A Systematic Review of Privacy-Preserving Approaches in Deep Learning Applications

MARTIN NOCKER*, Institute of Computer Science, University of Rostock, Germany and Department of Medical, Health & Sports Engineering, MCI The Entrepreneurial School, Austria
MARTIN BECKER, Department of Mathematics and Computer Science, Marburg University, Germany and Hessian.AI, Darmstadt University of Technology, Germany
OLAF WOLKENHAUER, Institute of Computer Science, University of Rostock, Germany, Leibniz Institute for Food Systems Biology, Technical University of Munich, Germany, and Stellenbosch Institute for Advanced Study, Stellenbosch University, South Africa
PASCAL SCHÖTTLE, Josef Ressel Centre for Security Analysis of IoT Devices, MCI The Entrepreneurial School, Austria

As machine learning becomes increasingly important in areas that handle sensitive data, preserving data privacy is crucial. We present a comprehensive systematic literature review of privacy-preserving deep learning applications. We categorize research papers in this field from 2017 to May 2024, distinguishing between inference and training, and considering single and multiple data owners. Our taxonomy offers a structured approach to understanding state-of-the-art methods and their suitability for various applications. We further provide real-world examples of each scenario to illustrate their practical relevance and discuss challenges and limitations. In addition to reviewing the existing research literature on privacy-preserving deep learning, we identify gaps in current knowledge and suggest future research directions. Our findings show that execution times are steadily decreasing, making privacy-preserving deep learning increasingly practical. Defending models against privacy attacks still poses challenges and often leads to decreased utility, e.g., in terms of accuracy, one of many trade-offs discussed in this paper. By systematizing a vast body of existing work and providing practical guidelines for technique selection and deployment, our paper serves as a valuable resource for experts, practitioners, and newcomers alike, facilitating cross-disciplinary collaboration and advancements in privacy-preserving deep learning.

JAIR Track: Surveys

JAIR Associate Editor: Ferdinando Fioretto

JAIR Reference Format:
Martin Nocker, Martin Becker, Olaf Wolkenhauer, and Pascal Schöttle. 2026. A Systematic Review of Privacy-Preserving Approaches in Deep Learning Applications. *Journal of Artificial Intelligence Research* 87, Article 23 (November 2026), 53 pages. DOI: 10.1613/jair.1.21094

*Corresponding Author.

Authors' Contact Information: Martin Nocker, ORCID: 0000-0002-6967-8800, martin.nocker@uni-rostock.de, Institute of Computer Science, University of Rostock, Rostock, Germany and Department of Medical, Health & Sports Engineering, MCI The Entrepreneurial School, Innsbruck, Austria; Martin Becker, ORCID: 0000-0003-4296-3481, martin.becker@uni-rostock.de, Department of Mathematics and Computer Science, Marburg University, Marburg, Germany and Hessian.AI, Darmstadt University of Technology, Darmstadt, Germany; Olaf Wolkenhauer, ORCID: 0000-0001-6105-2937, olaf.wolkenhauer@uni-rostock.de, Institute of Computer Science, University of Rostock, Rostock, Germany and Leibniz Institute for Food Systems Biology, Technical University of Munich, Freising, Germany and Stellenbosch Institute for Advanced Study, Stellenbosch University, Stellenbosch, South Africa; Pascal Schöttle, ORCID: 0000-0001-8710-9188, pascal.schoettle@mci.edu, Josef Ressel Centre for Security Analysis of IoT Devices, MCI The Entrepreneurial School, Innsbruck, Austria.



This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2026 Copyright held by the owner/author(s).
DOI: 10.1613/jair.1.21094

1 Introduction

Machine learning (ML) and particularly deep learning (DL) have revolutionized data processing and analysis (Bachute and Subhedar 2021; Voulodimos et al. 2018). As DL applications increasingly utilize sensitive information, ensuring data and model privacy during training and inference becomes crucial in domains like medicine (Piccialli et al. 2021) or finance (Ahmed et al. 2022). This paper presents a comprehensive exploration of privacy-preserving techniques in DL through a systematic review of the research landscape.

As a foundation, we present a systematic framework of DL use cases focusing on preserving the privacy of involved parties, termed privacy-preserving deep learning (PPDL) (Shokri and Shmatikov 2015). We distinguish between inference and training tasks, with our taxonomy exhaustively covering all practically relevant combinations. For inference, we identify four categories (I1–I4) encompassing all combinations of single or multiple data owners and single or multiple model owners. For training, we consider all settings where data are held centrally or distributed and where the exchanged private information consists of either training data samples or model parameters (T1–T4). Every practical PPDL use case maps to one of our categories. Table 1 summarizes our PPDL taxonomy.

Here ”exhaustive“ refers to the coverage of use case scenarios rather than all possible data modalities. The majority of existing PPDL literature employs image data, reflecting the central role of benchmark datasets such as MNIST, CIFAR, and ImageNet in DL research. Our systematic review predominantly discusses results on image data, reflecting this uneven distribution in the surveyed literature. We highlight works evaluating other data modalities in dedicated subsections where available.

Our objective is to categorize research published between 2017 and May 2024 according to these use case scenarios, shedding light on strategies to protect sensitive data during inference and training. This enables researchers and practitioners to make informed decisions for their specific use cases. The taxonomy we present is not just a cataloging exercise, but serves as a structured approach to comprehend state-of-the-art methods and identify suitable techniques for specific applications.

This systematic approach enables us to identify trends in the literature, compare reported evaluations to discuss practical implications and relative advantages, and pinpoint areas requiring further exploration. We systematize PPDL work by practical use cases rather than technology-driven taxonomy, supported by real-world examples. This paper serves as a valuable resource for domain experts, practitioners, and newcomers, facilitating cross-disciplinary collaboration and advancement in PPDL.

We do not explicitly exclude trusted execution environments (TEE) but focus on work that utilizes homomorphic encryption (HE), secure multi-party computation (MPC), and/or differential privacy (DP). In this study, we analyze the literature to address the following research questions for each DL use case of our taxonomy: What technologies are deployed and how do they contribute to safeguarding the privacy of sensitive data? What is the current state-of-the-art in PPDL within the literature?

Our findings show that an increasing number of frameworks approach practical execution times and increasingly implement GPU support, further facilitating the practical deployment of PPDL. Privacy-preserving training has evolved from simple, fully connected neural networks to complex DL models like AlexNet and VGG-16. However, PPDL inherently involves fundamental trade-offs between privacy protection, computational efficiency, and model utility. Stronger privacy guarantees typically require greater computational overhead or result in reduced model accuracy. These privacy-utility-efficiency trade-offs represent core design considerations that are analyzed throughout this paper.

The remainder of this paper is organized as follows: Section 2 provides a roadmap to help different audiences navigate the paper effectively. Section 3 outlines the systematic review methodology employed in this study. Section 4 reviews related work, followed by an exploration of essential background in Section 5, covering topics such as DL, privacy-preserving technologies, and security considerations. Sections 6 and 7 present a detailed

Table 1. Taxonomy of privacy-preserving deep learning (PPDL) use cases. PPDL applications are categorized based on whether they target inference or training, and whether the setting involves single or multiple data owners. Besides differentiating between single or multiple data owners, inference categories (I1–I4) distinguish between single (I1 & I2) or multiple model owners (I3 & I4), while training categories (T1–T4) differentiate between scenarios where training data samples are exchanged (T2 & T4) or model parameters are exchanged (T1 & T3). Each use case is examined in detail in its respective section. This taxonomy enables efficient identification of suitable privacy-preserving techniques for specific applications.

	Single Data Owner	Multiple Data Owners
Inference	**I1:** Deep-Learning-as-a-Service (1 : 1 setting) (Section 6.1)	**I2:** Multiple (n) Data Owners (n : 1 setting) (Section 6.2)
	I3: Multiple (m) Model Owners (1 : m setting) (Section 6.3)	**I4:** Multiple Model & Data Owners (n : m setting) (Section 6.4)
Training	**T1:** Local Training – Secure Deployment (Section 7.1)	**T3:** Collaborative Learning (Section 7.3)
	T2: Training-as-a-Service (Section 7.2)	**T4:** Commissioned Training (Section 7.4)

analysis of the taxonomy of PPDL use cases for inference and training, respectively. Individual use cases are thoroughly discussed, highlighting key challenges, methodologies, and relevant literature. Section 8 discusses our findings and open research questions, while Section 9 concludes the paper.

2 Roadmap for Readers

This paper is intended to serve a diverse audience, including newcomers, practitioners, and PPDL domain experts. To help readers navigate this paper effectively, we provide the following guide:

Newcomers to the field of PPDL may want to start with the background information provided in Section 5 to build a solid foundation. Additionally, the overview and real-world examples of use cases in Sections 6 and 7 can help newcomers understand the practical applications of PPDL.

For newcomers aiming to become practitioners or domain experts, we recommend (1) a focused study of the background sections and follow-up readings, (2) hands-on experiments using public benchmarks and libraries, and (3) reproducing implementations from papers to gain practical experience.

Practitioners interested in applying privacy-preserving methods can focus on the threat models, implementation summaries, and concluding insights of each use case section (Sections 5–7). Each use case is structured to include real-world examples, threat models, privacy goals, and practical implementations. Pay special attention to the concluding summaries and open questions in each section. Practitioners should note that this paper focuses on technical deployment criteria. Compliance with applicable regulatory frameworks, such as GDPR (European Union 2016), HIPAA (United States Congress 1996), or the EU AI Act (European Union 2024), constitutes an additional, domain-specific layer of deployment considerations that is beyond the scope of this review.

Domain experts seeking technical depth will find the most value in the detailed discussion of protocols and frameworks in Sections 6 and 7. For those identifying gaps in the literature and exploring future research directions, the open questions at the end of each use case section and the discussion in Section 8 will be most relevant.

3 Review Methodology

We followed the two-step method of Webster and R. T. Watson (2002) for a systematic literature review. Initially, we identified relevant literature through a comprehensive database search using a combination of essential

keywords and filters. The results are complemented by a manual forward-backward search as a second step. The following query string was applied to the Scopus database[1] to retrieve our set of papers:

```
TITLE-ABS-KEY ( ("privacy" OR "private") AND ( "inference" OR "training" ) AND ( "deep
     learning" OR "neural network" ) AND ( "homomorphic encryption" OR "differential
    privacy" OR "multiparty computation" OR "multi-party computation" OR "two-party
    computation" ) AND ( "implement" OR "implemented" OR "implementation" OR "
    evaluate" OR "evaluation" ) ) AND NOT TITLE ( "sok" OR "survey" OR "attack" OR
     "attacks" ) AND PUBYEAR > 2016 AND ( LIMIT-TO ( LANGUAGE, "English" ) )
```

We require the title, abstract, or keywords to contain `("privacy" OR "private")`, `("inference" OR "training") AND ("deep learning" OR "neural network")`, targeting DL use cases, privacy-preserving technologies `("homomorphic encryption" OR "differential privacy" OR "multiparty computation" OR "multi-party computation" OR "two-party computation")`, and `"implement" OR "implementation" OR "evaluate" OR "evaluation"`, seeking practical solutions over theoretical results.

We did not include `"federated learning"` in the search string, as federated learning (FL) is considered out of scope for this review: we target studies explicitly combining privacy-preserving cryptographic or statistical techniques (HE, MPC, DP) with deep learning, rather than architectural paradigms such as FL. Its inclusion would broaden the scope beyond what is manageable for a focused systematic review. FL-based papers combining one of the targeted privacy-preserving technologies are nonetheless captured through the existing query terms and the manual forward-backward search. We excluded titles containing `"sok" OR "survey" OR "attack" OR "attacks"` as these fall into related work. Papers centered on privacy attacks were excluded as we focus on privacy-protecting solutions. We constrained searches to English articles published between 2017 and the query date.

The query returned 326 unique publications as of May 2024. We evaluated abstracts and eliminated papers not matching our criteria for privacy-preserving technologies or DL use cases, examining the full text when necessary. This excluded 107 papers. The remaining publications were complemented with relevant papers from a manual forward-backward search. The final set of 291 papers was subject to a thorough analysis and categorization process based on our proposed taxonomy.

We do not discuss every publication for readability. Instead, we include papers that cover the main state-of-the-art, such that excluded papers do not include substantial additional insights. We provide a complete list of all papers assigned to their corresponding use case via GitHub[2].

Table 2 illustrates the number of papers per category and year resulting from our comprehensive literature search. Papers are categorized based on their main use case. Nevertheless, papers are discussed in multiple use case sections if applicable. Therefore, we provide a complete list of papers that address multiple use cases with our GitHub repository, exhibiting the papers' main use cases, and further addressed use cases. The numbers in Table 2 show that the number of publications within PPDL is increasing substantially over time. Notably, the majority of these papers are concentrated on three specific use cases: I1, T1, and T3. Meanwhile, moderate research activity is observed in categories I2, I3, and T2, whereas use cases I4 and T4 have received little to no research attention. These observations already highlight research gaps, which are discussed in more detail in the subsequent sections of this paper.

The structure of each use case section is as follows: We start with a use case description, outlining the participating parties, e.g., data owners and model owners, and their respective objectives. We provide dedicated threat model discussions covering threat model adaptations, privacy goals, and key vulnerabilities specific to each use case. Next, to demonstrate the applicability of our taxonomy, we provide a concrete real-world example

[1] Scopus: https://www.scopus.com (Accessed: 2026-07-17)
[2] https://github.com/MartinNoc/PPDL-Review

Table 2. Distribution of reviewed papers by use case category and year. Numbers of publications are presented for each inference use case (I1 – I4) and training use case (T1 – T4) from 2017 to 2024. There is a clear upward trend in research on PPDL over time, with a notable increase in publications. Particularly, use cases I1, T1, and T3 have attracted significant attention, while I4 and T4 show little to no research activity.

	2017	2018	2019	2020	2021	2022	2023	2024*	Total
Inference									
I1 Deep-Learning-as-a-Service	1	6	10	10	19	15	26	13	100
I2 Multiple Data Owners ($n:1$)	0	1	1	2	2	2	5	2	15
I3 Multiple Model Owners ($1:m$)	0	1	0	3	3	1	3	1	12
I4 $n:m$ Setting	0	0	0	0	0	0	0	0	0
Training									
T1 Secure Deployment	0	0	4	5	9	9	21	4	52
T2 Training-as-a-Service	1	1	3	4	1	2	4	0	16
T3 Collaborative Learning	3	4	3	10	13	21	25	13	92
T4 Commissioned Training	0	0	1	1	0	1	1	0	4
Total	5	13	22	35	47	51	85	33	291

∗ does not cover the entirety of 2024

for each use case. Subsequently, the main part of each use case section discusses the use case literature and introduces supplementary concepts as needed. Within subsections, we include practical guidance paragraphs that highlight key insights, trade-offs, and deployment considerations for practitioners. A concluding summary highlights which techniques are best suited to addressing specific PPDL challenges. We conclude each use case section by presenting open questions to guide future research.

4 Related Work

Research on PPDL has gained considerable attention in recent years, driven by the increasing use of sensitive data in DL applications. Several surveys, reviews, and systematization of knowledge (SoK) papers have examined PPDL, particularly focusing on the privacy-preserving technologies and privacy challenges associated with DL. Surveys summarize the existing literature, giving a broad overview of recent developments, while SoK papers organize and structure knowledge into frameworks and taxonomies at a higher level of abstraction.

Several SoK papers have contributed to systematizing PPDL. For example, L. K. Ng and Chow (2023) offer a detailed analysis of cryptographic methods for neural network computations, evaluating state-of-the-art frameworks from a technological perspective. Although their work provides an in-depth performance evaluation, it is largely confined to cryptographic approaches. Cabrero-Holgueras and Pastrana (2021) broaden the scope by categorizing privacy-preserving techniques at a high level, encompassing a wide range of methods, including both DP and cryptography. Other systematizations, such as Khan et al. (2024), focus on the training phase, particularly emphasizing the role of HE and MPC in protecting privacy. Papernot, McDaniel, A. Sinha, et al. (2018) offer a detailed systematization of threat models and privacy attacks, but discuss HE and DP only briefly. Similarly, Salem et al. (2023) approach the field from a novel perspective by introducing a game-based formalism to model privacy attacks, yet their focus remains on attacks rather than comprehensive privacy-preserving solutions. Azraoui et al. (2020) structure neural network computation based on cryptographic techniques, without considering DP.

All mentioned SoK papers use a technology-oriented structure, as opposed to our use case taxonomy of PPDL applications.

Several survey papers have also contributed to the field. Qin et al. (2024) present a thorough model of the ML process, considering the privacy implications of data collection and preprocessing steps, with an emphasis on cryptographic primitives. B. Liu et al. (2021) provide an extensive discussion of privacy attacks on models and training data, but overlook privacy concerns related to inference input data. Other surveys, such as Jegorova et al. (2023) and X. Liu, L. Xie, et al. (2021), offer only brief overviews of privacy-preserving techniques. X. Liu, L. Xie, et al. (2021) do not explore mixed-method approaches, e.g., integrating cryptography with DP. Boulemtafes et al. (2020) propose a leveled approach to categorize PPDL technologies, offering a taxonomy that shares similarities with ours. However, their framework places emphasis on the role of a central computation server, neglecting serverless approaches and failing to account for scenarios with multiple data owners during inference. From a deployment-environment perspective, Nocker, Merkle, et al. (2026) survey PPDL methods for IoT systems, evaluating DL-as-a-Service and collaborative learning under resource-constrained conditions.

Our work differs from previous surveys and SoK papers in two key aspects: First, previous works have largely adopted a technology-centric perspective to categorize the literature. These efforts often organize research based on specific privacy-preserving techniques. While offering valuable technological insights, most works lack guidance for practical application scenarios. Our work diverges by introducing a comprehensive use case-driven taxonomy of DL applications, spanning both inference and training phases. This approach enables researchers and practitioners to efficiently identify suitable privacy-preserving techniques tailored to their specific needs while understanding the inherent challenges of each use case.

Second, most works focus on a subset of privacy-preserving technologies, particularly cryptographic methods, without fully addressing the utility of DP or the need for hybrid approaches. Cryptographic techniques such as HE and MPC are commonly explored, but their integration with DP, which is essential for practical, balanced privacy-utility trade-offs, is frequently underrepresented. Our work seeks to fill this gap by considering a wider range of approaches, including cryptographic techniques, DP, and hybrid solutions that integrate multiple methodologies. This comprehensive perspective provides a better understanding of the trade-offs between privacy, utility, and computational efficiency in PPDL.

5 Preliminaries

This section provides the foundational concepts necessary for discussing privacy-preserving deep learning. First, deep learning is introduced in Section 5.1, followed by discussions on various privacy-preserving techniques. In particular, homomorphic encryption is covered in Section 5.2, secure multi-party computation in Section 5.3, and differential privacy in Section 5.4. Each technique plays a fundamental role in securing data during inference and model training. A comparison between differential privacy and cryptographic methods, i.e., homomorphic encryption and secure multi-party computation, is provided in Section 5.5. Finally, Section 5.6 examines the DL scenarios considered, along with the associated threat model and key management strategies.

5.1 Deep Learning

Deep learning (DL) uses deep neural networks (NNs or DNNs), processing data through layers of artificial neurons (Goodfellow et al. 2016; Schmidhuber 2015). Each layer applies a linear or non-linear function, with outputs from one layer serving as the input for subsequent layers.

5.1.1 Types of Layers. Linear layers comprise fully connected (dense) layers and convolutional layers. Dense layers involve weights connecting each input to every output neuron in the next layer. Convolutional layers, often used in image processing, use kernels with learnable weights that capture local patterns in the input data. Such networks are termed convolutional NNs (CNNs) (Zeiler and Fergus 2014). Non-linear layers frequently follow

linear layers. Pooling layers reduce spatial data, with maxpool selecting the maximum value in a local region and meanpool computing the average. Sigmoid and ReLU are popular so-called activation functions, introducing non-linearities necessary for complex problems.

Training adjusts the weights denoted as θ to align the model's predictions with the desired output for a set of training samples. This is measured with a loss function $\mathcal{L}(\theta)$ that quantifies prediction errors for parameters θ. For complex networks with non-convex loss functions, mini-batch stochastic gradient descent (SGD) is utilized. Here, at each step, a random batch of examples is used to estimate the gradient of the loss concerning θ, followed by updating θ in the direction of the negative gradient, aiming for a local minimum. This procedure is called backpropagation.

5.1.2 DNN Architectures. DNN architectures exhibit varying complexities, often defined by the number of layers and parameters. Popular CNNs used in benchmarks include LeNet, AlexNet, SqueezeNet, VGG, and ResNet. These architectures were primarily developed and optimized for image classification tasks, reflecting the dominant role of computer vision in DL research.

LeNet is a relatively simple network with about 60k parameters made up of two convolutional layers with sigmoid activation, each followed by a maxpool layer, and three fully connected layers at the end (LeCun, Bottou, et al. 2002). AlexNet is composed of five convolutional layers with ReLU activation, three fully connected layers, and three intermediate maxpool layers, totaling approximately 62M parameters (Krizhevsky et al. 2012). SqueezeNet consists of convolution layers, maxpool layers and fire modules, which contain 1×1 and 3×3 convolutions. SqueezeNet was designed to be compact, having only 1.2M parameters (Iandola et al. 2016). VGG-l and ResNet-l come in various sizes, where l denotes the total number of convolutional and dense layers, e.g., ResNet-18 exhibits 17 convolutional layers and one fully connected layer. Standard sizes l are 11, 13, 16, and 19 for VGG and 18, 34, 50, 101, and 152 for ResNet. ResNet includes "residual blocks" which skip layers to allow gradients to flow directly through the layers (He et al. 2016; Simonyan and Zisserman 2015).

5.1.3 Benchmark Datasets. Benchmark datasets validate DNN performance in terms of classification accuracy, latency, and throughput. The prevalence of image-based datasets in DL benchmarking has shaped both architectural development and research focus within the field. Among the most widely recognized datasets, MNIST contains 28×28 grayscale images for handwritten digit recognition (LeCun, Cortes, et al. 2010). FashionMNIST features 28×28 grayscale images of fashion items in ten classes (Xiao, Rasul, et al. 2017). CIFAR-10 and CIFAR-100 contain 32×32 color images for object classification across ten and 100 classes, respectively (Krizhevsky 2009). ImageNet comprises over a million images spanning a thousand object categories for large-scale image classification (Deng et al. 2009). SVHN (Street View House Numbers) contains house numbers from Google Street View images (Netzer et al. 2011).

5.2 Homomorphic Encryption

Homomorphic encryption (HE) allows for computations on encrypted data without decryption. Additive HE schemes, e.g., Paillier (1999), support additions only, i.e., $\mathbb{D}_{\text{sk}}(\mathbb{E}_{\text{pk}}(x_1) \oplus \mathbb{E}_{\text{pk}}(x_2)) = x_1 + x_2$, where $+$ and $\oplus$ are plaintext and ciphertext addition, respectively, and $\mathbb{E}$ and $\mathbb{D}$ are the encryption and decryption functions using the public key pk and secret key sk, respectively. We abbreviate $\mathbb{E}_{\text{pk}}(x)$ by $[x]_{\text{pk}}$ and omit the key pk unless necessary. ElGamal (1985) allows for multiplications only. Leveled HE (LHE) schemes support a limited number of multiplications and unlimited additions. Lastly, fully HE (FHE) schemes support arbitrary addition and multiplication operations. HE adds substantial memory and runtime overhead compared to cleartext operations.

