Data privacy preservation and uncertainty estimation in machine learning

A Thesis Submitted for the Degree of Doctor of Philosophy

By

Wanxin Sui

Department Electronic and Electrical Engineering, Brunel University London

2026

Declaration of Authorship

I hereby affirm that the content presented in this paper has not been previously submitted to any other institution or location for the purpose of earning a degree. The material included in this work has been developed under the supervision of my suervisor and reflects the outcomes of discussions and collaborative efforts with experts in the field.

"After years of dedication, countless late nights, and moments of doubt, the journey is finally coming to an end. As I prepare to close this chapter, I find myself reflecting on the quiet strength of perseverance and the beauty of the challenges that have shaped me. Every step, every experiment, every piece of advice from my mentors has left a lasting mark on my soul. This place, once unfamiliar, has become a second home filled with memories of growth, friendships, and endless curiosity. I leave with a heart full of gratitude—for the lessons learned, for the hands that supported me, and for the path that lies ahead. These years will forever remain etched in my heart, guiding me in the future."

Wanxin Sui

Abstract

This thesis addresses the challenges of data privacy in the field of machine learning, with a focus on privacy threats and uncertainty estimation in decentralized learning environments. As data grows exponentially and machine learning models are widely adopted, the challenge of effectively using data while ensuring privacy protection has become paramount. To tackle this issue, the thesis proposes a task-adaptive privacy protection method that combines differential privacy and local differential privacy techniques, dynamically adjusting the noise level to maximize model utility while ensuring privacy protection. Additionally, this thesis explores privacy attacks in decentralized learning, including reconstruction attacks on Decentralized Gradient Descent (D-GD) and Gossip averaging protocols, and proposes corresponding defense strategies. To improve model robustness, a normalizing flow-based uncertainty estimation method is introduced to detect anomalous predictions and apply additional privacy measures. Experiments demonstrate the effectiveness of these methods in various application scenarios, including real estate valuation and breast cancer detection. Ultimately, this thesis proposes a multi-layer defense mechanism that combines privacy protection and uncertainty estimation, offering stronger privacy protection and model robustness in complex decentralized learning scenarios.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Maozhen Li, who not only opened the door to academic research for me but also provided invaluable guidance and inspiration throughout my academic journey. Professor Li is not just an academic mentor but also a kind-hearted elder who offered unwavering support and care. His wisdom, patience, and foresight deeply influenced me, helping me overcome various challenges and giving me the strength to keep moving forward. I am truly honored to have been his student. I am also very grateful to Professor Hongying Meng for his guidance and support during my study. I also extend my heartfelt thanks to my family—my parents, for their endless emotional and financial support during the most difficult moments of my research, and my husband, whose unwavering love and encouragement have been my constant source of strength. I am grateful for the companionship and encouragement of my friends, who have been a source of strength during the long and sometimes lonely path of research. Lastly, I am sincerely grateful to the scholars and experts in my field, whose invaluable advice and guidance have broadened my horizons and deepened my understanding. Their mentorship has taught me to think critically, solve problems effectively, and continue growing in this field.

Contents

List of Figures

List of Tables

List of Abbreviations

ML	Machine Learning
DP	Differential Privacy
LDP	Local Differential Privacy
FIPP	Fair Information Practice Principles
GDPR	General Data Protection Regulation
IoT	Internet of Things
CNN	Convolutional Neural Network
DNN	Deep Neural Network
MSE	Mean Squared Error
RNN	Recurrent Neural Network
D-GD	Decentralized Gradient Descent

List of Symbols

$\mathbb{E}$	Expected value
i.i.d.	independent and identical distributions
$\langle \cdot \rangle$	Denote expectations

Dedicated to my family…

Chapter 1

Introduction

This chapter briefly describes the background, challenges, motivations, major contributions to the problem investigated and the structure of this thesis.

1.1 Background

Given the exponential increase in data and the extensive use of machine learning [1], data privacy protection has become a critical issue. Many machine learning models [2], [3], [4], [5] rely on training with large amounts of data, which often contain personal sensitive information. However, the sharing and use of this information may lead to privacy breaches, especially when the dataset contains personally identifiable features. To balance the contradiction between data usage and privacy protection, privacy-preserving technologies such as Differential Privacy (DP) have emerged, providing a mathematically rigorous way to protect data privacy.

At the same time, decentralized learning (such as federated learning [6], [7], [8], [9], [10], [11] and decentralized gradient descent [12], [13], [14]) has emerged as a new paradigm in machine learning, aiming to reduce privacy risks and improve system robustness by training models collaboratively among participating nodes without centralized raw data. However, despite the theoretical avoidance of centralized data storage, decentralized learning still faces potential privacy threats in practical applications. Specifically, when model

updates are shared among participating nodes, attackers may exploit these updates to reconstruct private data, leading to privacy breaches. This risk is even more significant in scenarios where the network structure is sparse or node connections are limited.

In addition to privacy protection, uncertainty estimation is also an important challenge facing the current machine learning community. In the context of data complexity and task diversity, quantifying the uncertainty of model predictions becomes crucial. Uncertainty estimation not only helps improve the robustness of the model in adversarial environments and out-of-distribution (OOD) data but also provides further support for privacy protection. For example, by identifying predictions with high uncertainty, the system can take additional privacy protection measures to reduce the risk of potential privacy breaches.

Therefore, the current machine learning community faces numerous challenges in decentralized learning, privacy protection, and uncertainty estimation. How to effectively protect data privacy while quantifying and addressing model uncertainty in decentralized learning environments has become a comprehensive issue that urgently needs to be resolved. By organically combining differential privacy, uncertainty estimation, and decentralized learning, this thesis aims to explore solutions to these challenges.

1.2 Research Issues and Challenges

Although existing privacy protection methods, such as differential privacy and local differential privacy (LDP), can theoretically protect data privacy, they still face many challenges in practical applications. In particular, when dealing with multidimensional data and complex machine learning tasks, balancing privacy protection and model performance becomes especially difficult. Existing methods often reduce data utility while protecting privacy,

thereby affecting model prediction accuracy.

Moreover, there are many unique issues and challenges in decentralized learning. Since nodes in decentralized learning cannot directly access each other's data, model training relies on local communication between nodes. However, this local communication mechanism tends to cause delays in model updates and information loss, especially in scenarios with sparse network structures. Limited communication between nodes can severely affect the convergence speed and overall performance of the model. The uncertainty and asynchrony of communication between nodes also increase the risk of data leakage, particularly in situations where nodes frequently go offline or have unstable connections, allowing attackers to gradually obtain information about other nodes' data by monitoring the nodes.

Another significant challenge is the heterogeneity of participating nodes. In real-world applications, nodes may have significant differences in computational power, data distribution, and network connectivity. This heterogeneity can lead to uneven contributions from nodes during model training, affecting the convergence and performance of the global model. In addition, heterogeneity introduces extra complexity for privacy protection, and how to achieve efficient collaborative learning while ensuring the privacy of different nodes remains an open research question.

In terms of uncertainty estimation, many existing methods typically assume access to global information, which is difficult to achieve in decentralized learning. Each node can only access local data, making it very challenging to effectively assess global uncertainty. Furthermore, current uncertainty estimation methods often perform poorly in terms of accuracy and robustness when dealing with high-dimensional and multimodal data, especially in the presence of OOD data and adversarial samples. In such cases, the model may severely underestimate its prediction uncertainty, leading to unreliable decisions. Therefore, designing algorithms capable of effectively quantifying

uncertainty in decentralized environments, allowing each node to accurately assess prediction uncertainty based on local information, is a significant challenge in current research.

1.3 Research Motivation

The motivation of this study is to explore how to better protect data privacy in the context of decentralized learning while maintaining model efficiency. By combining the advantages of differential privacy and local differential privacy, this thesis proposes a task-adaptive privacy protection method that aims to intelligently adjust the noise level for specific tasks involving multidimensional data, thereby maximizing model utility while maintaining privacy protection. In addition, this thesis conducts an in-depth analysis of existing privacy attacks to better understand their mechanisms and design more effective defense strategies accordingly.

Decentralized learning alone may not prevent data privacy leaks, one of our motivations is to design an enhanced privacy protection method to address this issue. The motivation is to supplement the shortcomings of decentralized learning with additional privacy protection mechanisms, ensuring that communication and model updates between nodes can be conducted without compromising private data. Specifically, this thesis proposes an enhanced privacy protection method that combines differential privacy with collaborative defense strategies, aiming to reduce the possibility of attackers reconstructing private data through model updates, thereby balancing efficient communication and privacy protection.

In terms of uncertainty estimation, the goal is to develop an uncertainty quantification method suitable for decentralized environments, enabling each node to effectively evaluate the model's uncertainty based on only local data.

This is crucial for improving the model's robustness in adversarial environments and with OOD data. This thesis combines Normalizing Flow models to propose a new uncertainty estimation method, which can enhance the model's sensitivity to anomalous data and improve its ability to handle complex tasks.

Moreover, this research aims to establish a multi-layer defense mechanism by combining privacy protection and uncertainty estimation. When predictions with high uncertainty are identified, the system can take additional privacy protection measures to further reduce potential privacy leakage risks. This combination will help us achieve higher privacy protection and model robustness in complex decentralized learning scenarios.

Compared to fixed-noise schemes or purely Bayesian inference–based approaches, our method has two key distinctions: first, task adaptivity, whereby noise injection intensity is automatically tuned according to local data distribution and measured model uncertainty; second, joint optimization, in which privacy loss and uncertainty estimation error are simultaneously minimized within a unified update framework. This design enables our approach to maintain a stable privacy budget while dynamically responding to model prediction uncertainty, achieving a superior balance between privacy preservation and robustness in complex scenarios such as adversarial attacks and large-scale distributed networks.

1.4 Main Contributions

Firstly, this thesis explores privacy threats in decentralized learning, focusing specifically on vulnerabilities in Decentralized Gradient Descent (D-GD) [12], [13], [14] and Gossip Averaging protocols [15], [16], [17], [18]. this thesis describes in detail how malicious nodes can exploit these protocols to reconstruct private data from neighboring and distant nodes, highlighting

the privacy challenges that decentralized learning still faces. These findings laid the foundation for developing stronger privacy protection mechanisms in subsequent chapters.

Secondly, this thesis proposes a task-adaptive privacy protection method aiming at balancing privacy protection and model utility in multidimensional data scenarios. The proposed method dynamically adjusts the noise level based on task requirements to ensure ϵ-LDP (Local Differential Privacy) while minimizing task loss. Experimental evaluations show that this approach outperforms benchmark methods in terms of task accuracy, particularly under different privacy constraints in applications such as real estate valuation and breast cancer detection.

Lastly, Data privacy preservation adds uncertainty to machine learning algorithms. For this purpose, this thesis investigates uncertainty estimation in decentralized learning, using Normalizing Flow models to improve model robustness in adversarial and out-of-distribution (OOD) scenarios. By quantifying aleatoric and epistemic uncertainty, the proposed uncertainty-aware framework helps detect predictions with high uncertainty, allowing additional privacy protection measures to be taken, thus enhancing overall model reliability and privacy protection.

The main contributions of this thesis are as follows:

- This thesis conducts a detailed analysis of privacy attacks in decentralized learning and proposed corresponding defense strategies, particularly in the context of Gossip averaging and decentralized gradient descent protocols.

- This thesis proposes a task-adaptive local differential privacy method to improve model performance while protecting privacy in multidimensional data scenarios.

- This thesis researches uncertainty estimation in deep learning and used Normalizing Flow models to improve the model's robustness in adversarial environments and with OOD data.

In the following sections of this thesis, this thesis provides a detailed introduction to these specific topics.

1.5 Structure of the Thesis

The rest of this thesis is organised as follows:

- Chapter 2 reviews the literature related to this study, including differential privacy, decentralized learning, and privacy threats.
- Chapter 3 provides a detailed introduction to privacy threats and attack mechanisms in decentralized learning.
- Chapter 4 proposes a task-adaptive privacy protection method and discusses its application in multidimensional data scenarios.
- Chapter 5 explores uncertainty estimation methods in deep learning and demonstrates improvements based on Normalizing Flow models.
- Chapter 6 concludes the study and discusses future research directions.

Chapter 2

Literature Review

2.1 Data privacy

Privacy refers to private matters that an individual is unwilling to share with others or discuss in public [19]. Research shows that there is no single definition of privacy, as definitions vary across contexts such as location, discipline, and time [20]. There is no universally accepted definition of data privacy, so organizations are at risk of having incomplete data privacy [20]. Privacy is a concept that encompasses a variety of social contexts, and the general concept of privacy is very intuitive when viewed from the daily lives of citizens [21]. Researchers often treat privacy and data confidentiality as different concepts. Data privacy refers to the appropriate use of available data by any individual or organization, as distinct from data security, which ensures confidentiality, integrity, and availability of data [22]. The distinction between privacy and data confidentiality becomes blurred in practice because privacy refers to the right of an individual to control the information collected about themselves, while confidentiality refers to the responsibility of the data manager (i.e., the agent responsible for data development, such as a statistical agency) not to disclose the personal information and/or identity of the subjects to unauthorized parties [23]. It is believed that the problem is that data privacy is considered more complex than data security, and therefore many policymakers choose to avoid defining data privacy [20]. In this

study, the three elements of data privacy from the US NIST and the German ULD were compared with an aim that these elements could become the basis for a common definition of data privacy in the future. The analysis concludes that there are two different approaches to defining data privacy: one focuses on the actual implementation of data privacy protection measures (NIST) and the other focuses on defining the highest standards that data processors must comply with (ULD) [20]. Facebook announced a massive restructuring and strategic change to put privacy at the heart of its strategy [24]. It includes: What is data privacy? What does data privacy include? What privacy protection methods and tools are being developed around the world? How to combine privacy with technology? The questions raised above encompass the main concerns.

2.2 Evolution of Data Privacy Protection

The concept of the 'right to privacy' did not emerge in international law until after World War II, when it first appeared in Article 12 of the Universal Declaration of Human Rights, which states that no one shall be subjected to arbitrary interference with their privacy, family, home or correspondence [25]. The concept of privacy rights began to take shape, focusing primarily on the protection of personal life. In 1890, Samuel Warren and Louis Brandeis published 'The Right to Privacy' in the Harvard Law Review, which first articulated the concept of privacy rights. Warren and Brandeis began their article with the fundamental principle that 'individuals should be adequately protected in person and property.' They acknowledged that this is a changing principle that has been readjusted over the centuries in response to political, social, and economic changes. Warren and Brandeis argued that the court had no reason to prohibit the publication of the letters, either under

existing doctrine or under property rights; instead, they argued that the principle of privacy is to protect personal writings and any other intellectual or emotional creations [26]. In the 1970s, with the widespread adoption of computer technology, the ability to collect and process personal data increased significantly, raising concerns about data privacy. People's lives have undergone tremendous changes due to the penetration of information technology. From the agricultural era of farming to the development of cities, and then to the information revolution, people have gradually adopted new technologies. These changes are rapidly changing our lifestyle and social structure. In 1973, the U.S. introduced the Fair Information Practice Principles (FIPP), which became the foundation for data privacy protection. FIPP is a set of organizational rules designed to protect individual privacy and has been at the heart of privacy discussions for the past 50 years. FIPP focuses primarily on fairness and privacy, that is, control over access to other people's information, while also taking into account the need for anti-commercial preferences. Ideally, if businesses and governments follow these guidelines when collecting, storing, sharing, and using personal information, people will be able to enjoy privacy when interacting with these large institutions [27]. Fair Information Practice Principles (FIPP)-based rules form the basis of U.S. sectoral regulations for privacy protection, covering federal government matters other than law enforcement and national security, while Europe has also adopted FIPP-based regulations throughout its private economy [27]. In 1974, the U.S. passed the Privacy Act, regulating government handling of personal data. The Privacy Act Amendments establish fair information practice guidelines to govern the collection, maintenance, use, and dissemination of personal information maintained by federal agencies in their systems of records. The Privacy Act requires agencies to make information available to the public about their systems of records by publishing it in the Federal Register and prohibits the disclosure of personal records in systems of records without

the individual's written consent. In addition, the Act provides individuals with ways to access and amend their records and establishes recordkeeping requirements for various agencies [1]. In 1980, the Organisation for Economic Co-operation and Development (OECD) released the Guidelines on the Protection of Privacy and Transborder Flows of Personal Data, outlining eight privacy protection principles that had a significant global impact. As information technology penetrates into all aspects of economic and social life, and computer data processing becomes increasingly important and influential, the Organisation for Economic Cooperation and Development (OECD) decided in 1980 to publish international policy guidelines on the protection of privacy and cross-border flows of personal data. In these guidelines: a) 'Data Controller' means a party that has the power under domestic law to determine the content and purpose of personal data, regardless of whether the data is collected, stored, processed or transmitted by that party or its agents; b) 'Personal Data' means any information related to an identified or identifiable individual (data subject); c) 'Cross-border flows of personal data' means the transfer of personal data between countries [28]. In 1995, the European Union adopted the Data Protection Directive (95/46/EC), providing a legal framework for data privacy protection among member states and setting conditions for cross-border data transfers. The European Commission will review the list of countries currently deemed to provide an adequate level of protection for personal data. At the same time, data protection authorities will explore data protection certification at national and EU level to award seals and logos to services to increase consumer confidence [28]. Japan's Act on the Protection of Personal Information (APPI) is one of the oldest data protection laws in Asia, having been enacted in 2003 to provide a legal basis for the collection, use, and management of personal data. In

[1] https://www.justice.gov/opcl/privacy-act-1974

September 2015, a series of high-profile data breaches rocked Japan, prompting a major revision of the Act, indicating that its original requirements were no longer adequate to meet today's needs [28]. The revised APPI came into effect on May 30, 2017.

2.3 Major Data Privacy Incidents

On April 13, 2023, the Irish Data Protection Authority fined Meta (formerly Facebook) €1.2 billion for transferring personal data to the United States, the largest General Data Protection Regulation (GDPR) fine to date [17], it is reflecting the serious nature of the violations and the EU's commitment to enforcing data privacy standards. GDPR sets out the rules companies must follow when transferring user data outside the EU [2].

In March 2023, TikTok CEO Shou Zi Chew appeared before the U.S. House Committee on Energy and Commerce, where he faced intense questioning from U.S. lawmakers [29]. Throughout 2024, TikTok faced ongoing scrutiny and legal challenges in various countries over concerns about data security and privacy. In February, the app was banned on government devices in the U.S. and several other countries due to fears that user data could be accessed by the Chinese government. These actions highlight the geopolitical dimensions of data privacy issues.

In April 2024, a major healthcare provider in the U.S. reported a data breach that exposed the personal and medical information of over 10 million patients. The breach, caused by a ransomware attack, underscored the vulnerabilities in the healthcare sector and the critical need for robust data security measures. The following data may have been compromised: patient ID, social security number, name, address, phone number, spousal information, driver's license number, credit card number with expiration date,

[2] BBC News: Meta fined €1.2 billion over data transfer to U.S.

demographic information, personal health data, health assessment test results, health insurance information, prescription information, clinical notes, lab tests, etc. [3].

In January 2024, Amazon was fined $50 million by the French data protection authority (CNIL) for failing to comply with cookie consent regulations under GDPR. The fine highlighted ongoing issues with how major tech companies handle user consent and data tracking .

2.4 General Data Protection Regulation

In 2016, the EU passed the General Data Protection Regulation (GDPR), the most stringent and comprehensive data privacy regulation to date, covering data subject rights, data processor obligations, cross-border data transfers, and more. The EU GDPR privacy law came into effect in May 2018 and applies to any organization that collects and processes the personal information of EU citizens within or outside the EU [30]. GDPR introduces significant changes on how personal data is processed, the guidance and auditing methods of data protection authorities, and the personal rights of data subjects [31]. GDPR proposes the principle of data minimization, requiring organizations to collect only data that is absolutely necessary for the intended purpose and not encouraging additional data collection. This principle is consistent with the concept of privacy design. At the same time, organizations must also ensure the accuracy of the personal data they process and take measures to promptly correct inaccurate information to improve the reliability of the data. In order to prevent unauthorized access, changes or disclosures, organizations must implement appropriate technical and organizational measures to protect the integrity and confidentiality of personal data [32].

[3] 14 biggest healthcare data breaches, Accessed: 26 July 2024

This Regulation establishes guidelines for safeguarding the privacy and rights of individuals concerning the processing of their personal data. It aims to ensure the free movement of such data within the European Union while simultaneously protecting fundamental rights, particularly the right to personal data protection. The Regulation prohibits any restrictions on the movement of personal data within the Union based on concerns for individuals' data protection, thus promoting a balance between data privacy and the free flow of information [4].

Its key principles are:

- **Lawfulness, Fairness, and Transparency**: Personal data processing must adhere to legal standards, treat individuals fairly, and maintain transparency about data usage.
- **Purpose Limitation**: Personal data should only be collected for specific, legitimate purposes and should not be used in ways that are incompatible with those purposes, except under certain conditions for specific purposes like research or archiving.
- **Data Minimisation**: Organizations should collect only the personal data that is necessary for the intended purposes, minimizing the amount of data collected and retained.
- **Accuracy**: Personal data should be kept accurate and up to date, with organizations taking steps to rectify any inaccuracies promptly.
- **Storage Limitation**: Personal data should only be stored for as long as necessary for the purposes for which it was collected, with extended storage for specific purposes requiring appropriate safeguards.
- **Integrity and Confidentiality**: Personal data should be processed securely, protecting against unauthorized access, loss, or damage, with

[4] Norwegian Data Protection Authority, 2018

organizations implementing necessary measures to safeguard data integrity and confidentiality.

Personal data: personal data encompasses any information pertaining to an identified or identifiable individual, known as the data subject. An identifiable person can be directly or indirectly identified using various identifiers, including but not limited to, their name, identification number, location data, online identifiers, or specific factors related to their physical, physiological, genetic, mental, economic, cultural, or social identity. This definition underscores the broad scope of personal data and highlights the importance of protecting individuals' privacy and rights in an increasingly data-driven world.

Processing: Processing refers to any action or series of actions carried out on personal data or collections of personal data, whether automated or not. These actions include gathering, recording, organizing, structuring, storing, adapting, altering, retrieving, consulting, using, disclosing through transmission, distributing, or otherwise providing access to, aligning or merging, restricting, erasing, or destroying personal data.

Controller: The controller refers to an individual or entity, whether a natural person or a legal entity, a public authority, agency, or any other organization. The controller, either independently or in collaboration with others, is responsible for deciding the purposes and methods of processing personal data. In cases where Union or Member State law dictates the purposes and methods of processing, the controller's identity or the criteria for selecting the controller may be specified by Union or Member State law.

In conclusion, GDPR underscores the recognition of data protection as a fundamental right and emphasizes the importance of respecting fundamental rights and freedoms in all data processing activities. It builds upon the harmonization efforts initiated by Directive 95/46/EC to establish a balanced framework that ensures data protection while facilitating the free movement

of personal data within the Union. This framework aims to achieve a harmonized level of data protection across Member States, despite potential variations in national laws. Cooperation among Member States is essential to effectively exchange personal data while ensuring a high level of protection. The Regulation establishes clear standards, powers, and sanctions to ensure control and certainty in data processing activities. Its adoption into national law ensures its enforceability and effectiveness in safeguarding individuals' rights.

