations, ICLR 2019*, 2019.

[KB15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[KKP20] Vinayak Kulkarni, Mitali Kulkarni, and Atul Pant. Survey of personalization techniques for federated learning. *arXiv preprint arXiv:2003.08673*, 2020.

[KL23] Wendy Kim and Xander Lee. Transfer learning with dp-sgd for privacy-preserving deep learning. *Neural Computation*, 35(7):890–910, 2023.

[KLN+11a] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM J. Comput.*, 40(3):793–826, 2011.

[KLN+11b] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

[KMA+21] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konecny, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard G. M. Morris, Satyen Kale, Patrick P. K. Lee, Karthik Natarajan, Virginia Smith, A. T. Suresh, Florian Tramèr, Mingqing Chen, and Jakub Konečný. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1-2):1–210, 2021.

[KMY+16] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated optimization: Distributed machine learning with communication efficiency. 2016.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1106–1114, 2012.

[KST12] Daniel Kifer, Adam Smith, and Abhradeep Thakurta. Private convex empirical risk minimization and high-dimensional regression. In *Proceedings of the 25th Annual Conference on Learning Theory, COLT 2012*, volume 23 of *JMLR Proceedings*, pages 25.1–25.36, 2012.

[KVH23] László Kurmos, Bálint Vass, and András Horváth. Differentially private curriculum learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 206 of *Proceedings of Machine Learning Research*, pages 7956–7976. PMLR, 2023.

[KW+13] Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[LC24] Carol Li and David Chen. Lower precision training with dp-sgd for privacy-preserving deep learning. *IEEE Transactions on Neural Networks*, 35(2):123–140, 2024.

[LCR+21] Shen Li, Xiaoyu Chen, Sathya N. Ravi, Bang An, and Vikas K. Garg. Differentially private neural architecture search. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021*, volume 139 of *Proceedings of Machine Learning Research*, pages 6565–6574. PMLR, 2021.

[LeC15] Yann LeCun. Deep learning & convolutional networks. In *2015 IEEE Hot Chips 27 Symposium (HCS), Cupertino, CA, USA, August 22-25, 2015*, pages 1–95. IEEE, 2015.

[LGL+21] Renju Liu, Luis Garcia, Zaoxing Liu, Botong Ou, and Mani B. Srivastava. Secdeep: Secure and performant on-device deep learning inference framework for mobile and iot devices. In *IoTDI '21: International Conference on Internet-of-Things Design and Implementation, Virtual Event / Charlottesville, VA, USA, May 18-21, 2021*, pages 67–79. ACM, 2021.

[LHZ+23] Yijing Li, Nan He, Yaqiong Zhang, Huimin Li, and Chengyi Zhang. Optimization of capital structure for urban water ppp projects using mezzanine financing: Sustainable perspective. *Journal of Construction Engineering and Management*, 149(8):04023059, 2023.

[LM24] Zoe Lee and Alice Ma. Variational inference with differential privacy for generative models. *Machine Learning*, 113(1):456–470, 2024.

[LPBK20] Julien Launay, Iacopo Poli, François Boniface, and Florent Krzakala. Direct feedback alignment scales to modern deep learning tasks and architectures. *Advances in neural information processing systems*, 33:9346–9360, 2020.

[LTLH22] Xuechen Li, Florian Tramèr, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. In *International Conference on Learning Representations, ICLR 2022*, 2022.

[LX24] Karen Liu and Michael Xu. Compute-memory optimization for privacy-preserving deep learning. *IEEE Transactions on Computers*, 73(4):234–250, 2024.

[LZ23] David Lin and Eve Zhu. Contrastive pre-training with differential privacy for deep learning. *International Conference on Learning Representations*, pages 123–140, 2023.

[MAE+17] H. Brendan McMahan, Galen Andrew, Úlfar Erlingsson, Hubert Eichner, Ilya Mironov, Nicolas Papernot, Steve Chien, and Kunal Talwar. Learning

differentially private recurrent language models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017.

[MHS21] Takayuki Miura, Satoshi Hasegawa, and Toshiki Shibahara. MEGEX: data-free model extraction attack against gradient-based explainable AI. *CoRR*, abs/2107.08909, 2021.