The security of modern HE schemes is based on the ring-learning-with-errors (RLWE) hardness assumption (Lyubashevsky et al. 2010), i.e., noise is added during encryption to the ciphertext. Correct decryption relies on keeping the noise level below a specific threshold. Additions minimally affect the noise level, but multiplications

add a substantial amount of noise, hence, the applicability of HE is determined by the multiplicative depth. The bootstrapping operation is used to reduce the noise level of a ciphertext but is computationally expensive (Gentry 2009).

As HE schemes only support additions and multiplications, performing PPDL is a non-trivial task using HE. Non-linear layers like activation functions must be approximated using polynomials, preferably of low degree, to minimize the multiplicative depth. The same holds for the computation of cost functions and gradient calculations.

LHE schemes like BFV (Fan and Vercauteren 2012) and CKKS (Cheon et al. 2017) use batching, which packs a vector of complex numbers into a single ciphertext. This enables elementwise operations on all encrypted values at once for more efficient implementations, particularly of linear layers. While computational performance decreases with increasing vector sizes (Laine 2017), batching is still a powerful technique to increase efficiency. TFHE (Chillotti, Gama, et al. 2020) is an FHE scheme built for Boolean gates with fast bootstrapping. However, TFHE does not support batching. Subsequent extensions enable support for arithmetic circuits and introduce programmable bootstrapping (Chillotti, Joye, et al. 2021), which acts as a lookup table, i.e., ciphertexts can be evaluated by arbitrary univariate functions during bootstrapping, e.g., non-linear activations.

5.3 Secure Multi-Party Computation

Secure multi-party computation (MPC) enables multiple parties to compute functions over private inputs without revealing inputs to each other. MPC uses cryptographic primitives, communication rounds, and preprocessing phases, with communication as the main bottleneck.

Additive **secret sharing (SS)** splits a secret $x \in \mathbb{Z}_p$ into n shares within the ring $\mathbb{Z}_p$, i.e., the integers mod p. For example, for $n=2$ and a random number from the ring, $r \in \mathbb{Z}_p$, the two shares are $\langle x \rangle_0 = r$ and $\langle x \rangle_1 = x - r \mod p$. Additive SS perfectly hides the original message x. Additions between secrets and multiplications with constants can be done individually by the shareholders. Multiplying two secret shares, i.e., $\langle x \cdot y \rangle$ with $\langle x \rangle_i$ and $\langle y \rangle_i$ at party i requires one round of communication and Beaver triplets (Beaver 1992) which need to be precomputed in the preprocessing phase, which also includes other preparations to speed up the online computation.

Garbled circuits (GC) (Yao 1986) are a two-party protocol to evaluate a function on the parties' private inputs with constant communication rounds. The function is represented as a Boolean circuit, which grows exponentially with the computational complexity. GC is less efficient than additive SS for addition and multiplication, but performs better for logical operations like AND and XOR.

Oblivious transfer (OT) is a building block for MPC protocols and is used in GC and for generating Beaver triplets. A receiver obliviously selects a message x_i from the set of k messages $x_1, \ldots, x_k$ while hiding i from the sender. Oblivious table lookups are realized using OT with $O(k)$ communication complexity (Dessouky et al. 2017).

HE serves as a primitive for MPC. Parties encrypt their inputs using an HE scheme before computing on the ciphertexts. The result is obtained through a distributed decryption protocol. The decryption key is split among the parties, which requires collaboration for successful decryption (Damgård et al. 2012).

In the context of PPDL, the primary focus lies in improving the efficiency of MPC protocols for neural network operations. This involves optimizing communication complexity, reducing the number of communication rounds, and developing specialized protocols for common DL operations such as matrix multiplications, convolutions, and non-linear activations. The architecture of the DL model is disclosed in MPC as the network structure must be known to all parties to execute the protocol, however, the model weights remain private inputs.

5.4 Differential Privacy

Differential privacy (DP) is a formalism to quantify the degree to which the privacy of each individual in a dataset is preserved while releasing the output of a data analysis algorithm (Dwork, Kenthapadi, et al. 2006; Dwork,

McSherry, et al. 2006). It guarantees that, regardless of side knowledge, an adversary's knowledge gain is the same whether the individual's data were included in the database. The definition uses the concept of adjacent databases, which are two datasets that differ by exactly one entry. In the following, a mechanism $\mathcal{M}$ is used to denote a randomized algorithm that outputs information about a sensitive dataset. A mechanism $\mathcal{M}$ satisfies (ϵ, δ)-DP if for any pair of adjacent datasets d, d' and for any set of outputs S in the range of $\mathcal{M}$, it holds that

$$\Pr[\mathcal{M}(d) \in S] \leq e^{\epsilon} \Pr[\mathcal{M}(d') \in S] + \delta\,. \tag{1}$$

If $\delta = 0$, it is simply called ϵ-DP. Lower values of ϵ mean stronger privacy. The parameter δ bounds the probability that the privacy guarantee expressed by ϵ may be violated., i.e., smaller δ values indicate stronger privacy. The total amount of privacy loss allowed when performing a series of DP operations is called the privacy budget. When multiple DP mechanisms are applied sequentially, the privacy parameters accumulate, and the privacy budget constrains the total ϵ (and δ) that can be consumed. The randomization of algorithms is predominantly achieved by adding noise during the computation process. Given a function f, the noise magnitude to satisfy ϵ-DP can be calibrated using the sensitivity of f, which is defined as the maximal absolute distance $S_f = |f(d) - f(d')|$ for any two adjacent databases d and d'. For simplicity, we consider a real-valued function f. However, the noise addition reduces the accuracy of the algorithm's results. Therefore, the design of differentially private mechanisms that achieve a good trade-off between privacy protection and result accuracy is crucial (Balle and Y.-X. Wang 2018; Geng and Viswanath 2016; M. Kumar, Rossbory, et al. 2019).

Regarding DL, randomization is carried out via either adding random noise to the inputs or outputs or modifying the DL algorithm itself. DP preserves the privacy of the training dataset to ensure that an adversary cannot infer any single data instance by observing model parameters or outputs. Differentially private stochastic gradient descent (DP-SGD) is a widely used gradient perturbation technique to train a DNN that is robust against privacy attacks, as noise is added to the gradients during training (Abadi et al. 2016). To satisfy DP, the noise has to be large enough to mask the effect of any possible record in a mini-batch. Without any noise limitation, this could change a gradient by a large amount, resulting in a useless model. Therefore, gradients are clipped to a maximum magnitude, which is computationally expensive for individual gradients and cancels the advantage of parallel GPU computation. Goodfellow (2015) presents a fast per-example gradient clipping method, however, only for fully connected NNs. More recent approaches support various layers (J. Lee and Kifer 2021) or improve the computation of per-sample gradient norms (Bu et al. 2021; H. Liu, C. Li, et al. 2022; Subramani et al. 2021).

Local vs. Global DP: Local differential privacy (LDP) applies noise directly at the data source before transmission, while global DP (GDP) adds noise to aggregated results at a trusted central curator. LDP provides stronger privacy guarantees by eliminating trust requirements, but typically achieves lower utility.

Input vs. Output Privacy: Input privacy protects sensitive data during computation (e.g., inference inputs), while output privacy prevents inference about training data from model outputs or parameters.

Synthetic Data Generation: An important application of DP in machine learning involves generating synthetic datasets that preserve statistical properties while protecting individual privacy. Synthetic data generation typically employs DP mechanisms during the generation process to ensure privacy guarantees (Jordon et al. 2018). By training generative models with DP-SGD or using DP mechanisms to post-process synthetic samples, these approaches can create datasets that reduce direct exposure of sensitive training data while maintaining utility for downstream machine learning tasks. The effectiveness of differentially private synthetic data generation depends on the quality of the underlying generative model and the privacy budget allocation during the generation process.

5.5 Differential Privacy vs. Cryptographic Methods

DP and cryptographic techniques (HE and MPC) both aim to preserve privacy, but with distinct protection goals and capabilities.

Cryptographic techniques provide input privacy by protecting data during computations among multiple parties, e.g., inference inputs or model parameters. Adversaries gain only minimal information about private inputs when intercepting (intermediate) results or collaborating with some parties (Gilad-Bachrach et al. 2016; Rouhani et al. 2018). While DP can also protect private inputs through local DP mechanisms, the necessary amount of noise to achieve privacy may distort the data excessively, making it unsuitable for safeguarding sensitive DL inputs. However, DP excels at providing output privacy for training DL models that do not reveal individual training data during subsequent inferences (Papernot, Thakurta, et al. 2021). Global DP enables centralized noise addition with better utility than local approaches when a trusted curator is available. Cryptographic techniques cannot address this concern, leaving models vulnerable to privacy attacks. However, implementing DP in practice can be challenging as the sensitivity of algorithms and the definitions of adjacency vary (J. Wang, J. Zhang, et al. 2018). Lastly, cryptographic techniques add substantial computational and memory overhead, compared to DP.

5.6 Security Considerations

We consider the following scenarios of PPDL. In the inference case, there are n inference data owners holding sensitive data x_i, and m model owners with a pretrained model M_j. The parties aim to obliviously computing $f(x)$, where $f(x) = M(x)$ in the simplest case of one model owner and one data owner, and an aggregation of $M_j(x_1, \ldots, x_n)_{1 \leq j \leq m}$ in the case of multiple data and model owners. During the computation, data of individual participants should not be disclosed to any other party. Scenarios, including multiple model owners, might also include a helping computing server that is not trusted.

In training, we consider n training data owners to train a DL model using their data. We also take external servers for training computation and coordination support into account, which manage communication and aggregate (intermediate) results. Privacy objectives depend on what private information is exchanged: either raw training data samples must be protected during transfer to external parties, or model parameters and updates must be safeguarded during collaborative computation. While models represent learned patterns from training data, the actual private information exposed and requiring protection differs between scenarios: direct data exposure in outsourced training versus model weight exposure in collaborative training.

We do not make any assumptions about how data is distributed among data owners.

5.6.1 Threat Model. In the context of PPDL, adversaries are typically categorized into passive (honest-but-curious) and active (malicious) adversaries (Attrapadung et al. 2022; Lehmkuhl et al. 2021; Mohassel and Y. Zhang 2017). Passive adversaries follow the protocol (honest) but try to extract as much sensitive information as possible (curious). Their capabilities include intercepting and analyzing transmitted data between parties and performing statistical analyses. Active adversaries may deviate from the protocol and exhibit more powerful capabilities. They can inject false or manipulated data, launch denial-of-service attacks, and introduce fake identities into the system to increase their influence within it. Preserving privacy against active adversaries is substantially harder.

Moreover, n-party protocols with t adversaries distinguish between an honest majority if $t < n/2$ and a dishonest majority otherwise. In contrast to costly schemes required for dishonest majorities (Keller and Sun 2022), honest majority settings allow for efficient protocols. The non-colluding assumption, frequently linked to helper servers, implies that parties do not reveal their secrets to any other party. Otherwise, colluding parties could combine their knowledge and reveal sensitive information, compromising the system's integrity (Kamara et al. 2011; Mohassel and Y. Zhang 2017).

Security attacks on ML systems aim to compromise the models' integrity, availability, or functionality, whereas privacy attacks try to uncover sensitive information in training data, model parameters, or inferences while retaining the model's functionality. Security attacks involve poisoning and evasion attacks (Biggio and Roli 2018). Poisoning injects manipulated samples into the training phase (Xiao, Biggio, et al. 2015). Evasion attacks craft adversarial examples to mislead classification (Szegedy et al. 2014). While attacks targeting the security

of ML models are important, we focus on privacy attacks in our systematic literature review, as defined in the work of Rigaki and Garcia (2023). Membership inference attacks try to reveal whether a specific sample was part of the training dataset (Shokri, Stronati, et al. 2017). Model extraction attacks, also model stealing, aim to obtain model parameters or the architecture (Papernot, McDaniel, Goodfellow, et al. 2017; Tramèr, F. Zhang, et al. 2016). Reconstruction attacks try to infer sensitive features about training samples (Fredrikson et al. 2015). Property inference attacks extract dataset properties that are not explicitly encoded as features, e.g., the gender distribution in a medical dataset (Ateniese et al. 2015). Adversaries' knowledge can be categorized as white-box, black-box, or gray-box, depending on their access to model internals and training data (Biggio and Roli 2018). In the white-box setting, the adversary has full knowledge of the model architecture, parameters, and often the training procedure. This enables the strongest privacy attacks, such as exact reconstruction of training data or highly accurate membership inference. In the black-box setting, the adversary has no knowledge of the internals and can only issue queries to the model. Privacy attacks in this case are limited to query-based strategies, such as black-box membership inference or model extraction via repeated querying. Gray-box settings represent everything in between these two extremes, where the adversary knows some but not all aspects of the system, e.g., the architecture and implementation details but not the trained parameters or the original data. This limits the attack surface: property inference and membership inference remain possible, but direct parameter inspection is not feasible. White-box settings provide a useful worst-case bound, while black- and gray-box settings more closely reflect practical deployment scenarios of PPDL.

5.6.2 Key Management. Implementations based on cryptographic methods, i.e., HE and MPC, often involve a key or secret setup phase. This is commonly simplified using a trusted key distribution authority (KDA) which provides the parties with the necessary keys, e.g., in the works of Lyu (2020) and X. Zhang, Ji, et al. (2017). This rather strong assumption can be difficult to establish in practice. However, it can be realized, e.g., using Group Diffie-Hellman protocol (Steiner et al. 1996). To mitigate the risk of man-in-the-middle attacks, secure communication channels must be used. Furthermore, such a single point of failure can be avoided by splitting the single KDA into multiple KDAs, each distributing a share of the secret. Combining the shares results in the secret. This setup is secure as long as not all KDAs collude.

The security level of cryptographic techniques is specified by λ, implying approximately 2^λ operations are required to break the encryption. For HE, λ is a function of the encryption parameters which affects the scheme's performance. NIST recommends a minimum of $\lambda = 128$ bits of security (Barker 2020).

6 Inference

This section covers the four inference use cases (I1–I4) in our PPDL taxonomy. We consider one or multiple data owners together with one or multiple model owners. The parties' objective is to evaluate trained models of model owners on the sensitive data of data owners. Depending on the underlying technology, this may be done for a single instance or for multiple instances in parallel. Several modern HE schemes support batching, where several plaintext samples can be packed into one ciphertext, enabling simultaneous inference on multiple inputs. By contrast, most MPC protocols operate on a per-instance basis, though parallelization across multiple sessions or vectorized extensions of the protocol are possible in principle. Sections 6.1 and 6.3 discuss use cases with a single data owner, whereas Sections 6.2 and 6.4 consider multiple data owners. Similarly, scenarios with a single model owner are discussed in Sections 6.1 and 6.2, multiple model owners are covered in Sections 6.3 and 6.4. A high-level visualization of all inference use cases is illustrated in Figure 1.

Threat Model: Inference scenarios assume honest-but-curious adversaries who follow protocols correctly but attempt to extract sensitive information from observed communications and intermediate results. Data owners seek to protect input privacy, meaning that sensitive inference data should not be disclosed. Model owners aim to

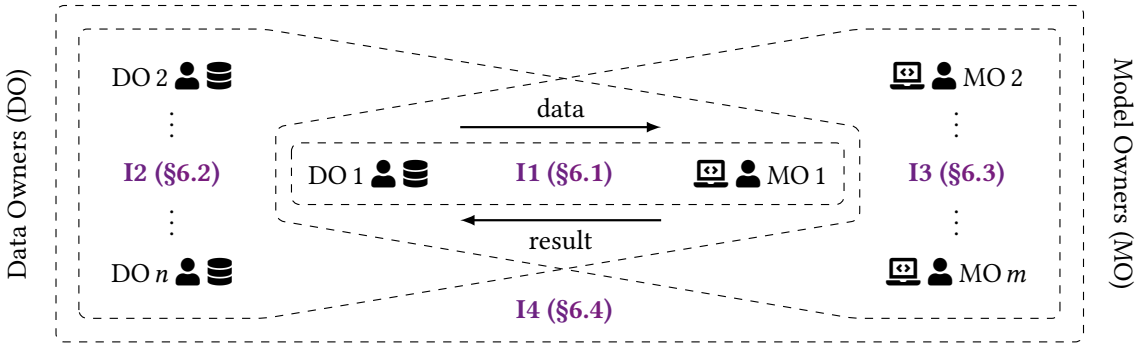


Fig. 1. Visualization of inference scenarios I1–I4, where data owners ( ) hold sensitive data and model owners ( ) own a pretrained model. The parties aim to obliviously evaluating models on the private inputs. I1 represents the simplest scenario, involving a single data owner and a single model owner, while I4 reflects a more complex setting with multiple data and model owners. Detailed analyses of each use case are provided in the respective sections of the paper.

preserve model privacy, meaning that the model architecture and parameters should remain confidential. External servers or aggregators, when present, are considered untrusted and may collude with a subset of participants.

Privacy Goals: Primary objectives include input privacy to prevent inference about sensitive input data, model privacy to protect proprietary model parameters and architectures, and output privacy to limit information leakage from inference results. Secondary goals involve minimizing communication overhead and maintaining computational efficiency while preserving privacy guarantees.

Vulnerabilities: Key attack vectors include input reconstruction, where adversaries attempt to recover sensitive input data from observed computations, output analysis attacks that infer private information from inference results, and communication interception that reveals data patterns. Additional threats include collusion between parties and side-channel attacks exploiting computational patterns, particularly in multi-party scenarios requiring trusted coordination.

6.1 I1: Deep-Learning-as-a-Service

This fundamental scenario involves a model owner (also called a server) and a single data owner (client). The client holds sensitive data denoted as x, while the model owner maintains the model M and offers inference services. When utilizing DL, this is referred to as deep-learning-as-a-service (DLaaS). The primary objective is to compute $M(x)$ for the client while protecting x and $M(x)$ from disclosure to the server. Training powerful models is not trivial and involves monetary costs. Consequently, these models are often considered intellectual property. Owners want to protect them from disclosure.

Threat Model Adaptations: I1 focuses on bilateral privacy between client and server, where both parties are honest-but-curious about model parameters and input data, respectively. Primary vulnerabilities include input reconstruction from computation patterns and model extraction through repeated queries.

Example: Cancer diagnostics – Secure and accurate tumor detection and classification are essential for patient treatment. DNNs have achieved state-of-the-art performance in such tasks, including breast cancer recurrence prediction (Son et al. 2021), melanoma occurrence detection (Jain et al. 2021), and analysis of high-dimensional genetic data (Hong et al. 2022). Protecting patient privacy is essential. Therefore, data must not be transmitted in plain text. Data owners should only receive their inference results, without any information about the model.

6.1.1 Homomorphic Encryption. HE emerges as a suitable solution for oblivious inference. Clients encrypt input data and send it to the server, which computes inference results on encrypted data using local unencrypted model weights. Most HE schemes support only additions and multiplications, making non-linear operations like ReLU, sigmoid, or maxpool challenging. This limitation drives two primary strategies: polynomial approximations and scheme-switching approaches.

Foundational approaches and polynomial approximations: CryptoNets (Gilad-Bachrach et al. 2016) established the first complete private NN inference using polynomial-only operations. By employing square functions as activations and a scaled meanpool for pooling, CryptoNets demonstrated feasibility but suffered from impractical latency (300 s for MNIST), revealing the need for HE-friendly architectural modifications. CryptoNets was evaluated on desktop hardware (Intel Xeon E5-1620, 3.5GHz, 16GB RAM). Building upon CryptoNets, LoLa (Brutzkus et al. 2019) and Nandakumar et al. (2019) achieved improvements through optimized ciphertext representations tailored for NN operations, demonstrating that computational efficiency can be enhanced through careful data layout optimization. Nandakumar et al. (2019) evaluated their approach on high-performance server hardware (dual-socket Xeon E5-2698 v3, 32 cores, 250GB RAM).

The evolution toward sophisticated polynomial approximations is exemplified by LHDNN (Qian et al. 2023), introducing LotHps activations based on weighted low-degree Hermite polynomials (Spanier et al. 1988, Chapter 24). This approach demonstrates competitive accuracy on medical datasets while highlighting the persistent challenge of balancing approximation quality with computational efficiency. J. Lee, E. Lee, et al. (2023) advance polynomial approximations using minimax techniques for ReLU and maxpool operations, illustrating the trade-off between approximation complexity and accuracy.

Noise management: A fundamental challenge emerges from noise accumulation in deep networks. While CryptoNets and LoLa demonstrate end-to-end inference on encrypted data without exposing inputs or model internals, noise growth limits feasible network depth, leading to interactive protocols. CryptoDL (Hesamifard et al. 2018) addresses this through intermediate decryption and re-encryption steps, trading computational efficiency for deeper network support. However, this introduces a privacy-efficiency trade-off: while maintaining input and parameter confidentiality, re-encryption frequency may indirectly leak architectural information. Evaluated on server-class hardware (12-core CPU, 48GB RAM), CryptoDL demonstrates the feasibility of interactive protocols for deeper networks.

The bootstrapping approach, exemplified by J.-W. Lee et al. (2022) using CKKS with bootstrapping for ResNet-20 on CIFAR-10, offers an alternative noise management strategy. While avoiding interactive protocols, bootstrapping incurs substantial computational overhead, making it suitable primarily for scenarios prioritizing non-interactive computation over efficiency. J.-W. Lee et al. (2022) demonstrated GPU acceleration (CUDA implementation) on server hardware (Intel Xeon E5-2630, 2.20GHz, 482GB RAM), achieving 2.2× speedup over CPU but requiring 120GB RAM and 5.04 days training time. AutoFHE (Ao and Boddeti 2024) advances this direction by optimizing bootstrapping placement through mixed-degree polynomial activations.

Scheme-switching approaches: Leveraging multiple HE schemes within a single computation can result in performance improvements. Glyph (Lou, B. Feng, et al. 2020) pioneered this approach by switching between TFHE for table lookups (enabling accurate sigmoid and softmax computation) and BGV for linear arithmetic. This hybrid strategy addresses single-scheme limitations: each HE scheme has optimal use cases, and combining them captures the benefits of both. T.-L. Liu et al. (2023) extend this concept by switching between CKKS and FHEW/TFHE, demonstrating that scheme-switching can be generalized beyond specific activation functions.

These approaches represent a paradigm shift from fitting all operations into a single HE scheme toward designing protocols that leverage the strengths of different schemes. The success suggests future HE-based inference systems will likely employ multi-scheme architectures rather than optimizing within single-scheme constraints.

Model optimization strategies: Recognizing HE computational limitations, researchers have developed model optimization techniques specifically for encrypted inference. Pruning and quantization emerge as critical strategies, with binarized neural networks (BNNs) representing an extreme but effective approach. J. Zhou et al. (2020) demonstrate that binarizing inputs and weights to $\{0, 1\}$ enables efficient bit-wise operations, while REDsec (Folkerts et al. 2023) explores ternary networks with weights constrained to $\{-1, 0, 1\}$. FHE-DiNN (Bourse et al. 2018) achieves latency improvements (under one second) by applying TFHE to BNNs, but supports only small networks with reduced accuracy. FHE-DiNN was evaluated on consumer-grade hardware (Intel Core i7-4720HQ, 2.60GHz, single core). This illustrates a critical trade-off: aggressive quantization can achieve practical latency but may limit model expressiveness. HE-PEx (Aharoni, Baruch, et al. 2023) and SpENCNN (Ran et al. 2023) attempt to balance this trade-off through pruning and packing techniques. S. Sinha et al. (2022) combine quantization with bitsliced data packing to reduce bootstrapping overhead.