The Digital Services Act (DSA) cooperation policy aims to enhance coordination and collaboration among EU member states to ensure effective implementation of the DSA. Digital Services Coordinators (DSCs) in each member state are responsible for monitoring and enforcing DSA regulations, sharing information, conducting joint investigations, and addressing cross-border digital service challenges [33]. This cooperation mechanism improves regulatory consistency, enhances user protection, and promotes fair competition. Algorithm transparency and accountability are key components, with the European Centre for Algorithmic Transparency (ECAT) providing scientific and technical support to the European Commission for overseeing systemic obligations of Very Large Online Platforms (VLOPs) and Very Large Online Search Engines (VLOSEs) [5]

2.5 Differential Privacy

Differential Privacy (DP) is a mathematical framework designed to protect individual privacy when releasing statistical information from datasets. It allows data holders to share aggregate patterns of a group while limiting the exposure of specific details about individuals [34], [35]. This is accomplished

[5] European Centre for Algorithmic Transparency, Accessed: 26 July 2024

by introducing carefully designed noise into the statistical computations, ensuring that the utility of the results is preserved while theoretically restricting the ability to make inferences about any particular individual within the dataset.

DP can also be understood as a constraint imposed on algorithms that publish aggregated information from a statistical database, thereby limiting the disclosure of private information contained in the records. For instance, some government agencies use DP algorithms to release demographic data or other statistical summaries while ensuring the confidentiality of survey responses. Companies also employ these algorithms to collect user behavior data while controlling what even internal analysts can access.

In essence, an algorithm is considered differentially private if an observer cannot determine whether a particular individual's data was used in the computation based solely on the output. DP is often discussed in relation to identifying individuals whose information may be present in a database. Although it does not directly address identification or re-identification attacks, differentially private algorithms are theoretically proven to resist such threats[36].

With the rapid advancement of machine learning and artificial intelligence, data privacy has become a critical issue. Machine learning models rely on vast amounts of data for training, which often includes sensitive personal information. A major challenge is how to effectively utilize this data without compromising privacy. DP, a mathematically rigorous framework, offers a solution to this problem. It enables the protection of individual data privacy when releasing statistical information or training models, thereby balancing the needs of both machine learning and data privacy [6], [37], [38], [39].

DP works by injecting carefully calibrated noise into data analysis or machine learning computations, making it impossible for those accessing the

analysis results to determine whether a particular individual's data was used for training or calculations. This method theoretically provides strong resistance against various privacy attacks, such as re-identification attacks. Under the protection of DP, even the data holders or analysts themselves cannot determine whether a specific individual's information is included in the dataset [40], [41].

2.6 Decentralized Learning

2.6.1 The Basic Architecture and Advantages of Decentralized Learning

Decentralized learning allows multiple nodes to share and collaboratively train models directly, eliminating the reliance on a central server. This distributed architecture enhances data privacy and system resilience because data is retained locally, reducing the risk associated with centralized storage while avoiding single points of failure. However, this architecture faces challenges such as low communication efficiency and difficulty in model convergence, especially when data is non-IID (non-independent and identically distributed) among nodes [42]. To address these issues, researchers have proposed efficient communication protocols based on network topology and optimization algorithms adapted to asynchronous environments to improve model convergence speed and overall performance [43], [44].

2.6.2 Privacy Protection and Adversarial Challenges in Decentralized Learning

While decentralized learning reduces the risk of data leakage, sharing gradient information can still expose nodes to privacy attacks, such as reconstruction attacks and adversarial attacks. To enhance privacy protection, differential privacy methods are often employed to add noise to gradients, limiting the possibility of attackers reconstructing original data from the gradients. Furthermore, the introduction of Byzantine fault tolerance algorithms effectively filters out malicious node updates, improving model robustness in the face of adversarial attacks. Techniques like Local Differential Privacy (LDP) and homomorphic encryption further enhance privacy protection in decentralized learning, particularly in scenarios requiring highly secure communication between nodes [43], [44], [45], [46], [47], [48].

2.6.3 Technological Innovations and Application Scenarios of Decentralized Learning

In the research of decentralized learning, technological innovations addressing communication overhead, asynchronous updates, and personalized requirements have received considerable attention. For example, Graph Neural Networks (GNNs) [49] are used to model node topology, enhancing the understanding of information flow and the efficiency of parameter updates. The design of adaptive learning rates and personalized models allows each node to dynamically optimize based on local data. Moreover, decentralized learning has wide applications in smart cities, edge computing, and healthcare, effectively ensuring privacy protection and real-time response. In healthcare, for instance, decentralized learning allows for local training and collaboration, reducing data transmission to a central server while meeting strict privacy requirements [11], [49], [50].

2.7 Uncertainty Estimation in Machine Learning

2.7.1 Background and Types of Uncertainty Estimation

Data privacy preservation adds uncertainty to machine learning algorithms. Uncertainty estimation is a technique used to quantify the uncertainty in model predictions, which is crucial for enhancing the safety, reliability, and interpretability of machine learning systems. There are primarily two types of uncertainty: epistemic uncertainty and aleatoric uncertainty. Epistemic uncertainty arises due to incomplete knowledge of the model and can be reduced by acquiring more data. For instance, Bayesian Neural Networks (BNNs) introduce probability distributions over neural network weights to capture this type of uncertainty [51]. Aleatoric uncertainty, on the other hand, is due to inherent noise in the data and cannot be reduced by increasing the dataset size. It is typically handled by modeling the observation noise, such as using probabilistic models to add noise parameters to the observations [52].

2.7.2 Common Methods for Uncertainty Estimation

Common methods for uncertainty estimation include Bayesian Neural Networks, Monte Carlo Dropout, deep ensembles, and probabilistic prediction-based models. Bayesian Neural Networks are powerful yet computationally expensive, capable of handling epistemic uncertainty [53]. Monte Carlo Dropout approximates Bayesian inference by enabling dropout during inference, significantly reducing computational complexity [51]. Deep ensembles train multiple models with different initializations and combine their predictions to capture diversity and uncertainty, which is particularly effective

for improving model robustness [54]. Additionally, probabilistic prediction-based models handle aleatoric uncertainty by modeling output as a probability distribution, which captures the inherent noise in observational data [52].

2.7.3 Applications and Challenges of Uncertainty Estimation

Uncertainty estimation has important applications in various fields. In autonomous driving, it helps identify high-risk scenarios, while in medical diagnostics, it can flag predictions that need further manual review, enhancing system reliability [55]. Uncertainty estimation is also useful for detecting adversarial attacks and outliers, thus improving model robustness against unknown inputs [56]. However, uncertainty estimation still faces several challenges in practical applications, such as high computational complexity, large storage requirements, and instability in high-dimensional data. Researchers are exploring more efficient and resource-friendly approaches to improve the applicability and reliability of uncertainty estimation in real-world systems.

2.8 Related Works to this Research

The use of random walk algorithms for privacy preservation in decentralized learning effectively balances privacy and accuracy, though lacking real-world scenarios [57]. Differences in privacy risks between centralized and decentralized systems show that decentralized systems are more resilient but require advanced privacy mechanisms [58]. DP in deep learning models poses challenges in balancing privacy and utility, requiring more optimized algorithms [59]. The integration of DP in decentralized deep learning shows effectiveness in enhancing privacy but faces challenges with heterogeneous data [60]. DP has been applied to decision tree-based federated learning models demonstrates acceptable accuracy loss but degradation due to DP

noise [61]. Federated matrix factorization methods are privacy-preserving but show trade-offs in performance, particularly with sparse data [62]. Adaptive privacy mechanisms in federated learning enhance privacy without significantly affecting utility, but implementation complexity remains a challenge [63]. A review of privacy-preserving machine learning techniques recommends hybrid approaches for better real-world integration [64]. Privacy-preserving decentralized federated learning using Shamir's secret sharing addresses privacy in time-varying graphs but struggles with latency [65]. Practical federated learning systems require privacy and communication optimizations [64]. A survey of decentralized federated learning (DFL) highlights its scalability, privacy, and efficiency advantages, though lacking practical evaluations [65]. Decentralized deep learning using consensus algorithms ensures privacy and synchronization but faces challenges in high-latency environments[66]. Privacy-preserving decentralized federated learning using methods like secret sharing and DP shows effectiveness but requires careful tuning to avoid delays, especially in larger networks [67].

2.8.1 Comparison and Contrast of Findings

The literatures collectively illustrate a common emphasis on balancing privacy and performance in decentralized learning. Privacy preservation is a core trend across all efforts, with multiple approaches such as DP, random walks, and secret sharing being applied ([57], [59], [60], [61], [62], [64], [67]). However, each method presents unique trade-offs: DP, though widely used, often results in degraded model utility due to added noise ([59], [61]), whereas random walk and consensus algorithms aim to mitigate these impacts but lack real-world deployment evidence ([57], [66]). Hybrid approaches

are suggested for better integration of privacy measures, particularly in practical scenarios ([63]). A notable theoretical divergence arises regarding decentralized vs. centralized systems: decentralized models are portrayed as inherently more resilient to privacy breaches ([58], [68]), but they also require sophisticated mechanisms to address communication efficiency and scalability ([65], [68]). The scalability and reliability of decentralized federated learning are recurring themes, with most works pointing to challenges in high-latency environments and difficulties in ensuring consistent performance across heterogeneous data ([60], [64], [66], [67]).

2.8.2 Implications for Research

These findings support and challenge various aspects of this research on privacy-preserving decentralized learning. The emphasis on balancing privacy and model utility directly aligns with the focus on achieving optimal trade-offs between these two objectives. [59], [61] and [69] highlight the challenges of maintaining utility in the face of differential privacy, which aligns with developing more efficient noise mechanisms to mitigate this issue. Moreover, [58] and [68]'s findings regarding the resilience of decentralized systems support the decision to explore decentralized learning frameworks as a more robust alternative to centralized ones. However, the challenges related to scalability, communication efficiency, and practical implementation ([65], [66], [68]) provide a critical perspective, underscoring the importance of addressing these gaps to ensure that the proposed privacy-preserving methods are viable in real-world applications.

2.9 Summary

This chapter provided a comprehensive review of recent research in decentralized learning, differential privacy, and uncertainty estimation in machine

learning. It started with an overview of the evolution of data privacy protection and its development in international law, including significant regulations such as the GDPR. This chapter then delved into the basics of DP and its applications in decentralized learning, highlighting the challenges in balancing data privacy and model utility. In decentralized learning, researchers have explored how decentralized structures can effectively reduce privacy risks but also face potential risks when sharing gradient information. Various privacy protection mechanisms, such as Local Differential Privacy (LDP) and homomorphic encryption, are also discussed as methods to enhance privacy in decentralized learning.

Furthermore, the chapter discussed the importance of uncertainty estimation in machine learning and its applications in model prediction. Uncertainty estimation helps improve model robustness in adversarial environments and with out-of-distribution data. Common estimation methods, such as Bayesian Neural Networks, Monte Carlo Dropout, and deep ensembles, are reviewed in detail. Finally, this chapter summarized the main findings and controversies in existing literature, comparing the strengths and limitations of decentralized versus centralized systems in terms of privacy protection, scalability, and system performance. These discussions provide the theoretical foundation for subsequent research and highlight issues like privacy and communication efficiency that need further exploration in future studies.

In summary, this chapter established a solid theoretical foundation for understanding privacy protection and uncertainty estimation in decentralized learning. In the following chapters, this thesis will discuss how to verify the privacy leakage problem of decentralized learning through attack mechanisms, and further propose an task adaptive privacy protection algorithm for multi-dimensional data.

Chapter 3

Analysis of Privacy Threats in Decentralized Learning

This chapter delves into privacy attacks in decentralized learning, particularly attacks on Decentralized Gradient Descent (D-GD) [12], [13] and Gossip Averaging protocols [15], [70], [71]. It will also introduce relevant foundational knowledge to help understand the background and mechanisms of these attacks.

3.1 Overview of Decentralized Learning

Decentralized Learning

Decentralized learning is a distributed machine learning framework that does not rely on a central server [14]. It involves multiple nodes learning locally on their own data and exchanging model updates through peer-to-peer communication with neighboring nodes, thereby gradually forming a global model across the network. This approach eliminates the need for a central server, enhancing the system's robustness, especially in large-scale distributed scenarios such as the Internet of Things (IoT) and edge computing [14]. Each node shares update information with its neighbors, allowing the system to approximate a global optimal solution without the need to clean or transfer

data. Furthermore, decentralized learning is highly fault-tolerant, as there is no single point of failure, ensuring continuous model training even when nodes fail or communication is interrupted.

Federated Learning

Federated learning is a distributed learning paradigm that relies on a central coordinating server, aiming to train a global model while preserving data privacy [7], [8]. The key idea is that multiple devices (such as smartphones or IoT devices) independently train models on their local data without uploading the data to a central server. Each node sends its local model updates to the central server, which aggregates these updates (typically using algorithms like FedAvg [72]) and then distributes the updated global model back to the nodes. This centralized aggregation method allows federated learning to utilize the computational power of distributed devices while keeping user data private. Federated learning is widely applied in multi-device systems, such as mobile and smart devices, where privacy preservation is critical.

Differences and Similarities

The primary difference between decentralized learning and federated learning lies in the reliance on a central node. Federated learning depends on a central server to aggregate and distribute model updates, whereas decentralized learning achieves model updates through peer-to-peer communication without the need for central coordination. In federated learning, communication typically occurs indirectly through the central server, while decentralized learning uses direct communication between neighboring nodes, offering higher fault tolerance.

Despite these differences, both methods share common goals and mechanisms, focusing on protecting data privacy and reducing communication

costs through local computation. In practice, decentralized learning and federated learning can be combined, for example, by optimizing communication with a decentralized topology while using a few central nodes for coordination, thereby enhancing system robustness and efficiency. Recent literature suggests many intersections between the two approaches, with opportunities to leverage each other's strengths to address the challenges of large-scale distributed systems.

Although decentralized learning is widely regarded as offering stronger privacy protection due to its decentralized architecture and peer-to-peer communication model, this architecture also introduces certain potential privacy risks. Since each node can only observe updates from its neighbors, attackers may need to infer other nodes' contributions indirectly. However, this does not mean decentralized learning is entirely immune to privacy attacks. In some cases, an attacker may track update patterns across multiple communication rounds and gradually reconstruct the individual contributions of more distant nodes, especially in sparsely connected network topologies. Furthermore, while the repeated propagation of updates might introduce additional noise or information mixing, malicious nodes could exploit this by analyzing transmission paths or the characteristics of data mixing to extract sensitive information. Therefore, despite its inherent privacy-preserving advantages, decentralized learning still faces various privacy attack risks, highlighting the importance of further research into defensive mechanisms.

3.2 The risks of decentralized learning

In decentralized learning, while its peer-to-peer architecture eliminates reliance on a central server, enhancing scalability and robustness, it still faces significant privacy risks. In algorithms like Decentralized Gradient Descent (D-GD) [12], [13], [16], [73], nodes collaborate by sharing local model updates

with their neighbors without sharing raw data. Intuitively, this mechanism should enhance privacy protection because each node can only observe the contributions of its immediate neighbors, and data from more distant nodes is mixed and propagated multiple times before being indirectly observed. However, this assumption does not hold true in some cases.

For instance, attackers can exploit the Gossip Averaging protocol by leveraging model update patterns, treating each message as a data point in a system of linear equations, enabling the reconstruction of private data from multiple nodes, including non-neighboring ones. Similarly, the Decentralized Gradient Descent algorithm is vulnerable to analogous attacks, where an attacker can track the propagation of gradient updates, gradually reconstructing the gradient information of target nodes. This gradient information can then be used in conjunction with gradient inversion attacks such as Gradient Inversion [9], [74], [75] to fully reconstruct the original data.

To assess the potential vulnerabilities specific to decentralization, this thesis focuses on the strongest type of privacy leakage, namely data reconstruction, performed by the weakest type of attackers: honest-but-curious nodes. These attacker nodes follow the protocol but attempt to reconstruct as much data as possible from other nodes using only their legitimate observations within the protocol.

Therefore, the decentralized design does not inherently guarantee data privacy, as it remains vulnerable to privacy attacks that exploit communication between nodes and model updates. In the following sections, this thesis delves into the specifics of these attacks, including Gossip Averaging and D-GD attacks, and analyze how these attacks leverage data updates to achieve data reconstruction.

3.2.1 Reconstruction attacks in federated learning

Privacy attacks on server-based federated algorithms are a key area of active research. Starting from the possibility of reconstructing data using gradient descent [76], known as gradient inversion, several improvements have been proposed [9], [74], [75]. These include techniques that can separate gradients aggregated over large batches [10]. In the following attack on D-GD, this thesis primarily focuses on reconstructing gradients based on the observations from attacker nodes. In the next step, this thesis treats gradient inversion attacks as a black box to further reconstruct data from the reconstructed gradients. Thus, the proposed work complements gradient inversion attacks and stands to benefit from advances in this area.

3.3 Setting of Decentralized Algorithms

In this section, we introduce the setup for decentralized learning and the algorithms we focus on. We assume a fixed, undirected, and connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $|\mathcal{V}| = n$, and each edge $\{u, v\} \in \mathcal{E}$ represents that nodes u and v can communicate. The neighbors of a node u are denoted as $\mathcal{N}(u) = \{v : \{u, v\} \in \mathcal{E}\}$. Each node u holds a private value $x_u \in \mathbb{R}^d$. For notational simplicity, we assume in the following sections that $d = 1$, but this can be easily generalized to the vector case.

This simplification of representing local datasets with a single element has also been applied in the work of Pasquini [77], and is appropriate in the context of decentralized averaging, where each node only has one private value. In the case of Decentralized Gradient Descent, recent research has introduced reconstruction attacks that exploit gradients aggregated over multiple data points [10], [78]. These methods can be effectively utilized as a black box in our attack, allowing us to abstract away this aspect and instead focus on the specific characteristics of decentralized learning.

3.3.1 Gossip Averaging protocol

In this study, we focus on synchronous gossip-based algorithms, where at each iteration, every node calculates a weighted average of its own value and the values of its neighboring nodes, based on the weights provided by a predefined gossip matrix.

Definition (Gossip matrix). The gossip matrix $W \in [0,1]^{n\times n}$ is a doubly stochastic matrix applied over the graph $\mathcal{G}$, satisfying $W^\top \mathbf{1} = W\mathbf{1} = \mathbf{1}$. $W_{uv} > 0$ holds only if there is an edge between nodes u and v.

Gossip averaging. The gossip matrix is used to iteratively compute the average of private values $(x_v)_{v\in\mathcal{V}}$. Each node v is initialized with $\theta_v^0 = x_v$, and the gossip averaging iteration [70] is expressed as:

$$\theta^{t+1} = W\theta^t. \tag{3.1}$$

This process converges to the global average at a geometric rate, with the convergence speed governed by the spectral gap of W [70].

Accelerated gossip. The accelerated version of the gossip algorithm [17] achieves faster convergence by replacing the multiplication by W with a polynomial of W. It is important to note that this acceleration does not affect the information accessible to a node, as results obtained via accelerated gossip can be transformed back to the non-accelerated version through a simple linear transformation. For clarity, we will focus on the non-accelerated version in our discussions.

3.3.2 Decentralized Gradient Descent

Decentralized Gradient Descent (D-GD). In decentralized learning, each node aims to optimize an objective function of the form $f(\theta) = \sum_{v=1}^{n} L(\theta, x_v)$, where θ represents the model's parameters and L is a differentiable loss function. One of the most widely used algorithms to achieve this is D-GD [12],

[13]. In each iteration of D-GD, every node performs a local gradient update with a learning rate $\eta > 0$, followed by a gossip averaging step with its neighbors. Let θ_v^0 be the initial parameter setting for each node v. The local gradient of node v at iteration t, scaled by η, is given by

$$g_v^t = -\eta \nabla_{\theta_v} L(\theta_v^t, x_v). \tag{3.2}$$

The gradient update step in D-GD is then expressed as

$$\theta^{t+\frac{1}{2}} = \theta^t + g_v^t, \tag{3.3}$$

and the gossip averaging step is represented as

$$\theta^{t+1} = W\theta^{t+\frac{1}{2}}. \tag{3.4}$$

Under certain conditions, D-GD converges to a global or local optimum of the objective function f [13].

3.4 Defining Privacy Attacks and Threat Models

3.4.1 Overview of Privacy Attacks

The risk of data leakage in decentralized learning has been acknowledged by the research community, as demonstrated by numerous proposed defenses against privacy breaches. Many of these methods are based on differential privacy, where nodes introduce noise during their updates. Earlier research provided local differential privacy guarantees mainly through local noise addition [79], [80]. However, more recent studies show that decentralized learning can enhance these baseline privacy protections under an adaptive threat model [81], [82]. Additionally, other methods introduce correlated noise to

restrict the ability to extract information from local updates [18], [83], [84], [85], [86].

The proposed work demonstrates that privacy attacks in decentralized learning pose a tangible threat. Regarding such attacks, we identify two recent studies by Pasquini [77] and Dekker [87]. Pasquini attack focuses solely on direct neighbors, similar to existing attacks in federated learning. On the other hand, Dekker. explored a different scenario where nodes perform a sequence of secure aggregations with their neighbors without considering a learning objective. In contrast, the proposed attack is capable of reconstructing data from more distant nodes, effectively addressing the unique privacy challenges in decentralized settings.

3.4.2 Threat Models

In this work, we focus on "honest-but-curious" attackers, which represent a subset of nodes that adhere to the protocol but aim to extract as much information as possible from their observations. The set of attacker nodes is denoted as $A \subseteq V$, and their neighbors, excluding the attackers themselves, are represented by $\mathcal{N}(A) = \bigcup_{a \in A} \mathcal{N}(a) \setminus A$. Typically, we assume that the attacker nodes correspond to the first $|A|$ nodes. When $|A| > 1$, we assume that the knowledge is fully shared among attackers, even if there are no direct edges connecting them in the network graph. Additionally, the attackers are assumed to have knowledge of the network graph and the gossip matrix, an assumption that is often reasonable in real-world cases such as social networks. The relevance of this assumption is further discussed in Appendix A.