[Mie99] Kaisa Miettinen. *Nonlinear Multiobjective Optimization*, volume 12 of *International Series in Operations Research Management Science*. Springer US, 1999.

[Mir17] Ilya Mironov. Rényi differential privacy. In *30th IEEE Computer Security Foundations Symposium, CSF 2017, Santa Barbara, CA, USA, August 21-25, 2017*, pages 263–275. IEEE Computer Society, 2017.

[ML24] Bob Ma and Carol Lin. Pretrained language models with dp-sgd for privacy-preserving nlp. *Transactions of the Association for Computational Linguistics*, 12:345–360, 2024.

[MMR+17a] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics (AISTATS)*, pages 1273–1282. PMLR, 2017.

[MMR+17b] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 2017.

[MNMG21] Ninareh Mehrabi, Muhammad Naveed, Fred Morstatter, and Aram Galstyan. Exacerbating algorithmic bias through fairness attacks. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 8930–8938. AAAI Press, 2021.

[MRTZ18] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private models with secure federated learning. In *Federated Learning Workshop, NeurIPS 2018*, 2018.

[MSCS19] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019*

IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019, pages 691–706. IEEE, 2019.

[MSDCS19] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 691–706. IEEE, 2019.

[MT07] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, Rhode Island, USA*, pages 94–103. IEEE Computer Society, 2007.

[MTKC22] Harsh Mehta, Abhradeep Thakurta, Alexey Kurakin, and Ashok Cutkosky. Large scale transfer learning for differentially private image classification. *arXiv preprint arXiv:2205.02973*, 2022.

[MXZ24] Hugh Brendan McMahan, Zheng Xu, and Yanxiang Zhang. A hassle-free algorithm for strong differential privacy in federated learning systems. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 842–865, 2024.

[MZ17] Payman Mohassel and Yupeng Zhang. Secureml: A system for scalable privacy-preserving machine learning. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 19–38. IEEE, 2017.

[NSH19] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE symposium on security and privacy (SP)*, pages 739–753. IEEE, 2019.

[NT23] Vankamamidi Srinivasa Naresh and Muthusamy Thamarai. Privacy-preserving data mining and machine learning in healthcare: Applications, challenges, and solutions. *WIREs Data. Mining. Knowl. Discov.*, 13(2), 2023.

[PAE+17] Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian J. Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[PAF+23] Martin Pelikan, Sheikh Shams Azam, Vitaly Feldman, Jan Silovsky, Kunal Talwar, Tatiana Likhomanenko, et al. Federated learning with differential privacy for end-to-end speech recognition. *arXiv preprint arXiv:2310.00098*, 2023.

[Pap19] Nicolas Papernot. Machine learning at scale with differential privacy in {TensorFlow}. In *2019 {USENIX} Conference on Privacy Engineering Practice and Respect ({PEPR} 19)*, 2019.

[PFM+21] Carlo Pasquini, Marco Flocco, Shahrokh Mahpod, Andrea Apicella, Pietro Conca, Emanuele Formaggio, Michelangelo Stanzani Maserati, Randall Balestriero, and Mauro Conti. Exploring vulnerability of medical image diagnosis models to reconstruction attacks. *arXiv preprint arXiv:2112.09421*, 2021.

[PG23] Yvonne Park and Zach Gao. Random process priors for differentially private deep learning. *Conference on Computer Vision and Pattern Recognition*, pages 234–248, 2023.

[PGM+19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035, 2019.

[PNMM21] Aman Priyanshu, Rakshit Naidu, Fatemehsadat Mireshghallah, and Mohammad Malekzadeh. Efficient hyperparameter optimization for differentially private deep learning. *arXiv preprint arXiv:2108.03888*, 2021.

[PNS20] Santisudha Panigrahi, Anuja Nanda, and Tripti Swarnkar. A survey on transfer learning. In *Intelligent and Cloud Computing: Proceedings of ICICC 2019, Volume 1*, pages 781–789. Springer, 2020.