Ishiyama et al. (2022) provide quantitative evidence for this trade-off, demonstrating that high-degree activation functions increase accuracy only marginally while substantially increasing latency. This suggests that for HE-based inference, simpler activation functions often provide an optimal balance between security, efficiency, and accuracy.

Algorithmic and hardware optimizations: Beyond cryptographic innovations, researchers have optimized fundamental operations within HE constraints. Balla and Koushanfar (2023) advance homomorphic matrix multiplication algorithms, while X. Liu, H. Li, et al. (2023) leverage the Winograd algorithm to reduce convolution complexity. These optimizations demonstrate that algorithmic improvements can complement cryptographic advances.

The integration of specialized hardware represents another optimization dimension. GPU acceleration (Folkerts et al. 2023) and FPGA implementations (X. Ren et al. 2023) show that HE computations benefit from parallel processing capabilities. This hardware-software co-design approach suggests practical HE-based inference systems will require optimization across multiple layers, from cryptographic protocols to hardware architecture.

Real-world applications: While most HE-based inference research evaluates on standard benchmarks (MNIST, CIFAR-10), several works demonstrate practical applicability. Medical applications show particular promise: Hong et al. (2022) apply shallow networks to cancer classification using The Cancer Genome Atlas (TCGA) dataset, achieving meaningful accuracy on genomic data from over 11 000 patients, while LHDNN (Qian et al. 2023) demonstrates skin cancer classification on medical imaging data. These medical applications highlight computational constraints limiting deployment to simple architectures. Beyond healthcare, p2Detect (L. Wu et al. 2023) demonstrates applicability to smart meter data for electricity theft detection, and Izabachène et al. (2019) implement face recognition using Hopfield networks, though 0.6 s inference time falls short of real-time requirements of 100 ms.

These applications reveal key challenges: datasets require extensive HE-compatible preprocessing, applications are limited to shallow networks due to computational overhead, and domain-specific accuracy requirements constrain acceptable privacy-utility trade-offs. Successful deployments occur where privacy benefits outweigh computational costs.

Practical implications: Single-scheme approaches face limitations, driving the development of hybrid strategies combining multiple HE schemes (e.g., Glyph) or model optimization techniques (e.g., BNNs). The most promising direction combines scheme-switching, strategic quantization, and hardware acceleration. For practitioners, the choice depends on specific requirements: single-scheme approaches suit simple models with moderate accuracy requirements, multi-scheme approaches benefit complex operations requiring high accuracy, and quantized approaches enable sub-second inference at the cost of model expressiveness. With current performance ranging from seconds (optimized BNNs) to minutes (full-precision networks), HE is well-suited to non-interactive, latency-tolerant applications where input confidentiality is paramount. It is unsuitable for real-time applications, large participant sets, or deployments where computational overhead cannot be justified.

6.1.2 Secure Multi-Party Computation. MPC protocols are more efficient with respect to memory and computational overhead compared to HE but require communication among parties. An important distinction is that parties involved in MPC protocols possess knowledge of the computation, i.e., the NN architecture. The standard MPC approach involves collaboration between client and server through the exchange of secret shares or the execution of garbled circuits. MPC solutions enhance the efficiency of runtime-critical subprotocols, such as matrix multiplication and non-linear functions. The primary bottleneck occurs during evaluation of non-linear layers, like ReLU activation functions (Diaa et al. 2024).

Foundational approaches: DeepSecure (Rouhani et al. 2018) established early MPC-based inference feasibility using pure GC, demonstrating applications on audio (ISOLET) and motion sensor (Smart-Sensing) data beyond standard benchmarks. Despite practical results, the approach revealed fundamental scalability limitations from exponential circuit growth, motivating subsequent hybrid approaches and specialized optimizations. Addressing the accessibility gap, CrypTen (Knott et al. 2021) exposes MPC primitives through ML-native abstractions. Benchmarks on text classification, speech recognition, and image classification demonstrate efficient private inference under a semi-honest threat model.

Optimization strategies: Building on DeepSecure, several works addressed computational bottlenecks through complementary strategies. Diaa et al. (2024) replaced expensive ReLU activations with polynomial approximations in single-round protocols, directly improving MPC efficiency. MD-ML (B. Yuan et al. 2024) enhanced security for malicious adversaries in dishonest majority settings, tolerating corruption of up to $n - 1$ parties. COINN (Hussain et al. 2021) combines secret sharing for matrix operations with GC for maxpool/ReLU, incorporating quantized networks and genetic algorithms for optimal parameterization, enabling private inference for 100+ layer networks. The approach achieves competitive performance while jointly optimizing architectural and cryptographic efficiency.

Specialized architectures and data modalities: Different neural architectures require specialized MPC optimizations. SiRnn (D. Rathee, M. Rathee, Goli, et al. 2021) provides the only efficient framework for oblivious recurrent neural network (RNN) inference (L. K. Ng and Chow 2023), extending beyond image benchmarks to time series, speech, and sequential data. SepMM (Tung-Lin and Pei-Yuan 2023) optimizes secure matrix multiplication in two-party protocols, with integration into SiRnn demonstrating improved communication costs through composable privacy primitives. CryptoSPN (Treiber et al. 2020) explores probabilistic queries in sum-product networks using GC, demonstrating MPC applicability to tabular data and probabilistic reasoning beyond traditional neural architectures.

Modern architectures: Recent works address MPC application to transformer architectures dominant in natural language processing (NLP) and computer vision. Y. Liu and Su (2024), Zeng et al. (2023), and D. Chen, Y. Zhang, et al. (2023) improve MPC efficiency for transformers, with D. Chen, Y. Zhang, et al. (2023) proposing reduced-dimension approximate normalized attention (RNA) modules for efficient vision transformer computation. These works handle text data and demonstrate MPC adaptation to modern architectures, though requiring architectural modifications for practical performance.

Protocol design implications: An essential decision in MPC-based inference involves choosing between two-party protocols and n-party computation using external non-colluding servers. Many works utilize n external servers (Mohassel and Y. Zhang 2017; Riazi, Weinert, et al. 2018), typically with $n \in \{2, 3, 4\}$, where inputs and models are split into secret shares across servers for collective execution. While more servers enable flexible protocols and improved efficiency, this increases the attack surface since compromising a subset can break security for most frameworks. We refer to Section 6.2 for non-colluding nPC protocols.

In contrast, two-party protocols involving only the data and model owner offer simpler deployment in scenarios with limited architectural capabilities, though they may sacrifice some computational efficiency. This trade-off between deployment complexity and performance represents a key practical consideration.

Practical implications: MPC-based inference demonstrates strong practical potential through computational efficiency advantages over HE, with broad applicability across domains including audio, motion sensing, and text processing. The evolution from pure GC to modern hybrid frameworks illustrates systematic addressing of architectural bottlenecks, with three key insights: (1) architecture-specific optimizations often outperform generic approaches, (2) hybrid strategies combining different MPC primitives optimize operations effectively, and (3) the two-party vs. n-party choice should balance deployment constraints with performance requirements. MPC is well-suited to interactive, low-latency applications where communication infrastructure is reliable and architecture disclosure is acceptable. It is unsuitable where model confidentiality is a hard requirement or where persistent communication channels cannot be guaranteed.

6.1.3 Differential Privacy. DP is unsuitable for private inference scenarios due to the incompatibility between noise injection and inference accuracy requirements. Unlike training scenarios, where noise can be absorbed across many samples and epochs, inference typically involves single data points, where any noise addition directly degrades prediction quality. This limitation explains the limited research interest and the community's focus on cryptographic approaches for inference privacy.

The Arden framework (J. Wang, J. Zhang, et al. 2018) represents one of the few attempts to apply DP for inference input protection, incorporating noise into data attributes and using stochastic feature nullification before cloud transmission. Results demonstrate the fundamental privacy-utility trade-off: even with relatively high privacy budgets ($\epsilon = 3.5$ for CIFAR-10, $\epsilon = 0.7$ for MNIST), accuracy degradation remains substantial. Accuracy degradation varies across datasets, with complex datasets like CIFAR-10 experiencing considerable drops even with relaxed privacy budgets.

In conclusion, meaningful privacy guarantees typically render inference results unusable. Conversely, relaxed budgets that preserve utility provide minimal privacy protection. This illustrates DP's unsuitability for inference scenarios, contrasting with its effective integration in training through techniques like DP-SGD.

6.1.4 Trusted Execution Environments. TEEs face challenges when used as standalone solutions in PPDL, requiring strong trust in chip manufacturers and representing a shift of trust from the model owner to the chip manufacturer.

Slalom (Tramèr and Boneh 2019) performs computations inside the TEE and outsources heavy matrix multiplications to a co-located untrusted processor via Freivalds' algorithm (Freivalds 1977). Slalom was evaluated on desktop hardware (Intel Core i7-6700, 3.40GHz, 8GB RAM) with SGX support and co-located Nvidia TITAN XP GPU. eNNclave (Schlögl and Böhme 2020) partitions the model into an "extractor" and a "classifier", with the classifier operating within a trusted enclave while the extractor forward pass can be computed locally on the client side. eNNclave was evaluated on workstation hardware (Intel i7-9700, 3GHz, 8 cores, 64GB RAM) with Linux SGX support.

6.1.5 Mixed Frameworks. Mixed frameworks exploit complementary strengths of different privacy-preserving techniques: HE excels at linear operations (matrix multiplication, convolution) while MPC efficiently handles non-linear functions (activations, comparisons). This natural division aligns with NN structure, enabling technique-switching strategies that optimize each operation type independently.

Foundational HE+MPC approaches: Early frameworks established technique-switching solutions. MiniONN (J. Liu et al. 2017) pioneered this approach by combining HE for linear operations with GC for non-linear activations, demonstrating feasibility without requiring model modifications. MiniONN was evaluated on distributed desktop hardware (Intel Core i5, 3.30GHz server, Intel Core i5, 3.20GHz client). GAZELLE (Juvekar et al. 2018) advanced this concept by introducing periodic re-encryption to handle deeper networks, where the data owner remains online to refresh ciphertexts and prevent noise accumulation. GAZELLE was evaluated on AWS instances (Intel Xeon E5-2666 v3, 2.90GHz, 7.5GB RAM). This re-encryption pattern enables deeper models but introduces architectural disclosure risks through interaction patterns.

Optimization advances: Building on GAZELLE, several frameworks addressed scalability and deployment challenges. Delphi (Mishra et al. 2020) introduced GPU acceleration for convolutional layers, while Cheetah (Z. Huang et al. 2022) focused on optimizing HE-MPC switching overhead. Delphi was evaluated on AWS c5.2xlarge instances (Intel Xeon 8000 series, 3.0GHz, 16GB RAM) with Tesla V100 GPUs. Cheetah was evaluated on Alibaba Cloud instances (2.70GHz processor, 16GB RAM) with varying network conditions (384 MBps LAN, 44 MBps WAN). CrypTFlow2 (D. Rathee, M. Rathee, N. Kumar, et al. 2020) prioritized practical deployment through TensorFlow integration, while Muse (Lehmkuhl et al. 2021) enhanced security by supporting malicious adversaries. GForce (L. K. Ng and Chow 2021) represents the current state-of-the-art in efficiency, utilizing GPU-accelerated secret sharing protocols. GForce was evaluated on Google Cloud VMs (Intel Xeon Skylake, 2GHz, 52GB RAM, Nvidia V100 GPU).

HE+TEE strategies: A complementary direction combines HE with TEEs to address computational bottlenecks. P. Liu and W. Zhang (2022) delegates de/re-encryption tasks to TEEs while maintaining HE-encrypted computation in the cloud, though concentrating all keys in the TEE creates single points of failure. Xiao, Q. Zhang, et al. (2021) partitions computation more granularly, computing linear layers via HE and non-linear operations within TEEs, requiring careful orchestration of encryption state transitions. Chex-Mix (Natarajan et al. 2023) takes a conservative approach by placing the entire computation within TEEs while maintaining strong adversarial assumptions.

Practical implications: Mixed frameworks offer compelling advantages but introduce deployment complexities. The HE+MPC approach provides strong cryptographic guarantees but demands technique-switching orchestration and often requires data owner availability for re-encryption. The HE+TEE approach reduces computational overhead but shifts trust assumptions to hardware manufacturers and introduces single points of failure. For practitioners, HE+MPC frameworks suit scenarios prioritizing cryptographic security with acceptable computational costs, while HE+TEE approaches benefit resource-constrained deployments where hardware trust is acceptable. Current performance leaders like GForce demonstrate that sub-second online inference is achievable, though substantial offline preprocessing remains necessary for optimal performance.

6.1.6 HE/MPC-efficient Architectures. Specialized NN architectures address inefficiencies in applying privacy-preserving techniques to standard networks, focusing on activation function optimization and architectural search strategies tailored to cryptographic constraints.

Activation function alternatives: SieveNet (Q. Wang et al. 2021) addresses the MPC bottleneck of ReLU evaluation by introducing SieveLayer, an MPC-efficient alternative that maintains comparable accuracy while reducing online latency. SieveNet was evaluated on distributed workstation hardware (Intel Core i7-8700, 3.6GHz, with Nvidia GTX 1080Ti GPU for servers). This work demonstrates that activation function design directly impacts privacy-preserving efficiency, motivating further exploration of cryptography-friendly non-linearities.

Architecture optimization: HEMET (Lou and L. Jiang 2021) reveals that conventional mobile-optimized architectures can be counterproductive for privacy-preserving inference due to increased multiplicative depth. The framework introduces NN architecture search specifically for HE constraints, complemented by layer coefficient merging to minimize circuit depth. HEMET was evaluated on high-performance server hardware (Intel Xeon Gold 5120, 2.2GHz, 56 cores, 256GB DRAM). This optimization demonstrates considerable latency improvements through HE-aware architectural design.

Fundamental operation redesign: PolyKervResNets (Aremu and Nandakumar 2023) replaces core operations (convolution, ReLU, maxpool) with polynomial kernel convolution blocks (kervolution). The kervolution operation $k(x, w) = \sum_{j=0}^{d_p} c_p^{d_p - j} (x^T w)^j$ provides inherent HE/MPC compatibility while balancing non-linearity through polynomial degree d_p and coefficient c_p. Evaluations show inference latency of 0.26 s on CIFAR-10 using modified VGG-16 architecture, representing current state-of-the-art performance on workstation-class hardware (Intel Xeon Silver 4215, 2.50GHz, 32GB RAM).

Practical implications: Architecture-aware design is crucial for practical privacy-preserving inference, with approaches ranging from activation replacement to full architectural search and fundamental operation redesign. Specialized architectures achieve order-of-magnitude latency improvements over naive applications to standard networks and are well-suited when model modification is permissible. They are unsuitable in deployments requiring strict compatibility with off-the-shelf architectures or where retraining costs are prohibitive.

6.1.7 Compilers. Compiler frameworks address the gap between theoretical advances and practical deployment by abstracting cryptographic complexity and automating optimization decisions.

Early compilers focused on specific technique automation. EVA (Dathathri et al. 2020) automates HE parameter derivation for CKKS-based computation, while XONN (Riazi, Samragh, et al. 2019) specializes in BNN compilation for GC protocols. The nGraph-HE compiler (Boemer, Costache, et al. 2019; Boemer, Lao, et al. 2019) pioneered TensorFlow integration but required manual parameter setting. Subsequent frameworks prioritized developer accessibility. SEALion (Elsloo et al. 2019) introduced heuristic parameter search for BFV schemes, while CrypTFlow (N. Kumar et al. 2020) automated TensorFlow-to-MPC compilation. HE-MAN (Nocker, Drexel, et al. 2023) and HeLayers (Aharoni, Adir, et al. 2023) emphasized standard format support (ONNX) and tensor packing optimization, demonstrating evolution toward production-ready tools. A remaining barrier is the mismatch between C++/Rust-based FHE libraries and the Python-dominated ML ecosystem. Nocker, Henke, et al. (2025) address this with the Tailored Wrapper Architecture, a framework-agnostic blueprint for efficient Python frontends, demonstrating significant inference speedups over naive wrappers. Recent compilers address multi-scheme optimization. HEIR (Bian et al. 2024) automates cross-scheme HE optimization, while HECO (Viand et al. 2023) introduces end-to-end optimization extending beyond previous scope limitations.

Practical implications: Compilers successfully democratize privacy-preserving inference by enabling developers to apply these techniques without cryptographic expertise, and are well-suited for teams without deep cryptographic knowledge or for rapid prototyping across diverse model types. However, performance trade-offs persist between automated convenience and hand-optimized efficiency, making compilers less appropriate where latency is critical and maximum performance must be extracted. In such cases, specialized hand-optimized implementations remain preferable.

6.1.8 Summary. Private NN inference technologies exhibit fundamental trade-offs between privacy guarantees, computational efficiency, and deployment complexity. HE provides the strongest cryptographic privacy for both inputs and models but incurs substantial computational overhead, with current implementations achieving seconds to minutes of latency. MPC offers superior computational efficiency but requires inter-party communication and architectural disclosure, achieving sub-second inference with optimized frameworks. DP proves incompatible with inference scenarios due to accuracy degradation from required noise injection, limiting its practical applicability.

Mixed frameworks emerge as the most promising direction, combining complementary strengths of different techniques to achieve both strong privacy guarantees and practical performance, with state-of-the-art systems demonstrating sub-second latency on standard benchmarks. Specialized architectures and compiler frameworks address deployment barriers by optimizing operations for privacy-preserving computation and abstracting cryptographic complexity for practitioners.

Despite these advances, privacy-preserving inference faces persistent challenges in balancing the privacy-utility-efficiency trilemma. While techniques prevent direct data exposure, they do not inherently protect against inference attacks like membership inference, requiring complementary training-time defenses as discussed in Section 7.1. The field's advancement depends on addressing deployment complexity and establishing systematic frameworks for technology selection based on application-specific requirements. Future research should address the following open questions:

Q1: Can novel activation functions or computational patterns match or improve the accuracy of current polynomial approximations?

Q2: How can MPC protocols be designed to preserve computational privacy, potentially through techniques like architecture obfuscation or encrypted circuit representations, without sacrificing efficiency?

Q3: What systematic frameworks can guide practitioners in technology selection for privacy-preserving inference, considering the complex trade-offs between security guarantees, performance requirements, deployment constraints, and regulatory compliance?

Q4: Can automated compilation techniques achieve performance parity with hand-optimized implementations while maintaining usability, or does a fundamental trade-off exist between abstraction convenience and computational efficiency?

6.2 I2: Multiple Data Owners

In this scenario, $n > 1$ data owners possess complementary attributes x_i of identical samples, corresponding to vertical data partitioning. The computational objective is to evaluate $M(x_1, \ldots, x_n)$ while ensuring that: (1) data owners learn only the final result without accessing other parties' input data or model parameters; and (2) model owners perform computations obliviously, gaining no knowledge about input data or results.

Threat Model Adaptations: I2 introduces multi-party coordination challenges where data owners must protect their individual input features from other participants. Key vulnerabilities include collusion between data owners to reconstruct complete input profiles and potential information leakage when combining data from multiple institutions during collaborative computation.

Example: Anti-money laundering – Financial institutions collaboratively leverage vertically partitioned customer data to enhance fraud detection accuracy. Protocols integrate sensitive information from multiple institutions (transaction histories, account balances, behavioral patterns) while ensuring customer data remains confidential and is not disclosed to any external party, including participating institutions.

6.2.1 Homomorphic Encryption. HE provides three approaches for handling multiple data owners:

(1) Shared secret key approach: All data owners encrypt and decrypt using a common secret key. While operationally simple, this introduces security vulnerabilities, as compromising any single party or communication channel exposes all encrypted data. Key establishment requires a multi-party Diffie-Hellman protocol with $n-1$ communication rounds. Despite these risks, this approach provides flexibility for dynamic party participation and demonstrates robustness against participant dropout.

(2) Multi-key HE (MKHE): Each party independently generates public and secret key pairs, enabling distributed key generation without coordination overhead. Input data is encrypted sequentially under all participants' public keys, producing nested ciphertexts of the form $[[x]_{\mathrm{pk}_A}]_{\mathrm{pk}_B}$. This ciphertext expansion degrades computational performance proportionally to the number of participants. However, MKHE allows parties to join at any stage. Decryption requires distributed computation where each party generates partial decryption shares using their private key, followed by collaborative reconstruction of the plaintext result. Implementations employ noise flooding techniques or GC to prevent secret key leakage during decryption (Kwak et al. 2022).

(3) Multi-party HE (MPHE): Participants collaboratively generate a common public key through distributed key generation, typically by summing individual public keys: $\mathrm{pk} = \sum_i \mathrm{pk}_i$. Data encryption uses this shared public key $[x]_{\mathrm{pk}}$, yielding ciphertext sizes and computational costs comparable to single-party HE schemes.

The primary limitation is the requirement for fixed participant sets during the setup phase, as the distributed key generation protocol must complete before any encryption operations. Like MKHE, decryption necessitates collaborative computation of partial decryption shares. MPHE represents a special case of threshold HE, denoted as (t, n)-threshold HE, where t specifies the minimum number of parties required for successful decryption. Unlike standard MPHE, threshold schemes do not require all parties to collaborate for decryption, and allow for parties to join later (Sugizaki et al. 2023).

The security properties of threshold schemes depend on the relationship between t and n. In particular, robust secret sharing protocols can withstand malicious participants submitting incorrect shares when $t < n/3$, achieved through standard error correction techniques (Ben-Or et al. 1988). Extending robustness to $t < n/2$ incurs substantial computational overhead (Personal communication with Nigel Smart, October 2023). Robust secret sharing becomes theoretically impossible when $n < 2t$ (Fehr and C. Yuan 2020). Standard MPHE corresponds to the case where $t = n$, requiring all parties for decryption.

Performance and practical considerations: MPHE is more efficient in terms of ciphertext size and computational cost compared to MKHE, but requires a fixed set of participants established during the setup phase. MKHE provides more flexibility through dynamic party participation, however, with higher computational overhead. Recent advances have achieved linear complexity for MKHE schemes (Akın et al. 2023; T. Kim et al. 2022; Kwak et al. 2022). Despite these theoretical improvements, empirical studies evaluating real-world deployments of both MPHE and MKHE remain limited.

The snu-mghe repository[3] provides a reference implementation of the current state-of-the-art MKHE scheme (T. Kim et al. 2022), supporting both BFV and CKKS. By contrast, several surveyed works employ outdated MKHE constructions: Kuang et al. (2020) propose utilizing the scheme of Clear and McGoldrick (2015), which has been demonstrated to be impractical due to its approach of encrypting individual bits within large ciphertexts (H. Chen, Chillotti, et al. 2019; H. Chen, Dai, et al. 2019).