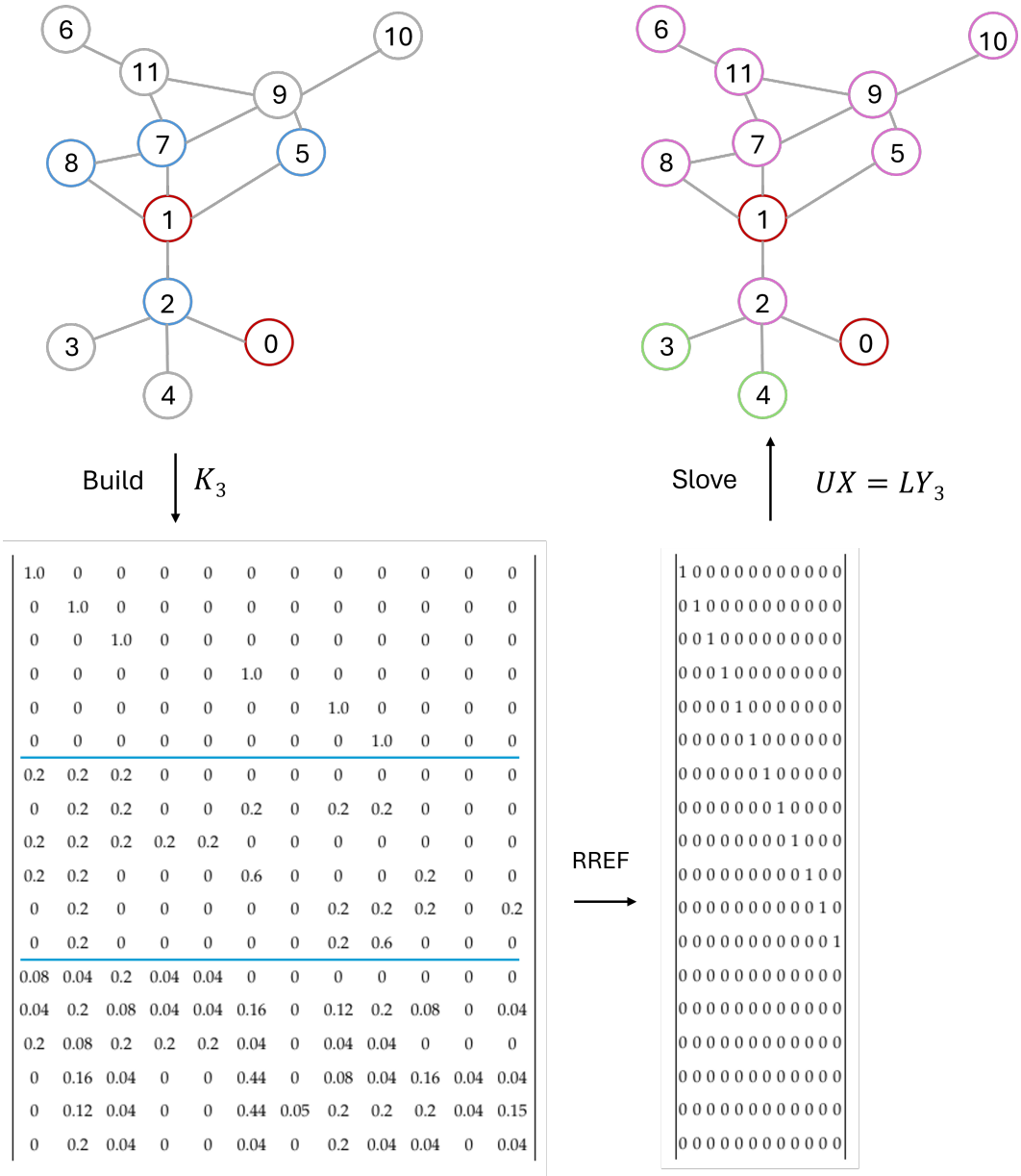

FIGURE 3.1: A summary of our attack on gossip averaging. Attackers 0 and 1 (red) receive updates from nodes 2, 5, 7, and 8 (blue). After multiple iterations, this process results in the knowledge matrix K_3. The reduced row echelon form (RREF matrix U) reveals that only nodes 3 and 4 are non-reconstructible (green), while the private values of all other nodes (purple) are leaked.

3.4.3 Privacy Attacks on Gossip Averaging Protocols

In this section, we present our attack on the gossip averaging process. The main idea behind our attack is that each message θ_t^v received by an attacker

node $a \in A$ from one of its neighbors $v \in \mathcal{N}(a)$ corresponds to a linear equation. In this equation, the unknowns are the private values of the nodes in the graph, and the coefficients depend on the gossip matrix W, which is assumed to be known by the attackers. Our attack involves precisely constructing this system of linear equations $K_T X = Y_T$ and solving it to reconstruct as many private values as possible. A visual summary of this reconstruction process is provided in Figure 3.1.

Gathering linear equations

For a given gossip matrix W and the set of attackers $\mathcal{A}$, we describe how attackers systematically construct a system of linear equations to capture the private values they gather during T iterations of gossip averaging. Specifically, the attackers receive $|\mathcal{A}| + T \cdot |\mathcal{N}(\mathcal{A})|$ values, each of which can be associated with a linear combination of the n private inputs. Initially, the attackers are aware of their inputs, corresponding to the first $|\mathcal{A}|$ values. At each round t of gossip, attackers receive a value $\theta_v^{(t)}$ from each neighbor $v \in \mathcal{N}(\mathcal{A})$. These values represent linear combinations of private inputs, with the weights determined by the powers of the gossip matrix.

Formally, this system of linear equations is represented as $K_T X = Y_T$, where $X = (x_0, \ldots, x_n) \in \mathbb{R}^n$ denotes the vector of private values the attackers aim to reconstruct, and K_T, Y_T are defined accordingly.

Knowledge matrix and observation vector

The knowledge matrix $K_T \in \mathbb{R}^{|\mathcal{A}|+T\cdot|\mathcal{N}(\mathcal{A})|\times n}$ is defined through Algorithm 1. The observation vector $Y_T \in \mathbb{R}^{|\mathcal{A}|+T\cdot|\mathcal{N}(\mathcal{A})|}$ is formed by stacking the attackers' private values together with the messages $\theta_v^{(t)}$ received from their neighbors, where $0 \leq t \leq T-1$ and $v \in \mathcal{N}(\mathcal{A})$. The view of the attackers is represented by the pair (K_T, Y_T).

It is important to note that K_T depends solely on the gossip matrix, while Y_T reflects the private values.

To make this concrete, let us consider a simple example. Suppose there is a graph with n nodes, and the attacker is located at position 0. The attacker knows their private input x_0, stored in the first entry of Y_T, which corresponds to the insertion of the one-hot vector $(1, 0, \ldots, 0)$ into the first row of K_T. If nodes 1 and 2 are the attacker's neighbors, they will send $\theta_1^{(0)} = x_1$ and $\theta_2^{(0)} = x_2$ to the attacker during iteration 0 (stored in the second and third entries of Y_T), corresponding to the insertion of $(0, 1, 0, \ldots, 0)$ and $(0, 0, 1, 0, \ldots, 0)$ into the second and third rows of K_T. In iteration 1, they will send:

$$\theta_1^{(1)} = \sum_j W_{j,1} x_j \quad \text{and} \quad \theta_2^{(1)} = \sum_j W_{j,2} x_j, \tag{3.5}$$

corresponding to the rows $(W_{0,1}, \ldots, W_{n-1,1})$ and $(W_{0,2}, \ldots, W_{n-1,2})$. Generally, at each iteration t, the attacker receives values $\theta_1^{(t)} = \sum_j W_{j,1}^{(t)} x_j$ and $\theta_2^{(t)} = \sum_j W_{j,2}^{(t)} x_j$ from their neighbors, as determined by the gossip averaging algorithm, and these values are stored in the corresponding rows of K_T and Y_T.

Algorithm 1 Construction of Knowledge matrix

Require: The graph G, the set of attackers $\mathcal{A}$, the number of iterations T

Initialization: K_T, an uninitialized matrix of dimensions $m \times n$, where $m = |\mathcal{A}| + T \cdot |\mathcal{N}(\mathcal{A})|$
for each $v \in \mathcal{A}$ **do**
  $K_T[v, :] \leftarrow e_v$, where e_v represents a one-hot vector of length n that has a value of 1 at the v-th position
end for
$i \leftarrow |\mathcal{A}|$
for t from 0 to $T - 1$ **do**
  for each $v \in \mathcal{N}(\mathcal{A})$ **do**
    $K_T[i, :] \leftarrow W^t[v, :]$
    $i \leftarrow i + 1$
  end for
end for
return K_T

Reconstructible node

In the context of a network graph G and a set of attackers $\mathcal{A}$, a node v is considered reconstructible by $\mathcal{A}$ after T iterations if, in the RREF form U of K_T, the row corresponding to v is a vector with a single 1 in one position and 0s in all other positions.

Solving the system of linear equations

Recovering private values involves solving the equation $K_T X = Y_T$, where X represents the unknown vector. Since this system is constructed such that the private values satisfy the equation, it guarantees a non-empty solution set. When K_T is full rank (i.e., rank n), the solution is unique, meaning attackers can fully reconstruct the private values of all nodes. If K_T is not full

rank, attackers can still reconstruct a subset of the private values and deduce relationships among those that cannot be fully reconstructed.

To solve this system, we decompose K_T into its Reduced Row Echelon Form (RREF), represented as $K_T = L^{-1}U$, where U is the unique RREF of K_T and L satisfies $UX = LY_T$. RREF, often taught in algebra courses for Gauss-Jordan elimination [88], transforms the block of matrix U corresponding to reconstructible nodes into the identity matrix, while the remaining rows contain linear equations linking the values of non-reconstructible nodes. Therefore, solving $K_T X = Y_T$ is equivalent to solving $UX = LY_T$, where for reconstructible nodes, trivial equations $1 \times X_v = (LY_T)_v$ hold. This decomposition clearly identifies the nodes that the attack can reconstruct even before executing the algorithm, as attackers only need to construct K_T without requiring Y_T.

3.4.4 Privacy Attacks in Decentralized Gradient Descent

In this part, we introduce our attack on Decentralized Gradient Descent (D-GD). Our approach consists of two primary steps: first, we reconstruct the gradients, and then we utilize the reconstructed gradients to recover the data points. The second step, involving the recovery of data points, can be accomplished by using existing gradient inversion attacks as a black box (see, for example, [9], [10], [76]). Therefore, our main focus in this section is on the gradient reconstruction process.

To reconstruct the gradients of nodes, we build upon the gossip averaging attack presented in Section 3.4.3. However, additional challenges arise in the case of D-GD. While private values remain constant across iterations in gossip averaging, gradients in D-GD evolve over time. As a result, each step introduces new unknowns into the equations, making it difficult to find an

exact solution through direct equation solving. Therefore, attacking D-GD requires additional steps:

- Lowering the number of unknown variables by presuming similarity in the gradients;
- Adjusting the construction method of the knowledge matrix to reconstruct the gradients g_v^t instead of the model parameters θ^t;
- Eliminating the attackers' contributions to thereby lower the total noise in the approximate reconstruction.

(i) Gradient similarity. Our reconstruction attack is based on the assumption that the gradients of a node across iterations can be expressed as a combination of a fixed component and a random component.

Assumption 3.4 (Noise-signal gradient decomposition). For each node $v \in \mathcal{V}$, we assume that the gradient update can be decomposed as follows:

$$g_v^t = -\eta \nabla_{\theta_v} L(\theta_v^t, x_v) = g_v + N_v^t, \tag{3.6}$$

where N_v^t is a zero-mean random variable with variance σ^2, and the constant term g_v is specific to node v but remains the same across iterations.

It is worth noting that this assumption is generally not satisfied in real-world scenarios, but as we will discuss in Section 6, the algorithm demonstrates robustness in practice even when this assumption is slightly violated (particularly when the gradients evolve sufficiently slowly across iterations).

Connection to differential privacy.

Assumption 3.4 naturally captures scenarios where noise is added to fulfill differential privacy requirements [79]. In fact, this can be interpreted as averaging under local differential privacy. The accuracy of the reconstruction is

directly related to the noise variance and, consequently, to the chosen privacy budget.

(ii) Knowledge matrix construction. Denoting the set of target nodes as $\mathcal{T} = \mathcal{V} \setminus \mathcal{A}$, we rewrite the gossip matrix W as follows:

$$W = \begin{pmatrix} W_{\mathcal{A},\mathcal{A}} & W_{\mathcal{A},\mathcal{T}} \\ W_{\mathcal{T},\mathcal{A}} & W_{\mathcal{T},\mathcal{T}} \end{pmatrix} \tag{3.7}$$

We now express the values of $\theta^{t+\frac{1}{2}}$ shared by nodes during the execution of the algorithm in terms of this decomposition.

Proposition 3.4 (Closed-form of D-GD updates). For the D-GD algorithm described by eq. 3.3 and eq. 3.4, we have:

$$\theta^{t+\frac{1}{2}} = \left(\sum_{i=0}^{t} W_{\mathcal{T},\mathcal{T}}^{i} \right) g_{\mathcal{T}} + \sum_{i=0}^{t} W_{\mathcal{T},\mathcal{T}}^{i} N_{\mathcal{T}}^{t-i} + \sum_{i=0}^{t-1} W_{\mathcal{T},\mathcal{T}}^{t-1-i} W_{\mathcal{T},\mathcal{A}} \theta_{\mathcal{A}}^{i+\frac{1}{2}} \tag{3.8}$$

Proof: The proof is completed by induction, applying eq. 3.3 and eq. 3.4, and rearranging the terms.

This formula leads to a more complex computation of the knowledge matrix K_T (as described in Algorithm 2) compared to the one used for gossip averaging, as we aim to reconstruct the gradients $g = (g_v)_{v \in \mathcal{V}}$ rather than the model parameters θ^t.

Algorithm 2 Building the knowledge matrix for D-GD

Require: The graph G, the set of attackers $\mathcal{A}$, the set of targets $\mathcal{T} = \mathcal{V} \setminus \mathcal{A}$, the number of iterations T

Initialization: K_T, an empty matrix of size $m \times n$ where $m = T \cdot |\mathcal{N}(\mathcal{A})|$
$i \leftarrow 0$
for t from 0 to $T-1$ **do**
  for each $v \in \mathcal{N}(\mathcal{A})$ **do**
    $K_T[i,:] \leftarrow \left(\sum_{j=0}^{t} W_T^j\right)[v - |\mathcal{A}|,:]$
    $i \leftarrow i + 1$
  end for
end for
return K_T

(iii) Attackers' contributions removal. Given Y_T, the concatenated vector of updates received by the attackers up to iteration T, the attackers need to preprocess it to remove their own contributions. Algorithm 3 shows how to compute $\hat{Y}_T$ from Y_T and the gossip matrix.

Gradient reconstruction. Using the concepts introduced earlier, reconstructing the gradients can be formulated as a Generalized Least Squares (GLS) problem:

$$K_T g + \epsilon_T = \hat{Y}_T, \tag{3.9}$$

where ϵ_T represents the noise with covariance Σ_T, a non-diagonal covariance matrix resulting from the aggregation of noise from various nodes at each step. The formula and detailed algorithm for computing this covariance matrix please refer to [89].

The gradient reconstruction that minimizes the squared error is given by:

$$\hat{g} = \left(K_T^\top \Sigma_T^{-1} K_T\right)^{-1} K_T^\top \Sigma_T^{-1} \hat{Y}_T. \tag{3.10}$$

Under Assumption 3.4, this estimator is unbiased, and its variance is $\left(K_T^\top \Sigma_T^{-1} K_T\right)^{-1}$.

Algorithm 3 Eliminating the Influence of Attackers

Require: The gossip matrix W of the graph G, the set of attackers $\mathcal{A}$, the set of targets $\mathcal{T} = \mathcal{V} \setminus \mathcal{A}$, the number of iterations T, the model dimension d, the received updates Y_T, and the concatenated vector of the updates sent by the attackers $\theta_{\mathcal{A}} = (\theta_{\mathcal{A}}^{\frac{1}{2}}, \ldots, \theta_{\mathcal{A}}^{T-\frac{1}{2}})$

1: Initialize $\hat{Y}_T \in \mathbb{R}^{T \times |\mathcal{N}(\mathcal{A})| \times d}$

2: Initialize $B \in \mathbb{R}^{|\mathcal{T}| \times d}$ with all elements set to zero

3: **for** $t = 0, 1, \ldots, T-1$ **do**

4: $\quad \hat{Y}_T[t,:] \leftarrow Y_T[t,:] - B[\mathcal{N}(\mathcal{A}),:]$

5: $\quad B \leftarrow W_{T,\mathcal{T}} B + W_{T,\mathcal{T}} \theta_{\mathcal{A}}^{t+\frac{1}{2}}$ {Subtracting the contribution of the attackers}

6: **end for**

7: **return** $\hat{Y}_T$

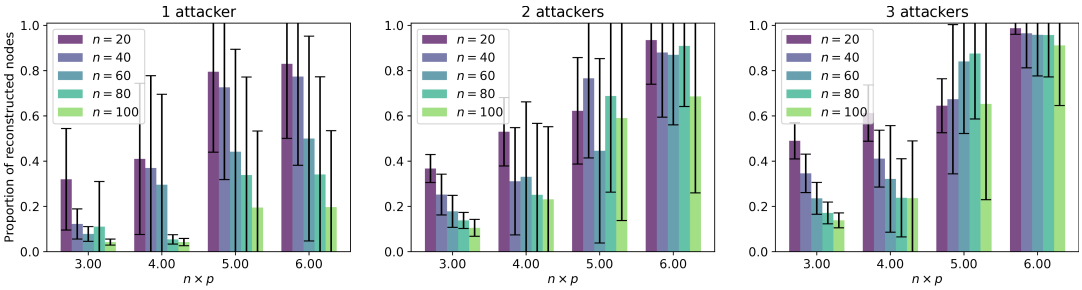

FIGURE 3.2: In Erdős-Rényi graphs with varying node counts n and edge probabilities p, the average ratio of reconstructed nodes is assessed for 1, 2, or 3 attacking nodes. Error bars provide the standard deviations computed from 20 random graph samples.

3.5 Experiments

This section presents evidence that our attacks against gossip averaging and decentralized gradient descent are practically successful. We test the proposed methods on synthetic as well as real-world graphs, and in every case,

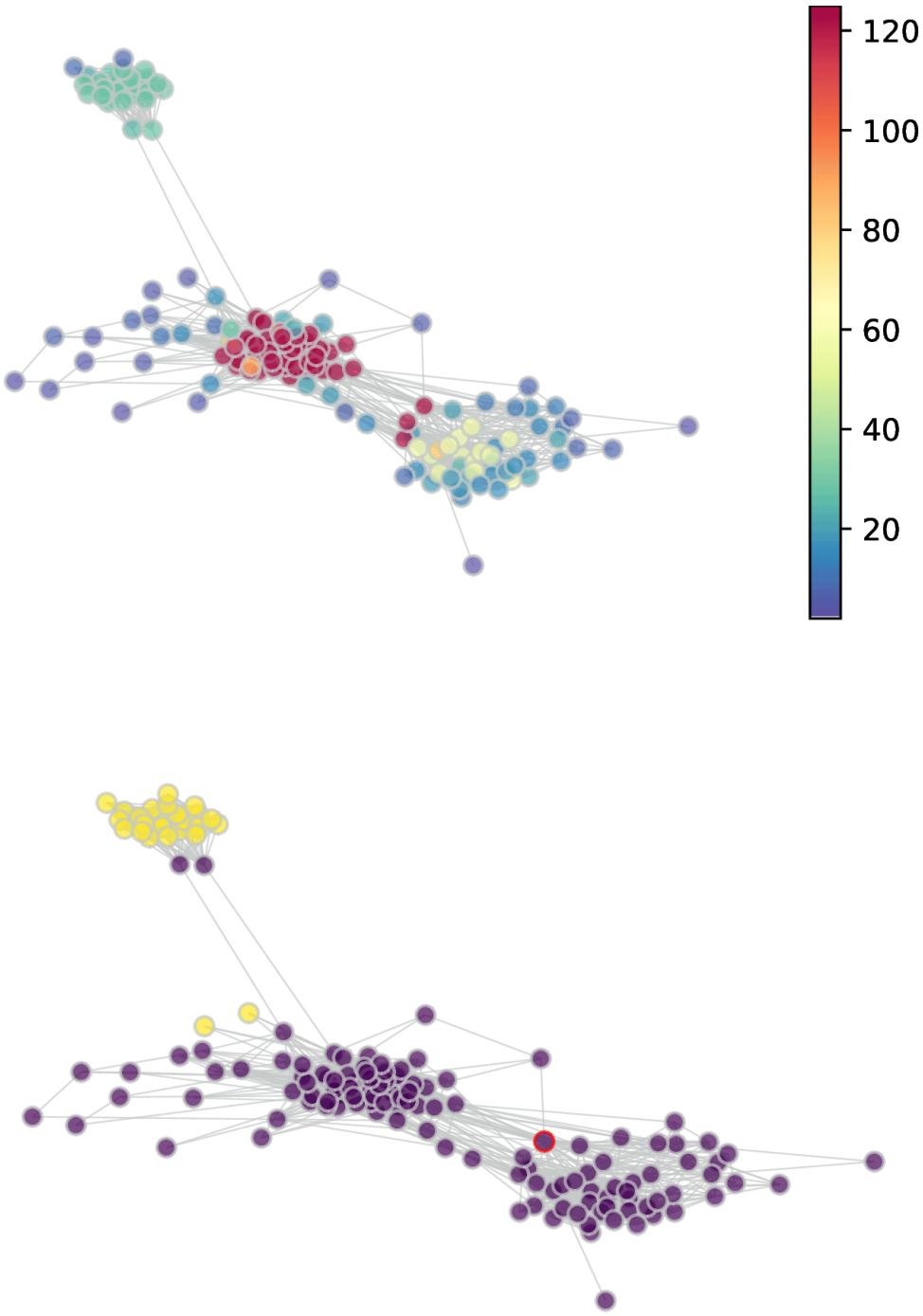

FIGURE 3.3: Reconstruction attack on the Facebook Ego Graph 414. In the top image, each node is colored based on how many of the 147 other nodes it can reconstruct. The bottom image provides a detailed view where the node highlighted in red is the attacker, with purple indicating successfully reconstructed nodes and yellow showing the non-reconstructed ones.

we achieve successful reconstructions.

FIGURE 3.4: Reconstruction attacks on several Facebook Ego graphs using gossip averaging, where attackers were chosen randomly. The node highlighted in red is the attacker, with successfully reconstructed nodes shown in purple, and non-reconstructed ones in yellow.

3.5.1 Gossip Averaging attack

Synthetic graphs. We construct Erdős-Rényi graphs with varying numbers of nodes (n) and different probabilities for edges (p). Additionally, the number of attacker nodes is adjusted between 1 and 3. Figure 3.2 illustrates the proportion of nodes reconstructed for each configuration. It is evident that even a single attacker node can typically reconstruct a substantial number of nodes, extending beyond its immediate neighborhood. The proportion of reconstructed nodes rises as both the graph's connectivity and the number of attackers increase.

Real-world graphs. We analyze graphs derived from the Facebook Ego dataset [90], where each node represents a friend of a user (the central user is excluded from the graph) and edges denote friendship connections between these nodes. Typically, these graphs form several distinct communities, each representing a specific interest group. Our findings indicate that node reconstruction is more common within individual clusters but can also occur across nodes from different clusters. Figure 3.3 presents an example, with

additional Ego graphs detailed in Figrue 3.4.