[PS21] Marlotte Pannekoek and Giacomo Spigler. Investigating trade-offs in utility, fairness and differential privacy in neural networks. *arXiv preprint arXiv:2102.05975*, 2021.

[PSL19] Venkatadheeraj Pichapati, Reza Shokri, and Lingjuan Lyu. Adaclip: Adaptive clipping for private stochastic gradient descent. In *Workshop on Privacy Preserving Machine Learning (PPML@NeurIPS)*, 2019.

[PSM+18] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with pate. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC,*

Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net, 2018.

[PTS+21a] Nicolas Papernot, Abhradeep Thakurta, Shuang Song, Steve Chien, and Úlfar Erlingsson. Tempered sigmoid activations for deep learning with differential privacy. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*, pages 9161–9169. AAAI Press, 2021.

[PTS+21b] Nicolas Papernot, Abhradeep Thakurta, Shuang Song, Steve Chien, and Úlfar Erlingsson. Tempered sigmoid activations for deep learning with differential privacy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9312–9321, 2021.

[PVG+11] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[RBPT21] Théo Ryffel, Francis Bach, David Pointcheval, and Andrew Trask. Functional encryption for privacy-preserving federated learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 23389–23400, 2021.

[RFA23] Benladgham Rafika, Hadjila Fethallah, and Belloum Adam. A pareto-optimal privacy-accuracy settlement for differentially private image classification. In *2023 5th International Conference on Pattern Analysis and Intelligent Systems (PAIS)*, pages 1–7. IEEE, 2023.

[RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[RHW86] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[RHZ+22] Zakaria Rguibi, Abdelmajid Hajami, Dya Zitouni, Amine Elqaraoui, and Anas Bedraoui. Cxai: Explaining convolutional neural networks for medical imaging diagnostic. *Electronics*, 11(11), 2022.

[SAB15] Muhammad Sabt, Mohamed Achemlal, and Abdelmadjid Bouabdallah. Trusted execution environment: what it is, and what it is not. *arXiv preprint arXiv:1506.05461*, 2015. Often cited as TRUST 2015.

[SHK+14] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, 2014.

[SK19] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[SLA12] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, pages 2951–2959, 2012.

[SLL18] Min Jae Shin, Jung Hee Lee, and Kristen LeFevre. Differentially private recommendation system using randomized response. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2189–2198, 2018.

[SMDH13] Ilya Sutskever, James Martens, George E. Dahl, and Geoffrey E. Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 1139–1147. JMLR.org, 2013.

[SMS20] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multi-task optimization under privacy. In *International Conference on Learning Representations (ICLR)*, 2020.

[SP24] Tina Sun and Victor Park. Public data priors for differentially private deep learning. *International Journal of Computer Vision*, 132(3):678–690, 2024.

[SSP+03] Patrice Y Simard, David Steinkraus, John C Platt, et al. Best practices for convolutional neural networks applied to visual document analysis. In *Icdar*, volume 3. Edinburgh, 2003.

[SSSS17a] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *Proceedings of the IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 3–18. IEEE Computer Society, 2017.

[SSSS17b] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

[STR+18] Hoo-Chang Shin, Neil A Tenenholtz, Jameson K Rogers, Christopher G Schwarz, Matthew L Senjem, Jeffrey L Gunter, Katherine P Andriole, and Mark Michalski. Medical image synthesis for data augmentation and

anonymization using generative adversarial networks. In *Simulation and Synthesis in Medical Imaging: Third International Workshop, SASHIMI 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Proceedings 3*, pages 1–11. Springer, 2018.

[SZ22] Oliver Smith and Paul Zhao. Gradient clipping with noise for differentially private deep learning. *Annual Conference on Neural Information Processing Systems*, 35:123–140, 2022.

[TAL16] Sasha Targ, Diogo Almeida, and Kevin Lyman. Resnet in resnet: Generalizing residual architectures. *arXiv preprint arXiv:1603.08029*, 2016.

[TCB+19] Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. *Advances in neural information processing systems*, 32, 2019.

[TGL+20] Neil C Thompson, Kristjan Greenewald, Keeheon Lee, Gabriel F Manso, et al. The computational limits of deep learning. *arXiv preprint arXiv:2007.05558*, 10, 2020.