6.2.2 Secure Multi-Party Computation. Multiple data owners can utilize MPC in two ways. First, each data owner acts as a party in an MPC protocol. This ensures privacy, but data owners have to be online throughout the protocol. Moreover, protocols for large numbers of parties take considerably more time than protocols involving only a few parties and do not scale well with increasing input data size (Mohassel and Y. Zhang 2017). Notably, F. Liu et al. (2024) performs the online inference phase of a four-layer CNN across 63 parties in 0.1 seconds, where the model is secretly shared among up to 63 data owners.

A more common approach involves partitioning data from all owners into secret shares, distributing them among n non-colluding servers (typically $n = 2$–4). Subsequently, these servers execute an nPC protocol. Table 3 provides a comprehensive comparison of nPC frameworks and reveals distinct patterns in their design and application focus.

Two-party protocols ($n = 2$): SecureML, QUOTIENT, and ABY2.0 demonstrate feasibility but are restricted to relatively small models due to overhead in handling non-linear operations through quantization and MPC-friendly approximations. These protocols prioritize broad applicability across diverse use cases, with QUOTIENT uniquely demonstrating comprehensive real-world dataset evaluation spanning health and finance domains. SecBNN targets BNNs with specialized protocols for linear and non-linear layers, improving efficiency to enable inference on larger networks.

Three-party protocols ($n = 3$): These introduce honest-majority assumptions to reduce communication overhead while enabling more sophisticated operations. Chameleon optimizes preprocessing of multiplication triples, while ABY3 and SecureNN combine arithmetic and binary secret sharing to support convolutional layers efficiently. Recent frameworks broaden applicability: Privformer targets transformer-based NLP models, Meteor optimizes core cryptographic primitives, and BANNERS specializes in BNNs.

[3] https://github.com/SNUCP/snu-mghe (Accessed: 2026-07-17)

Table 3. Overview of *n*PC frameworks categorized by the number of computation servers. Dataset usage is indicated by filled circles (●) for supported datasets and empty circles (○) for unsupported ones. The Use Case column specifies whether frameworks target general applicability or specific domains. The Specialized Datasets column lists domain-specific datasets when applicable. Security properties specify adversary models.

Framework	Use Case	MNIST	CIFAR-10	ImageNet	Specialized Datasets	Security†
Two-Party Protocols ($n = 2$)						
SecureML (Mohassel and Y. Zhang 2017)	General	●	○	○	–	sh
QUOTIENT (Agrawal et al. 2019)	Health & Finance	●	○	○	MotionSense, Thyroid, Breast Cancer, Skin Cancer MNIST, German Credit	sh
ABY2.0 (Patra, Schneider, et al. 2021)	General	●	○	○	–	sh
SecBNN (H. Chen, H. Li, et al. 2024)	General	●	●	○	BNNs	sh
Three-Party Protocols ($n = 3$)						
Chameleon (Riazi, Weinert, et al. 2018)	Network Optimization	●	●	○	–	sh
ABY3 (Mohassel and Rindal 2018)	General	●	○	○	–	sh/mal
SecureNN (Wagh, Gupta, et al. 2019)	Network Optimization	●	○	○	–	sh
Blaze (Patra and Suresh 2020)	Healthcare	○	○	○	Parkinson's (audio)	mal
Banners (Ibarrondo et al. 2021)	Binary Networks	●	●	○	BNNs	mal
CryptGPU (Tan et al. 2021)	General	●	●	●	Tiny ImageNet	sh/mal
Adam in Private (Attrapadung et al. 2022)	General	●	●	○	–	sh/mal
LEGO (Z. Zhou et al. 2022)	General	●	●	○	FashionMNIST	sh/mal
Meteor (Dong et al. 2023)	General	●	●	○	–	sh
Privformer (Akimoto et al. 2023)	NLP/Transformers	○	○	○	Transformer embeddings	sh
R. Wang et al. (2024)	Computer Vision	○	○	○	PASCAL VOC2012	sh
Four-Party Protocols ($n = 4$)						
Flash (Byali et al. 2019)	Binary Networks	●	○	○	BNNs	mal
Trident (Chaudhari et al. 2020)	General	●	○	○	–	mal
Fantastic Four (Dalskov et al. 2021)	General	●	○	●	–	mal
SWIFT (Koti et al. 2021)	General	●	●	○	–	mal
Keller and Sun (2022)	General	●	●	○	–	sh/mal

†: sh…semi-honest, mal…malicious

Four-party protocols ($n = 4$): These emphasize throughput and malicious security, with Flash, Swift, Fantastic Four, and Trident all reducing online latency compared to 2PC and 3PC protocols, though requiring stricter trust assumptions. Malicious security is supported by all but one framework. CryptGPU demonstrates that hardware acceleration through GPU mapping can outperform protocol-level optimizations alone, achieving the lowest reported online latencies.

The progression from 2PC to 4PC reveals key trade-offs: increasing server count enables better parallelism and security guarantees but requires more complex trust assumptions and coordination. Domain-specific optimizations prove crucial for practical performance, while hardware acceleration emerges as a critical factor for deployment viability.

6.2.3 Differential Privacy. DP faces limitations in multi-data owner inference scenarios that extend beyond single data owner challenges. The core challenge lies in determining where and how to apply DP noise when multiple parties contribute vertically partitioned data: noise injection at the level of individual data owners degrades combined inference quality multiplicatively, while centralized noise addition requires a trusted aggregator

that undermines distributed privacy objectives. This incompatibility, combined with complex privacy budget coordination challenges, explains the scarcity of research in this area.

J. Kim and Lim (2023) compute user density distributions from location information, protecting sensitive location data by adding DP noise. They report mean average errors of below 1% ($\epsilon \in [0.1, 1.5]$) evaluated on synthetic data and Seoul metro passenger density data. The Latent algorithm (Arachchige et al. 2019) and DeepGuess (Tchaye-Kondi et al. 2023) add noise before data aggregation and final layer evaluation.

These works represent specialized applications rather than general-purpose solutions, confirming that cryptographic approaches provide more viable paths for preserving privacy while maintaining utility in collaborative inference scenarios.

6.2.4 Summary. Privacy-preserving inference with multiple data owners introduces challenges that extend beyond single-party scenarios, requiring sophisticated coordination mechanisms and stronger security assumptions. Each technology presents distinct trade-offs that constrain practical deployment.

HE offers three paradigms: shared secret keys provide simplicity at the cost of severe security vulnerabilities, multi-key HE (MKHE) enables dynamic participation with computational penalties, and multi-party HE (MPHE) achieves single-party efficiency but restricts flexibility. Recent advances have achieved linear complexity for MKHE schemes (Akın et al. 2023; T. Kim et al. 2022; Kwak et al. 2022). NIST has initiated guidelines for multi-party threshold cryptosystems[4], indicating growing institutional recognition. MPC demonstrates remarkable performance, achieving 2.14 s online latency for VGG-16 on CIFAR-10 (Tan et al. 2021). CryptGPU was evaluated on GPU-optimized cloud infrastructure (NVIDIA Tesla V100, 8-core Xeon E5-2686, 61GB RAM) with high-bandwidth networking (1.25GB/s, 0.2ms latency). However, MPC solutions require non-colluding server assumptions. DP faces incompatibilities due to multiplicative noise degradation and privacy budget coordination challenges in distributed settings, explaining the scarcity of publications in this area.

Practical implications: HE approaches offer strong theoretical guarantees but differ in deployment fit: MPHE outperforms MKHE for fixed participant sets due to linear efficiency gains and is the preferred choice for stable collaborations, while MKHE better accommodates dynamic participation at higher computational cost. MPC achieves practical performance, with 3PC and 4PC protocols delivering sub-second inference times for moderate models, and is well-suited when non-colluding server infrastructure is available. DP proves unsuitable for this use case due to multiplicative noise degradation and privacy budget coordination challenges. Current cryptographic approaches are unsuitable when neither fixed participant sets nor non-colluding server assumptions can be established upfront. Future research should address the following open questions:

Q5: How can MPHE and MKHE schemes be systematically integrated into end-to-end protocols for practical deployment in distributed inference scenarios?

Q6: What novel cryptographic techniques can enable efficient collaborative decryption while mitigating noise flooding attacks and maintaining provable security guarantees?

Q7: Can practical alternatives to the non-colluding server assumption be developed without compromising security or increasing computational costs?

6.3 I3: Multiple Model Owners

In this scenario $m > 1$ model owners hold models denoted $M_1, \ldots, M_m$. Each model owner independently conducts private inference on their models and the input of one data owner, resulting in m local results that require private aggregation. The privacy objective is to ensure the confidentiality of the input data, prevent the transfer of

[4] https://csrc.nist.gov/Projects/threshold-cryptography (Accessed: 2026-07-17)

models, and ensure that the final result is only accessible to the data owner.

Threat Model Adaptations: I3 introduces trust concerns regarding the entity responsible for combining results from multiple model owners. Primary vulnerabilities include collusion between model owners and the party aggregating results, which could enable exhaustion of privacy protections and inference of individual model behaviors during the ensemble combination process.

Example: Aggregated diagnosis from multiple hospitals – Several hospitals develop models using their local datasets. For privacy reasons, the models or the underlying training data must not be disclosed to other parties. To leverage the joint knowledge, each hospital evaluates its model on the same sensitive input while ensuring input privacy. The individual results are aggregated for the final diagnosis. This approach finds application in medical image classification (Jarin and Eshete 2021).

6.3.1 PATE and Variations. Private aggregation of teacher ensembles (PATE) (Papernot, Abadi, et al. 2017) establishes the foundational framework for privacy-preserving model ensemble inference. The protocol operates through distributed teacher models trained on distinct sensitive datasets, with students querying these teachers through a differentially private aggregation mechanism. Teachers compute local inferences, after which the results undergo noisy aggregation on a trusted aggregator to protect against membership inference and model extraction attacks, and confident outcomes are returned to train student models. PATE's architecture-agnostic design enables broad applicability, though query limits imposed by privacy budget constraints restrict practical deployment scale.

In PATE, training occurs locally on sensitive data, but student labels emerge through inference processes, justifying its inclusion in this inference-focused section rather than Section 7.

Core PATE improvements enhance specific framework components to address fundamental limitations. Privacy mechanism refinements include scalable DP aggregation with tighter privacy analysis (Papernot, Song, et al. 2018) and individualized privacy budgets that reduce query rejection rates (Boenisch, Mühl, et al. 2023). Accuracy improvements incorporate generative adversarial networks during student training (Q. Zhang et al. 2020) and mobile-optimized architectures for resource-constrained deployment (Lyu and C.-H. Chen 2020). Domain-specific adaptations extend the framework to multi-label classification scenarios (Dziedzic et al. 2023) and graph neural networks (H. Huang et al. 2023), demonstrating PATE's versatility across problem domains.

Cryptographic PATE variants address the trusted aggregator vulnerability through cryptographic protection mechanisms. SPEED (Grivet Sébert et al. 2021) eliminates aggregator trust requirements by shifting noise addition to teachers and performing oblivious aggregation via TFHE encryption, achieving an overhead of less than four seconds per query while preventing aggregator-based membership inference attacks. M. Kumar, W. Zhang, et al. (2023) pursue similar decentralization through DP-trained individual models, encrypting outputs with TFHE and utilizing fuzzy logic aggregation. Their advanced KAHM approach (M. Kumar, Moser, et al. 2024) introduces kernel affine hull machines that generate synthetic samples within the DP affine hull, providing additional protection for original data while maintaining model utility.

Input privacy extensions address client-side privacy through complementary cryptographic techniques. PRICURE (Jarin and Eshete 2021) implements dual-server input protection using model and input shares distributed across non-colluding servers, with trusted aggregation under client public keys. Their analysis reveals that increased party participation enables higher noise tolerance, reducing individual model sensitivity to aggregation perturbations. CaPC (Choquette-Choo, Dullerud, et al. 2021) provides comprehensive input encryption through GC-based inference, employing a privacy guardian for DP noise injection. While ensuring complete input confidentiality, this approach incurs substantial computational overhead due to cryptographic operations and secure aggregation protocols.

6.3.2 Summary. Multiple model owner scenarios center on secure aggregation of individual inference results, primarily addressed through the PATE framework and its variations. The core challenge lies in preventing aggregator-student collusion that can bypass privacy protections through noise pattern analysis.

Three technological approaches emerge: Pure DP (PATE variants) offers theoretical guarantees but suffers from query budget limitations and trusted aggregator vulnerabilities. Cryptographic methods (SPEED, PRICURE, CaPC) eliminate trust requirements with practical performance but require non-colluding servers. Hybrid approaches combining DP with cryptographic techniques show the most promising results by addressing multiple threat models simultaneously.

Practical implications: Avoid pure PATE in adversarial settings due to trusted aggregator vulnerabilities. It suits only scenarios where computational simplicity is prioritized over security and a trusted aggregator can realistically be assumed. Cryptographic variants such as SPEED deliver superior security at acceptable computational costs and are well-suited when non-colluding servers are available and high-security requirements must be met. Deployment success depends on privacy budget management and balancing ensemble size against privacy protection effectiveness. Organizations in regulated or adversarial domains should prioritize cryptographic variants and are discouraged from pure DP approaches where aggregator trust cannot be guaranteed.

Q8: Can decentralized aggregation protocols be designed to eliminate single points of failure while maintaining computational efficiency and provable security guarantees?

Q9: Can MKHE or MPHE schemes provide privacy guarantees for collaborative model aggregation without requiring trusted third parties or non-colluding server assumptions?

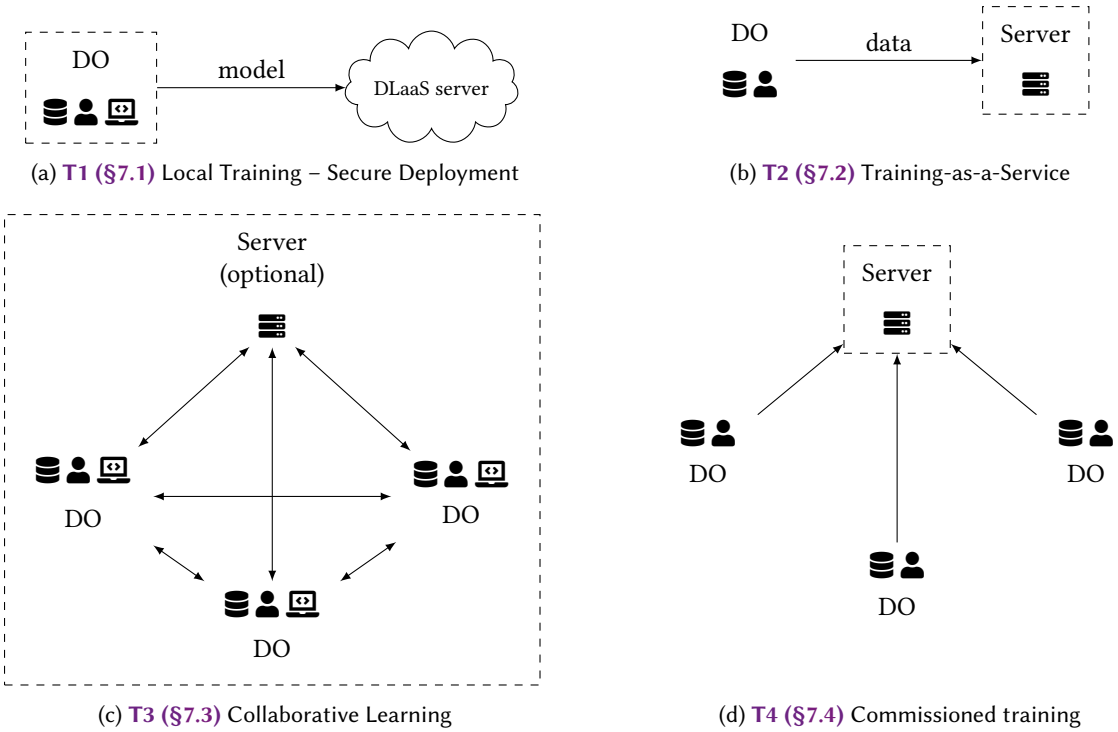
6.4 I4: Multiple Model & Data Owners

In this configuration, multiple model owners hold models $M_1, \ldots, M_m$, and multiple data owners hold vertically split data shares $x_1, \ldots, x_n$. Their goal is to privately compute an aggregation of m individual private inference results $M_j(x_1, \ldots, x_n)_{1 \leq j \leq m}$. As before, input data should not be disclosed to any other party, while the privacy of model owners is upheld by ensuring that their models are not transferred.

Threat Model Adaptations: I4 combines all previous vulnerabilities, requiring coordination between multiple data owners and multiple model owners. The threat surface expands with potential for complex collusion patterns between data owners, model owners, and external aggregators.

Example: Distributed medical diagnosis with split data – This example extends the case from I3 (multiple model owners). Here, patient data is vertically divided and distributed across different doctors as a result of individual treatments. To achieve the most accurate and comprehensive result, these separate datasets need to be collectively analyzed by models situated at various medical institutions. The individual inference results must be combined to yield the final prediction.

Currently, no research papers address this scenario, which represents the most complex configuration in our taxonomy by combining challenges from both I2 (secure data aggregation) and I3 (private result aggregation). The absence of research reflects compounding technical barriers: protocols must coordinate nested privacy-preserving operations, including data reconstruction, private model evaluation, and secure ensemble aggregation. Each layer introduces cascading cryptographic overhead and trust assumptions that may render practical deployment infeasible with current techniques. As privacy-preserving technologies already pose challenges in simpler scenarios (I1–I3), major advances in cryptographic primitives may be a prerequisite for addressing this scenario effectively.



(a) **T1 (§7.1)** Local Training – Secure Deployment

(b) **T2 (§7.2)** Training-as-a-Service

(c) **T3 (§7.3)** Collaborative Learning

(d) **T4 (§7.4)** Commissioned training

Fig. 2. Visualization of training scenarios T1–T4. Data owners (DO,) possess sensitive data and aim to train a robust DL model without revealing private data to other parties. External servers ( ) facilitate or support training, coordinate communication, and aggregate (intermediate) results. Arrows indicate the exchange of either model parameters (T1 & T3) or training data samples (T2 & T4), representing the primary privacy assets requiring protection in each scenario. Detailed analyses of each use case are provided in the corresponding sections of the paper.

Q10: Assuming practical deployment of I1 to I3, does this scenario correspond to a straightforward combination of existing solutions, or do new challenges arise?

7 Training

This section covers the four training use cases (T1–T4) in our PPDL taxonomy. We consider one or multiple training data owners that possess sensitive data, intending to train a model using their private data. Sections 7.1 and 7.2 discuss use cases with a single data owner, whereas Sections 7.3 and 7.4 consider multiple data owners. Furthermore, the scenarios are categorized based on the type of private information that is exchanged and requires protection: scenarios where model parameters or updates are shared (Sections 7.1 and 7.3) versus scenarios where raw training data samples are transferred (Sections 7.2 and 7.4). A high-level visualization of all training use cases is illustrated in Figure 2.

The type of exchanged private information fundamentally determines the most effective privacy-preserving techniques. In scenarios where model parameters or updates are shared (T1 and T3), DP emerges as the primary technique, as it directly addresses model-based privacy attacks such as membership inference and reconstruction

attacks on training data. Cryptographic methods like HE and MPC can provide additional protection layers for model updates during collaborative training, but DP remains the primary mechanism for protecting against post-training privacy attacks. Conversely, in scenarios where raw training data samples are transferred (T2 and T4), cryptographic techniques become indispensable. Applying DP to raw training data would require such substantial noise that model utility would be severely compromised, making HE and MPC the practical choice for protecting data during transmission and computation. This fundamental distinction reflects the different nature of privacy risks: model-based attacks require statistical protection through DP, while data transmission requires computational protection through cryptography.

Threat Model: Training scenarios consider honest-but-curious adversaries attempting to extract sensitive information from training data, intermediate model parameters, or gradient updates. Data owners protect training data privacy, meaning that individual samples should not be recoverable. External servers and collaborating parties may observe communications, intermediate computations, and final model parameters, but follow protocol specifications correctly.

Privacy Goals: Core objectives include training data privacy to prevent reconstruction of individual training samples, gradient privacy to protect intermediate learning updates, and model utility preservation to maintain training effectiveness while ensuring privacy. Additional goals encompass communication efficiency in collaborative settings and resistance to attacks such as membership inference on the trained model.

Vulnerabilities: Primary attack vectors include membership inference to determine if specific data was used for training, model inversion to reconstruct training inputs from model parameters, property inference to learn dataset statistics, and gradient-based reconstruction attacks. In collaborative settings, additional risks include collusion between participants and information leakage through communication patterns or metadata analysis.

7.1 T1: Local training – Secure deployment

In this scenario, a single data owner trains a DL model locally on sensitive data with the objective of subsequently deploying the model for inference services while ensuring robustness against privacy attacks. The fundamental challenge lies in preventing adversaries from extracting sensitive information about the training data through membership inference, model inversion, or reconstruction attacks. Since the final trained model parameters are exposed to external users during deployment, protecting these model parameters against privacy attacks on the underlying training data becomes the primary concern.

Threat Model Adaptations: T1 focuses on post-deployment privacy attacks where trained models face extensive querying by external adversaries. Primary concerns include membership inference attacks targeting training data privacy and model inversion attacks attempting to reconstruct training samples from deployed model parameters.

Example: Medical Classifier Deployment – A hospital trains a model for medical diagnostics, e.g., to detect pneumonia from chest radiographs (Ziller et al. 2021). Once trained, the model is deployed as a diagnostic service for external clinicians. Because the training data contains sensitive patient information, the released model must be robust against membership inference or reconstruction attacks.

7.1.1 Differential Privacy. DP-SGD is a widely employed technique for perturbing the training process to protect against privacy attacks. The main challenge lies in balancing privacy and utility in terms of model accuracy by injecting minimal noise while maintaining high accuracy and strong privacy guarantees.

The literature addressing this challenge can be grouped into three broad lines of work: (1) Standard DP-SGD and variants, (2) leveraging "public" data for improved DP training, and (3) leveraging pretrained models. We first focus on works that report results on CIFAR-10, the most commonly used benchmark, allowing for meaningful comparison across approaches. For CIFAR-10, there exists a standard split (50 000 training, 10 000

test images), which we assume unless otherwise stated. Following this analysis, we expand the discussion to examine applications beyond CIFAR-10, including specialized architectures, diverse datasets, and domain-specific implementations.

(1) DP-SGD variants: A large body of work modifies DP-SGD itself, targeting activation and loss function design. Papernot, Thakurta, et al. (2021) replace ReLU with tempered sigmoids $\phi_{s,T,o}(x) = s/(1 + e^{-Tx}) - o$ to reduce the impact of single samples and reduce the exploding gradient issue. $\phi_{2,2,1}$ yields the tanh function. DPLIS (W. Wang et al. 2021) proposes smoothing the loss function, which reduces performance variation across training runs and achieves higher accuracy. Shamsabadi and Papernot (2023) introduce a three-way loss function that replaces cross-entropy, combining squared error, a loss to emphasize challenging examples, and a regularization penalty to stabilize training by mitigating extreme weight updates. As networks with more parameters require proportionally stronger perturbation, the following works leverage gradient sparsity. Hölzl et al. (2023) use sparse models by design, employing equivariant CNNs that require fewer parameters. Ito et al. (2023) improve the utility-privacy trade-off by reducing the gradient dimension due to the low-rank and sparse structure of NNs. DP-SAT (Park et al. 2023) is a sharpness-aware training method that exploits the geometric properties of the loss landscape, seeking flat minima that are more robust to noise during DP training. Flat minima help mitigate the negative effects of per-example gradient clipping and Gaussian noise addition. For $\epsilon < 2$, considerable accuracy loss must be expected, with reported results consistently below 65%. Once $\epsilon \geq 2$, models achieve markedly better performance, typically in the range of 71%–81%, with the upper bound reached at $\epsilon = 8$.