TABLE 3.1: Analysis of the correlation between attacker centrality and the proportion of nodes it is able to reconstruct.

Centrality	**Erdos-Renyi graph**	**Ego graph**
Degree	0.94	0.94
Eigenvector	0.78	0.63
Betweenness	0.81	0.65

Effect of Node Features It is intuitively easier to attack nearby nodes than distant ones. We quantify this by evaluating the impact of attacker centrality on its ability to reconstruct nodes. Centrality measures are frequently used in graph mining to assess a node's importance. We test two types of graphs. First, we randomly sample Erdős-Rényi graphs with $n = 50$ and $p = 0.08$, excluding those that are not fully connected and assuming node 0 as the attacker, since all nodes are treated equally during graph construction. Second, for a fixed Facebook Ego graph, each node is tested as the attacker. To evaluate the relationship between node centrality and the proportion of nodes it can reconstruct, we use Spearman correlation, a non-parametric measure based on rank statistics. Table 3.1 shows that for both types of graphs, degree centrality is most correlated with the proportion of reconstructed nodes. Interestingly, other centrality measures that capture structural properties beyond immediate neighbors, such as eigenvector and betweenness centrality, also exhibit strong correlations.

3.5.2 Decentralized Gradient Descent attack

We now move on to the more challenging case of D-GD. We begin by focusing on the Cifar10 dataset [91], using a model that consists of a fully connected layer, softmax activation, a bias term, and cross-entropy loss (i.e., logistic regression). For this simple model, one can reconstruct a data point from its gradient in closed form (see [92] Lemma 6.1). This allows us to focus on

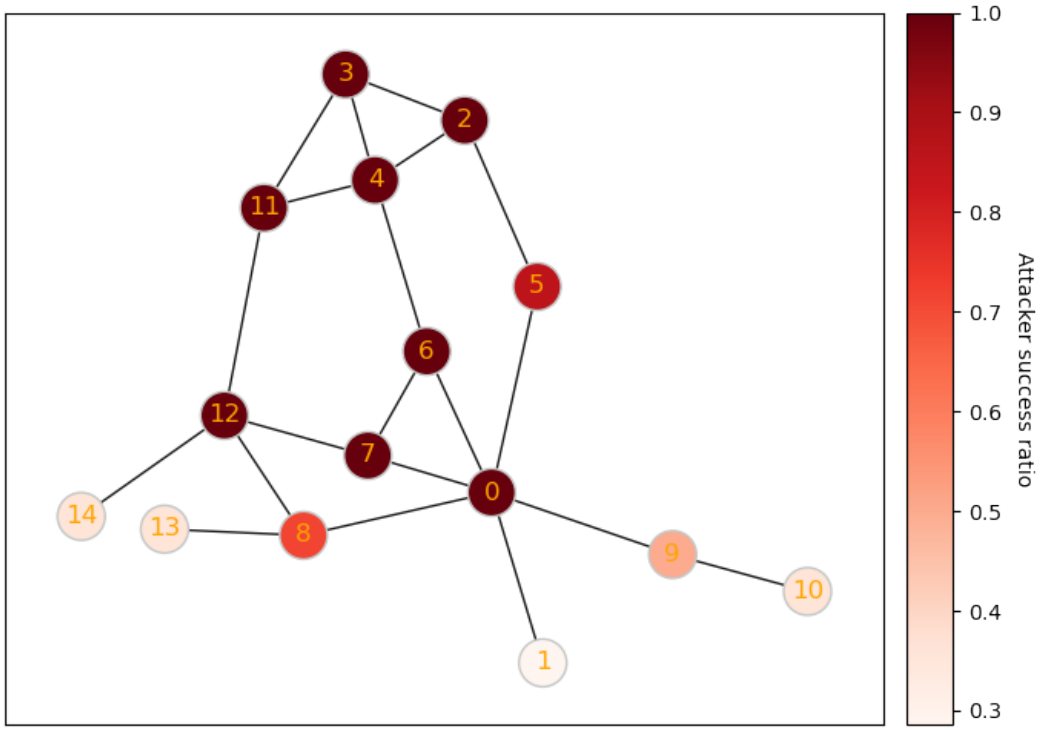

FIGURE 3.5: Attacks using D-GD to reconstruct the Florentine graph (Cifar10, logistic regression model, learning rate 10^{-5}). Each node's color represents the success rate when that node is chosen as the attacker. The success rate is defined by the proportion of nodes where the reconstructed image achieves a PSNR greater than 10, based on an average of 10 trials.

the core of our attack (reconstructing gradients), avoiding the inherent errors caused by gradient inversion attacks in more complex models.

We start the attack when the model is close to convergence to ensure stable gradients. To make sure attackers gather sufficient information about other nodes in the knowledge matrix, we run D-GD for a number of steps roughly equal to the diameter of the graph. First, we apply our attack to the classic Florentine graph [93], which consists of $n = 15$ nodes, describing marital relations between families in 15th-century Florence. As shown in Figure 3.5 and Figure 3.6, most nodes (except those at the network's edge) can reconstruct a large portion of other nodes with high visual accuracy.

We test the upper limit of reconstruction by employing a line graph of 30 nodes, with the attacker positioned at one extremity. As shown in Figure

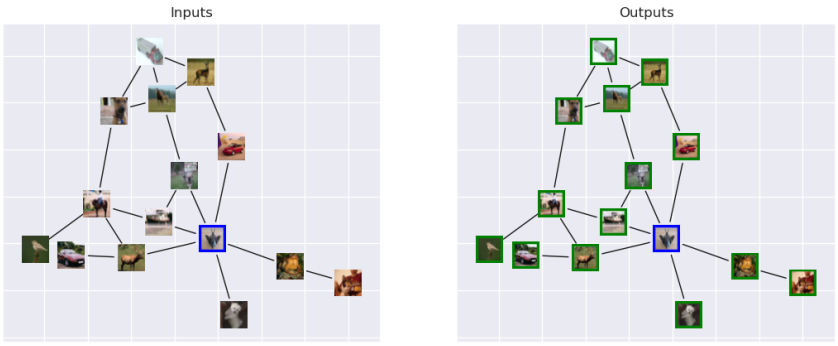

FIGURE 3.6: This figure illustrates the D-GD reconstruction attack on the Florentine graph, where the true image inputs are on the left and their reconstructions are on the right. The attacker node is marked with blue borders. Nodes enclosed in green are successfully reconstructed, while those enclosed in red are not.

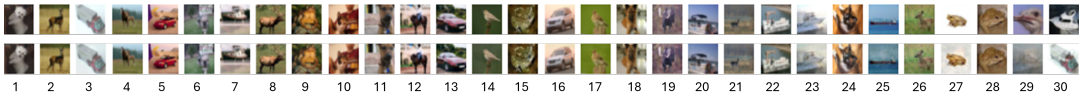

FIGURE 3.7: D-GD reconstruction attack on a line graph consisting of 30 nodes, where the attacker is positioned at one end of the graph. The top row displays the actual inputs for the rest nodes, arranged by their proximity to the attacker, the second row display the output of the attack.

3.7, the results surpass initial expectations: although the gradients from distant nodes undergo several combinations before reaching the attacker, our method can still disentangle the contributions of different nodes, enabling meaningful reconstruction up to a distance of 28. This observation is further reinforced by reconstruction metrics across multiple runs (see Figure 3.9).

Subsequently, we apply a more complex model, requiring the use of gradient inversion attacks. On the MNIST dataset, we use a small convolutional neural network and the black-box gradient inversion attack from Geiping [9]. Using the same 31-node line graph as in the previous experiment, we observe in Figure 3.8 that our method can naturally rely on black-box gradient inversion to reconstruct data from more complex models. In this case, reconstructions are accurate are working well.

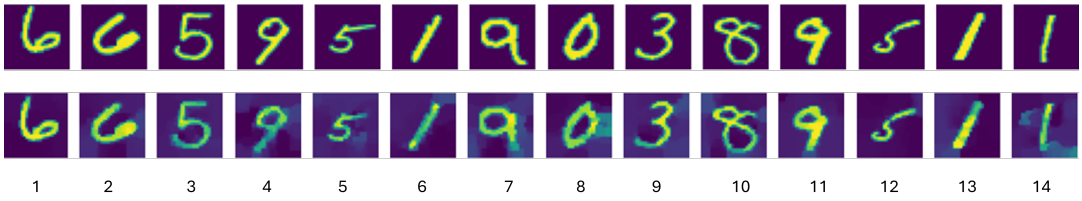

FIGURE 3.8: D-GD reconstruction attack on a line graph consisting of 14 nodes of mnist data, where the attacker is positioned at one end of the graph. The top row displays the actual inputs for the rest nodes, the second row display the output of the attack. The results show that the attack reconstruction works well.

We also note that the performance of our D-GD attack is highly sensitive to several parameters. First, having similar local parameters θ_v across nodes enables more accurate reconstructions since the observed values are influenced primarily by gradients rather than by parameter variations. This condition is easily satisfied by initializing all nodes with identical parameters (a standard practice in D-GD) or waiting until the system approaches convergence. Second, the learning rate plays a critical role: it must be small enough to prevent gradients from fluctuating significantly across iterations.

3.6 Summary

In this study, we found that there is a certain vulnerability in the data when using decentralized learning algorithms. Specifically, we demonstrated how a node can successfully attack and reconstruct the data of a distant node by leveraging the communication structure of the gossip protocol. We also noted that the topology of the graph and the position of the attacker have a significant impact on the success rate of the attack. We recognize that relying solely on decentralized mechanisms is insufficient to adequately protect sensitive data. Therefore, to enhance privacy protection, we recommend combining decentralized algorithms with additional defense mechanisms such

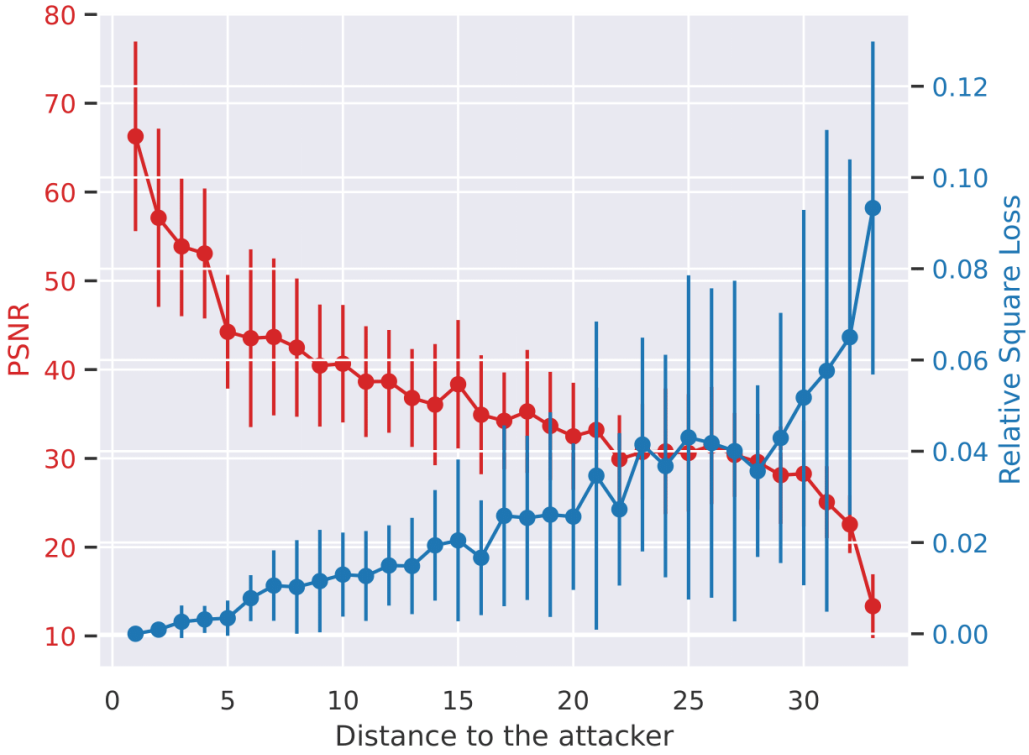

FIGURE 3.9: The plot shows the reconstruction accuracy by using PSNR (Peak Signal-to-Noise Ratio) and error (relative square distance) as a function of the distance between the victim and the attacker in a D-GD line graph experiment (details in Figure 3.7). It includes the mean and standard deviation computed over 100 experimental runs.

as differential privacy. In the following chapters, we will explore the relationship between differential privacy guarantees and the success rate of reconstruction attacks through formal experiments and analyses.

Chapter 4

Task-adaptive Privacy Preservation for Multi-dimensional Data

As the data used in machine learning tasks increases, user privacy becomes increasingly important. Differential privacy (DP) [94] is a leading technique for data protection, and its local variant local differential privacy (LDP) offers stronger privacy guarantees without needing trusted third parties. Companies like Google, Apple, and Microsoft have successfully deployed LDP in tasks such as basic frequency or histogram estimation, where data is limited to discrete variables. LDP shows potential for more complex applications, such as in healthcare, power grids, and the Internet of Things (IoT), where richer data attributes are involved in sophisticated ML tasks. However, current LDP methods may not perform well in these scenarios due to the challenges of adding noise to multidimensional, variable data, which can degrade task performance. To solve this issue, in this chapter we proposes a task-aware LDP approach that adjusts noise based on the relevance of data attributes to the task, leading to improved accuracy while maintaining privacy under a fixed budget.

4.1 Local differential privacy

Local Differential Privacy (LDP) is a variant of DP designed to protect user privacy without the need for a trusted third party. In LDP, users perturb their data locally before sending it to the data collector, ensuring that the data the collector receives cannot be used to directly infer the original input [37].

For any two possible inputs x and x', and any output set $S \subseteq \mathcal{Y}$, a randomized algorithm $\mathcal{M}$ satisfies ϵ-LDP if, for all inputs x and x', the following inequality holds:

$$\Pr[\mathcal{M}(x) \in S] \leq e^{\epsilon} \Pr[\mathcal{M}(x') \in S] \tag{4.1}$$

Where:

- $\mathcal{M}$ is a randomized algorithm, or the local perturbation mechanism.
- x and x' are two different inputs from the user.
- ϵ is the privacy budget, which controls the strength of the privacy protection. Smaller ϵ values provide stronger privacy.
- $S \subseteq \mathcal{Y}$ is a possible output set.

LDP ensures that, for any two possible inputs x and x', the outputs generated by the perturbation mechanism $\mathcal{M}$ have similar probability distributions. In other words, the data collector cannot distinguish the user's original input by observing the perturbed outputs. This mechanism guarantees privacy for each user, as even if the data collector sees the perturbed data, they cannot easily infer the true data.

The main advantage of LDP in practice is that it does not rely on a trusted third party, as each user is responsible for perturbing their own data independently.

4.2 Noise mechanism

Noise Mechanism is a core tool in the implementation of differential privacy, designed to ensure privacy protection by adding random noise to the data. In the framework of differential privacy, the role of noise mechanisms is to obscure the processed data so that even if an adversary obtains the perturbed data, they cannot accurately infer the user's original data, thus safeguarding individual privacy. The underlying idea of noise mechanisms is to introduce an appropriate level of randomness to balance data privacy and utility. The main goal is to perturb the data in such a way that an observer cannot identify specific information about an individual in the dataset, while still preserving the data's statistical value for analysis and learning. Noise mechanisms control the amount of noise based on the privacy budget ϵ; a smaller privacy budget means stronger privacy protection and larger noise, which makes the data more obscured, while a larger privacy budget leads to less noise and data closer to its true value. The basic principle behind noise mechanisms relies on the concept of sensitivity, which is defined as the maximum change in the output caused by altering a single data point in the dataset. The noise mechanism generates noise based on the function's sensitivity and the privacy budget ϵ, ensuring that even if a single data point changes, it does not significantly affect the output's probability distribution. In the following subsections, this thesis introduces several common noise mechanisms and the basic ideas behind them.

4.2.1 Laplace Mechanism

In order to release a sensitive function $g : \mathcal{X} \mapsto \mathbb{R}^Z$ while ensuring ϵ-LDP for $\epsilon > 0$, the Laplace mechanism is a commonly employed technique. It operates by introducing Laplace noise to the function g:

$$\mathcal{M}_{\text{Lap}}(x, g, \epsilon) = g(x) + \text{Lap}^Z(\mu = 0, b = \frac{\Delta_1 g}{\epsilon}), \tag{4.2}$$

where $\text{Lap}^Z(\mu, b)$ represents a vector of dimensionality Z, whose components are independent and identically distributed (i.i.d.) Laplace random variables, each with mean μ and scale parameter b, resulting in a variance of $2b^2$. Furthermore,

$$\Delta_1 g = \max_{x, x' \in \mathcal{X}} \|g(x) - g(x')\|_1 \tag{4.3}$$

captures the sensitivity of the function g under the ℓ_1 norm.

4.2.2 Gaussian Mechanism

The gaussian mechanism is similar to the Laplace mechanism but adds noise drawn from a Gaussian distribution (i.e., normal distribution). In some cases, the Gaussian mechanism provides better robustness and flexibility than the Laplace mechanism, especially when the input data has high uncertainty.

Given a function f and privacy budget ϵ, the Gaussian mechanism perturbs the output as follows:

$$\mathcal{M}(x) = f(x) + \mathcal{N}(0, \sigma^2) \tag{4.4}$$

Where:

- σ is the standard deviation, typically related to the privacy budget ϵ and sensitivity Δf, calculated as:

$$\sigma \geq \frac{\Delta f \sqrt{2 \ln(1.25/\delta)}}{\epsilon}$$

Here, δ is the probability of privacy leakage.

The Gaussian mechanism is commonly used to protect numerical data, particularly in the (ϵ, δ)-differential privacy framework, as it allows more flexibility in balancing privacy and utility. It is widely applied in statistical aggregation and gradient perturbation during machine learning model training.

4.2.3 Exponential Mechanism

The exponential Mechanism is designed for non-numerical data and is particularly useful when the output is categorical or comes from a finite set of choices. Its core idea is to select an output weighted by a utility function, ensuring privacy while enhancing task accuracy.

Given a utility function $q(x, r)$ (which measures the quality of output r for input x) and privacy budget ϵ, the exponential mechanism selects an output r with the following probability:

$$\Pr[\mathcal{M}(x) = r] \propto \exp\left(\frac{\epsilon q(x, r)}{2\Delta q}\right) \tag{4.5}$$

Where:

- Δq is the sensitivity of the utility function, representing the maximum possible change in $q(x, r)$ when a single entry in the input dataset is modified.

The exponential mechanism is widely used in ranking, classification, and recommendation systems. For example, in recommendation systems, it can protect user privacy while recommending the most suitable items based on user behavior.

4.2.4 Randomized Response Mechanism

The randomized response mechanism is one of the earliest differential privacy mechanisms, specifically designed for protecting the privacy of categorical data. It was originally developed for social science surveys to protect user privacy and reduce reporting bias. The mechanism provides users with two options: give a truthful answer or a random one, making it impossible for data collectors to determine the user's true intent.

Suppose a user's original data is $x \in \{0, 1\}$, the randomized response mechanism processes the data as follows:

$$\Pr[\mathcal{M}(x) = x] = \frac{e^{\epsilon}}{e^{\epsilon} + 1}, \quad \Pr[\mathcal{M}(x) \neq x] = \frac{1}{e^{\epsilon} + 1} \tag{4.6}$$

Where:

- ϵ is the privacy budget, controlling the probability of reporting the true value.

The randomized response mechanism is primarily used for protecting categorical or discrete data, particularly in surveys and questionnaires. It effectively reduces the risk of user privacy leakage while maintaining the statistical accuracy of the overall data.

Different noise mechanisms in the implementation of differential privacy have their unique application scenarios and advantages. The Laplace Mechanism and Gaussian Mechanism are primarily used for the protection of numerical data, operating under ϵ-differential privacy and (ϵ, δ)-differential privacy frameworks, respectively. The Exponential Mechanism is suitable for handling non-numerical data, especially performing well in classification tasks and recommendation systems. Meanwhile, the Randomized Response Mechanism is particularly effective in handling discrete data, social surveys,

and questionnaires. These mechanisms strike a balance between privacy protection and data utility, ensuring that the data retains its statistical and machine learning value while providing privacy guarantees, for more details please refer to the literature [37].

4.2.5 Explanatory Examples

Differential Privacy: Querying Average

Suppose you have a dataset containing the ages of individuals and you want to compute the average age while ensuring individual privacy. You can apply differential privacy by adding noise to the true average.

One commonly used noise mechanism in differential privacy is the Laplace mechanism, and its formula is:

$$\tilde{f}(D) = f(D) + \text{Lap}\left(\frac{\Delta f}{\epsilon}\right)$$

where:

- $f(D)$ is the query result on the dataset D (such as the average age).
- Δf is the sensitivity, which represents the maximum impact a single data point can have on the query result. For an average query, Δf is typically $\frac{\text{range}}{n}$, which is the range of the data divided by the number of data points.
- ϵ is the privacy parameter.
- $\text{Lap}(b)$ is the Laplace distribution with parameter b, generating symmetric noise around 0.

For example, if the range of ages in a dataset is 0 to 100, and there are 100 data points, the sensitivity $\Delta f = \frac{100}{100} = 1$. If $\epsilon = 1$, noise drawn from

the Laplace distribution $\mathrm{Lap}(1)$ would be added to the computed average, ensuring differential privacy.

Local Differential Privacy: Randomized Response Mechanism

Suppose each user is asked to answer a sensitive question, such as Do you smoke? To protect privacy, the randomized response mechanism is applied:

- The user answers truthfully with probability p, and randomly generates an answer with probability $1 - p$.

This mechanism satisfies ϵ-differential privacy. The mathematical formula is:

$$\Pr[\text{truthful answer}] = p, \quad \Pr[\text{random answer}] = 1 - p$$

For binary questions (such as Yes/No), let $p = \frac{e^{\epsilon}}{1+e^{\epsilon}}$, and this mechanism ensures privacy under the definition of ϵ-differential privacy.

For example, if $\epsilon = 1$, then $p = \frac{e^1}{1+e^1} \approx 0.731$, meaning the user will provide the truthful answer with about 73% probability and a random answer with 27% probability. This way, even if the collector sees the user's response, they cannot be sure of its authenticity.