[TKP19] Reza Torkzadehmahani, Peter Kairouz, and Benedict Paten. DP-GAN: Differentially private generative adversarial network to synthesize artificial tabular data. *arXiv preprint arXiv:1911.04427*, 2019.

[TLB+19] David Tellez, Geert Litjens, Péter Bándi, Wouter Bulten, John-Melle Bokhorst, Francesco Ciompi, and Jeroen Van Der Laak. Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology. *Medical image analysis*, 58:101544, 2019.

[TRM+20] Om Thakkar, Swaroop Ramaswamy, Rajiv Mathews, Matthew Andrea, and H. Brendan McMahan. Understanding the tradeoffs in differentially private machine learning. In *Workshop on Theory and Practice of Differential Privacy (TPDP@ICML)*, 2020.

[TTK21] Harsh P. Thakur, Parth Thaker, and Ketan Kotecha. Differentially private collaborative filtering using user-based locality sensitive hashing. In *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, pages 890–895, 2021.

[TZJ+16] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 601–618, 2016.

[VLHT24] Abhishek Vyas, Po-Ching Lin, Ren-Hung Hwang, and Meenakshi Tripathi. Privacy-preserving federated learning for intrusion detection in iot environments: A survey. *IEEE Access*, 12:127018–127050, 2024.

[VNP+20] Anamaria Vizitiu, Cosmin Ioan Nita, Andrei Puiu, Constantin Suciu, and Lucian Mihai Itu. Applying deep neural networks over homomorphic encrypted medical data. *Comput. Math. Methods Medicine*, 2020:3910250:1–3910250:26, 2020.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[W+19] Chris Waites et al. Pyvacy: Privacy algorithms for pytorch, 2019.

[WBK19] Yu-Xiang Wang, Borja Balle, and Shiva Kasiviswanathan. Subsampled rényi differential privacy and analytical moments accountant. *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics, AISTATS 2019*, 89:1225–1233, 2019.

[WL23] Eve Wang and Frank Liu. Efficient gradient clipping for differentially private optimization. *Advances in Neural Information Processing Systems*, 36:456–470, 2023.

[WPL+17] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106, 2017.

[WSS15] Yuting Wang, Shuang Song, and Dawn Song. Differentially private bayesian inference for exponential families. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics, AISTATS 2015*, volume 38, pages 1034–1043, 2015.

[XKG19] Cong Xie, Sanmi Koyejo, and Indranil Gupta. Asynchronous federated optimization. 2019.

[XLW+18] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739*, 2018.

[XSW+24] Zifei Xu, Sayeh Sharify, Tristan Webb, Xin Wang, et al. Understanding the difficulty of low-precision post-training quantization of large language models. *arXiv preprint arXiv:2410.14570*, 2024.

[XWD+18] Linjun Xie, Kun Wang, Anupam Datta, Peter Z. Kairouz, and Wang Lie. Differentially private generative adversarial network. In *International Conference on Learning Representations, ICLR 2018, Workshop Track*, 2018.

[XZ23] Nancy Xu and Oliver Zhou. Differentially private transformer training for speech synthesis. *Interspeech*, pages 567–580, 2023.

[Yan10a] Xin-She Yang. *Nature-Inspired Metaheuristic Algorithms*. Luniver Press, 2010.

[Yan10b] Xin-She Yang. A new metaheuristic bat-inspired algorithm. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, volume 284 of *Studies in Computational Intelligence*, pages 65–74. Springer, 2010.

[Yao86] Andrew C Yao. How to generate and exchange secrets. In *27th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 162–167. IEEE, 1986.

[YG12] Xin-She Yang and Amir Hossein Gandomi. Bat algorithm: a novel approach for global engineering optimization. *Engineering Computations*, 29(5):464–483, 2012.

[YHO+19] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[YK23] Paul Yang and Rachel Kim. Federated learning with differential privacy for large-scale deep learning. *IEEE Transactions on Information Forensics and Security*, 18(5):345–360, 2023.