(2) Leveraging public data: A second group of works improves the privacy-utility trade-off by assuming access to a certain quantity of "public" data, defined as non-sensitive data that can be used without privacy constraints or information leakage about the private training data. Xiao, Wan, et al. (2023) combine DP-SGD with hybrid clipping, calibrating gradient clipping with 2 000 ImageNet samples. Ganesh et al. (2023) train models to find flat minima by training on public data before DP-SGD on private data to improve robustness against noise and enhancing accuracy. Nasr et al. (2023) designate a fraction of CIFAR-10 as public, augment it, and use it for pretraining and extending the training dataset. Depending on the public set size, they report substantial improvements. For example, with 4% of public data, augmented to 40 000 instances, accuracy increases to 72.1% at $\epsilon = 1$ (previously achieved with $\epsilon > 2$). However, these approaches rely on strong assumptions about public data availability. Except for $\epsilon = 1$, only Nasr et al. (2023) in the above mentioned setting (4% public data augmented to 40 000 instances) report an improvement that marginally exceeds the strongest DP-SGD baselines. While public data strategies can yield utility improvements, their practical applicability is constrained by the challenge of acquiring non-sensitive data that is both relevant to the sensitive training domain and available in sufficient quantities. This limitation is evident in the research literature, where studies often artificially partition standard benchmark datasets into "public" and "private" subsets rather than utilizing truly independent non-sensitive datasets.

(3) Leveraging pretrained models: A third line of research departs from training models end-to-end with DP-SGD by leveraging pretrained representations. Large non-private models serve either as feature extractors or as initializations for fine-tuning under DP. This strategy has yielded strong improvements: D. Yu et al. (2021) use embeddings from ResNet152 pretrained on ImageNet, reaching >90% accuracy at low privacy budgets ($\epsilon \in [0.1, 0.5]$). C. Feng et al. (2023) report accuracies above 95% when fine-tuning a WideResNet-28-10 model (Zagoruyko and Komodakis 2017). Park et al. (2023) reach comparable performance using DeiT (Touvron et al. 2021) and CrossViT architectures (C.-F. R. Chen et al. 2021), while X. Zhang, Bu, et al. (2024) report the highest accuracy with 97.5% at $\epsilon = 2$. Simpler approaches, such as Tramèr and Boneh (2021), employ ScatterNet features (Oyallon and Mallat 2015) with DP training. These results show that when high-quality pretrained models are available, private fine-tuning nearly closes the gap to non-private training. Compared to the modest gains from "public" data, pretrained-model approaches yield substantial accuracy improvements, but privacy is guaranteed only for the fine-tuning phase, while pretraining data and models are treated as non-sensitive.

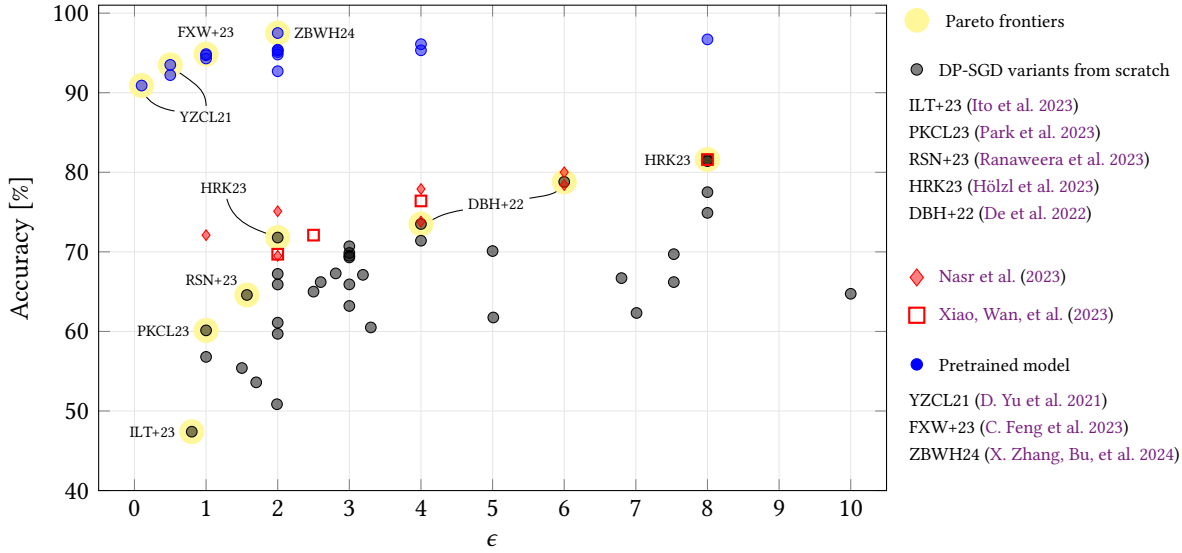


Fig. 3. CIFAR-10 classification accuracy versus privacy budget ϵ ($\delta = 10^{-5}$) for differentially private training methods (top-left ones are better). Results are grouped into (i) DP-SGD and variants (black circles, Pareto frontiers highlighted), (ii) approaches leveraging limited public data (red markers), and (iii) approaches leveraging pretrained models (blue circles, Pareto frontiers highlighted). Pretrained-model approaches consistently achieve the best accuracy-privacy trade-offs, while public-data strategies provide moderate improvements under restrictive assumptions.

Figure 3 illustrates reported CIFAR-10 classification accuracies as a function of the privacy budget ϵ under (ϵ, δ)-DP with $\delta = 10^{-5}$. Results are grouped according to the three categories of approaches discussed. Works that train models from scratch using standard DP-SGD or its variants are shown as black circles, with Pareto frontiers highlighted. Approaches that incorporate auxiliary "public" data are shown in red, with Xiao et al. represented by squares and Nasr et al. by diamonds. These methods generally provide modest gains over DP-SGD baselines, except at $\epsilon = 1$, where Nasr et al. report considerable improvement. For this group, no separate Pareto frontier is plotted, as the limited number of works (two) does not justify an additional curve and would compromise visual clarity. Methods that leverage pretrained models are depicted as blue circles, with Pareto frontiers highlighted. These approaches consistently achieve the highest accuracies, approaching non-private performance even at small privacy budgets. The corresponding numerical results, including exact ϵ values and accuracies for all works depicted, are summarized in Table 4. Results are taken directly from the original works, restricted to CIFAR-10 with the standard dataset split and $\delta = 10^{-5}$, enabling meaningful comparison across methods.

These results show the fundamental privacy-utility trade-off for DP: smaller values of ϵ correspond to stronger protection but typically yield substantial utility loss, while larger values relax privacy in favor of higher accuracy. Operating at very high ϵ can formally satisfy DP with minimal noise injection, yielding near non-private performance, but the resulting privacy protection is practically negligible and may be insufficient for sensitive applications.

For practitioners aiming to implement private training, the reviewed works suggest three practical pathways depending on available resources and assumptions. If no auxiliary data or pretrained models can be used, carefully tuned standard DP-SGD remains the baseline, and developers should benchmark against reported Pareto frontiers (e.g., De et al., Hölzl et al.) to calibrate expectations. When small amounts of trusted public data are available,

Table 4. Privacy budget and accuracy of DP training implementations on CIFAR-10. $\delta = 10^{-5}$. Pareto frontiers are highlighted as displayed in Figure 3.

Reference	ϵ	Accuracy
De et al. (2022)	1.00	56.80
	2.00	65.90
	3.00	70.70
	4.00	73.50
	6.00	78.80
	8.00	81.40
C. Feng et al. (2023)	3.00	69.51
Hölzl et al. (2023)	2.00	71.80
	8.00	81.60
Ito et al. (2023)	0.80	47.40
	1.70	53.60
	3.30	60.50
	6.80	66.70
H. Liu, C. Li, et al. (2022)	3.00	65.91
	3.19	67.11
Papernot, Thakurta, et al. (2021)	7.53	66.20
Park et al. (2023)	1.00	60.13
	2.00	67.23
	3.00	69.86
Ranaweera et al. (2023)	1.57	64.58
	2.81	67.29
	7.53	69.71
Shamsabadi and Papernot (2023)	1.50	55.40
	3.00	63.20
Tramèr and Boneh (2021)	2.60	66.20
	3.00	69.30
W. Wang et al. (2021)	1.99	50.85
	5.01	61.75
	7.01	62.32
	10.00	64.73
Xiao, Wan, et al. (2023)	2.00	61.10
	2.50	65.00
	4.00	71.40
	8.00	77.50
D. Yu et al. (2021)	2.00	59.70
	5.00	70.10
	7.00	74.90

Reference	ϵ	Accuracy
Using Public Data		
Nasr et al. (2023)	1.00	72.10†
	2.00	75.10†
	4.00	77.90†
†: 4% augmented to 40K samples	6.00	80.00†
	2.00	69.50‡
‡: 4% augmented to 5K samples	4.00	73.80‡
	6.00	78.40‡
Xiao, Wan, et al. (2023)	2.00	69.70
	2.50	72.10
	4.00	76.40
	8.00	81.60
Using Pretrained Models		
De et al. (2022)	1.00	94.70
	2.00	95.40
	4.00	96.10
	8.00	96.70
C. Feng et al. (2023)	1.00	94.85
	2.00	95.11
	4.00	95.33
Park et al. (2023)	0.50	92.21
	2.00	95.32
Tramèr and Boneh (2021)	2.00	92.70
D. Yu et al. (2021)	0.10	90.90
	0.50	93.50
	1.00	94.30
	2.00	94.80
X. Zhang, Bu, et al. (2024)	2.00	97.50

techniques such as hybrid clipping or public-data pretraining can improve accuracy and are relatively easy to integrate. Finally, if access to high-quality pretrained models is acceptable within the application's trust model, fine-tuning or feature extraction offers the most effective strategy, yielding near non-private accuracy with strong privacy guarantees on the fine-tuned stage. Thus, the choice of method should be guided primarily by the availability and trust assumptions surrounding auxiliary data and pretrained representations.

Beyond CIFAR-10, differentially private training has been explored across diverse architectures, domains, and optimization strategies.

Graph neural networks: Differentially private graph neural networks (GNNs) protect against inference attacks targeting node or edge information. Mueller et al. (2023) evaluate their implementation on five U.K. Biobank datasets, reporting accuracy drops of 2–10% at $\epsilon = 1$ and 6–18% for $\epsilon = 0.5$ ($\delta < 10^{-3}$) compared

to non-private SGD. Sajadmanesh and Gatica-Perez (2024) compare node- and edge-level DP across multiple benchmark datasets, finding edge DP achieves moderate utility loss (2–18% at $\epsilon = 1$, $\delta < 5 \cdot 10^{-7}$) while node DP performs poorly even at higher budgets (4–28% at $\epsilon = 8$, $\delta < 3.8 \cdot 10^{-5}$). Edge-level DP achieves superior utility compared to node-level DP, as it offers a strictly weaker guarantee by protecting individual edges rather than all edges incident to a node. Utility degradation nonetheless remains substantial under both notions.

Natural language processing: M. Du et al. (2023) propose DP-Forward for fine-tuning language models, perturbing embedding matrices during forward passes rather than gradients. This approach outperforms DP-SGD under comparable privacy parameters, achieving moderate accuracy decreases of 0.3–2.8% for $\epsilon = 1$ and 0.25–2.1% for $\epsilon = 3$. The embedding-level perturbation strategy suggests that domain-specific DP adaptations can improve utility retention compared to generic gradient perturbation methods.

Medical applications: Healthcare domains present both compelling use cases and stringent accuracy requirements for DP training. Islam et al. (2022) employ objective perturbation for DP-autoencoders in cancer classification tasks ($\epsilon = 0.5$–2.0). The deepee framework (Ziller et al. 2021) demonstrates contrasting results across medical tasks: chest radiography classification shows performance degradation (ROC-AUC dropping from 0.960 to 0.848–0.882 for Renyi-DP $\epsilon = 0.64$–2.81, $\delta = 10^{-5}$), while liver tissue segmentation maintains comparable performance (Dice coefficient of 0.943 vs. 0.950 non-private). These findings highlight task-dependent sensitivity to DP noise, with segmentation tasks appearing more robust than classification.

Alternative optimization strategies: Differentially Private Follow-The-Regularized-Leader (DP-FTRL) provides an alternative to DP-SGD by accumulating gradients over time and injecting correlated noise, potentially reducing variance and improving convergence stability. Choquette-Choo, H. B. McMahan, et al. (2023) extend DP-FTRL to multi-epoch training on CIFAR-10 and Stackoverflow datasets ($\epsilon = 2$–17.7, $\delta = 10^{-6}$), achieving 24.94% accuracy at $\epsilon = 8.84$ vs. 25.43% baseline, demonstrating compatibility with modern DL practices. While promising, comprehensive empirical comparisons with DP-SGD across diverse tasks remain limited.

Practical implications: DP's utility impact depends not only on privacy parameters (ϵ, δ) but also on model architecture, dataset characteristics, and task complexity, making deployment decisions highly context-dependent. For practitioners, three pathways exist depending on available resources. (1) Standard DP-SGD suits settings without auxiliary data or pretrained models but should be benchmarked against reported Pareto frontiers to calibrate expectations. (2) Public data strategies offer moderate gains and are most beneficial at strong privacy levels ($\epsilon = 1$). (3) Pretrained model fine-tuning provides the most effective accuracy-privacy trade-off and is well-suited when access to high-quality pretrained representations is compatible with the application's trust model. Beyond the choice of training strategy, domain- and architecture-specific considerations apply: graph-based applications should select privacy granularity (node-level vs. edge-level) according to the threat model, and NLP applications should prefer embedding-level perturbation over gradient-based approaches. DP training is unsuitable for safety-critical tasks where accuracy thresholds are strict. Chest radiography classification, for example, may not tolerate the utility loss from DP training, while segmentation tasks with inherent noise robustness can accommodate privacy protection. Finally, translating application-level privacy requirements into concrete (ϵ, δ) parameters remains an open research challenge, with emerging methods relating privacy budgets to adversarial success probabilities (Pankova and Laud 2022) and developing interpretability frameworks (Mehner et al. 2021; Nanayakkara et al. 2023).

7.1.2 Synthetic Data. Synthetic data is an upcoming trend for PPDL applications. Generative adversarial networks (GANs) are increasingly being used for synthetic data generation with DP protection for sensitive training data.

GANObfuscator (C. Xu et al. 2019) adds noise to the discriminator gradients during training for image synthesis and applies semi-supervised classification on MNIST ($\epsilon = 2$–8, $\delta = 10^{-5}$). Sensitive medical data are generated under privacy protection, e.g., longitudinal patient data (D. Lee et al. 2020) for EHR diagnosis prediction using LSTM (recall 44.8 vs. 46.2 non-private, $\epsilon = 1.06$, $\delta = 0.001$) or EEG signals for motor imagery classification

(Debie et al. 2020) using PP-GAN (accuracy drops from 77.9% to 72.1%, $\epsilon = 0.01$, $\delta = 10^{-3}$). Hu et al. (2024) give a comprehensive overview of privacy-preserving data synthesis, unifying statistical and DL-based methods.

Instead of applying noise directly into the GAN training process, DPGOMI (D. Chen, Cheung, et al. 2023) first maps the training data to a latent space before introducing them into a differentially private GAN for image generation on CIFAR-10 and SVHN ($\epsilon = 1$–50, $\delta = 10^{-5}$). At $\epsilon = 10$, classification accuracy drops from baseline 86% to 73% on CIFAR-10.

Training on synthetic data generated under DP offers an alternative to DP-SGD, potentially improving utility by avoiding direct noise injection into model parameters. However, the quality of downstream models depends on the fidelity of the synthetic data, which may poorly capture important correlations. Rosenblatt et al. (2020) show that the effectiveness of different DP generative models depends on the privacy budget and dataset, with some methods outperforming DP-SGD in specific cases. This highlights a future research direction: determining under what conditions synthetic data may consistently offer advantages over DP-SGD.

7.1.3 Summary. Local training with privacy-preserving techniques centers on DP as the primary defense against privacy attacks targeting deployed models. Cryptographic approaches are not suitable for this single-party scenario since no collaboration is required. DP-SGD provides formal privacy guarantees but introduces utility loss, particularly at strong privacy levels ($\epsilon < 2$), with CIFAR-10 accuracy consistently below 65%.

Three approaches emerge with varying applicability: Standard DP-SGD variants achieve moderate improvements through architectural modifications but remain constrained by fundamental noise-utility trade-offs. Public data strategies provide limited benefits except at specific privacy budgets ($\epsilon = 1$). Pretrained model fine-tuning achieves the best accuracy–privacy trade-offs, reaching over 95% accuracy even at low privacy budgets ($\epsilon \in [0.1, 0.5]$), though privacy protection only applies to the fine-tuning phase.

Domain-specific applications reveal varying utility impacts: GNNs exhibit substantially higher utility costs under node-level than edge-level DP, consistent with the former's strictly stronger privacy guarantee. Medical applications show task-dependent sensitivity. For practitioners, pretrained model fine-tuning provides the most effective strategy when compatible with trust models, while carefully tuned DP-SGD serves as baseline protection with expected accuracy drops of 10–35%. Translating privacy requirements into specific (ϵ, δ) parameters remains challenging and requires domain-specific evaluation.

Q11: How to translate privacy requirements of DL applications to DP parameters?

Q12: Under what conditions and for which datasets does differentially private synthetic data consistently provide better utility than DP-SGD?

7.2 T2: Training-as-a-Service

The effectiveness of DL models relies on the quality and availability of sufficient training data, computational power, and domain knowledge. When data owners lack computational resources or expertise, they can outsource training to third parties, termed Training-as-a-Service (TaaS). In this scenario, sensitive training data samples must be transferred to external training providers, making the protection of raw training data during transmission and computation the primary privacy challenge. After training, models are returned to data owners or remain at training servers in encrypted form.

Threat Model Adaptations: T2 introduces external server trust concerns where training providers may attempt to extract individual training data samples during outsourced computation. Key vulnerabilities include training data reconstruction from intermediate computations and unauthorized access to sensitive data during processing.

Example: Internet of Things (IoT) model training – IoT devices typically generate large amounts of data but have limited memory and computing power, hence, they have to outsource DL model training. If the data is sensitive, the external party should not receive the data in clear text.

7.2.1 Homomorphic Encryption. HE for outsourced training faces fundamental computational challenges, leading to two research directions: full model training and transfer learning approaches.

(1) Training entire models: Early works demonstrate feasibility but reveal practical limitations. CryptoDL (Hesamifard et al. 2018) trains fully connected networks on encrypted data by approximating activations with low-degree polynomials. CryptoDL was evaluated on server-class hardware (12-core CPU, 48GB RAM). Without bootstrapping, the system requires data owners to be online for periodic decryption, creating communication bottlenecks. MNIST training achieves 95.1% accuracy but requires 2.9 hours per iteration, while CIFAR-10 reaches 91.5% accuracy. PrivFT (Al Badawi et al. 2020) addresses communication overhead through non-interactive training using CKKS bootstrapping. However, computational costs remain prohibitive: training a 2.5M parameter spam classifier requires 5.04 days on GPU with 80-bit security. PrivFT was evaluated on server-class hardware requiring 120GB RAM for the spam classification task. Nandakumar et al. (2019) reduce MNIST training from 1.5 days to 40 minutes via parallelization on high-performance server hardware (dual-socket Intel Xeon, 32 cores, 250GB RAM), though sacrificing model capacity and input fidelity.

(2) Transfer learning and fine-tuning: Computational burden drives adoption of transfer learning: non-sensitive pretraining followed by encrypted fine-tuning. PrivGD (Jin et al. 2020) fine-tunes only final layers on encrypted data for machine fault diagnosis, completing ten epochs in 3.85 hours with 80-bit security. PrivGD was evaluated on server hardware (Intel Xeon Platinum 8170, 2.10GHz, 26 cores, 188GB RAM). Glyph (Lou, B. Feng, et al. 2020) fine-tunes multiple layers using scheme-switching (TFHE for lookups, BGV for arithmetic). Applied to skin cancer classification, the system requires seven days for 15 epochs with 80-bit security. Glyph was evaluated on high-performance server hardware (Intel Xeon E7-8890 v4, 2.2GHz, dual-socket, 24 threads per socket, 256GB DRAM). HETAL (S. Lee et al. 2023) achieves state-of-the-art 128-bit security while completing fine-tuning within one hour using transformer architectures and early stopping across diverse datasets. HETAL was evaluated on server hardware (Intel Xeon Gold 6248, 2.50GHz, 64 threads, Nvidia Ampere A40 GPU) with GPU-accelerated HE library support.

7.2.2 Differential Privacy. DP research for outsourced training remains limited due to challenges in third-party training scenarios. W. Xu et al. (2023) address outsourced GNN training through feature matrix perturbation, reporting accuracies approaching non-private baselines but relying on exceptionally high privacy budgets ($\epsilon = 30$). Even at $\epsilon = 10$, half their experiments show 10% utility reduction despite weak privacy guarantees.

The scarcity of DP research, combined with these limitations, suggests challenges in TaaS. Noise injection conflicts with the accuracy expectations of outsourced services, while coordinating privacy budgets between data owners and training providers introduces complexity absent in local training.

7.2.3 Summary. Training-as-a-Service with privacy protection relies primarily on HE, as MPC typically requires multiple non-colluding servers (addressed in Section 7.3). Full HE training faces prohibitive costs, requiring hours to days for simple models with sub-128-bit security. Transfer learning emerges as the practical alternative, enabling encrypted fine-tuning while maintaining reasonable times and achieving 128-bit security. MPC frameworks handling T3 (collaborative learning) automatically support T2 as a special case, making dedicated T2 MPC implementations redundant. Progress from CryptoDL's communication-heavy approach to HETAL's efficient transformer fine-tuning shows advancement toward practical deployment, though computational overhead remains the primary barrier. DP shows limited applicability due to challenges between noise injection and accuracy expectations.

Practical implications: Transfer learning with HE represents the most viable approach for sensitive data outsourcing. Organizations should prioritize 128-bit security systems and allocate computational budgets for multi-hour to multi-day training times. This approach works best when pretraining data is non-sensitive and only fine-tuning requires privacy protection. Full HE training remains computationally prohibitive for production deployment regardless of data modality and is unsuitable unless milestones in HE efficiency are achieved that substantially reduce current training times.

Synthetic data generation can provide additional T2 protection using DP mechanisms, though research focuses mainly on local training (T1) with limited exploration of outsourced training contexts.

Q13: What milestones need to be achieved to train models using large datasets encrypted under HE in reasonable time and with appropriate security strength, e.g., 128 bits?

Q14: Using HE, it is difficult to implement gradient clipping. How to efficiently implement DP-SGD in this scenario?