4.3 Problem Setup

In this section, we present the proposed task-adaptive privacy preservation issue, as shown in Figure 4.1. Let $y = f(x) \in \mathbb{R}^n$ represent the task output for each original data sample x, where f stands for the task function. To ensure ϵ-LDP for every data point x, its true value must remain hidden from the task function. Instead, a perturbed version of x, denoted as $\hat{x}$, is utilized as the input to the task function, leading to a task output $\hat{y} = f(\hat{x})$. The main

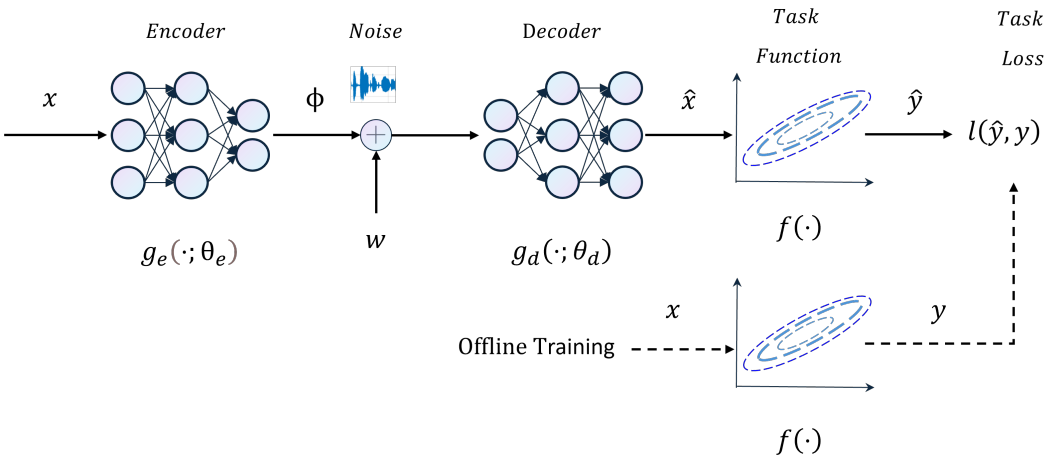

FIGURE 4.1: The overall architecture for addressing the task-adaptive privacy preservation problem.

objective is to minimize the overall task loss $L = \mathbb{E}[l(\hat{y}, y)]$, which arises due to the difference between the perturbed input $\hat{x}$ and the actual data x, with x being sampled from distribution $\mathcal{D}_x$. The task loss function l is responsible for measuring the discrepancy between the task output $\hat{y}$ and the true output y, and can take common forms such as ℓ_2-norm or cross-entropy loss. Importantly, privacy is not directly addressed during the offline training phase, where the true data x is used to optimize certain parameters C. However, once the model is trained and deployed online, LDP for user data is ensured.

In more detail, x is first transformed into a latent variable $\phi \in \mathbb{R}^Z$ via an encoder function $\phi = g_e(x; \theta_e)$, where θ_e is the set of parameters for the encoder. The latent representation ϕ is then subjected to Laplace noise, denoted by a noise vector $w \in \mathbb{R}^Z$. In this case, ϕ is treated as the sensitive output of the function g, in accordance with Equation (2). After the addition of noise, x is reconstructed from $\phi + w$ through a decoder function $\hat{x} = g_d(\phi + w; \theta_d)$, where θ_d represents the parameters of the decoder. In practical settings, the decoder operates on the user's side and is typically designed to be lightweight (e.g., using a linear function or a simple one-layer neural network).

The optimal task-adaptive input $\hat{x}$ minimizes the task loss L while satisfying ϵ-LDP. That is, the task-adaptive privacy problem requires the joint design of both the encoder and decoder, i.e., finding appropriate values for Z, θ_e, θ_d, in a way that minimizes L while preserving ϵ-LDP. Formally, this can be expressed as:

$$\begin{aligned} &\min_{Z,\theta_e,\theta_d} L = \mathbb{E}_{x,w}[l(\hat{y}, y)], \\ &\text{s.t.} \quad y = f(x), \\ &\qquad \hat{y} = f(g_d(g_e(x;\theta_e) + w;\theta_d)), \\ &\qquad x \sim \mathcal{D}_x, \quad w \sim \text{Lap}^Z(0, \frac{\Delta 1 g_e}{\epsilon}). \end{aligned} \tag{4.7}$$

The challenge with this task-adaptive LDP problem stems from the conflict between the measurement of the overall task loss L, which considers the average performance based on the distribution $\mathcal{D}_x$, and the preservation of LDP, which is focused on the worst-case privacy guarantee based on the input data X.

Benchmark Approaches. Here, we introduce two common strategies to achieve ϵ-LDP protection. The first strategy, called a task-unaware approach, involves adding noise directly to the normalized x[1]. For simplicity, we assume that x is already normalized, so that $Z = n$ and $g_e(x) = x$. The second method, known as the privacy-unaware approach, applies noise to the latent variable ϕ, which is derived from solving the problem defined in Equations 4.7. In this case, $Z \leq n$ is predetermined, and w is set to be a zero vector. This implies that privacy considerations are disregarded during the encoder design, and as a result, an appropriate choice for Z must be made in advance. Otherwise, one might always conclude that a larger Z (allowing more information transfer in the absence of noise) is preferable. Both benchmark methods still require the optimal decoder parameters θ_d to be found for input $\phi + w$.

4.4 Method Description

In scenarios where the problem becomes more intricate, deriving an analytical solution for the task-aware privacy preservation issue is particularly challenging, especially when the encoder function g_e, decoder function g_d, and task function f are modeled using neural networks. To tackle this, we propose a gradient-based learning approach.

Algorithm General Settings for Task-adaptive Algorithm for ϵ-LDP Preservation

Require: Privacy budget ϵ and Z

1: Set up initial values for encoder/decoder parameters θ_e, θ_d and noise vector w

2: **for** $\tau \in \{0, 1, \ldots, N_{\text{epochs}} - 1\}$ **do**

3: Update θ_e and θ_d with $-(\nabla_{\theta_e}\mathcal{L} + 2\eta\theta_e)$ and $-\nabla_{\theta_d}\mathcal{L}$, respectively, by one or multiple steps

4: Recalculate Δ_{1g_e} and draw a new sample of w from the distribution $\text{Lap}^Z(0, \Delta_{1g_e}/\epsilon)$

5: **end for**

6: **return** θ_e, θ_d and Δ_{1g_e}

In general, Z should be chosen carefully: it should not be too small (as larger values of Z often yield better solutions), nor too large (since this can introduce unnecessary complexity). In practice, the appropriate choice of Z is determined on a case-by-case basis.

When updating the encoder parameter θ_e, we include an ℓ_2 regularization term $\eta||\theta_e||_F^2$, where η is a positive constant. This regularization helps prevent the norm $||\theta_e||_F^2$ from growing indefinitely, which could otherwise lead to increasing the scale of ϕ and reducing the task loss $\mathcal{L}$. However, this direction is undesirable, as it would proportionally increase the noise variance σ_w^2, compromising ϵ-LDP guarantees.

Additionally, the time complexity of computing $\Delta_1 g_e$ is quadratic in the number of data samples, which can become computationally expensive for large datasets. In such cases, dividing the data into mini-batches or utilizing parallel processing can help to reduce the overall computation time.

4.5 Experiments

Our evaluation examines the performance of the proposed task-adaptive approach in comparison to the benchmark methods, which is task-unaware and privacy-unaware methods. The task-unaware method adds noise uniformly to all data attributes without considering their relevance to specific tasks and the privacy-unaware method completely ignores privacy protection when optimizing task performance, adding no noise at all. We utilize three applications and their corresponding datasets from the UCI Machine Learning Repository: hourly household power consumption estimation, real estate price prediction, and breast cancer diagnosis [95].

The experiments were conducted on a personal laptop equipped with an Intel(R) Core(TM) i7-10750H processor, an NVIDIA GeForce RTX 3070 Laptop GPU, and 32 GB of memory. Our implementation utilizes Pytorch as the underlying framework, with the Adam optimizer applied across all applications, and a learning rate set to 10^{-3}. All datasets employed in the evaluation are publicly available through the UCI Machine Learning Repository [95] and have been anonymized using standard protocols.

For the task function f, we employed a one-hidden-layer feedforward neural network with an input size of n, a hidden layer size of $1.5n$, and an output size of 1 in both the real estate valuation and breast cancer detection tasks. The hidden and output layers used Rectified Linear Unit (ReLU) activation. Our experiments demonstrated that this network architecture performed well, achieving near-zero loss when compared to the ground truth x

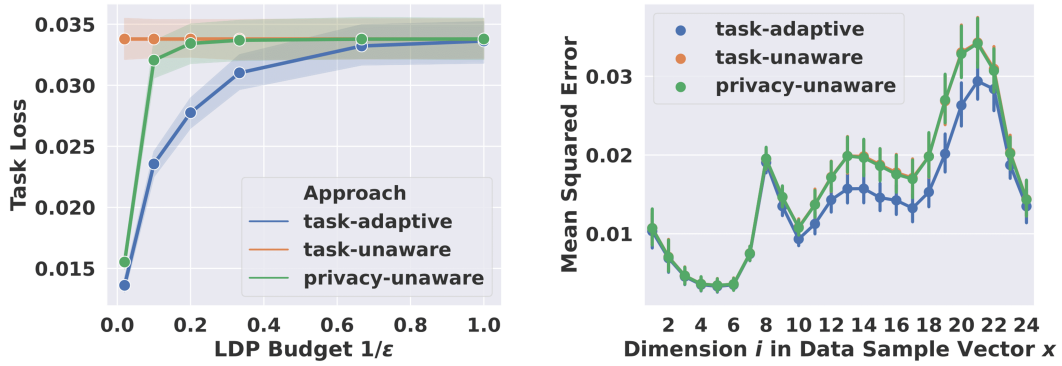

FIGURE 4.2: Comparison of Task-Adaptive, Task-Unaware, and Privacy-Unaware Methods in Estimating Hourly Household Power Consumption under LDP Constraints.

and y. To avoid overfitting, we opted not to use a deep neural network, as adding more layers (e.g., two layers) did not improve task performance.

For the gradient-based learning algorithm described in the main context, we set $\eta = 0.2$ for the real estate valuation task and $\eta = 0.001$ for the breast cancer detection task. In both tasks, we performed 15 updates to θ_e and θ_d per epoch.

4.5.1 Estimation of Hourly Average Household Power Usage

We begin by considering a mean estimation task, which is based on the measurement of individual household electricity usage over a period of four years [96]. Each data sample $x \in \mathbb{R}^{24}$ represents a time series that contains the hourly power consumption of a household for a single day. Our aim is to estimate the average hourly power consumption over N days. As discussed in Section 4.1, the overall task loss can be defined as follows:

$$\mathcal{L} = \mathbb{E}_{x\sim\mathcal{D}_x}\left[||K(\hat{x}-x)||_2^2\right] = \sum_{i=1}^{24} k_i^2 \mathbb{E}_{x\sim\mathcal{D}_x}\left[(\hat{x}_i - x_i)^2\right] \tag{4.8}$$

In this equation, $K = \text{diag}(k_1, k_2, \cdots, k_{24})$ reflects the significance of each hour's contribution to the mean estimation. In our experiments, we set $k_i = 2$ for the hours $i \in \{9, 10, \cdots, 20\}$ (representing daytime hours), while $k_i = 1$

for the remaining hours (nighttime). For this task, we employ a linear encoder and decoder model. Since the problem is framed using a linear model with MSE as the task loss, we apply the solutions presented in Section 4.1 for the three different approaches (with $Z = 3$ chosen for the privacy-unaware approach).

Our experimental results are shown in the Figure 4.2. First, the left graph compares the task loss for the three approaches under different LDP budgets. As observed, with the increase of the LDP budget $\frac{1}{\epsilon}$, the task loss for all methods decreases, but the Task-adaptive method consistently outperforms the other two approaches. Specifically, when $\frac{1}{\epsilon} = 0$, the Task-adaptive method achieves a task loss of around 0.005, while the Task-unaware and Privacy-unaware methods exhibit task losses of approximately 0.015 and 0.035, respectively. As the LDP budget increases, the performance gap among the three approaches narrows, particularly when $\frac{1}{\epsilon} \geq 0.6$, where the task losses for all methods stabilize. This suggests that under higher privacy budgets, the Task-adaptive method can effectively reduce the task loss while maintaining privacy protection.

The right graph presents the mean squared error (MSE) across different hourly dimensions. It is evident that the Task-adaptive method achieves significantly lower MSE during the daytime hours (i.e., dimensions 9 to 20) compared to the other methods, particularly between dimensions 10 and 14, where MSE is reduced by approximately 20%. This demonstrates the superiority of the Task-adaptive approach during periods of high power consumption. In contrast, during the nighttime hours (e.g., dimensions 2 to 6 and 21 to 24), the MSEs of all three methods are relatively similar, ranging between 0.005 and 0.025. This result can be attributed to the Task-adaptive method's ability to optimize the importance of different dimensions, particularly for the task-relevant periods of higher power consumption during

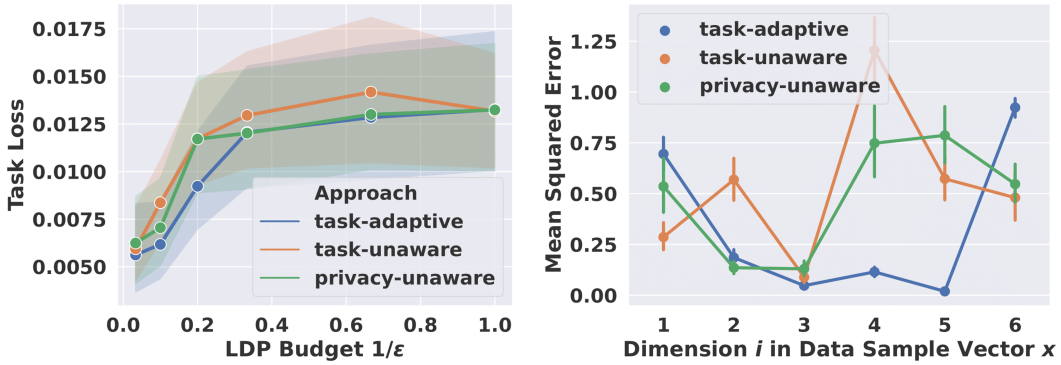

FIGURE 4.3: Comparison of Task Loss and Dimensional MSE for Task-Adaptive, Privacy-unaware and task-unaware Methods Under LDP in Real Estate Valuation.

the day, whereas the other methods are less flexible in addressing such task-specific variations. Overall, the Task-adaptive method significantly reduces estimation errors during critical periods while ensuring privacy.

4.5.2 Real Estate Valuation

Next, we address two specific problems: real estate valuation and breast cancer detection. Since neither of these problems relies on a linear model with MSE task loss, we apply the proposed gradient-based learning approach, as detailed mentioned above, to tackle both tasks. To ensure a fair comparison, we set $Z = 3$ for both our task-adaptive method and the privacy-unaware approach.

Real Estate Valuation. In this case, we utilize historical real estate valuation data from Taiwan [97], which comprises more than 400 samples. Each data point $x \in \mathbb{R}^6$ consists of six key attributes that significantly influence the value of a property, such as transaction date, house age, and geographic location, among others. The target variable $y \in \mathbb{R}$ corresponds to the property's valuation. Initially, we train a one-hidden-layer feedforward neural network regression model, leveraging the true values of x and y, which serves as our

task function f. We then optimize the ℓ_2 loss between $\hat{y}$ and y, using a linear encoder-decoder architecture.

This experimental data shows the task loss under different LDP budgets and the mean squared error (MSE) across six dimensions for the three methods. From the Figure 4.3 left side, the Task-adaptive method significantly outperforms the Task-unaware and Privacy-unaware methods at lower LDP budgets (where privacy protection is stronger), showing the lowest task loss and more stable performance. As the LDP budget increases, the task losses for the three methods gradually converge, but the Task-adaptive method maintains a certain advantage, especially at lower budgets where its performance is notably superior. In contrast, the Task-unaware method consistently has the highest task loss across all budgets, reflecting poorer adaptability, while the Privacy-unaware method performs similarly to the Task-unaware method at lower budgets but shows slightly better performance at higher budgets.

In the Figure 4.3 right side, the Task-adaptive method exhibits lower MSE across most dimensions, particularly in dimensions 2 and 3, where it significantly outperforms the Task-unaware and Privacy-unaware methods. This indicates that the Task-adaptive method is better at adapting to different dimensions and effectively reducing error. The Task-unaware method shows greater fluctuations in some dimensions, particularly in dimensions 3 and 5, where its MSE increases substantially, indicating instability in handling certain dimensions. The Privacy-unaware method performs more consistently in some dimensions but overall is still less accurate than the Task-adaptive method. Thus, the Task-adaptive method demonstrates stronger robustness and accuracy in handling multi-dimensional tasks. Mean square error Loss Evolution for Task-Adaptive Method Under Different Privacy Budgets is as shown in Figure 4.4. This chart illustrates the training loss (MSE) of the Task-Adaptive method under varying privacy budgets (ϵ) across different epochs.

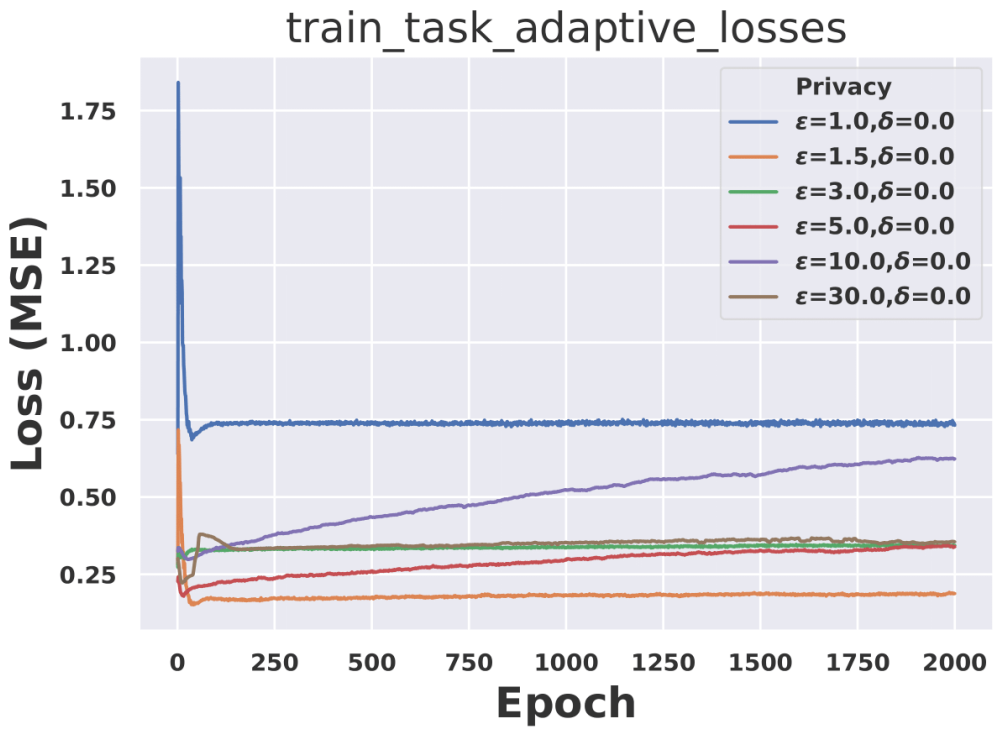

FIGURE 4.4: MSE Loss Evolution for Task-Adaptive Method Under Different Privacy Budgets.

The privacy budgets range from $\epsilon = 1.0$ to $\epsilon = 30.0$, representing different levels of privacy protection. In the early stages of training (roughly the first 250 epochs), all curves show a rapid decrease in MSE, indicating that the model converges quickly during this phase. However, as the training progresses, the long-term performance varies significantly depending on the privacy budget. For smaller privacy budgets (e.g., $\epsilon = 1.0$ and $\epsilon = 1.5$), the MSE remains relatively high, and the curves exhibit more fluctuations, reflecting the trade-off between stronger privacy protection and reduced model accuracy. For medium privacy budgets (e.g., $\epsilon = 3.0$ and $\epsilon = 5.0$), the MSE decreases further, and the model demonstrates greater stability and lower loss over time. With larger privacy budgets (e.g., $\epsilon = 10.0$ and $\epsilon = 30.0$), the MSE is the lowest, and the model converges earlier, indicating that with weaker privacy constraints, the model can more accurately fit the data.

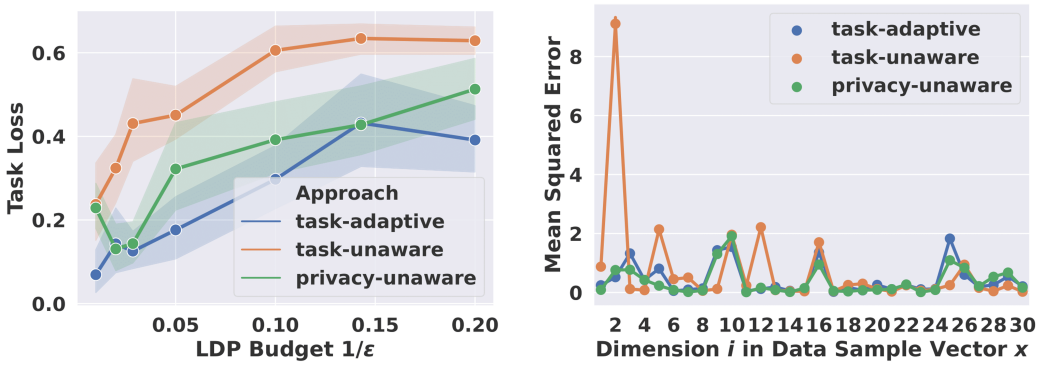

FIGURE 4.5: Task Loss and MSE Comparison Across LDP Budgets for Breast Cancer Detection.

4.5.3 Breast Cancer Detection

Breast Cancer Detection. In this study, we employ a widely recognized breast cancer diagnostic dataset [98] from Wisconsin, consisting of over 500 samples. Each sample $x \in \mathbb{R}^{30}$ includes 30 features that represent 10 characteristics of a cell nucleus. The target variable y is a binary label indicating whether the diagnosis is malignant or benign. We begin by training a one-hidden-layer feedforward neural network classification model using the true values of x and y, which serves as our task function f. Our objective is to minimize the cross-entropy loss between $\hat{y}$ and y, with both the encoder and decoder structured as one-hidden-layer feedforward neural networks.

This set of experimental results demonstrates the performance of three methods in the breast cancer detection task under different LDP budgets. The Figrue 4.5 left part shows that the task-adaptive method consistently achieves the lowest task loss across all privacy budgets, particularly in low-budget (high privacy protection) scenarios, where its loss is significantly lower than the other two methods. Although the task loss increases as the privacy budget grows, the increase is relatively small, indicating that this method strikes a good balance between privacy protection and task accuracy. In contrast, the Task-unaware method exhibits a consistently higher task loss across all budgets, with a significant increase in loss at higher privacy budgets,

showing its inability to maintain high accuracy under privacy constraints. The Privacy-unaware method performs slightly better at low budgets, but its loss gradually increases as the budget grows.

The Figure 4.5 right part presents the mean squared error (MSE) across different data dimensions. The task-adaptive method demonstrates stability across most dimensions, maintaining a low MSE, particularly in key dimensions like dimension 2 and 3, where its error is significantly lower than the other methods. In contrast, the Task-unaware method shows large fluctuations in some dimensions, especially in dimension 2, where there is a sharp increase in MSE, indicating instability. The Privacy-unaware method shows relatively stable performance across most dimensions, with smaller fluctuations in MSE, but the error tends to increase in higher dimensions. Overall, the Task-adaptive method exhibits outstanding accuracy and stability in the breast cancer detection task, especially under higher privacy protection requirements, clearly outperforming the other methods.