[YSS+21] Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Misha Laskin, Albert Gordon, Xiang Gao, John Nguyen, Pierre Fernandez, and et al. Opacus: User-friendly differential privacy library in pytorch. `https://github.com/pytorch/opacus`, 2021.

[YSW+23] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

[YYW+23] Hongmiao Yan, Lin Yao, Yun Wang, Xiaoxiong Zhong, and Junsuo Zhao. A stationary random process based privacy-utility tradeoff in differential privacy. In *2023 International Conference on High Performance Big Data and Intelligent Systems (HDIS)*, pages 178–185. IEEE, 2023.

[YZC+21] Da Yu, Huishuai Zhang, Wei Chen, Jian Yin, and Tie-Yan Liu. Large scale private learning via low-rank reparametrization. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021*, volume 139 of *Proceedings of Machine Learning Research*, pages 12213–12224. PMLR, 2021.

[YZL20] Zuobin Ying, Yun Zhang, and Ximeng Liu. Privacy-preserving in defending against membership inference attacks. In Benyu Zhang, Raluca Ada Popa, Matei Zaharia, Guofei Gu, and Shouling Ji, editors, *PPMLP'20: Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice, Virtual Event, USA, November, 2020*, pages 61–63. ACM, 2020.

[ZDC24] Ying Zhao, Jia Tina Du, and Jinjun Chen. Scenario-based adaptations of differential privacy: A technical survey. *ACM Computing Surveys*, 56(8):1–39, 2024.

[ZH23] Frank Zhu and Grace Huang. Dp-sgd with fine-tuning for privacy-preserving nlp. *Empirical Methods in Natural Language Processing*, pages 567–580, 2023.

[Zhu23] Yuqing Zhu. *Making Differential Privacy Practical via Modern Privacy Accountings and Data-adaptive Analysis*. University of California, Santa Barbara, 2023.

[ZL07] Qingfu Zhang and Hui Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.

[ZLH19] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

[ZLL+21] Junge Zhang, Runsheng Liu, Thomas P. Y. Lin, Nathalie Baracaldo, Bu Yuan, and Ruoxi Jia. Differentially private bayesian deep learning via functional variational inference. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021*, pages 13827–13840, 2021.

[ZLYQ20] Jiliang Zhang, Chen Li, Jing Ye, and Gang Qu. Privacy threats and protection in machine learning. In Tinoosh Mohsenin, Weisheng Zhao, Yiran Chen, and Onur Mutlu, editors, *GLSVLSI '20: Great Lakes Symposium on VLSI 2020, Virtual Event, China, September 7-9, 2020*, pages 531–536. ACM, 2020.

[ZQD+20] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.

[ZS24] Quentin Zhou and Robert Sun. Marginal-based synthetic data generation with differential privacy. *Journal of Artificial Intelligence Research*, 80(2):123–145, 2024.

[ZW24] Alice Zhang and Bob Wu. Bayesian networks with differential privacy for deep learning. *Journal of Machine Learning Research*, 25(3):1–25, 2024.

[ZWHC21] Ruicheng Zhao, Yuzheng Wang, Chaoyang He, and Li Chen. Multi-objective optimization for differentially private deep learning. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 847–856, 2021.

[ZXCZ18] Zhichao Zhang, Shugong Xu, Shan Cao, and Shunqing Zhang. Deep convolutional neural network with mixup for environmental sound classification. In *Chinese conference on pattern recognition and computer vision (prcv)*, pages 356–367. Springer, 2018.

[ZY24] Ian Zhao and Jane Yang. A holistic evaluation framework for differentially private models. *ACM Transactions on Privacy and Security*, 27(1):15–30, 2024.

[ZZK+20] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020.

[ZZL+20] Yang Zhao, Jun Zhao, Meng Li, Kwok-Yan Lam, Congcong Miao, and Min Wu. Local differential privacy based federated learning for internet-of-things. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, pages 762–771, 2020.

[ZZX+12] Jingchen Zhang, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, and Marianne Winslett. Functional mechanism: regression analysis under differential privacy. *Proc. VLDB Endow.*, 5(11):1364–1375, 2012.