7.3 T3: Collaborative Learning

In this scenario, multiple data owners collaboratively train a model on their joint data without disclosing private data to other parties, possibly with untrusted computation servers. This approach trains more accurate models on collective knowledge while preserving training data privacy. The key characteristic of collaborative learning is that model parameters, gradients, or other model updates are exchanged between participants rather than raw training data. Upon completion, all data owners receive the final model.

Threat Model Adaptations: T3 extends vulnerabilities to multi-party collaboration where participants must avoid collusion while protecting individual training contributions from model parameter and gradient exchanges. Key concerns include gradient-based reconstruction attacks, model inversion through shared parameters, and information leakage through communication patterns of model updates.

Example: Collaborative medical model training – Hospitals often disallow medical records to leave their premises, demanding collaborative model training without sharing sensitive patient data. This enhances healthcare quality, safeguards patient privacy, and ensures that institutions remain in compliance with data protection laws (Kaissis et al. 2021; Senanayake et al. 2022).

7.3.1 Secure Multi-Party Computation. MPC distributes training data via secret shares across n independent, non-colluding servers executing nPC protocols that compute model shares for reconstruction. While achieving good performance, data owners must trust server operators. Establishing such settings remains challenging.

Table 5 summarizes performance for various nPC frameworks across benchmark models and datasets. FCNN refers to fully connected networks with two hidden layers of 128 neurons each (Mohassel and Y. Zhang 2017). CNN refers to a network with one convolution and two fully connected layers (Riazi, Weinert, et al. 2018). LeNet, AlexNet, and VGG-16 are described in Section 5.1.2. These frameworks build upon each other as depicted in Figure 4, revealing key trends despite hardware and network differences.

Early two-party solutions like SecureML (Mohassel and Y. Zhang 2017) rely on activation approximations with limited efficiency, while subsequent frameworks like QUOTIENT (Agrawal et al. 2019) and LEGO (Z. Zhou et al. 2022) improve performance through weight ternarization and GPU acceleration. Keller and Sun (2022) bridge 2PC feasibility and 3PC performance gaps.

Three-party protocols like SecureNN (Wagh, Gupta, et al. 2019) and ABY3 (Mohassel and Rindal 2018) combine arithmetic and binary secret sharing to reduce communication. SecureNN achieves the highest MNIST accuracy

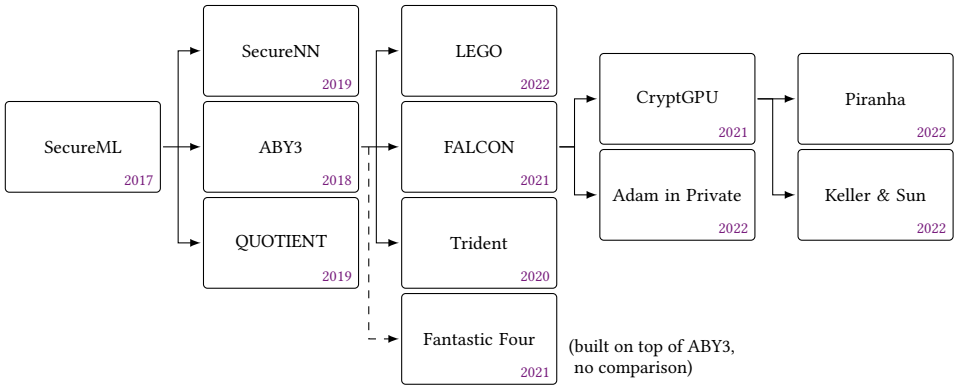


Fig. 4. Benchmarking relationships between frameworks of Table 5. Arrows indicate comparisons where the framework at the tip uses the root framework for comparison. Arrows are exclusively depicted for comparisons with the most recent framework. Fantastic Four does not compare against any other of the listed frameworks, but is built on top of ABY3.

with LeNet (99.15%). Falcon (Wagh, Tople, et al. 2021) reduces training time through optimized convolutional primitives.

Recent work emphasizes optimizers and training strategies: Adam in Private (Attrapadung et al. 2022) outperforms Falcon on AlexNet and VGG-16 with 3.5× and 12.1× speedups, while CryptGPU (Tan et al. 2021) achieves 20–30× speedups via GPU exploitation compared to Falcon across AlexNet and VGG-16 on CIFAR-10. Four-party frameworks like Fantastic Four (Dalskov et al. 2021) and Trident (Chaudhari et al. 2020) demonstrate additional throughput gains with stronger trust assumptions. Piranha (J.-L. Watson et al. 2022) provides general-purpose GPU compilation for cross-framework comparisons.

Three key trends emerge in *n*PC collaborative training: two-party to three-party transitions improve computational efficiency and model accuracy through combined arithmetic and binary secret sharing (SecureNN (Wagh, Gupta, et al. 2019), ABY3 (Mohassel and Rindal 2018)); GPU acceleration becomes essential for larger architectures, with 20–30× performance improvements (LEGO (Z. Zhou et al. 2022), CryptGPU (Tan et al. 2021)); and security-performance trade-offs become pronounced, with maliciously secure variants incurring 2–4× slowdown compared to honest-but-curious assumptions.

7.3.2 Federated Learning. FL eliminates non-colluding server requirements but introduces communication overhead and privacy vulnerabilities. In standard FL, clients perform local training and transmit updates to central servers for aggregation (B. McMahan et al. 2017). While initially considered privacy-preserving, research demonstrates that sensitive data can be reconstructed from model updates by passive adversaries (Boenisch, Dziedzic, et al. 2023; Lycklama et al. 2023).

Privacy protection in FL requires additional safeguards. Gradient pruning offers effective defense by transmitting only gradient subsets, thereby reducing privacy exposure and communication costs. Alternative approaches modify training parameters or incorporate cryptographic techniques and DP, though introducing computational overhead and accuracy trade-offs (Melis et al. 2019; L. Zhu et al. 2019).

Early cryptographic FL implementations use additive HE for secure gradient aggregation. Trieu Phong et al. (2017) require secure channels between server and clients when data owners share encryption keys, while Ma et al. (2018) introduce double encryption for server-only decryption. Y. Li, Lai, et al. (2024) address shared key vulnerabilities through multi-key HE schemes with increased complexity. Recent approaches like ScionFL (Ben-Itzhak et al. 2024) leverage MPC for secure aggregation on quantized inputs, while Lotto (Z. Jiang et al.

Table 5. Overview of *n*PC collaborative learning frameworks over different model architectures and datasets for the LAN setting (a) and the WAN setting (b). Honest but curious setting in the top part, malicious security in the bottom part. For each model and dataset, the left column states the training time in minutes per epoch unless otherwise stated, the middle column the maximum accuracy, and the right column the number of epochs to reach this accuracy. The last column gives details about the model's activation function for SecureML (Mohassel and Y. Zhang 2017), the used optimizer for Keller and Sun (2022), and the training batch size for Trident (Chaudhari et al. 2020). Highest measured accuracies per model and dataset are highlighted in bold.

(a) LAN setting

Framework	n	FCNN (MNIST) t	acc	e	CNN (MNIST) t	acc	e	LeNet (MNIST) t	acc	e	AlexNet (CIFAR-10) t	acc	e	VGG-16 (CIFAR-10) t	acc	e	details (see caption)
SecureML (Mohassel and Y. Zhang 2017)	2	28.1	0.9340	15		-			-			-			-		ReLU
		23.4	0.9310	15		-			-			-			-		square
QUOTIENT (Agrawal et al. 2019)	2	523.2	0.9504	10		-			-			-			-		
LEGO (Z. Zhou et al. 2022)	2	135.3	0.9440	15		-		14.4	0.9800	10		-			-		
Keller and Sun (2022)	2	62.4	0.9780	15		-			-			-			-		SGD
		94.8	**0.9810**	15		-			-			-			-		AMSGrad
SecureNN (Wagh, Gupta, et al. 2019)	3	4.1	0.9340	15		-		119.8	**0.9915**	15		-			-		
ABY3 (Mohassel and Rindal 2018)	3	3.0	0.9400	15	60.0‖*	0.9900*			-			-			-		
Falcon (Wagh, Tople, et al. 2021)	3	0.7	0.9742	15	1.7	**0.9781**	15	24.2	0.9685	15	52.6			2,023.2			
Adam in Private (Attrapadung et al. 2022)	3	2.0	0.9564	1		-			-		52.3	0.6000*	2*	719.2	0.7000*	5*	
CryptGPU (Tan et al. 2021)	3		-			-		17.3	0.9397	5	75.8	**0.5960**	1	12.1†			
Fantastic Four (Dalskov et al. 2021)	3	0.5	0.9500	20		-			-			-			-		
Piranha (J.-L. Watson et al. 2022)	≥ 2	0.8	0.9656	10		-		7.2	0.9710	10	11.1	0.4071	10	198.0	**0.5426**	10	
Keller and Sun (2022)	3	0.5	0.9780	15	2.4	0.9800	15	5.7	0.9850	5	26.7	**0.6490**	10		-		SGD
		1.5	**0.9810**	15	3.1	**0.9860**	15	8.6	**0.9900**	5	41.2	0.6360	10		-		AMSGrad
			-			-			-		40.5	0.6470	10		-		Adam
Falcon (Wagh, Tople, et al. 2021)	3	2.2			4.7			48.8			188.0			7,320.0			
Adam in Private (Attrapadung et al. 2022)	3	9.5	0.9561	1		-			-		250.4	0.6000*	2*	2,691.4	0.7000*	5*	
Fantastic Four (Dalskov et al. 2021)	3	7.5	0.9500	20		-			-			-			-		
	4	0.7	0.9500	20		-			-			-			-		
Trident (Chaudhari et al. 2020)	4	23.0‡			10.5‡				-			-			-		128
		13.6‡			5.6‡				-			-			-		256
		7.7‡			3.0‡				-			-			-		512

‖ total runtime in minutes; ∗ estimated value (not measured); † runtime in seconds/iteration; ‡ runtime in iterations/second

(b) WAN setting

Framework	n	FCNN (MNIST) t	acc	e	CNN (MNIST) t	acc	e	LeNet (MNIST) t	acc	e	AlexNet (CIFAR-10) t	acc	e	VGG-16 (CIFAR-10) t	acc	e	details (see caption)
QUOTIENT (Agrawal et al. 2019)	2	3044.4	0.9504	10		-			-			-			-		
LEGO (Z. Zhou et al. 2022)	2	2075.2	0.9440	15		-			-			-			-		
SecureNN (Wagh, Gupta, et al. 2019)	3	31.3	0.9340	15		-		368.0	**0.9915**	15		-			-		
Falcon (Wagh, Tople, et al. 2021)	3	15.0	**0.9742**	15	13.6	**0.9781**	15	74.0	0.9685	15	155.3			5,016.0			
Adam in Private (Attrapadung et al. 2022)	3	75.6	0.9558	1		-			-		830.6	0.6000*	2*	5,798.8	0.7000*	5*	
Falcon (Wagh, Tople, et al. 2021)	3	32.0			30.0			208.0			482.7			12,624.0			
Adam in Private (Attrapadung et al. 2022)	3	191.9	**0.9561**	1		-			-		2,663.0	0.6000*	2*	21,553.8	0.7000*	5*	
Trident (Chaudhari et al. 2020)	4	0.232‡			0.231‡				-			-			-		128
		0.232‡			0.228‡				-			-			-		256
		0.230‡			0.219‡				-			-			-		512

‖ total runtime in minutes; ∗ estimated value (not measured); † runtime in seconds/iteration; ‡ runtime in iterations/second

2024) targets malicious server settings through secure participant selection to prevent adversarial servers from strategically forming dishonest majorities. W. Du et al. (2023) advance threshold-HE schemes maintaining security below specified corruption thresholds.

7.3.3 Differentially Private Federated Learning (DP-FL). DP integration addresses FL privacy vulnerabilities while providing communication efficiency via gradient pruning. DP-FL research optimizes privacy-utility trade-offs while maintaining federated communication advantages.

Foundational algorithmic work has relaxed core assumptions of DP training in FL settings. Kairouz et al. (2021) propose DP-FTRL, an alternative to DP-SGD that achieves comparable privacy-utility trade-offs without requiring privacy amplification by sampling or shuffling.

Beyond foundational optimization, gradient compression and pruning combine privacy protection with reduced communication costs. Kerkouche et al. (2021) achieve 78% accuracy ($\epsilon = 1$) on FashionMNIST with 95% communication reduction through transmitting only low-frequency gradients. Medical applications show similar patterns, with in-hospital mortality prediction maintaining 69% accuracy ($\epsilon = 1$) versus 74% for non-private FL. SignDS-FL (X. Jiang et al. 2022) introduces Bernoulli sampling for gradient selection, achieving 4% accuracy loss at $\epsilon \geq 8$ on FashionMNIST while transmitting only indices to eliminate clipping requirements. Y. Wang et al. (2018) propose ternary gradient quantization for credit card fraud detection, achieving non-private AUC for $\epsilon = 0.5$–1. This strategy inherently provides defense against reconstruction attacks (Rigaki and Garcia 2023).

Client selection mechanisms address FL's distributed challenges. LDP-Fed (Truex et al. 2020) incorporates distance-based privacy requirements, achieving 86.8% accuracy using α-CLDP for FashionMNIST. SecProbe (Zhao et al. 2019) handles unreliable participants through utility-score assessment, achieving centralized accuracy with $\epsilon = 10$/epoch. Specialized architectures extend DP-FL: Tang et al. (2024) and X. Zhu et al. (2023) advance GNNs using electronic health records, while L. Liu et al. (2023) implement BNNs.

Domain-specific implementations reveal varying privacy-utility relationships. Stephanie et al. (2022) demonstrate transfer learning for medical imaging using lung cancer datasets, though scalability limitations emerge with increasing model size and participant count. Vertical FL variants like AdaVFL (Errounda and Y. Liu 2023) and FedVS (S. Li et al. 2023) address feature distribution scenarios while providing model privacy through MPC integration.

7.3.4 DP-FL with HE and/or MPC. Gong et al. (2020) extend Trieu Phong et al. (2017) by adding DP noise to gradients before server upload for MNIST and SVHN classification ($\epsilon = 1$–10), achieving 96.87%–96.99% on MNIST versus 96.92% baseline at $\epsilon = 10$. FedOpt (Asad et al. 2020) achieves 99.6% MNIST accuracy at $\epsilon = 1.0$ by transferring means of top or bottom gradient fractions. FedBoosting (H. Ren et al. 2024) uses Paillier encryption for visual text recognition, dropping from 94.34% to 92.23% with DP.

Lyu (2020) employ three-layer encryption for gradient protection: lightweight additive HE, symmetric encryption with data owner keys, and server public key encryption. They achieve accuracies of 89%–97% on MNIST and 66%–72% on CIFAR-10 for ϵ between 0.5 and 8 ($\delta = 10^{-5}$).

Pencil (X. Liu, Z. Liu, et al. 2024) is a privacy-preserving collaborative training framework for MNIST, CIFAR-10, and AGNews text classification ($\epsilon = 11.4$–114, $\delta = 10^{-5}$) that ensures both data and model privacy without relying on the assumption of non-colluding parties. At $\epsilon = 11.4$, test accuracy for CIFAR-10 drops by 14.6% for CNN, while ResNet drops by 4.8%. The framework integrates cryptographic primitives from HE and MPC, along with the addition of DP noise to protect sensitive information. In ∝MDL (X. Zhang, Ji, et al. 2017), each party holds a secret share generated by a Shamir (m, n)-threshold scheme for MNIST and SVHN classification ($\epsilon = 0.01$, 0.1, 1). Local gradients are encrypted using ElGamal encryption together with the secret share. When combining at least m shares at the server, the decryption results in a meaningful aggregation, while protecting individual updates. Additionally, gradients are perturbed using DP to account for membership inference attacks. Training fails to converge for SVHN at $\epsilon \leq 0.1$ but achieves reasonable accuracy at $\epsilon = 1$. Tian et al. (2022) propose a federated deep metric learning approach for user authentication using smartphone touch patterns and photos (TouchAuth: $\epsilon = 4$–8, CameraAuth: $\epsilon = 0.25$–1). TouchAuth achieves 80.55% accuracy at $\epsilon = 8$ vs. 80.88% baseline, while CameraAuth maintains 78.87% at $\epsilon = 1$ vs. 79.98% baseline. Data owners locally compute embeddings, which are collected by the server to compute the batch loss within a TEE, randomly excluding some embeddings. Gradients are then computed with added DP noise and returned to the clients for further gradient computation in the initial layers.

Medical applications include PriMIA (Kaissis et al. 2021) for pediatric chest radiograph classification using ResNet18 ($\epsilon = 6, \delta = 1.9 \cdot 10^{-4}$), achieving 85% versus 89% baseline accuracy, and FedMed-GAN (J. Wang, G. Xie, et al. 2023) for brain image synthesis maintaining baseline PSNR with $\epsilon = 28.83$–200.47 ($\delta = 10^{-5}$).

Real-world applications require handling both horizontal and vertical partitioning simultaneously. The U.S.-U.K. PETs Prize Challenge (Asif et al. 2024) exemplified this in financial crime detection. PV4FAD (Kadhe et al. 2023) addresses mixed partitioning through comprehensive frameworks combining HE, MPC, and DP, providing input privacy through cryptographic methods while ensuring output privacy against inference attacks. Healthcare (C. Yu et al. 2025) and load prediction applications (H. Liu, X. Zhang, et al. 2023) demonstrate hybrid FL across diverse partitioning scenarios.

7.3.5 Decentralized Solutions. Decentralized FL eliminates single points of failure by removing central aggregation servers, distributing coordination across participating parties. These approaches address centralized FL trust concerns while introducing communication complexity and coordination challenges.

Poseidon (Sav et al. 2021) demonstrates comprehensive decentralized training using multi-party HE within tree-structured coordination. The root party coordinates encrypted weight distribution while participants perform local gradient descent on encrypted data. The framework resists collusion among up to $N - 1$ parties with automatic cryptographic parameter optimization.

Peer-to-peer architectures offer direct party communication through secret sharing protocols. NeuroCrypt (Senanayake et al. 2022) distributes gradient shares among all participants, eliminating server dependencies with increased communication overhead. Medical imaging applications (brain tumor classification) demonstrate feasibility, though execution times scale with participant count.

Hybrid approaches combine multiple privacy techniques for complex threat models. The PEA protocol (Ruan et al. 2023) integrates MPC with DP through two-stage processing: feature extraction with local DP-SGD training, followed by MPC-based global aggregation. GNN applications extend decentralization to specialized architectures, with M. Lee et al. (2024) demonstrating privacy-preserving decentralized inference and training for graph-based learning tasks.

7.3.6 Summary. MPC-based solutions provide strong cryptographic guarantees but require a non-colluding server infrastructure. Computational and communication overhead scales with participant count, though GPU acceleration shows improvement. Traditional FL reduces infrastructure requirements while introducing communication bottlenecks during heterogeneous training. DP integration addresses collaborative learning privacy vulnerabilities but introduces accuracy degradation scaling with privacy strength. Gradient pruning and selective aggregation jointly optimize communication efficiency and privacy protection, though optimal trade-offs remain application-dependent. Combining DP with cryptographic methods provides comprehensive protection with increased implementation complexity. Decentralized architectures eliminate single points of failure but introduce coordination complexity and communication overhead. Tree-structured coordination and peer-to-peer secret sharing offer different approaches with distinct scalability characteristics. Research gaps include transformer and GNN support, optimization for larger participant counts, and standardized evaluation frameworks beyond CIFAR-10.

Practical implications: MPC-based solutions suit environments with non-colluding server access and high security requirements, while traditional FL enables easier deployment with central server coordination but needs additional privacy protection mechanisms. DP integration addresses privacy vulnerabilities but introduces accuracy trade-offs scaling with privacy strength. Gradient pruning and selective aggregation jointly optimize communication efficiency and privacy protection. Consider decentralized architectures when eliminating single points of failure is critical, accepting increased coordination complexity. Hybrid frameworks balance implementation complexity against enhanced privacy guarantees for specific application requirements. MPC-based solutions suit environments with non-colluding server access and high security requirements but become unsuitable

as participant counts grow or communication overhead exceeds deployment budgets. Traditional FL reduces infrastructure requirements and suits easier-deployment scenarios with central server coordination, but requires additional privacy protection mechanisms to defend against gradient reconstruction attacks. DP integration addresses these vulnerabilities at the cost of accuracy trade-offs that scale with privacy strength. Gradient pruning and selective aggregation can jointly optimize communication efficiency and privacy, and are particularly valuable in bandwidth-constrained settings. Decentralized architectures are well-suited when eliminating single points of failure is critical, at the cost of increased coordination complexity. Hybrid frameworks offer the most comprehensive protection but impose the highest implementation burden, and are suitable for applications where regulatory or threat model requirements demand coverage across multiple attack vectors.

Synthetic data generation can provide additional help in T3 scenarios using DP mechanisms for privacy-preserving datasets, though research focuses mainly on local training (T1) with limited multi-party collaborative context exploration.

Q15: At the time of our May 2024 literature cutoff, we did not identify any MPC-based implementations for collaborative training of transformer models using private data. While recent federated learning approaches have emerged for LLMs (see Section 8), can MPC protocols be adapted to efficiently support transformer architectures in collaborative training scenarios?

7.4 T4: Commissioned Training

Commissioned training represents a specialized scenario where multiple data owners contribute sensitive datasets to train a model that is ultimately controlled by a third party, such as a service provider or a designated model owner. Similar to T2, the primary privacy challenge involves protecting raw training data samples during transfer from multiple data owners to the commissioning party. Unlike traditional collaborative learning, participating data owners do not retain access to the final trained model, creating unique privacy and access control requirements.

Threat Model Adaptations: Commissioned training faces unique adversarial scenarios where the commissioning party could potentially compromise contributor privacy through analytical attacks on received training data samples. The threat model must account for data aggregation patterns, selective data access, and potential collusion between the commissioning party and external attackers seeking to reverse-engineer individual contributors' datasets.

Example: Predictive maintenance – Machine manufacturers provide predictive maintenance services by training DL models on operational data from multiple client facilities. While clients benefit from improved maintenance predictions, their process data contains proprietary information that cannot be shared directly. The manufacturer requires access to the trained model for service delivery, but individual clients must not access models trained on competitors' data.

Technical approaches: Commissioned training builds upon collaborative learning techniques but requires additional mechanisms for selective model access control. MPC protocols naturally support this through secret sharing, where non-colluding servers can transfer final model shares directly to the commissioning party (Keller and Sun 2022) or maintain distributed shares under third-party control for inference queries. HE frameworks enable model transfer through key-switching operations, re-encrypting model parameters under the commissioning party's public key as demonstrated in Poseidon (Sav et al. 2021).

Limited research explores alternative architectures: Arachchige et al. (2019) propose split learning, where data owners process convolutional layers locally and transmit randomized features through DP-style bit flipping

before server-side training. The DeepGuess framework (Tchaye-Kondi et al. 2023) advances this direction by training feature extractors without requiring public datasets, though evaluation remains at a proof-of-concept level.

Synthetic data generation can additionally help in T4 commissioned training scenarios, typically using DP mechanisms where data owners generate privacy-preserving synthetic datasets for model commissioning. However, current research focuses mostly on the local training scenario (T1), with limited exploration of synthetic data approaches specifically designed for commissioned training contexts.