4.6 Summary

This chapter introduced a task-adaptive privacy preservation method designed to improve the balance between privacy and utility, particularly for machine learning tasks that rely on rich, multi-dimensional user data. We presented an solution for a linear encoder-decoder model with mean squared error (MSE) task loss and develop a gradient-based learning algorithm to address more complex nonlinear scenarios. Our evaluation demonstrates that this task-aware approach outperforms benchmark methods in terms of overall task loss under various local differential privacy (LDP) budgets. Moreover, this chapter outlines several potential directions for future work. First, extending the analysis of the task-adaptive privacy preservation problem to

other LDP mechanisms, including approximate LDP, is a worthwhile endeavor. Additionally, exploring task-adaptive privacy preservation for different user groups in a distributed environment presents another avenue for further research. Lastly, we plan to investigate task-aware anonymized representations for multi-task learning.

Chapter 5

Uncertainty Estimation in Deep Learning

Preserving data privacy adds uncertainty to machine learning. Uncertainty estimation in deep learning [1] is a crucial research area that aims to quantify the uncertainty in model predictions while protecting data privacy. By applying normalizing flow methods, it is possible to more accurately estimate the predictive distribution of models and ensure sensitive data is not compromised, thereby providing more reliable and secure uncertainty measures. It is worth noting that, the results presented in this chapter have been published in [99].

5.1 Basis of Normalizing Flow

Normalizing Flows gained prominence through the work of Rezende and Mohamed [100] in variational inference, and through Dinh et al. [100] in the field of density estimation. The key idea behind normalizing flow model is build complex distribution from simple distribution via a flow of successive (invertible) transformations. By utilizing change of variable technique, from simple distribution transform to a better expressive distribution (equal probability).

5.1.1 Change of Variables Formula

In the context of normalizing flows, the objective is to transform simple distributions, which are easy to sample and evaluate, into more complex distributions learned from data. The change of variables formula plays a crucial role in determining the density of a random variable resulting from a deterministic transformation of another variable.

Change of Variables: Consider two random variables, Z and X, linked by a function $f : \mathbb{R}^n \to \mathbb{R}^n$, where $X = f(Z)$ and $Z = f^{-1}(X)$. The density function of X can then be expressed as:

$$p_X(x) = p_Z(f^{-1}(x)) \left| \det \left(\frac{\partial f^{-1}(x)}{\partial x} \right) \right| \tag{5.1}$$

Key points to consider include:

- The variables x and z must be continuous and share the same dimensionality.
- The Jacobian matrix, $\frac{\partial f^{-1}(x)}{\partial x}$, is an $n \times n$ matrix with entries defined by $\frac{\partial f^{-1}(x)}{\partial x_j}$.
- The determinant of a square matrix A is denoted by $\det(A)$.
- For an invertible matrix A, the relationship $\det(A^{-1}) = \det(A)^{-1}$ holds. Thus, for $z = f^{-1}(x)$, we get:

$$p_X(x) = p_Z(z) \left| \det \left(\frac{\partial f(z)}{\partial z} \right) \right|^{-1} \tag{5.2}$$

- If $\left| \det \left(\frac{\partial f(z)}{\partial z} \right) \right| = 1$, the mapping preserves volume, meaning the transformed distribution p_X retains the same "volume" as the original distribution p_Z.

5.1.2 Normalizing Flow Models

We are now ready to discuss normalizing flow models. Consider a directed latent-variable model involving observed variables X and latent variables Z. In the framework of a normalizing flow model, the mapping between Z and X, represented by $f_\theta : \mathbb{R}^n \to \mathbb{R}^n$, is deterministic and invertible such that $X = f_\theta(Z)$ and $Z = f_\theta^{-1}(X)$.

Using the change of variables, the marginal likelihood $p(x)$ is expressed as:

$$p_X(x;\theta) = p_Z(f_\theta^{-1}(x)) \left| \det \left(\frac{\partial f_\theta^{-1}(x)}{\partial x} \right) \right| \tag{5.3}$$

The term normalizing flow can be understood as follows: Normalizing refers to the fact that the change of variables yields a normalized density after applying an invertible transformation, while Flow signifies that the invertible transformations can be composed to form more complex invertible transformations. Unlike autoregressive models and variational autoencoders, deep normalizing flow models necessitate specific architectural structures: the dimensions of the input and output must be the same, the transformation must be invertible, and the computation of the determinant of the Jacobian must be efficient and differentiable. To learn more about the normalizing flow model, please refer to the literature [101], [102].

5.2 Normalizing Flow for Uncertainty Aware Regression

5.2.1 Prerequisite

In ML, regression refers to a supervised learning technique used to model the relationship between a dependent variable and one or more independent variables. The goal of regression is to find a function that can predict the value of the dependent variable based on the values of the independent variables. Differ from classification, the output of regression model is real-value attributes for the data instances, instead of the predefined classes that the data belong to. The quality of the regression model is typically evaluated based on metrics such as mean squared error, root mean squared error, and others. Formally, given a dataset $\mathcal{D}$, which is made up of N pair training examples, it is expressed as $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$. The optimization process is achieved by adjusting the values of the weights w in order to learn a functional f.

$$\min_{\boldsymbol{w}} J(\boldsymbol{w}); \quad J(\boldsymbol{w}) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_i(\boldsymbol{w}) \tag{5.4}$$

where $\mathcal{L}_i(\cdot)$ is the loss function. Sum of squared erros is the commonly used objective function, $\mathcal{L}_i(\boldsymbol{w}) = \frac{1}{2} \left\| y_i - f\left(\boldsymbol{x}_i; \boldsymbol{w}\right) \right\|^2$, it typically optimizes a model by rewarding correct predictions and penalizing incorrect ones. However, this cannot fit the potential noise and uncertainty estimates when the test data is completely different from the training data.

From the perspective of probabilities, it allows predictions to be made in face of uncertainty. Assume the targets y_i were drawn i.i.d. from Gaussian distribution with mean and variance parameters $\boldsymbol{\theta} = \left(\mu, \sigma^2\right)$. The objective of maximum likelihood estimation (MLE) is to train a model to determine

the value of parameter $\boldsymbol{\theta}$ that maximizes the probability of observing the target outputs, y, as given by the function $p\left(y_i \mid \boldsymbol{\theta}\right)$. This is accomplished by minimizing the loss function of negative log likelihood.

$$\mathcal{L}_i(\boldsymbol{w}) = -\log p(y_i \mid \underbrace{\mu, \sigma^2}_{\boldsymbol{\theta}}) = \frac{1}{2}\log\left(2\pi\sigma^2\right) + \frac{\left(y_i - \mu\right)^2}{2\sigma^2} \tag{5.5}$$

The learned parameter $\boldsymbol{\theta}$ will vary according to different datasets. Uncertainty is then estimated from the numerical properties of learned dataset in statistics. This kind of method, can only model the uncertainty inside the dataset, which is commonly referred to as aleatoric uncertainty, but does not have the ability to estimate the epistemic uncertainty [52]. Implicitly modeling the prior distribution, approaches such as ensemble [54] and dropout [51] have their limitations, as they may sacrifice the estimation of statistics for the sake of S samples. The model can learn hyperparameters of the prior distribution by explicitly placing priors over the likelihood function, which is the approach taken by a group of methods [103], [104], [105], [106], [107], without the need for sampling, it is possible to accurately represent both epistemic and aleatoric uncertainty.

5.2.2 Problem Formulation

The problem we are considering involves observed targets, y_i, drawn independent and identically distributed from a Gaussian distribution, aim to estimate the probabilistic values of unknown mean and variance $\left(\mu, \sigma^2\right)$ using a method similar to classic Maximum Likelihood Estimation (MLE) (Section. III.A). To achieve this, we introduce a prior distribution on $\left(\mu, \sigma^2\right)$. If it is assumed observations are sampled from a Gaussian, as described in Section. III.A, we use Inverse-Gamma prior and Gaussian prior for the unknown variance and mean respectively .

$$
\begin{aligned}
(y_1, \ldots, y_N) &\sim \mathcal{N}\left(\mu, \sigma^2\right) \\
\sigma^2 \sim \Gamma^{-1}(\alpha, \beta) \quad \mu &\sim \mathcal{N}\left(\gamma, \sigma^2 v^{-1}\right)
\end{aligned}
\tag{5.6}
$$

where $\beta > 0, \alpha > 1, v > 0, \gamma \in \mathbb{R}$, $\boldsymbol{m} = (\beta, \alpha, v, \gamma)$ and $\Gamma(\cdot)$ is refer to as gamma function. FlowNet will estimate a posterior distribution $q\left(\mu, \sigma^2\right) = p\left(\mu, \sigma^2 \mid y_1, \ldots, y_N\right)$. Assuming that the estimated distribution can be factorized [108], we obtain an approximation for the true posterior $q\left(\mu, \sigma^2\right) = q(\mu) q\left(\sigma^2\right)$. The approximation we use is in the form of the Normal Inverse-Gamma (N-Γ^{-1}) distribution, which is a Gaussian conjugate prior distribution.

$$
\begin{aligned}
p(\underbrace{\mu, \sigma^2}_{\boldsymbol{\theta}, \boldsymbol{\phi}} \mid \underbrace{\beta, \alpha, v, \gamma}_{\boldsymbol{m}}) = \frac{\beta^\alpha \sqrt{v}}{\Gamma(\alpha)\sqrt{2\pi\sigma^2}} \left(\frac{1}{\sigma^2}\right)^{\alpha+1} \\
\exp\left\{-\frac{2\beta + v(\gamma - \mu)^2}{2\sigma^2}\right\}.
\end{aligned}
\tag{5.7}
$$

The FlowNet model's parameterization is essential and relies on two main components. An encoder neural network f_θ is the first part of FlowNet, the inputs $\mathbf{X}^{(i)}$ is then mapped into a high-dimensional feature space. The second component is a normalizing flow model parameterized by ϕ, is used to learn a normalized sample density on this latent space. According to [109], one way to understand the parameters associated with the corresponding conjugate prior distribution is through the concept "pseudo observations". Such as, the variance of the N-Γ^{-1} distribution could thought of deriving from α pseudo observations accompany by a number $2v$ as sum of squared deviations and with sample mean α. Whereas, the mean is estimated from v pseudo observations accompany by a number γ as sample mean. Based on the stated perspective, we can define the total pseudo count, denoted by Φ,

of the target distribution as summation of all deduced pseudo observations count, which is equal to $2v$ plus α . It is important to note that the second part of FlowNet must be a proper normalized density function to make sure that the model's epistemic uncertainty increases, while sample lie out of known distribution. Our approach centers around the core concept of utilizing normalizing flows to parameterize distributions. Normalizing flows [101], such as radial flow [100], RealNVP [110] or MAF [111] , offer a flexible yet manageable family of distributions. It is worth noting that empowered with a sufficiently expressive and deep model [112], [113], normalizing flows can theoretically model any continuous distribution.

5.2.3 Estimation of Uncertainty

The two types of uncertainty in a prediction can be classified as aleatoric uncertainty, which is also known as statistical or data uncertainty, and epistemic uncertainty, which represents the lack of knowledge in the prediction. By using N-Γ^{-1} distribution, we can calculate the epistemic uncertainty, aleatoric uncertainty and prediction.

$$\underbrace{\mathbb{E}\left[\sigma^2\right] = \frac{\beta}{\alpha - 1}}_{\text{aleatoric}}, \quad \underbrace{\text{Var}[\mu] = \frac{\beta}{v(\alpha - 1)}}_{\text{epistemic}}, \quad \underbrace{\mathbb{E}[\mu] = \gamma}_{\text{prediction}}. \tag{5.8}$$

According to the nature of the N-Γ^{-1} distribution, we can understand aleatoric uncertainty and epistemic uncertainty as the mean of the variance and the variance of the mean, respectively.

5.2.4 Learning Target Distribution

After formalizing the use of N-Γ^{-1} distribution to obtain both epistemic and aleatoric uncertainty, our consequent step is to train a model that outputs outcome hyperparameters of this distribution. To make the learning process

clearer, we divide it into two distinct parts. The first part involves obtaining and maximizing model evidence to support for the observation, while the second part involves inflating uncertainty and minimizing evidence when prediction is incorrect. Broadly speaking, the first part involves fitting data to FlowNet, while the second part enforces a prior that removes inaccurate observation and inflates uncertainty.

To maximize the model fit, one can employ the Bayesian probability theory and utilize the marginal likelihood, also known as the model evidence. This quantity represents the probability of observing the data, y_i, given the values of the distribution parameters, $\boldsymbol{m}$, and is obtained by integrating over the possible values of the likelihood parameters, $\boldsymbol{\theta}, \boldsymbol{\phi}$:

$$\begin{aligned} p\left(y_i \mid \boldsymbol{m}\right) &= \frac{p\left(y_i \mid \boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{m}\right) p(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \boldsymbol{m})}{p\left(\boldsymbol{\theta}, \boldsymbol{\phi} \mid y_i, \boldsymbol{m}\right)} \\ &= \int_{\sigma^2=0}^{\infty} \int_{\mu=-\infty}^{\infty} p\left(y_i \mid \mu, \sigma^2\right) p\left(\mu, \sigma^2 \mid \boldsymbol{m}\right) \mathrm{d}\mu \mathrm{d}\sigma^2 \end{aligned} \tag{5.9}$$

Evaluating the model evidence is typically a challenging task as it requires integrating over the latent model parameters. However, if we use a N-Γ^{-1} prior for our Gaussian likelihood function, an analytical solution can be obtained:

$$
\begin{aligned}
&p\left(y_i \mid \boldsymbol{m}\right) = \int_{\boldsymbol{\theta},\boldsymbol{\phi}} p\left(y_i \mid \boldsymbol{\theta},\boldsymbol{\phi}\right) p(\boldsymbol{\theta},\boldsymbol{\phi} \mid \boldsymbol{m}) \mathrm{d}(\boldsymbol{\theta},\boldsymbol{\phi}) \\
&= \int_{\sigma^2=0}^{\infty} \int_{\mu=-\infty}^{\infty} p\left(y_i \mid \mu, \sigma^2\right) p\left(\mu, \sigma^2 \mid \boldsymbol{m}\right) \mathrm{d}\mu \mathrm{d}\sigma^2 \\
&= \int_{\sigma^2=0}^{\infty} \int_{\mu=-\infty}^{\infty} p\left(y_i \mid \mu, \sigma^2\right) p\left(\mu, \sigma^2 \mid \gamma, v, \alpha, \beta\right) \mathrm{d}\mu \mathrm{d}\sigma^2 \\
&= \int_{\sigma^2=0}^{\infty} \int_{\mu=-\infty}^{\infty} \left[\sqrt{\frac{1}{2\pi\sigma^2}} \exp\left\{-\frac{(y_i-\mu)^2}{2\sigma^2}\right\}\right] \\
&\left[\frac{\beta^{\alpha}\sqrt{v}}{\Gamma(\alpha)\sqrt{2\pi\sigma^2}} \left(\frac{1}{\sigma^2}\right)^{\alpha+1} \exp\left\{-\frac{2\beta + v(\gamma-\mu)^2}{2\sigma^2}\right\}\right] \mathrm{d}\mu \mathrm{d}\sigma^2 \\
&= \int_{\sigma^2=0}^{\infty} \frac{\beta^{\alpha}\sigma^{-3-2\alpha}}{\sqrt{2\pi}\sqrt{1+1/v}\Gamma(\alpha)} \exp\left\{-\frac{2\beta + \frac{v(y_i-\gamma)^2}{1+v}}{2\sigma^2}\right\} \mathrm{d}\sigma^2 \\
&= \int_{\sigma=0}^{\infty} \frac{\beta^{\alpha}\sigma^{-3-2\alpha}}{\sqrt{2\pi}\sqrt{1+1/v}\Gamma(\alpha)} \exp\left\{-\frac{2\beta + \frac{v(y_i-\gamma)^2}{1+v}}{2\sigma^2}\right\} 2\sigma \mathrm{d}\sigma \\
&= \frac{\Gamma(1/2+\alpha)}{\Gamma(\alpha)} \sqrt{\frac{v}{\pi}} (2\beta(1+v))^{\alpha} \\
&\left(v\left(y_i-\gamma\right)^2 + 2\beta(1+v)\right)^{-\left(\frac{1}{2}+\alpha\right)} \\
&p\left(y_i \mid \boldsymbol{m}\right) = \mathrm{St}\left(y_i; \gamma, \frac{\beta(1+v)}{v\alpha}, 2\alpha\right).
\end{aligned}
\tag{5.10}
$$

The Student-t distribution with degrees of freedom v_{St}, scale σ^2_{St} and location μ_{St} is denoted by $\mathrm{St}\left(y; \mu_{\mathrm{St}}, \sigma^2_{\mathrm{St}}, v_{St}\right)$, where y represents the input. The negative logarithm of the model evidence is expressed as the loss function $\mathcal{L}_i^{\mathrm{NLL}}(\boldsymbol{w})$.

$$
\begin{aligned}
\mathcal{L}_i^{\mathrm{NLL}}(\boldsymbol{w}) = \frac{1}{2}\log\left(\frac{\pi}{v}\right) - \alpha\log(\Omega) + \left(\alpha+\frac{1}{2}\right)\log \\
\left(\left(y_i-\gamma\right)^2 v + \Omega\right) + \log\left(\frac{\Gamma(\alpha)}{\Gamma\left(\alpha+\frac{1}{2}\right)}\right)
\end{aligned}
\tag{5.11}
$$

where $\Omega = 2\beta(1+v)$. By maximizing the model evidence, a neural network can be trained to output parameters of the N-Γ^{-1} distribution that fit the input observations. The loss function $\mathcal{L}_i^{\text{NLL}}(\boldsymbol{w})$ serves as an objective for this training process.

To regularize the training process (penalty on incorrect evidence), a technique is introduced where an incorrect evidence penalty is applied to minimize evidence on incorrect predictions. In the setting of classification, its effectiveness has been proven [106]. For the regression case, a similar minimization involves $KL[p(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \boldsymbol{m}) \| p(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \tilde{\boldsymbol{m}})]$, where $\tilde{m}$ represents the parameter belong to arbitrary N-Γ^{-1} prior with zero evidence . However, the KL between arbitrary N-Γ^{-1} and N-Γ^{-1} with zero evidence prior is undefined, these approaches to regularizing evidential learning are not applicable in regression. An alternative approach is that, by introducing some non-zero evidence (ϵ-evidence) to make the KL finite and defined. However, this would cause hypersensitivity to the selection of the ϵ value, leading to highly unstable training. Therefore, this alternative is not a practical solution. As a result, we employ methods that directly penalize incorrect evidence.

$$\mathcal{L}_i^{\text{R}}(\boldsymbol{w}) = |y_i - \mathbb{E}\left[\mu_i\right]| \cdot \Phi = |y_i - \gamma| \cdot (2v + \alpha) \tag{5.12}$$

The complete cost function, $\mathcal{L}_i(\boldsymbol{w})$, includes two distinct loss terms that serve to maximize and regularize evidence. A regularization coefficient (λ) is applied to these two terms to appropriately scale their contributions within the total loss.

$$\mathcal{L}_i(\boldsymbol{w}) = \mathcal{L}_i^{\text{NLL}}(\boldsymbol{w}) + \lambda \mathcal{L}_i^{\text{R}}(\boldsymbol{w}) \tag{5.13}$$

The regularization coefficient λ strikes a balance between the inflation of uncertainty and model fit. If λ is set to 0, the resulting estimate may be overly confident, while setting λ too high could lead to excessive inflation. During

training, the parameters m of target distribution is generated by the proposed model, with $\boldsymbol{m}_i$ being generated by the function $f(\boldsymbol{x}_i; \boldsymbol{w})$. Since each target y is associated with four parameters, our proposed model has four output neurons for each target y. To make certain that the constraints on (β, α, v) are enforced, we apply a softplus activation function (since $\alpha > 1$, with an additional +1 added). For other parameters, linear activation is used.

5.3 Experiments

All experiments were conducted on a workstation equipped with an Intel Core i7-10750H CPU running at 2.60 GHz (6 cores, 12 threads), an NVIDIA GeForce RTX 3070 Laptop GPU with 8 GB of dedicated memory, and 32 GB of DDR4 RAM. All models were implemented and trained using the TensorFlow framework. The details of the experimental parameters will be given in the description of the different experimental sections below.

5.3.1 Toy Dataset

We first validated our ideas on small dataset and compared with baseline methods. Following [54], [114], the toy dataset has inputs uniformly and randomly in the range of $[-4, 4]$. For each input x, the corresponding target y is computed as $y = x^3 + \epsilon_n$, where $\epsilon_n \sim \mathcal{N}(0, 3)$. We assessed aleatoric within ± 4 and epistemic ± 6 uncertainty estimation. We evaluated three normalizing flow methods - Radial[100], Planar[100] and RealNVP flow [110] in comparison with three baselines - PBP [114], Ensembling [54] and Dropout [51]. All the models were trained with the same parameters as $\eta = 5\text{e} - 3$ for Adam optimizer learning rate, batch size of 128 and train 5000 iterations, sampling based models [51], [54] employed $n = 5$ samples. As shown in Figure 5.1, Within the training range $[-4, 4]$, almost all methods are able to accurately predict aleatoric uncertainty. As going beyond the training range,

	RMSE				NLL			
Datasets	Dropout	Ensembles	Evidential	FlowNet	Dropout	Ensembles	Evidential	FlowNet
Boston	2.97 ± 0.19	3.28 ± 1.00	3.06 ± 0.16	$\mathbf{2.38 \pm 0.22}$	2.46 ± 0.06	2.41 ± 0.25	$\mathbf{2.35 \pm 0.06}$	$\mathbf{2.24 \pm 0.07}$
Concrete	$\mathbf{5.23 \pm 0.12}$	6.03 ± 0.58	5.85 ± 0.15	5.81 ± 0.19	$\mathbf{3.04 \pm 0.02}$	3.06 ± 0.18	$\mathbf{3.01 \pm 0.02}$	3.09 ± 0.02
Energy	1.66 ± 0.04	2.09 ± 0.29	2.06 ± 0.10	$\mathbf{0.93 \pm 0.17}$	1.99 ± 0.02	1.38 ± 0.22	1.39 ± 0.06	$\mathbf{1.10 \pm 0.09}$
Kin8nm	0.10 ± 0.00	0.09 ± 0.00	0.09 ± 0.00	$\mathbf{0.05 \pm 0.00}$	-0.95 ± 0.01	-1.20 ± 0.02	-1.24 ± 0.01	$\mathbf{-1.37 \pm 0.03}$
Naval	0.01 ± 0.00	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	-3.80 ± 0.01	-5.63 ± 0.05	-5.73 ± 0.07	$\mathbf{-5.99 \pm 0.07}$
Power	4.02 ± 0.04	4.11 ± 0.17	4.23 ± 0.09	$\mathbf{2.79 \pm 0.09}$	2.80 ± 0.01	2.79 ± 0.04	2.81 ± 0.07	$\mathbf{2.44 \pm 0.02}$
Protein	$\mathbf{4.36 \pm 0.01}$	4.71 ± 0.06	4.64 ± 0.03	$\mathbf{4.13 \pm 0.32}$	2.89 ± 0.00	2.83 ± 0.02	$\mathbf{2.63 \pm 0.00}$	$\mathbf{2.55 \pm 0.14}$
Yacht	1.11 ± 0.09	1.58 ± 0.48	1.57 ± 0.56	$\mathbf{0.75 \pm 0.18}$	1.55 ± 0.03	1.18 ± 0.21	1.03 ± 0.19	$\mathbf{0.62 \pm 0.11}$

TABLE 5.1: **RMSE** and negative log-likelihood (**NLL**) Benchmark tests summary in statistics. dropout sampling [51], model ensembling [54], evidential regression [107] and our proposed FlowNet. The best results for each dataset and metric are highlighted in bold, with a sample size of 5 for the baseline methods. On almost all datasets, FlowNet surpasses baseline methods in terms of NLL and RMSE performance.

which of greater than 4 and less than -4 depicted in Figure5.1, the epistemic uncertainty begins to increase. The methods based on the normalizing flows well bound the uncertainty in a small range near the ground truth, and the prediction of the baseline methods on the epistemic uncertainty grually fail.