Practical implications: Commissioned training is well-suited to commercial scenarios where multiple data owners contribute to a centrally controlled model, such as financial risk modeling, healthcare consortium research, or supply chain optimization, and where access control and contribution verification can be contractually established. It is unsuitable where current MPC or HE implementations cannot accommodate the required update protocols or verification mechanisms, as current implementations remain largely theoretical.

In summary, commissioned training represents an economically important but technically underdeveloped area requiring solutions that address unique challenges in access control, incentive alignment, and long-term protocol maintenance beyond current collaborative learning research.

8 Discussion

Our systematic review reveals substantial differences in maturity across the eight use case scenarios, with clear patterns emerging between inference and training applications.

Inference scenarios demonstrate varying levels of practical readiness. Use cases I1 (deep-learning-as-a-service) and I2 (multiple data owners) have achieved practical feasibility through mature HE and MPC implementations (Koti et al. 2021; L. K. Ng and Chow 2021; Tan et al. 2021). These scenarios benefit from well-established cryptographic protocols and computational optimizations that enable real-world deployment. In contrast, use case I3 (multiple model owners) presents ongoing challenges in balancing data owner privacy with inference efficiency, requiring more sophisticated protocols that protect input data while maintaining computational performance. Most notably, use case I4 (multiple data owners and multiple model owners) remains completely unexplored in current literature. This research gap becomes increasingly problematic as real-world applications evolve toward complex multi-party scenarios where both data and model ownership are distributed. Future applications will likely require privacy-preserving methods capable of managing these distributed ownership patterns while maintaining practical performance characteristics.

Practical adoption of inference scenarios remains uneven, but is no longer confined to research prototypes. Apple's Enhanced Visual Search uses HE to match an on-device image embedding, produced by a CNN, against a server-side landmark database without exposing the embedding to the server[5]. In healthcare, a home-monitoring system for fall and action detection applies HE to run a neural network directly on encrypted skeleton and pose data, allowing a cloud service to perform inference without accessing the raw video (M. Kim et al. 2022). At Meta, the CrypTen framework applies MPC to run standard DNNs, on encrypted inputs by splitting computation across non-colluding parties (Knott et al. 2021). These deployments concentrate almost exclusively in scenarios I1 and I2, mirroring the concentration of publications on these two categories in Table 2. Adoption also follows a pattern of regulatory influence, with healthcare and finance applications driving deployment due to stringent privacy requirements.

Training scenarios exhibit a different maturity landscape shaped by fundamental privacy-utility trade-offs. Use case T1 (local training with secure deployment) has demonstrated feasible implementations, though an unavoidable trade-off between data privacy and model accuracy persists across all approaches (see Fig. 3). The most effective strategy involves leveraging public data for pretraining followed by fine-tuning on sensitive data,

[5] https://machinelearning.apple.com/research/homomorphic-encryption (Accessed: 2026-07-17)

Table 6. Overview of privacy-preserving technologies and techniques across PPDL use cases for inference (I1-I4) and training (T1-T4) found in the literature, including support for different neural network architectures. Technologies that are not applicable in scenarios are indicated by "n.a.". Used privacy-preserving technologies, optimization techniques, and model types from the literature are marked with ●, while unsupported ones use ○. In the last row, ● indicates practical performance; ◐ means the use case is partly implemented efficiently but may need further research; ○ means no research has been done.

		I1	I2	I3	I4	T1	T2	T3	T4
Privacy-Preserving Technology	HE	●	●	●	○	n.a.	●	●	●
	MPC	●	●	●	○	n.a.	●*		●
	DP	●	●	●	○	●	●	●	○
	FL	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	●	n.a.
	TEE	●	○	○	○	n.a.	○	○	○
	Mixed	●	○	●	○	n.a.	○	●	○
Optimization Technique	Pruning	●	○	○	○	●	○	●	○
	Quantization	●	●	○	○	○	○	●	○
	Approximation	●	○	○	○	●	●	●	○
	Split Network	●	●	○	○	○	○	●	●
	Transfer Learning	n.a.	n.a.	n.a.	n.a.	●	●	●	○
Model Support	CNN	●	●	●	○	●	●	●	●
	Graph NN	○	○	●	○	●	●	●	○
	Transformer	●	●	○	○	●	●	○	○
Practical Performance		●	●	◐	○	●	●	●	◐

* No MPC solutions in T2, but T3 handles both.

which substantially enhances model performance compared to training from scratch with privacy constraints. Training-as-a-service (T2) currently achieves practical deployment only under restrictive assumptions: either non-colluding server availability for MPC protocols or transfer learning approaches that mitigate HE's computational overhead. Practitioners considering T2 deployments must carefully evaluate whether their infrastructure and trust models can accommodate these requirements. Collaborative learning (T3) has achieved practical implementations for MPC training, with frameworks like those developed by Keller and Sun (2022) demonstrating real-world applicability. MPC and FL approaches provide the foundation for preserving data privacy while maintaining model accuracy, though ongoing research continues to address communication overhead and privacy-accuracy optimization. Commissioned training (T4) represents a specialized variant of T3 where model access control introduces additional complexity, requiring further research to establish practical deployment frameworks.

Practical adoption on the training side is similarly uneven, but shows clear commercial traction concentrated in T3. The MELLODDY consortium used FL to train neural networks on molecular fingerprints across ten pharmaceutical companies, allowing partners to build shared drug-discovery models without exposing their proprietary assay data (Heyndrickx et al. 2024). In the EXAM project, twenty hospitals across five continents used FL to jointly train a model from chest X-rays with clinical data, to predict oxygen needs of COVID-19 patients without sharing patient records between sites (Dayan et al. 2021). Google applies FL together with DP to train the LSTM-based language model behind Gboard's next-word prediction (Z. Xu et al. 2023). This concentration of deployed systems in T3 mirrors the high publication volume for this category in Table 2.

Table 6 provides a comprehensive overview of the privacy-preserving techniques applied across inference and training use cases, as identified in the literature from our systematic review. The table also illustrates which

use cases incorporate optimization techniques. Pruning and quantization can be applied not only to models (e.g., traditional NN pruning in I1) but also to data and model updates (e.g., for model updates in T3). Approximation is commonly used by replacing non-linear functions with polynomial approximations. Split networks improve computational performance by processing the initial layers of a NN locally before offloading the remaining computation to an untrusted party, where privacy-preserving techniques are applied. Transfer learning is used to fine-tune a pretrained model using sensitive data, with the initial training performed on public datasets. Moreover, the table summarizes support for different NN architectures in the literature, including CNNs, GNNs, and transformer architectures.

Emerging Directions: LLMs and Foundation Models. Since our literature cutoff in May 2024, the PPDL landscape for large language models (LLMs) and foundation models (FMs) has evolved rapidly, though the fundamental challenges identified in our review persist with additional complexity introduced by model scale and architecture. Recent surveys (H. Li et al. 2024; Luo et al. 2025; Miranda et al. 2025) show that LLM and FM applications face both familiar PPDL challenges and novel privacy risks including enhanced memorization and extraction vulnerabilities. For private inference, cryptographic approaches (MPC and HE) continue to target transformer architectures, with recent work focusing on optimizing non-linear operations such as attention mechanisms, activation functions, and layer normalization that dominate computational overhead (W.-j. Lu et al. 2025; Pang et al. 2024). These efforts extend the protocol design and architectural optimization strategies observed in our reviewed literature. For private training of LLMs and FMs, DP continues to be the primary approach for protecting fine-tuning on sensitive data (Ding et al. 2024; Shi et al. 2022). Parameter-efficient fine-tuning methods have been used increasingly (Y. Li, W. Yu, et al. 2025). These methods reduce the number of trainable parameters, thereby decreasing the dimensionality of noise injection. FL has emerged as a complementary direction for collaborative LLM fine-tuning (X.-Y. Liu et al. 2025; Z. Zhang et al. 2023). These directions map naturally to our established taxonomy: private inference via MPC and HE extends I1 and I2 to transformer architectures, DP-based fine-tuning with LoRA addresses T1 at foundation model scale, and federated approaches extend T3 to LLMs. Future work must address the scale-specific optimization challenges and novel attack surfaces that distinguish these models from traditional DL architectures.

9 Conclusion

Our systematic literature review provides a comprehensive taxonomy of privacy-preserving deep learning (PPDL) applications, organizing research across eight distinct use case scenarios for both inference and training. This structured approach enables clear identification of mature techniques, emerging solutions, and critical research gaps that require future attention.

Newcomers: Entry into PPDL should begin with foundational scenarios that offer clear technical pathways and established evaluation frameworks. Use cases I1 (deep learning as a service) and T1 (local training with secure deployment) provide optimal starting points, combining manageable technical complexity with comprehensive literature coverage. The taxonomy presented here offers structured guidance for progressively advancing toward more complex multi-party scenarios.

Standard benchmark datasets, particularly CIFAR-10 and MNIST, serve as appropriate entry points for experimental work, though practitioners should recognize their limitations when transitioning to domain-specific applications. Healthcare, finance, and natural language processing domains present unique challenges extending beyond benchmark performance, requiring careful consideration of regulatory requirements, data characteristics, and accuracy thresholds.

Practitioners: Deployment decisions should align privacy-preserving techniques with specific use case requirements and infrastructure constraints. DP-SGD represents a widely used choice for centralized training

scenarios, providing formal privacy guarantees with well-understood utility trade-offs. MPC frameworks excel in collaborative learning environments where multiple data owners require strong cryptographic protection, though they necessitate non-colluding server infrastructure. HE proves most suitable for inference-as-a-service applications where input confidentiality is paramount and computational overhead can be accommodated.

The privacy-utility trade-off remains fundamental across all approaches, with small ϵ values providing stronger guarantees at the cost of substantial accuracy degradation. Leveraging pretrained models currently offers the most effective strategy for mitigating this trade-off, enabling near-baseline accuracy even under strong privacy constraints. System-level assumptions, including non-colluding server availability for MPC and key management protocols for HE, must be carefully evaluated against application trust models before deployment.

Deployment decisions must additionally account for the regulatory environment of the application domain. Legal frameworks such as GDPR, HIPAA, and the EU AI Act impose data protection obligations that technical privacy guarantees alone may not satisfy. For example, DP's (ϵ, δ) parameterization does not directly map to GDPR's notion of anonymization. Practitioners operating in healthcare, finance, or other regulated sectors should treat the technical taxonomy presented here as a necessary but not sufficient basis for deployment decisions, and seek legal counsel to align technique selection with applicable compliance requirements.

Domain experts: This review identifies critical research gaps limiting practical adoption in complex real-world scenarios. Use cases I4 (multiple data and model owners) and T4 (commissioned training) remain largely unexplored despite growing industry relevance. Addressing these gaps requires protocols that efficiently manage distributed ownership while maintaining practical performance.

Technical challenges persist across multiple dimensions: scaling MPC protocols beyond small participant sets, extending privacy-preserving techniques to modern architectures including transformers and GNNs, and enabling practical DP training at strong privacy levels ($\epsilon < 2$). Hybrid approaches combining MPC, HE, and DP show promise but introduce implementation complexity that requires systematic optimization.

Future research should prioritize standardized benchmarks beyond traditional datasets for meaningful cross-domain comparisons. The field requires interdisciplinary collaboration between cryptographers, DL researchers, and domain practitioners to align technical advances with real-world deployment requirements.

A central finding demonstrates that isolated privacy-preserving technologies rarely achieve practical deployment requirements. Each approach exhibits distinct strengths and limitations apparent only within specific use case contexts. The most successful implementations strategically combine multiple techniques, leveraging complementary properties to balance privacy, utility, and computational efficiency.

Recent technological developments provide encouraging evidence that current limitations can be overcome. Hardware acceleration, particularly GPU support, enables order-of-magnitude performance improvements for cryptographic protocols. Approximation techniques, model pruning, and transfer learning continue to enhance the practicality of privacy-preserving training. These advances collectively indicate PPDL is transitioning from a research possibility to a deployable technology.

The path forward requires continued innovation across algorithmic design, systems optimization, and interdisciplinary collaboration. By maintaining focus on real-world applicability while advancing theoretical foundations, the field can achieve its ultimate goal: making robust privacy guarantees integral components of production deep learning systems rather than academic considerations.

Acknowledgments

We thank the anonymous reviewers for their valuable feedback and suggestions that significantly improved the manuscript. Martin Becker is supported by the Federal Ministry of Education and Research of Germany (BMBF) under project number 01IS22077. The authors wish to thank Michael Rader, Ruben Hetfleisch, and Alessio

Montuoro for their support during the review. The financial support by the Austrian Federal Ministry of Labour and Economy, the National Foundation for Research, Technology and Development and the Christian Doppler Research Association is gratefully acknowledged.

References

M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. 2016. "Deep learning with differential privacy." In: *ACM Conference on Computer and Communications Security (CCS)*, 308–318.

N. Agrawal, A. Shahin Shamsabadi, M. J. Kusner, and A. Gascón. 2019. "QUOTIENT: Two-Party Secure Neural Network Training and Prediction." In: *ACM Conference on Computer and Communications Security (CCS)* (CCS '19), 1231–1247. ISBN: 9781450367479. DOI: 10.1145/3319535.3339819.

E. Aharoni, A. Adir, et al.. 2023. "HeLayers: A Tile Tensors Framework for Large Neural Networks on Encrypted Data." *Proceedings on Privacy Enhancing Technologies*, 2023, 1, 325–342.

E. Aharoni, M. Baruch, et al.. 2023. "Efficient Pruning for Machine Learning Under Homomorphic Encryption." In: *European Symposium on Research in Computer Security*. Springer, 204–225.

S. Ahmed, M. M. Alshater, A. El Ammari, and H. Hammami. 2022. "Artificial intelligence and machine learning in finance: A bibliometric review." *Research in International Business and Finance*, 61, 101646.

Y. Akimoto, K. Fukuchi, Y. Akimoto, and J. Sakuma. 2023. "Privformer: Privacy-preserving transformer with MPC." In: *IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 392–410.

Y. Akın, J. Klemsa, and M. Önen. 2023. "A practical TFHE-based multi-key homomorphic encryption with linear complexity and low noise growth." In: *European Symposium on Research in Computer Security*. Springer, 3–23.

A. Al Badawi, L. Hoang, C. F. Mun, K. Laine, and K. M. M. Aung. 2020. "PrivFT: Private and fast text classification with homomorphic encryption." *IEEE Access*, 8, 226544–226556.

W. Ao and V. N. Boddeti. Aug. 2024. "AutoFHE: Automated Adaption of CNNs for Efficient Evaluation over FHE." In: *33rd USENIX Security Symposium (USENIX Security 24)*. USENIX Association, Philadelphia, PA, (Aug. 2024), 2173–2190. ISBN: 978-1-939133-44-1.

P. C. M. Arachchige, P. Bertok, I. Khalil, D. Liu, S. Camtepe, and M. Atiquzzaman. 2019. "Local differential privacy for deep learning." *IEEE Internet of Things Journal*, 7, 7, 5827–5842.

T. Aremu and K. Nandakumar. 2023. "PolyKervNets: Activation-free Neural Networks For Efficient Private Inference." In: *IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*. IEEE, 593–604.

M. Asad, A. Moustafa, and T. Ito. 2020. "FedOpt: Towards communication efficiency and privacy preservation in federated learning." *Applied Sciences*, 10, 8, 2864.

H. Asif, S. Min, X. Wang, and J. Vaidya. 2024. "U.S.-U.K. PETs Prize Challenge: Anomaly Detection via Privacy-Enhanced Federated Learning." *IEEE Transactions on Privacy*, 1, 3–18. DOI: 10.1109/TP.2024.3392721.

G. Ateniese, L. V. Mancini, A. Spognardi, A. Villani, D. Vitali, and G. Felici. 2015. "Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers." *International Journal of Security and Networks*, 10, 3, 137–150. DOI: 10.1504/IJSN.2015.071829.

N. Attrapadung, K. Hamada, D. Ikarashi, R. Kikuchi, T. Matsuda, I. Mishina, H. Morita, and J. C. Schuldt. 2022. "Adam in Private: Secure and Fast Training of Deep Neural Networks with Adaptive Moment Estimation." *Proceedings on Privacy Enhancing Technologies*, 2022, 4, 746–767.

"SoK: Cryptography for Neural Networks." *Privacy and Identity Management. Data for Better Living: AI and Privacy: 14th IFIP WG 9.2, 9.6/11.7, 11.6/SIG 9.2.2 International Summer School, Windisch, Switzerland, August 19–23, 2019, Revised Selected Papers*. Springer International Publishing, Cham, 63–81. ISBN: 978-3-030-42504-3. DOI: 10.1007/978-3-030-42504-3_5.

M. R. Bachute and J. M. Subhedar. 2021. "Autonomous driving architectures: insights of machine learning and deep learning algorithms." *Machine Learning with Applications*, 6, 100164.

S. Balla and F. Koushanfar. 2023. "HELiKs: HE Linear Algebra Kernels for Secure Inference." In: *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security* (CCS '23). Association for Computing Machinery, Copenhagen, Denmark, 2306–2320. ISBN: 9798400700507. DOI: 10.1145/3576915.3623136.

B. Balle and Y.-X. Wang. 2018. "Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising." In: *International Conference on Machine Learning (ICML)*. PMLR, 394–403.

E. Barker. May 2020. "Recommendation for Key Management: Part 1 – General." In: *NIST Special Publication 800-57 Part 1 Revision 5*. National Institute of Standards and Technology (NIST), (May 2020).

D. Beaver. 1992. "Efficient multiparty protocols using circuit randomization." In: *Advances in Cryptology – CRYPTO '92*. Springer, 420–432.

Y. Ben-Itzhak et al.. 2024. "Scionfl: Efficient and robust secure quantized aggregation." In: *2024 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*. IEEE, 490–511. DOI: 10.1109/SaTML59370.2024.00031.

M. Ben-Or, S. Goldwasser, and A. Wigderson. 1988. "Completeness theorems for noncryptographic fault-tolerant distributed computations." In: *ACM Symposium on Theory of Computing (STOC)*, 1–10.

S. Bian, Z. Zhao, Z. Zhang, R. Mao, K. Suenaga, Y. Jin, Z. Guan, and J. Liu. 2024. "HEIR: A Unified Representation for Cross-Scheme Compilation of Fully Homomorphic Computation." In: *Symposium on Network and Distributed System Security (NDSS)*.

B. Biggio and F. Roli. 2018. "Wild patterns: Ten years after the rise of adversarial machine learning." In: *ACM Conference on Computer and Communications Security (CCS)*, 2154–2156.

F. Boemer, A. Costache, R. Cammarota, and C. Wierzynski. 2019. "nGraph-HE2: A high-throughput framework for neural network inference on encrypted data." In: *ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, 45–56.

F. Boemer, Y. Lao, R. Cammarota, and C. Wierzynski. 2019. "nGraph-HE: A graph compiler for deep learning on homomorphically encrypted data." In: *ACM International Conference on Computing Frontiers*, 3–13.

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

F. Boenisch, C. Mühl, R. Rinberg, J. Ihrig, and A. Dziedzic. 2023. "Individualized PATE: Differentially Private Machine Learning with Individual Privacy Guarantees." *Proceedings on Privacy Enhancing Technologies*, 2023, 1, 158–176. DOI: 10.56553/popets-2023-0010.

A. Boulemtafes, A. Derhab, and Y. Challal. 2020. "A review of privacy-preserving techniques for deep learning." *Neurocomputing*, 384, 21–45. DOI: https://doi.org/10.1016/j.neucom.2019.11.041.

F. Bourse, M. Minelli, M. Minihold, and P. Paillier. 2018. "Fast homomorphic evaluation of deep discretized neural networks." In: *Advances in Cryptology – CRYPTO 2018*. Springer, 483–512.

A. Brutzkus, R. Gilad-Bachrach, and O. Elisha. 2019. "Low latency privacy preserving inference." In: *International Conference on Machine Learning (ICML)*. PMLR, 812–821.

Z. Bu, S. Gopi, J. Kulkarni, Y. T. Lee, H. Shen, and U. Tantipongpipat. 2021. "Fast and memory efficient differentially private-SGD via JL projections." *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 19680–19691.

M. Byali, H. Chaudhari, A. Patra, and A. Suresh. 2019. *FLASH: Fast and Robust Framework for Privacy-preserving Machine Learning*. Cryptology ePrint Archive, Paper 2019/1365. (2019). https://eprint.iacr.org/2019/1365.

J. Cabrero-Holgueras and S. Pastrana. 2021. "SoK: Privacy-preserving computation techniques for deep learning." *Proceedings on Privacy Enhancing Technologies*, 2021.

H. Chaudhari, R. Rachuri, and A. Suresh. 2020. "Trident: Efficient 4PC framework for privacy preserving machine learning." In: *Symposium on Network and Distributed System Security (NDSS)*.

C.-F. R. Chen, Q. Fan, and R. Panda. 2021. "Crossvit: Cross-attention multi-scale vision transformer for image classification." In: *Proceedings of the IEEE/CVF international conference on computer vision*, 357–366.

D. Chen, Y. Zhang, S. Kundu, C. Li, and P. A. Beerel. 2023. "RNA-ViT: Reduced-Dimension Approximate Normalized Attention Vision Transformers for Latency Efficient Private Inference." In: *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*. IEEE, 1–9.

D. Chen, S.-c. S. Cheung, and C.-N. Chuah. 2023. "DPGOMI: Differentially Private Data Publishing with Gaussian Optimized Model Inversion." In: *2023 IEEE International Conference on Image Processing Challenges and Workshops (ICIPCW)*, 1–5. DOI: 10.1109/ICIPC59416.2023.10328345.

H. Chen, H. Li, M. Hao, J. Hu, G. Xu, X. Zhang, and T. Zhang. 2024. "SecBNN: Efficient Secure Inference on Binary Neural Networks." *IEEE Transactions on Information Forensics and Security*, 19, 10273–10286.

H. Chen, I. Chillotti, and Y. Song. 2019. "Multi-key homomorphic encryption from TFHE." In: *Advances in Cryptology – ASIACRYPT 2019*. Springer, 446–472.

H. Chen, W. Dai, M. Kim, and Y. Song. 2019. "Efficient multi-key homomorphic encryption with packed ciphertexts with application to oblivious neural network inference." In: *ACM Conference on Computer and Communications Security (CCS)*, 395–412.

J. H. Cheon, A. Kim, M. Kim, and Y. Song. 2017. "Homomorphic encryption for arithmetic of approximate numbers." In: *Advances in Cryptology – ASIACRYPT 2017*. Springer, 409–437.

I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène. 2020. "TFHE: fast fully homomorphic encryption over the torus." *Journal of Cryptology*, 33, 1, 34–91.

I. Chillotti, M. Joye, and P. Paillier. 2021. "Programmable bootstrapping enables efficient homomorphic inference of deep neural networks." In: *Cyber Security Cryptography and Machine Learning*. Springer, 1–19.

C. A. Choquette-Choo, N. Dullerud, A. Dziedzic, Y. Zhang, S. Jha, N. Papernot, and X. Wang. 2021. "CaPC Learning: Confidential and Private Collaborative Learning." In: *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=h2EbJ4_wMVq.