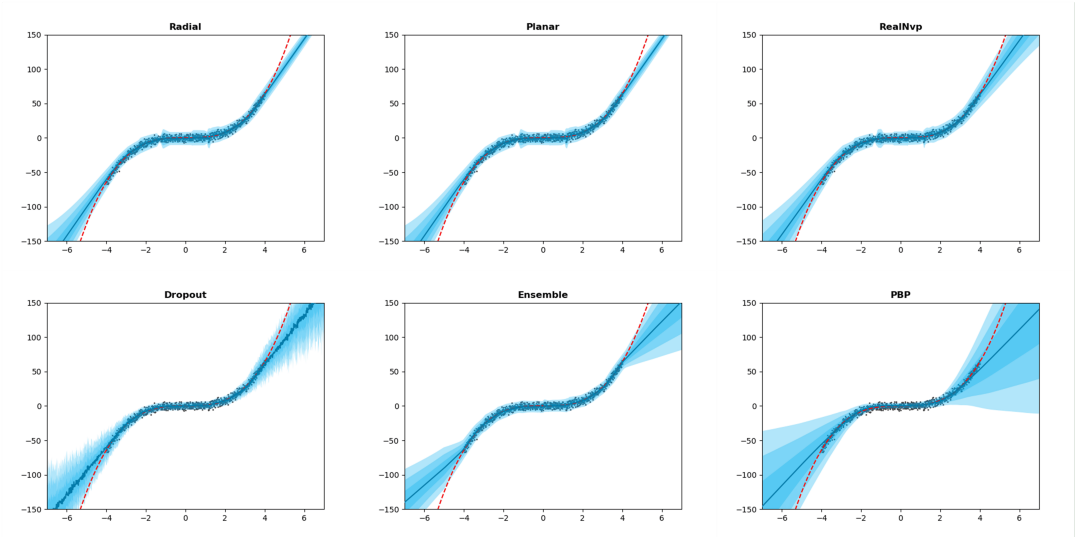

FIGURE 5.1: Toy dataset uncertainty estimation trained on $y = x^3 + \epsilon_n, \epsilon_n \sim \mathcal{N}(0, 3)$. The top three graphs are the uncertainty estimation of FlowNet based on various normalizing flows, and the bottom three graphs are the baseline methods. FlowNet is capable of bounding the epistemic uncertainty near the ground truth, whereas baseline methods were less accurate in prediction of epistemic uncertainty.

5.3.2 Real World Datasets

In this set of experiments, we followed the same experiment setup used by [51], [54]. We evaluated FlowNet with RealNvp realization in comparison with three baseline methods - Dropout [51], Ensembles [54] and Evidential [107] from the aspects of root mean squared error (RMSE) and negative log-likelihood (NLL). The results shows summary statistics in Table 5.1. On each data set, the top results among proposed method are shown in bold font. From Table 5.1, we can conclude that whether in terms of RMSE or NLL, FlowNet exceeds the baseline methods, almost in all data sets, except concrete. At the same time, we observed in the experiment that with the addition of more intermediate normalizing flow layers, the performance is further improved, and currently only one layer was employed.

5.3.3 Vision Tasks in Complex Scenes

We further evaluated the effectiveness of FlowNet on more complex vision tasks that are more close to real scenes. image depth estimation is the task of estimating the depth value (distance relative to the camera) of each pixel given a single (monocular) RGB image. This task has a wide range of applications in many fields such as virtual reality, semantic segmentation, automatic driving, and 3D reconstruction. Due to the lack of a single image for spatial information, object occlusion, movement, and the need to process high-dimensional data at the pixel level, this problem still remains challenging.

We employed NYU Depth v2 [115] dataset as training dataset. The NYU Depth v2 dataset is a large, publicly available dataset and widely used in the computer vision community as a benchmark for evaluating the performance of depth estimation algorithms. It contains diverse indoor scenes image pairs (e.g. office, libraries, etc.), where each pair consists of an RGB image and its

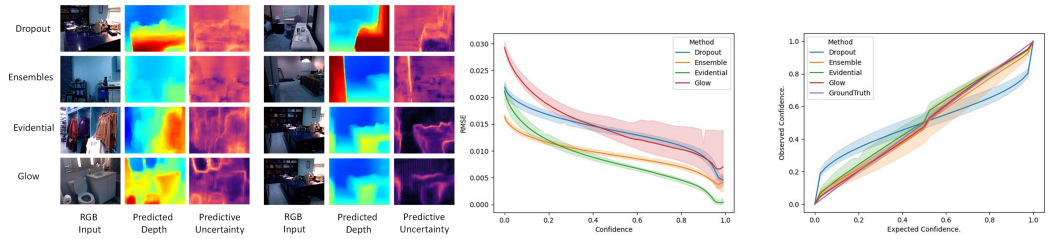

FIGURE 5.2: Illustration of epistemic uncertainty in depth estimation. (Left) An illustration of depth predictions and the estimation of uncertainty at the pixel level. (Middle) Relationship between observed error and prediction confidence level; usually inverse trend is desired. (Right) With inset shows calibration errors, model uncertainty calibration [120], where the ideal relationship between predicted uncertainty and actual uncertainty is y = x.

corresponding depth map. The depth maps were acquired using a Microsoft Kinect sensor, providing high-quality, dense depth information for each image in the dataset. FlowNet use U-Net [116] as the backbone, and in order to take a full advantage of the processing for the image dataset, we combined the Glow model [117] (with Invertible *1×1* Convolutions) to get the final output. The final layer outputs a single $H \times W$ activation map in the case of regression. Following the experiment setup with [107], the FlowNet model generates four outputs, corresponding to $(\beta, \alpha, v, \gamma)$ respectively, under restrictions. For the dropout implementation, spatial dropout uncertainty sampling [118], [119] was used.

We tested the model on unseen data in the subject of accuracy and predictive epistemic uncertainty. The predicted depth and predictive entropy are shown in Figure 5.2 left side, for randomly selected test images. An effective measure of epistemic uncertainty should be able to detect inaccuracies in predictions, which FlowNet effectively captures while providing clear confidence estimates. In contrast, dropout significantly underestimates uncertainty and ensembling sometimes overestimates it. Figure 5.2 Middle part clearly shows the inverse trend between observed error and prediction confidence. FlowNet, while being able to accurately predict epistemic uncertainty,

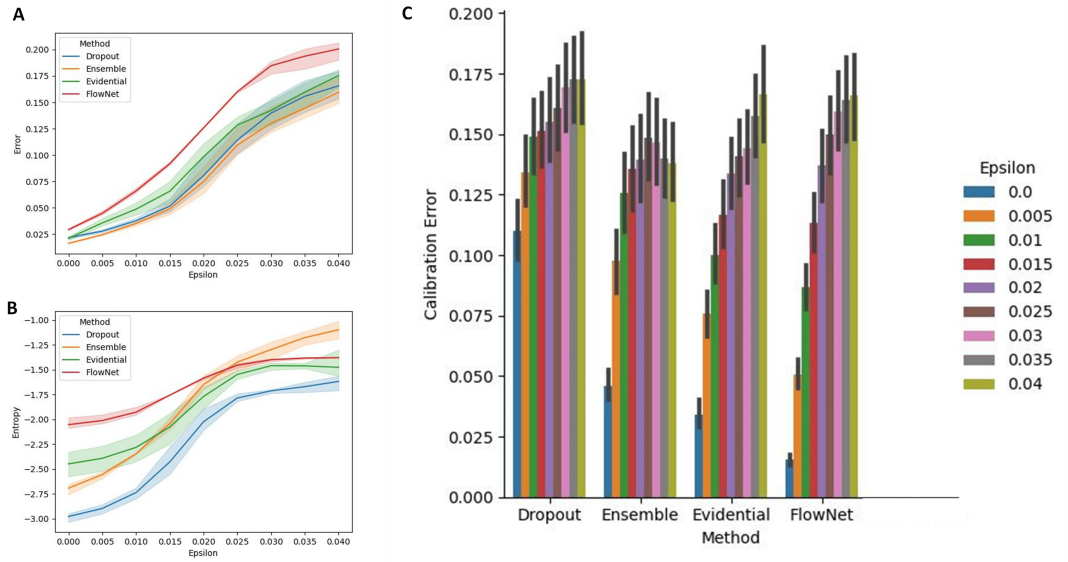

FIGURE 5.3: The robustness of uncertainty estimates under adversarial noise is explored. The relationship between adversarial noise and both the estimated epistemic uncertainty (B) and predictive error (A) is studied. (C) The calibration performance of various methods is compared visually as the noise level increases. FlowNet exhibits the highest calibration performance among the baseline methods.

has a prediction accuracy comparable to start-of-the-arts.

In Figure 5.2 right part, we further assess the accuracy of FlowNet on uncertainty estimates. The calibration curves are calculated as described in [120], with the ideal curve being $y = x$, Approximately 90% of the time, the target falls within a 90% certainty gap, as indicated. The results reveal that dropout method tends to overestimate confidence in low-confidence scences (0.126), while evidential (0.033) and ensembling (0.048) performs better but still falls short compared to FlowNet (calibration error: 0.015).

5.3.4 Resilience Against Adversarial Examples

We then examined the scenario of OOD detection where inputs are deliberately altered to produce incorrect predictions. To generate adversarial perturbations for our test set, we employed FGSM algorithm (detailed in [121]) with gradually increasing levels of noise, represented by Epsilon (ϵ). It is

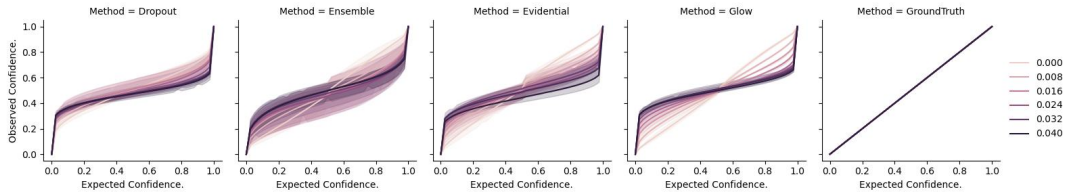

FIGURE 5.4: The relationship between Expected Confidence and Observed Confidence of FlowNet and baseline methods.

important to note that this experiment was not aimed at presenting a solution for advanced adversarial attacks, but rather to showcase that FlowNet accurately reflects heightened predictive uncertainty on samples that have undergone adversarial manipulations.

The results in Figure 5.3 A show that as adversarial noise is added, the absolute error of all methods increases. Additionally, Figure 5.3 B indicates that there is a positive effect of noise on our predictive uncertainty estimates. However, as noise levels continue to rise beyond a certain threshold, the ensemble method appears to exhibit better performance in terms of predictive uncertainty. This observation underscores the ensemble method's robustness to high noise levels, possibly due to its inherent diversity among multiple models, which can provide a broader perspective on uncertainty. Figure 5.3 C compares the calibration performance of various methods visually as the noise level increases. FlowNet achieves the best calibration performance compared to other baseline methods. Figure 5.4 shows the relationships between expected confidence and observed confidence. The regression model based on normalizing flow controls the uncertainty in a smaller range compared with other baseline methods.

5.3.5 OOD Sample Testing

The purpose of estimating uncertainty is to determine when a ML model encounters test samples that are not part of its training distribution or when its prediction cannot be relied upon. This section looks into the capacity

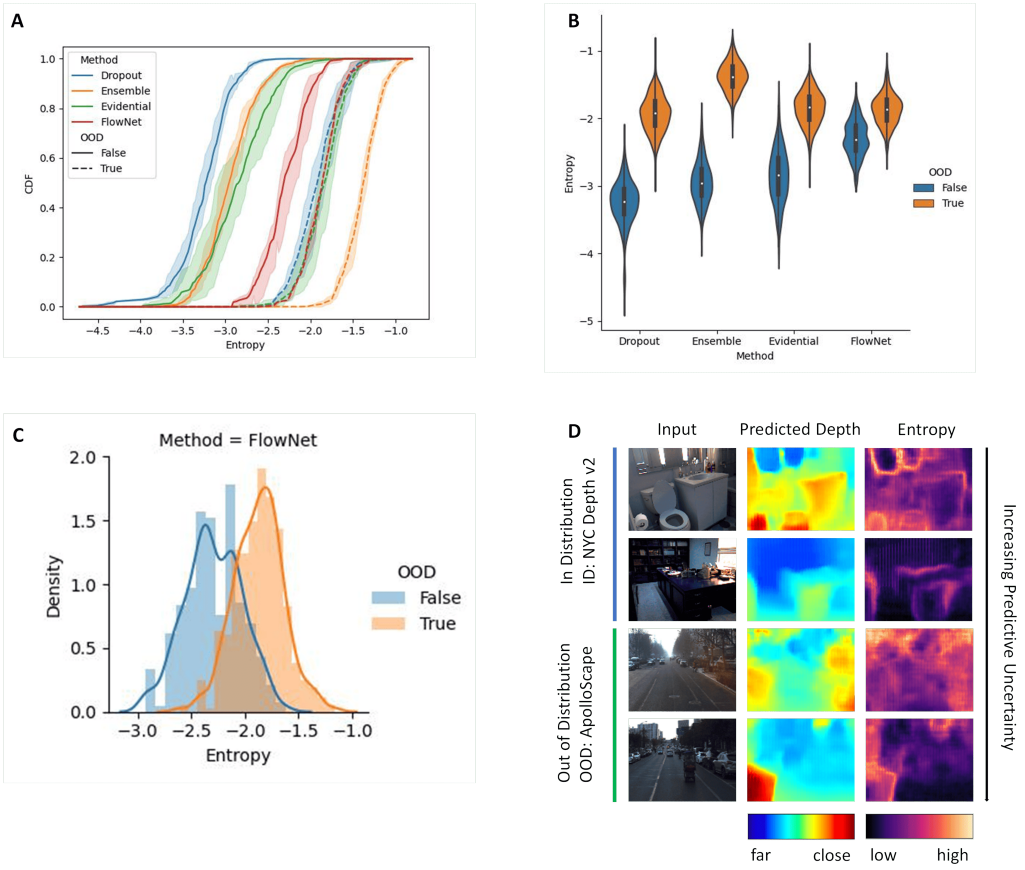

FIGURE 5.5: The uncertainty of OOD data is analyzed. FlowNet shows low uncertainty (entropy) on ID data and amplifies uncertainty on OOD data. (**A**) presents the cumulative density function (CDF) of ID and OOD entropy for the tested methods, and OOD detection was evaluated using AUC-ROC. (**B**) compares uncertainty (entropy) across the methods. (**C**) displays full density histograms of entropy estimated by FlowNet for ID and OOD data. (**D**) Examples of predictions including both ID and OOD data.

of FlowNet in dealing with heightened epistemic uncertainty in the case of OOD data, as evaluated on the ApolloScape [122] OOD dataset for outdoor driving scenes. It is important to emphasize that other techniques like Prior Networks [103], [104] feel necessity for OOD data to further guide the identification of instances with high uncertainty during the training process, while FlowNet only replies on ID data during training and does not have this restriction.

In order to test the model, we input both ID and OOD test datasets and further documented average entropy predicted for each test image. Figure

5.5 **A** displays for each test set and method, the entropy of the cumulative density function. All models performed as expected, with a positive shift, among all the models, FlowNet is competitive as shown in the results. The distribution of entropy is summarized in Figure 5.5 **B** using violin plots, again highlighting the clear distinction in uncertainty on OOD data. Figure 5.5 **C** shows the density distribution of the ID and OOD data. and Figure 5.5 **D** provides examples of predictions (both ID and OOD). These results indicate that FlowNet, without having OOD data during training, can effectively capture increased uncertainty on OOD data, matching the performance of established epistemic uncertainty estimation benchmarks.

5.4 Summary

This chapter presents a normalizing flow–based approach to uncertainty estimation, detailing how invertible transformations enable precise modeling of predictive distributions and analytic computation of both epistemic and aleatoric uncertainty. Through extensive experiments on regression benchmarks, adversarial robustness tests, and out-of-distribution detection, our proposed method consistently outperforms standard BNN approximation such as, MC Dropout, deep ensembles, and PBP—in predictive accuracy (NLL/RMSE), calibration (AUROC), demonstrating its strength for real-time, high-dimensional and adversarially challenging distributed learning scenarios.

Chapter 6

Conclusion and Future Work

This section summarises the work of the thesis and gives directions for future development.

6.1 Conclusion

This thesis presented a comprehensive exploration of key challenges in machine learning, specifically focusing on data privacy protection, privacy threats in decentralized learning, and uncertainty estimation. Several innovative methods were proposed to address these issues. First, the thesis systematically analyzed privacy attacks in decentralized learning, with particular emphasis on reconstruction attacks in Decentralized Gradient Descent (D-GD) and Gossip averaging protocols, revealing potential privacy risks within decentralized architectures and proposing effective defense mechanisms. These contributions provide a solid theoretical and practical foundation for safeguarding data privacy in decentralized environments. Additionally, the thesis introduced a task-adaptive privacy protection method that combines differential privacy and local differential privacy, allowing dynamic adjustment of noise levels based on task characteristics to maximize model utility while ensuring privacy. Experimental results showed that this approach outperforms existing baseline methods in multi-dimensional data scenarios, particularly in real estate valuation and breast cancer detection, where it achieves

higher task accuracy alongside strong privacy protection. Furthermore, the thesis explored uncertainty estimation methods, introducing a normalizing flow-based approach that significantly enhances model robustness when dealing with anomalous and out-of-distribution data. The experimental validation demonstrated that this method not only effectively identifies high-uncertainty predictions but also enhances overall model reliability when integrated with additional privacy measures.

Despite these contributions, there are some limitations to this work. The proposed methods were primarily evaluated on specific datasets and tasks, which may not fully capture the diversity and complexity of real-world scenarios. For instance, the effectiveness of the task-adaptive privacy protection method might vary under different data distributions and larger-scale deployments. Additionally, while the normalizing flow-based uncertainty estimation approach has shown promise, its computational complexity could limit its applicability to resource-constrained environments. Future work should focus on addressing these limitations by exploring more diverse datasets, optimizing computational efficiency, and extending the applicability of the proposed methods to other domains and larger-scale decentralized networks.

Throughout this thesis, our research has been driven by three primary objectives: enhancing privacy preservation, model robustness, and scalability in distributed machine learning systems. In Chapter 3, the proposed attack detection and defense framework was empirically validated against adversarial reconstruction attacks in decentralized settings. Chapter 4 introduced an adaptive joint differential and local differential privacy mechanism that achieves superior privacy–utility trade-offs across multidimensional tasks. Chapter 5 presented a normalizing flow–based uncertainty estimation approach, demonstrating its combined advantages in uncertainty quantification accuracy and real-time inference efficiency across regression, adversarial robustness, and out-of-distribution detection scenarios. These outcomes

directly fulfill the original PhD objectives and provide a strong foundation for application in large-scale, resource-constrained, or high-latency network environments.

6.2 Future Work

While this thesis has made significant progress in decentralized learning, privacy protection, and uncertainty estimation, there are numerous promising directions for future research.

First, in the realm of privacy protection, the task-adaptive local differential privacy method can be further optimized to handle more complex multi-dimensional data and real-world application scenarios. Advanced noise distribution mechanisms, such as those inspired by generative adversarial networks (GANs) [123] or variational autoencoders (VAEs) [124], could be explored to dynamically generate noise distributions that adapt to specific tasks and data distributions, potentially reducing the trade-off between privacy and utility. Moreover, privacy-preserving mechanisms like personalized differential privacy (PDP) [125] and federated learning with secure multiparty computation (SMPC) [126] could be integrated to provide tailored privacy levels based on individual node requirements.

Communication delays and computational overhead in decentralized learning remain significant challenges. Recent advances in asynchronous decentralized optimization, such as those leveraging graph neural networks (GNNs) for topology-aware communication, could help mitigate these issues [127]. Exploring efficient distributed communication protocols like gossip-based accelerated methods or communication-efficient SGD variants might also enhance the scalability and robustness of decentralized learning systems, particularly in environments with intermittent connectivity, such as edge computing or IoT networks.

In uncertainty estimation, future research can explore more adaptable models to handle diverse data types and tasks. Methods like Bayesian deep learning [128] and ensemble-based approaches [126] offer promise for improving robustness and reliability. Transformer architectures [3] could also

enhance uncertainty estimation, especially in high-dimensional data and real-time applications like healthcare and autonomous systems. Neural architecture search (NAS) [129] could further optimize models, balancing efficiency and accuracy.

Another focus is adapting privacy protection and uncertainty estimation for resource-constrained settings in federated and decentralized learning. Lightweight cryptographic protocols, like homomorphic encryption, combined with on-device machine learning, can improve scalability and security [130].

Future, with the growth of edge computing and IoT, there is a rising need for privacy-preserving and reliable decentralized learning. Edge-native algorithms and approaches like federated and swarm learning could address this [131], enabling applications in healthcare, smart grids, and autonomous vehicles. Blockchain-based solutions [132] may also enhance trust and resilience in these systems.

Finally, for stakeholders looking to deploy the proposed methods in industrial or research settings, we recommend: first, calibrating privacy budgets and uncertainty thresholds according to specific use cases and conducting small-scale pilot tests to validate performance and overhead; second, leveraging modern hardware accelerators (e.g., GPUs) and lightweight cryptographic protocols (such as homomorphic encryption or secure multi-party computation) to optimize system efficiency; and third, establishing continuous monitoring and logging mechanisms to track privacy risks and uncertainty metrics, enabling adaptive adjustments in dynamic environments to maintain robust and reliable operation.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[3] A Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[4] S Hochreiter, "Long short-term memory," *Neural Computation MIT-Press*, 1997.

[5] R. Kruse, S. Mostaghim, C. Borgelt, C. Braune, and M. Steinbrecher, "Multi-layer perceptrons," in *Computational intelligence: a methodological introduction*, Springer, 2022, pp. 53–124.

[6] A. El Ouadrhiri and A. Abdelhadi, "Differential privacy for deep and federated learning: A survey," *IEEE access*, vol. 10, pp. 22 359–22 380, 2022.

[7] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Computers & Industrial Engineering*, vol. 149, p. 106 854, 2020.

[8] P. M. Mammen, "Federated learning: Opportunities and challenges," *arXiv preprint arXiv:2101.05428*, 2021.

[9] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients-how easy is it to break privacy in federated learning?" *Advances in neural information processing systems*, vol. 33, pp. 16 937–16 947, 2020.

[10] S. Kariyappa et al., "Cocktail party attack: Breaking aggregation-based privacy in federated learning using independent component analysis," in *International Conference on Machine Learning*, PMLR, 2023, pp. 15 884–15 899.

[11] N. Rieke et al., "The future of digital health with federated learning," *NPJ digital medicine*, vol. 3, no. 1, pp. 1–7, 2020.

[12] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," *Advances in neural information processing systems*, vol. 30, 2017.

[13] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. Stich, "A unified theory of decentralized sgd with changing topology and local updates," in *International Conference on Machine Learning*, PMLR, 2020, pp. 5381–5393.

[14] H. Xu, K. P. Seng, L. M. Ang, and J. Smith, "Decentralized and distributed learning for aiot: A comprehensive review, emerging challenges and opportunities," *IEEE Access*, 2024.

[15] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE transactions on information theory*, vol. 52, no. 6, pp. 2508–2530, 2006.

[16] I. Hegedűs, G. Danner, and M. Jelasity, "Gossip learning as a decentralized alternative to federated learning," in *Distributed Applications and Interoperable Systems: 19th IFIP WG 6.1 International Conference, DAIS*

2019, Held as Part of the 14th International Federated Conference on Distributed Computing Techniques, DisCoTec 2019, Kongens Lyngby, Denmark, June 17–21, 2019, Proceedings 19, Springer, 2019, pp. 74–90.

[17] R. Berthier, F. Bach, and P. Gaillard, "Accelerated gossip in networks of given dimension using jacobi polynomial iterations," *SIAM Journal on Mathematics of Data Science*, vol. 2, no. 1, pp. 24–47, 2020.

[18] F. Hanzely, J. Konečnỳ, N. Loizou, P. Richtárik, and D. Grishchenko, "Privacy preserving randomized gossip algorithms," *arXiv preprint arXiv:1706.07636*, 2017.

[19] L. Yao-Huai, "Privacy and data privacy issues in contemporary china," *Ethics and Information Technology*, vol. 7, pp. 7–15, 2005.

[20] Q. Covert, D. Steinhagen, M. Francis, and K. Streff, "Towards a triad for data privacy," 2020.

[21] C. Véliz, *Privacy is power*. Melville House Brooklyn, 2021.

[22] S. Biswas, N. Khare, P. Agrawal, and P. Jain, "Machine learning concepts for correlated big data privacy," *Journal of Big Data*, vol. 8, no. 1, p. 157, 2021.

[23] V. J. Hotz et al., "Balancing data privacy and usability in the federal statistical system," *Proceedings of the National Academy of Sciences*, vol. 119, no. 31, e2104906119, 2022.

[24] E. Dreyfuss, "Facebook hires up three of its biggest privacy critics," *Wired. com, Ed., ed*, 2019.

[25] P. Hustinx, "Eu data protection law: The review of directive 95/46/ec and the proposed general data protection regulation," *University of Tartu. Data Protection Inspectorate, Tallinn*, 2013.

[26] S Warren and L. D. Brandeis, "Bthe right to privacyˆ," *Harvard Law Review*, vol. 15, no. 5, 1890.

[27] J. Harper, "Privacy and fair information practices: The struggle to protect threatened values," *American Enterprise Institute report*, 2021.

[28] A. Coos, "Data protection in japan: All you need to know about appi," *Endpoint Protector. April 5, 2022. URL: https://www. endpointprotector. com/blog/dataprotection-in-japan-appi/(Last accessed: 23.01. 2024)*, 2019.

[29] A. Bernot, D. Cooney-O'Donoghue, and M. Mann, "Governing chinese technologies: Tiktok, foreign interference, and technological sovereignty," *Internet Policy Review*, vol. 13, no. 1, 2024.

[30] R. N. Zaeem and K. S. Barber, "The effect of the gdpr on privacy policies: Recent progress and future promise," *ACM Transactions on Management Information Systems (TMIS)*, vol. 12, no. 1, pp. 1–20, 2020.

[31] A. Tsohou et al., "Privacy, security, legal and technology acceptance elicited and consolidated requirements for a gdpr compliance platform," *Information & Computer Security*, vol. 28, no. 4, pp. 531–553, 2020.

[32] S. S. Bakare, A. O. Adeniyi, C. U. Akpuokwe, and N. E. Eneh, "Data privacy laws and compliance: A comparative review of the eu gdpr and usa regulations," *Computer Science & IT Research Journal*, vol. 5, no. 3, pp. 528–543, 2024.

[33] I. Buri and J. van Hoboken, "The digital services act (dsa) proposal: A critical overview," *Digital Services Act (DSA) Observatory*, 2021.

[34] M. Hilton et al., "Differential privacy: A historical survey," *Cal Poly State University*, 2012.

[35] C. Dwork, "Differential privacy: A survey of results," in *International conference on theory and applications of models of computation*, Springer, 2008, pp. 1–19.

[36] C. Dwork, F. McSherry, K. Nissim, and A. 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 3*, Springer, 2006, pp. 265–284.

[37] M. Yang, T. Guo, T. Zhu, I. Tjuawinata, J. Zhao, and K.-Y. Lam, "Local differential privacy and its applications: A comprehensive survey," *Computer Standards & Interfaces*, p. 103 827, 2023.

[38] S. Z. El Mestari, G. Lenzini, and H. Demirci, "Preserving data privacy in machine learning systems," *Computers & Security*, vol. 137, p. 103 605, 2024.

[39] L. Huang, J. Wu, D. Shi, S. Dey, and L. Shi, "Differential privacy in distributed optimization with gradient tracking," *IEEE Transactions on Automatic Control*, vol. 69, no. 9, pp. 5727–5742, 2024.

[40] B. Ghazi, N. Golowich, R. Kumar, P. Manurangsi, and C. Zhang, "Deep learning with label differential privacy," *Advances in neural information processing systems*, vol. 34, pp. 27 131–27 145, 2021.

[41] M. Rigaki and S. Garcia, "A survey of privacy attacks in machine learning," *ACM Computing Surveys*, vol. 56, no. 4, pp. 1–34, 2023.

[42] C. Liu, N. Bastianello, W. Huo, Y. Shi, and K. H. Johansson, "A survey on secure decentralized optimization and learning," *arXiv preprint arXiv:2408.08628*, 2024.

[43] M. Assran, N. Loizou, N. Ballas, and M. Rabbat, "Stochastic gradient push for distributed deep learning," in *International Conference on Machine Learning*, PMLR, 2019, pp. 344–353.

[44] A. Bellet, R. Guerraoui, M. Taziki, and M. Tommasi, "Personalized and private peer-to-peer machine learning," in *International conference on artificial intelligence and statistics*, PMLR, 2018, pp. 473–481.

[45] C. Dwork, A. Roth, et al., "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[46] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *Advances in neural information processing systems*, vol. 30, 2017.

[47] P. Yao, D. Zhang, M. Guo, and X. Shao, "Hegd-fl: A privacy-preserving decentralized federated learning framework based on homomorphic encryption," in *2024 IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA)*, IEEE, 2024, pp. 26–33.

[48] G. K. Mahato, A. Banerjee, S. K. Chakraborty, and X.-Z. Gao, "Privacy preserving verifiable federated learning scheme using blockchain and homomorphic encryption," *Applied Soft Computing*, vol. 167, p. 112 405, 2024.

[49] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[50] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," *Advances in neural information processing systems*, vol. 30, 2017.

[51] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, PMLR, 2016, pp. 1050–1059.

[52] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" *Advances in neural information processing systems*, vol. 30, 2017.

[53] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012, vol. 118.

[54] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *Advances in neural information processing systems*, vol. 30, 2017.

[55] C. Leibig, V. Allken, M. S. Ayhan, P. Berens, and S. Wahl, "Leveraging uncertainty information from deep neural networks for disease detection," *Scientific reports*, vol. 7, no. 1, pp. 1–14, 2017.

[56] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in ai safety," *arXiv preprint arXiv:1606.06565*, 2016.

[57] E. Cyffers, A. Bellet, and J. Upadhyay, "Differentially private decentralized learning with random walks," *arXiv preprint arXiv:2402.07471*, 2024.

[58] C. Ji, S. Maag, R. Heusdens, and Q. Li, "Re-evaluating privacy in centralized and decentralized learning: An information-theoretical and empirical study," *arXiv preprint arXiv:2409.14261*, 2024.

[59] L. Demelius, R. Kern, and A. Trügler, "Recent advances of differential privacy in centralized deep learning: A systematic survey," *arXiv preprint arXiv:2309.16398*, 2023.

[60] H.-P. Cheng et al., "Towards decentralized deep learning with differential privacy," in *International Conference on Cloud Computing*, Springer, 2019, pp. 130–145.

[61] S. Maddock, G. Cormode, T. Wang, C. Maple, and S. Jha, "Federated boosted decision trees with differential privacy," in *Proceedings of the 2022 ACM SIGSAC conference on computer and communications security*, 2022, pp. 2249–2263.

[62] Z. Li, B. Ding, C. Zhang, N. Li, and J. Zhou, "Federated matrix factorization with privacy guarantee," *Proceedings of the VLDB Endowment*, vol. 15, no. 4, 2021.

[63] R. Xu, N. Baracaldo, and J. Joshi, "Privacy-preserving machine learning: Methods, challenges and directions," *arXiv preprint arXiv:2108.04417*, 2021.

[64] Y. Lu, Z. Yu, and N. Suri, "Privacy-preserving decentralized federated learning over time-varying communication graph," *ACM Transactions on Privacy and Security*, vol. 26, no. 3, pp. 1–39, 2023.

[65] Z. Xu et al., "Federated learning and analytics in practice: Algorithms, systems, applications, and opportunities," in *International Conference on Machine Learning*, 2023.

[66] J. Bayrooti, Z. Gao, and A. Prorok, "Differentially private decentralized deep learning with consensus algorithms," *arXiv preprint arXiv:2306.13892*, 2023.

[67] Y. Gao, L. Zhang, L. Wang, K.-K. R. Choo, and R. Zhang, "Privacy-preserving and reliable decentralized federated learning," *IEEE Transactions on Services Computing*, vol. 16, no. 4, pp. 2879–2891, 2023.

[68] E. T. M. Beltrán et al., "Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges," *IEEE Communications Surveys & Tutorials*, 2023.

[69] M. Talaei and I. Izadi, "Adaptive differential privacy in federated learning: A priority-based approach," *arXiv preprint arXiv:2401.02453*, 2024.

[70] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[71] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging," *SIAM journal on control and optimization*, vol. 48, no. 1, pp. 33–55, 2009.

[72] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, PMLR, 2017, pp. 1273–1282.

[73] K. Scaman, F. Bach, S. Bubeck, Y. T. Lee, and L. Massoulié, "Optimal algorithms for smooth and strongly convex distributed optimization in networks," in *international conference on machine learning*, PMLR, 2017, pp. 3027–3036.

[74] B. Zhao, K. R. Mopuri, and H. Bilen, "Idlg: Improved deep leakage from gradients," *arXiv preprint arXiv:2001.02610*, 2020.

[75] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *IEEE INFOCOM 2019-IEEE conference on computer communications*, IEEE, 2019, pp. 2512–2520.

[76] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *Advances in neural information processing systems*, vol. 32, 2019.

[77] D. Pasquini, M. Raynal, and C. Troncoso, "On the (in) security of peer-to-peer decentralized machine learning," in *2023 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2023, pp. 418–436.

[78] F. Boenisch, A. Dziedzic, R. Schuster, A. S. Shamsabadi, I. Shumailov, and N. Papernot, "When the curious abandon honesty: Federated learning is not private," in *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*, IEEE, 2023, pp. 175–199.

[79] Z. Huang, S. Mitra, and N. Vaidya, "Differentially private distributed optimization," in *Proceedings of the 16th International Conference on Distributed Computing and Networking*, 2015, pp. 1–10.

[80] X. Zhang, M. M. Khalili, and M. Liu, "Improving the privacy and accuracy of admm-based distributed algorithms," in *International conference on machine learning*, PMLR, 2018, pp. 5796–5805.

[81] Y. Yakimenka, C.-W. Weng, H.-Y. Lin, E. Rosnes, and J. Kliewer, "Straggler-resilient differentially-private decentralized learning," *IEEE Journal on Selected Areas in Information Theory*, 2024.

[82] E. Cyffers, M. Even, A. Bellet, and L. Massoulié, "Muffliato: Peer-to-peer privacy amplification for decentralized optimization and averaging," *Advances in Neural Information Processing Systems*, vol. 35, pp. 15 889–15 902, 2022.

[83] N. E. Manitara and C. N. Hadjicostis, "Privacy-preserving asymptotic average consensus," in *2013 European Control Conference (ECC)*, IEEE, 2013, pp. 760–765.

[84] Y. Mo and R. M. Murray, "Privacy preserving average consensus," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 753–765, 2016.

[85] P. Dellenbach, A. Bellet, and J. Ramon, "Hiding in the crowd: A massively distributed algorithm for private averaging with malicious adversaries," *arXiv preprint arXiv:1803.09984*, 2018.

[86] Y. Wang and A. Nedić, "Tailoring gradient methods for differentially private distributed optimization," *IEEE Transactions on Automatic Control*, vol. 69, no. 2, pp. 872–887, 2023.

[87] F. W. Dekker, Z. Erkin, and M. Conti, "Topology-based reconstruction prevention for decentralised learning," *arXiv preprint arXiv:2312.05248*, 2023.

[88] B. Noble, J. W. Daniel, et al., *Applied linear algebra*. Prentice-Hall Englewood Cliffs, NJ, 1977, vol. 477.

[89] A. E. Mrini, E. Cyffers, and A. Bellet, "Privacy attacks in decentralized learning," *arXiv preprint arXiv:2402.10001*, 2024.

[90] J. Leskovec and J. Mcauley, "Learning to discover social circles in ego networks," *Advances in neural information processing systems*, vol. 25, 2012.

[91] A. Krizhevsky, G. Hinton, et al., "Learning multiple layers of features from tiny images," 2009.

[92] S. Biswas, "Understanding and optimizing the trade-off between privacy and utility from a foundational perspective," Ph.D. dissertation, Ecole Polytechnique (EDX), 2023.

[93] R. L. Breiger and P. E. Pattison, "Cumulated social roles: The duality of persons and their algebras," *Social networks*, vol. 8, no. 3, pp. 215–256, 1986.

[94] C. Dwork, "Differential privacy," in *International colloquium on automata, languages, and programming*, Springer, 2006, pp. 1–12.

[95] D. Dua, C. Graff, et al., "Uci machine learning repository," 2017.

[96] G Hébrail and A Bérard, "Individual household electric power consumption data set, 2012. url https://archive. ics. uci. edu/ml/datasets/individual+ household+ electric+ power+ consumption," *Online: UCI Machine Learning Repository*,

[97] I.-C. Yeh and T.-K. Hsu, "Building real estate valuation models with comparative approach through case-based reasoning," *Applied Soft Computing*, vol. 65, pp. 260–271, 2018.

[98] W. N. Street, W. H. Wolberg, and O. L. Mangasarian, "Nuclear feature extraction for breast tumor diagnosis," in *Biomedical image processing and biomedical visualization*, SPIE, vol. 1905, 1993, pp. 861–870.

[99] B. Zhang, W. Sui, Z. Huang, M. Li, and M. Qi, "Normalizing flow based uncertainty estimation for deep regression analysis," *Neurocomputing*, vol. 585, p. 127 645, 2024.

[100] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *International conference on machine learning*, PMLR, 2015, pp. 1530–1538.

[101] I. Kobyzev, S. J. Prince, and M. A. Brubaker, "Normalizing flows: An introduction and review of current methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 11, pp. 3964–3979, 2020.

[102] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, "Normalizing flows for probabilistic modeling and inference," *Journal of Machine Learning Research*, vol. 22, no. 57, pp. 1–64, 2021.

[103] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," *Advances in neural information processing systems*, vol. 31, 2018.

[104] M. Gales and A. Malinin, "Reverse kl-divergence training of prior networks: Improved uncertainty and adversarial robustness," 2019.

[105] M. Biloš, B. Charpentier, and S. Günnemann, "Uncertainty on asynchronous time event prediction," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[106] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," *Advances in neural information processing systems*, vol. 31, 2018.

[107] A. Amini, W. Schwarting, A. Soleimany, and D. Rus, "Deep evidential regression," *Advances in neural information processing systems*, vol. 33, pp. 14 927–14 937, 2020.

[108] G. Parisi and R. Shankar, "Statistical field theory," 1988.

[109] M. Jordan, *The exponential family: Conjugate priors*, 2009.

[110] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," *arXiv preprint arXiv:1605.08803*, 2016.

[111] G. Papamakarios, T. Pavlakou, and I. Murray, "Masked autoregressive flow for density estimation," *Advances in neural information processing systems*, vol. 30, 2017.

[112] C.-W. Huang, D. Krueger, A. Lacoste, and A. Courville, "Neural autoregressive flows," in *International conference on machine learning*, PMLR, 2018, pp. 2078–2087.

[113] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, "Improved variational inference with inverse autoregressive flow," *Advances in neural information processing systems*, vol. 29, 2016.

[114] J. M. Hernández-Lobato and R. Adams, "Probabilistic backpropagation for scalable learning of bayesian neural networks," in *International conference on machine learning*, PMLR, 2015, pp. 1861–1869.

[115] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*, Springer, 2012, pp. 746–760.

[116] O. Ronneberger, P. Fischer, and T. 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*, Springer, 2015, pp. 234–241.

[117] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," *Advances in neural information processing systems*, vol. 31, 2018.

[118] A. Amini, A. Soleimany, S. Karaman, and D. Rus, "Spatial uncertainty sampling for end-to-end control," *arXiv preprint arXiv:1805.04829*, 2018.

[119] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 648–656.

[120] V. Kuleshov, N. Fenner, and S. Ermon, "Accurate uncertainties for deep learning using calibrated regression," in *International conference on machine learning*, PMLR, 2018, pp. 2796–2804.

[121] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[122] X. Huang et al., "The apolloscape dataset for autonomous driving," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 954–960.

[123] D. Saxena and J. Cao, "Generative adversarial networks (gans) challenges, solutions, and future directions," *ACM Computing Surveys (CSUR)*, vol. 54, no. 3, pp. 1–42, 2021.

[124] L. Girin, S. Leglaive, X. Bie, J. Diard, T. Hueber, and X. Alameda-Pineda, "Dynamical variational autoencoders: A comprehensive review," *arXiv preprint arXiv:2008.12595*, 2020.

[125] B. Niu, Y. Chen, B. Wang, Z. Wang, F. Li, and J. Cao, "Adapdp: Adaptive personalized differential privacy," in *IEEE INFOCOM 2021-IEEE conference on computer communications*, IEEE, 2021, pp. 1–10.

[126] M. Rahaman, V. Arya, S. M. Orozco, and P. Pappachan, "Secure multiparty computation (smpc) protocols and privacy," in *Innovations in Modern Cryptography*, IGI Global, 2024, pp. 190–214.

[127] P. Tam, I. Song, S. Kang, S. Ros, and S. Kim, "Graph neural networks for intelligent modelling in network management and orchestration: A survey on communications," *Electronics*, vol. 11, no. 20, p. 3371, 2022.

[128] H. Wang and D.-Y. Yeung, "A survey on bayesian deep learning," *ACM computing surveys (csur)*, vol. 53, no. 5, pp. 1–37, 2020.

[129] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," *Journal of Machine Learning Research*, vol. 20, no. 55, pp. 1–21, 2019.

[130] E. Moore, A. Imteaj, S. Rezapour, and M. H. Amini, "A survey on secure and private federated learning using blockchain: Theory and application in resource-constrained computing," *IEEE Internet of Things Journal*, 2023.

[131] S. Warnat-Herresthal et al., "Swarm learning for decentralized and confidential clinical machine learning," *Nature*, vol. 594, no. 7862, pp. 265–270, 2021.

[132] J. B. Bernabe, J. L. Canovas, J. L. Hernandez-Ramos, R. T. Moreno, and A. Skarmeta, "Privacy-preserving solutions for blockchain: Review and challenges," *Ieee Access*, vol. 7, pp. 164 908–164 940, 2019.

[133] B. Chen, C. Hawkins, K. Yazdani, and M. Hale, "Edge differential privacy for algebraic connectivity of graphs," in *2021 60th IEEE Conference on Decision and Control (CDC)*, IEEE, 2021, pp. 2764–2769.

Appendix A

Supplementary information regarding decentralization attacks

A.1 Analysis of the public knowledge assumption regarding the gossip matrix

In this study, we assume that attackers have knowledge of both the network graph and the gossip matrix. While these pieces of information may not always be entirely available in all use cases, we consider this assumption justified for the following reasons:

In general, it seems unreliable to assume that the network graph and gossip matrix W can be entirely concealed from the attacker nodes. Since attackers are part of the learning process, they must at least be aware of the data corresponding to their own row in the matrix. With this information, if there are enough attacker nodes, they could potentially deduce or infer a significant portion of the graph. Specifically, they can take advantage of the fact that W must be doubly stochastic. Moreover, it is challenging to maintain the privacy of a graph while releasing fundamental statistics, such as the spectral gap, the number of edges, triangles, or the degree distribution. For instance, Chen [133] provide an example where a node can deduce the edges of the

graph based on its own connections and the spectral gap; however, the spectral gap is often used to estimate how many gossip steps are required to reach a given level of precision. Therefore, considering the difficulty of accurately assessing the risk of graph reconstruction, we find it reasonable to assume that both the graph and the matrix W are public knowledge. This aligns with established research on differential privacy in decentralized learning, where it is commonly assumed that adversaries have access to both the network graph and the matrix W (see [82] for reference).

In many real-world contexts, the topology of the network is either publicly accessible or can be inferred from publicly available data. This applies to cases where nodes represent hospitals in collaborations across universities (as these are usually public knowledge), or financial institutions (for example, the SEC mandates the disclosure of financial connections), and in computations over social network graphs like Mastodon, or when learning occurs in blockchain systems or distributed ledger environments.