C. A. Choquette-Choo, H. B. McMahan, K. Rush, and A. Thakurta. 2023. "Multi-epoch matrix factorization mechanisms for private machine learning." In: *Proceedings of the 40th International Conference on Machine Learning* (ICML'23) Article 236. JMLR.org, Honolulu, Hawaii, USA, 40 pages.

M. Clear and C. McGoldrick. 2015. "Multi-identity and multi-key leveled FHE from learning with errors." In: *Advances in Cryptology – CRYPTO 2015*. Springer, 630–656.

A. Dalskov, D. Escudero, and M. Keller. 2021. "Fantastic four: Honest-Majority Four-Party secure computation with malicious security." In: *30th USENIX Security Symposium (USENIX Security 21)*, 2183–2200.

I. Damgård, V. Pastro, N. Smart, and S. Zakarias. 2012. "Multiparty Computation from Somewhat Homomorphic Encryption." In: *Advances in Cryptology – CRYPTO 2012*. Ed. by R. Safavi-Naini and R. Canetti. Springer Berlin Heidelberg, Berlin, Heidelberg, 643–662. ISBN: 978-3-642-32009-5.

R. Dathathri, B. Kostova, O. Saarikivi, W. Dai, K. Laine, and M. Musuvathi. 2020. "EVA: An encrypted vector arithmetic language and compiler for efficient homomorphic computation." In: *ACM Conference on Programming Language Design and Implementation (PLDI)*, 546–561.

I. Dayan et al.. 2021. "Federated learning for predicting clinical outcomes in patients with COVID-19." *Nature medicine*, 27, 10, 1735–1743.

S. De, L. Berrada, J. Hayes, S. L. Smith, and B. Balle. 2022. *Unlocking High-Accuracy Differentially Private Image Classification through Scale*. (2022). https://arxiv.org/abs/2204.13650 arXiv: 2204.13650 (cs.LG).

E. Debie, N. Moustafa, and M. T. Whitty. 2020. "A privacy-preserving generative adversarial network method for securing EEG brain signals." In: *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. "Imagenet: A large-scale hierarchical image database." In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

G. Dessouky, F. Koushanfar, A.-R. Sadeghi, T. Schneider, S. Zeitouni, and M. Zohner. 2017. "Pushing the Communication Barrier in Secure Computation using Lookup Tables." In: *Symposium on Network and Distributed System Security (NDSS)*.

A. Diaa et al.. Aug. 2024. "Fast and Private Inference of Deep Neural Networks by Co-designing Activation Functions." In: *33rd USENIX Security Symposium (USENIX Security 24)*. USENIX Association, Philadelphia, PA, (Aug. 2024), 2191–2208. ISBN: 978-1-939133-44-1. https://www.usenix.org/conference/usenixsecurity24/presentation/diaa.

Y. Ding, X. Wu, Y. Meng, Y. Luo, H. Wang, and W. Pan. July 2024. "Delving into Differentially Private Transformer." In: *Proceedings of the 41st International Conference on Machine Learning* (Proceedings of Machine Learning Research). Ed. by R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp. Vol. 235. PMLR, (July 2024), 11049–11071. https://proceedings.mlr.press/v235/ding24g.html.

Y. Dong, C. Xiaojun, W. Jing, L. Kaiyun, and W. Wang. 2023. "METEOR: Improved Secure 3-Party Neural Network Inference with Reducing Online Communication Costs." In: *ACM Web Conference 2023*, 2087–2098.

M. Du, X. Yue, S. S. M. Chow, T. Wang, C. Huang, and H. Sun. 2023. "DP-Forward: Fine-tuning and Inference on Language Models with Differential Privacy in Forward Pass." In: *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security* (CCS '23). Association for Computing Machinery, Copenhagen, Denmark, 2665–2679. ISBN: 9798400700507. DOI: 10.1145/3576915.3616592.

W. Du, M. Li, L. Wu, Y. Han, T. Zhou, and X. Yang. 2023. "A efficient and robust privacy-preserving framework for cross-device federated learning." *Complex & Intelligent Systems*, 9, 5, 4923–4937.

C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. 2006. "Our data, ourselves: Privacy via distributed noise generation." In: *Advances in Cryptology – EUROCRYPT 2006*. Springer, 486–503.

C. Dwork, F. McSherry, K. Nissim, and A. Smith. 2006. "Calibrating noise to sensitivity in private data analysis." In: *Theory of Cryptography Conference (TCC)*. Springer, 265–284.

A. Dziedzic, C. A. Choquette-Choo, N. Dullerud, V. M. Suriyakumar, A. S. Shamsabadi, M. A. Kaleem, S. Jha, N. Papernot, and X. Wang. 2023. "Private Multi-Winner Voting for Machine Learning." *Proceedings on Privacy Enhancing Technologies*, 2023, 1, 527–555. DOI: 10.56553/popets-2023-0031.

T. ElGamal. 1985. "A public key cryptosystem and a signature scheme based on discrete logarithms." *IEEE Transactions on Information Theory*, 31, 4, 469–472.

T. van Elsloo, G. Patrini, and H. Ivey-Law. 2019. *SEALion: a Framework for Neural Network Inference on Encrypted Data*. (2019). https://arxiv.org/abs/1904.12840 arXiv: 1904.12840 (cs.LG).

F. Z. Errounda and Y. Liu. 2023. "Adaptive differential privacy in vertical federated learning for mobility forecasting." *Future Generation Computer Systems*, 149, 531–546.

European Union. Apr. 2016. *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)*. OJ L 119, 4.5.2016, p. 1–88. Accessed: 2026-07-17. (Apr. 2016). https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng.

European Union. July 2024. *Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act)*. OJ L, 2024/1689, 12.7.2024. Accessed: 2026-07-17. (July 2024). https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng.

J. Fan and F. Vercauteren. 2012. *Somewhat Practical Fully Homomorphic Encryption*. Cryptology ePrint Archive, Paper 2012/144. (2012). https://eprint.iacr.org/2012/144.

S. Fehr and C. Yuan. 2020. "Robust secret sharing with almost optimal share size and security against rushing adversaries." In: *Theory of Cryptography Conference (TCC)*. Springer, 470–498.

C. Feng, N. Xu, W. Wen, P. Venkitasubramaniam, and C. Ding. 2023. "Spectral-DP: Differentially Private Deep Learning through Spectral Perturbation and Filtering." In: *IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 1944–1960.

L. Folkerts, C. Gouert, and N. G. Tsoutsos. 2023. "REDsec: Running Encrypted Discretized Neural Networks in Seconds." In: *30th Annual Network and Distributed System Security Symposium, NDSS 2023, San Diego, California, USA, February 27 - March 3, 2023*. The Internet Society.

M. Fredrikson, S. Jha, and T. Ristenpart. 2015. "Model inversion attacks that exploit confidence information and basic countermeasures." In: *ACM Conference on Computer and Communications Security (CCS)*, 1322–1333.

R. Freivalds. 1977. "Probabilistic Machines Can Use Less Running Time." In: *IFIP congress*. Vol. 839, 842.

A. Ganesh, M. Haghifam, M. Nasr, S. Oh, T. Steinke, O. Thakkar, A. Guha Thakurta, and L. Wang. July 2023. "Why Is Public Pretraining Necessary for Private Model Training?" In: *Proceedings of the 40th International Conference on Machine Learning* (Proceedings of Machine Learning Research). Vol. 202. PMLR, (July 2023), 10611–10627. https://proceedings.mlr.press/v202/ganesh23a.html.

Q. Geng and P. Viswanath. 2016. "The Optimal Noise-Adding Mechanism in Differential Privacy." *IEEE Transactions on Information Theory*, 62, 2, 925–951. DOI: 10.1109/TIT.2015.2504967.

C. Gentry. 2009. "Fully homomorphic encryption using ideal lattices." In: *ACM Symposium on Theory of Computing (STOC)*, 169–178.

R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing. 2016. "CryptoNets: Applying neural networks to encrypted data with high throughput and accuracy." In: *International Conference on Machine Learning (ICML)*. PMLR, 201–210.

M. Gong, J. Feng, and Y. Xie. 2020. "Privacy-enhanced multi-party deep learning." *Neural Networks*, 121, 484–496.

I. Goodfellow. 2015. "Efficient per-example gradient computations." *arXiv preprint arXiv:1510.01799*, 2015.

I. Goodfellow, Y. Bengio, and A. Courville. 2016. *Deep Learning*. MIT Press.

A. Grivet Sébert, R. Pinot, M. Zuber, C. Gouy-Pailler, and R. Sirdey. 2021. "SPEED: secure, PrivatE, and efficient deep learning." *Machine Learning*, 110, 4, 675–694. DOI: 10.1007/s10994-021-05970-3.

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

E. Hesamifard, H. Takabi, M. Ghasemi, and R. N. Wright. 2018. "Privacy-preserving machine learning as a service." *Proceedings on Privacy Enhancing Technologies*, 2018, 3, 123–142.

W. Heyndrickx, N. Ansari, M. Ashford, et al.. 2024. "MELLODDY: cross-pharma federated learning at unprecedented scale unlocks benefits in QSAR without compromising proprietary information." *Journal of Computer-Aided Molecular Design*, 38, 1, 1.

F. A. Hölzl, D. Rueckert, and G. Kaissis. 2023. "Equivariant Differentially Private Deep Learning: Why DP-SGD Needs Sparser Models." In: *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security* (AISec '23). Association for Computing Machinery, Copenhagen, Denmark, 11–22. ISBN: 9798400702600. DOI: 10.1145/3605764.3623902.

S. Hong, J. H. Park, W. Cho, H. Choe, and J. H. Cheon. 2022. "Secure tumor classification by shallow neural network using homomorphic encryption." *BMC genomics*, 23, 1, 284, 284. DOI: 10.1186/s12864-022-08469-w.

Y. Hu et al.. May 2024. "SoK: Privacy-Preserving Data Synthesis." In: *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, Los Alamitos, CA, USA, (May 2024), 6–6. DOI: 10.1109/SP54263.2024.00002.

H. Huang, L. Luo, B. Zhang, Y. Xie, C. Zhang, and J. Liu. 2023. "A PATE-based Approach for Training Graph Neural Networks under Label Differential Privacy." In: *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, 3427–3432. DOI: 10.1109/GLOBECOM54140.2023.10437079.

Z. Huang, W.-j. Lu, C. Hong, and J. Ding. 2022. "Cheetah: Lean and Fast Secure Two-Party Deep Neural Network Inference." In: *31st USENIX Security Symposium (USENIX Security 22)*, 809–826. ISBN: 978-1-939133-31-1. https://www.usenix.org/conference/usenixsecurity22/presentation/huang-zhicong.

S. U. Hussain, M. Javaheripi, M. Samragh, and F. Koushanfar. 2021. "COINN: Crypto/ML codesign for oblivious inference via neural networks." In: *ACM Conference on Computer and Communications Security (CCS)*, 3266–3281.

F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. 2016. *SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size*. (2016). https://arxiv.org/abs/1602.07360 arXiv: 1602.07360 [cs.CV].

A. Ibarrondo, H. Chabanne, and M. Önen. 2021. "BANNERS: Binarized neural networks with replicated secret sharing." In: *ACM Workshop on Information Hiding and Multimedia Security*, 63–74.

T. Ishiyama, T. Suzuki, and H. Yamana. 2022. "Latency-Aware Inference on Convolutional Neural Network Over Homomorphic Encryption." In: *International Conference on Information Integration and Web*. Springer, 324–337.

M. M. Islam, N. Mohammed, Y. Wang, and P. Hu. 2022. "Differential private deep learning models for analyzing breast cancer omics data." *Frontiers in Oncology*, 12, 879607.

R. Ito, S. P. Liew, T. Takahashi, Y. Sasaki, and M. Onizuka. 2023. "Scaling Private Deep Learning with Low-Rank and Sparse Gradients." *Journal of Information Processing*, 31, 748–757.

M. Izabachène, R. Sirdey, and M. Zuber. 2019. "Practical fully homomorphic encryption for fully masked neural networks." In: *International Conference on Cryptology and Network Security*. Springer, 24–36.

N. Jain, K. Nandakumar, N. Ratha, S. Pankanti, and U. Kumar. 2021. "CryptInfer: Enabling Encrypted Inference on Skin Lesion Images for Melanoma Detection." In: *International Conference on AI-ML Systems*, 1–7.

I. Jarin and B. Eshete. 2021. "PRICURE: privacy-preserving collaborative inference in a multi-party setting." In: *ACM Workshop on Security and Privacy Analytics*, 25–35.

M. Jegorova, C. Kaul, C. Mayor, A. Q. O'Neil, A. Weir, R. Murray-Smith, and S. A. Tsaftaris. 2023. "Survey: Leakage and Privacy at Inference Time." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45, 7, 9090–9108. DOI: 10.1109/TPAMI.2022.3229593.

X. Jiang, X. Zhou, and J. Grossklags. Oct. 2022. "SignDS-FL: Local differentially private federated learning with sign-based dimension selection." *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13, 5, (Oct. 2022), 1–22.

Z. Jiang, P. Ye, S. He, W. Wang, R. Chen, and B. Li. Aug. 2024. "Lotto: Secure Participant Selection against Adversarial Servers in Federated Learning." In: *33rd USENIX Security Symposium (USENIX Security 24)*. USENIX Association, Philadelphia, PA, (Aug. 2024), 343–360. ISBN: 978-1-939133-44-1. https://www.usenix.org/conference/usenixsecurity24/presentation/jiang-zhifeng.

C. Jin, M. Ragab, and K. M. M. Aung. 2020. "Secure transfer learning for machine fault diagnosis under different operating conditions." In: *International Conference on Provable Security*. Springer, 278–297.

J. Jordon, J. Yoon, and M. Van Der Schaar. 2018. "PATE-GAN: Generating synthetic data with differential privacy guarantees." In: *International conference on learning representations*.

C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan. 2018. "GAZELLE: A low latency framework for secure neural network inference." In: *27th USENIX Security Symposium (USENIX Security 18)*, 1651–1669.

S. R. Kadhe et al.. 2023. *Privacy-Preserving Federated Learning over Vertically and Horizontally Partitioned Data for Financial Anomaly Detection*. (2023). https://arxiv.org/abs/2310.19304 arXiv: 2310.19304 [cs.CR].

P. Kairouz, B. Mcmahan, S. Song, O. Thakkar, A. Thakurta, and Z. Xu. July 2021. "Practical and Private (Deep) Learning Without Sampling or Shuffling." In: *Proceedings of the 38th International Conference on Machine Learning* (Proceedings of Machine Learning Research). Ed. by M. Meila and T. Zhang. Vol. 139. PMLR, (July 2021), 5213–5225. https://proceedings.mlr.press/v139/kairouz21b.html.

G. Kaissis et al.. 2021. "End-to-end privacy preserving deep learning on multi-institutional medical imaging." *Nature Machine Intelligence*, 3, 6, 473–484.

S. Kamara, P. Mohassel, and M. Raykova. 2011. *Outsourcing Multi-Party Computation*. Cryptology ePrint Archive, Paper 2011/272. (2011). https://eprint.iacr.org/2011/272.

M. Keller and K. Sun. 2022. "Secure quantized training for deep learning." In: *International Conference on Machine Learning (ICML)*. PMLR, 10912–10938.

R. Kerkouche, G. Ács, C. Castelluccia, and P. Genevès. 2021. "Compression boosts differentially private federated learning." In: *IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 304–318.

T. Khan, M. Budzys, K. Nguyen, and A. Michalas. 2024. "SoK: Wildest Dreams: Reproducible Research in Privacy-preserving Neural Network Training." In: *Proceedings on Privacy Enhancing Technologies. Privacy Enhancing Technologies Symposium* 3. Vol. 2024.

J. Kim and B. Lim. 2023. "Effective and Privacy-Preserving Estimation of the Density Distribution of LBS Users under Geo-Indistinguishability." *Electronics*, 12, 4, 917.

M. Kim, X. Jiang, K. Lauter, E. Ismayilzada, and S. Shams. 2022. "Secure human action recognition by encrypted neural network inference." *Nature communications*, 13, 1, 4799.

T. Kim, H. Kwak, D. Lee, J. Seo, and Y. Song. 2022. *Asymptotically Faster Multi-Key Homomorphic Encryption from Homomorphic Gadget Decomposition*. Cryptology ePrint Archive, Paper 2022/347. (2022). https://eprint.iacr.org/2022/347.

B. Knott, S. Venkataraman, A. Hannun, S. Sengupta, M. Ibrahim, and L. van der Maaten. 2021. "Crypten: Secure multi-party computation meets machine learning." *Advances in Neural Information Processing Systems*, 34, 4961–4973.

N. Koti, M. Pancholi, A. Patra, and A. Suresh. 2021. "SWIFT: Super-fast and robust Privacy-Preserving machine learning." In: *30th USENIX Security Symposium (USENIX Security 21)*, 2651–2668.

A. Krizhevsky. 2009. *Learning multiple layers of features from tiny images*. Tech. rep. University of Toronto.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. 2012. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*, 25.

F. Kuang, B. Mi, and D. Huang. 2020. "A Secure Neural Network Prediction Model with Multiple Data Providers." In: *International Conference on Machine Learning for Cyber Security*. Springer, 272–286.

M. Kumar, B. A. Moser, and L. Fischer. 2024. "On Mitigating the Utility-Loss in Differentially Private Learning: A New Perspective by a Geometrically Inspired Kernel Approach." *Journal of Artificial Intelligence Research*, 79, 515–567.

M. Kumar, M. Rossbory, B. A. Moser, and B. Freudenthaler. 2019. "Deriving an optimal noise adding mechanism for privacy-preserving machine learning." In: *International Conference on Database and Expert Systems Applications*. Springer, 108–118.

M. Kumar, W. Zhang, L. Fischer, and B. Freudenthaler. 2023. "Membership Mappings for Practical Secure Distributed Deep Learning." *IEEE Transactions on Fuzzy Systems*, 31, 8, 2617–2631. DOI: 10.1109/TFUZZ.2023.3235440.

N. Kumar, M. Rathee, N. Chandran, D. Gupta, A. Rastogi, and R. Sharma. 2020. "CrypTFlow: Secure tensorflow inference." In: *IEEE Symposium on Security and Privacy (SP)*. IEEE, 336–353.

H. Kwak, S. Min, and Y. Song. 2022. *Towards Practical Multi-key TFHE: Parallelizable, Key-Compatible, Quasi-linear Complexity*. Cryptology ePrint Archive, Paper 2022/1460. (2022). https://eprint.iacr.org/2022/1460.

K. Laine. 2017. *Simple encrypted arithmetic library 2.3.1*. https://www.microsoft.com/en-us/research/uploads/prod/2017/11/sealmanual-2-3-1.pdf. Microsoft Research.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. 2002. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE*, 86, 11, 2278–2324.

Y. LeCun, C. Cortes, and C. Burges. 2010. *MNIST handwritten digit database*. (2010). http://yann.lecun.com/exdb/mnist/.

D. Lee, H. Yu, X. Jiang, D. Rogith, M. Gudala, M. Tejani, Q. Zhang, and L. Xiong. 2020. "Generating sequential electronic health records using dual adversarial autoencoder." *Journal of the American Medical Informatics Association*, 27, 9, 1411–1419.

J. Lee and D. Kifer. 2021. "Scaling up differentially private deep learning with fast per-example gradient clipping." *Proceedings on Privacy Enhancing Technologies*, 2021, 1.

J.-W. Lee et al.. 2022. "Privacy-preserving machine learning with fully homomorphic encryption for deep neural network." *IEEE Access*, 10, 30039–30054. DOI: 10.1109/ACCESS.2022.3159694.

J. Lee, E. Lee, J.-W. Lee, Y. Kim, Y.-S. Kim, and J.-S. No. 2023. "Precise approximation of convolutional neural networks for homomorphically encrypted data." *IEEE Access*, 11, 62062–62076. DOI: 10.1109/ACCESS.2023.3287564.

M. Lee, G. Yu, and H. Dai. 2024. "Privacy-Preserving Decentralized Inference With Graph Neural Networks in Wireless Networks." *IEEE Transactions on Wireless Communications*, 23, 1, 543–558. DOI: 10.1109/TWC.2023.3279442.

S. Lee, G. Lee, J. W. Kim, J. Shin, and M.-K. Lee. July 2023. "HETAL: Efficient Privacy-preserving Transfer Learning with Homomorphic Encryption." In: *Proceedings of the 40th International Conference on Machine Learning* (Proceedings of Machine Learning Research). Ed. by A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett. Vol. 202. PMLR, (July 2023), 19010–19035. https://proceedings.mlr.press/v202/lee23m.html.

R. Lehmkuhl, P. Mishra, A. Srinivasan, and R. A. Popa. 2021. "Muse: Secure inference resilient to malicious clients." In: *30th USENIX Security Symposium (USENIX Security 21)*, 2201–2218.

H. Li et al.. 2024. *Privacy in Large Language Models: Attacks, Defenses and Future Directions*. (2024). https://arxiv.org/abs/2310.10383 arXiv: 2310.10383 (cs.CL).

S. Li, D. Yao, and J. Liu. July 2023. "FedVS: Straggler-Resilient and Privacy-Preserving Vertical Federated Learning for Split Models." In: *Proceedings of the 40th International Conference on Machine Learning* (Proceedings of Machine Learning Research). Ed. by A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett. Vol. 202. PMLR, (July 2023), 20296–20311. https://proceedings.mlr.press/v202/li23an.html.

Y. Li, W. Yu, and J. Zhao. 2025. "PrivTuner with homomorphic encryption and LoRA: A P3EFT scheme for privacy-preserving parameter-efficient fine-tuning of AI foundation models." *IEEE Transactions on Wireless Communications*.

Y. Li, J. Lai, R. Zhang, and M. Sun. 2024. "Secure and efficient multi-key aggregation for federated learning." *Information Sciences*, 654, 119830.

B. Liu, M. Ding, S. Shaham, W. Rahayu, F. Farokhi, and Z. Lin. Mar. 2021. "When Machine Learning Meets Privacy: A Survey and Outlook." *ACM Comput. Surv.*, 54, 2, Article 31, (Mar. 2021), 36 pages. DOI: 10.1145/3436755.

F. Liu, X. Xie, and Y. Yu. Aug. 2024. "Scalable Multi-Party Computation Protocols for Machine Learning in the Honest-Majority Setting." In: *33rd USENIX Security Symposium (USENIX Security 24)*. USENIX Association, Philadelphia, PA, (Aug. 2024). ISBN: 978-1-939133-44-1. https://www.usenix.org/conference/usenixsecurity24/presentation/jiang-zhifeng.

H. Liu, X. Zhang, X. Shen, H. Sun, and M. Shahidehpour. 2023. "A Hybrid Federated Learning Framework With Dynamic Task Allocation for Multi-Party Distributed Load Prediction." *IEEE Transactions on Smart Grid*, 14, 3, 2460–2472. DOI: 10.1109/TSG.2022.3215742.

H. Liu, C. Li, B. Liu, P. Wang, S. Ge, and W. Wang. 2022. "Differentially Private Learning with Grouped Gradient Clipping." In: *ACM Multimedia Asia (MMAsia)* Article 14, 7 pages. ISBN: 9781450386074. DOI: 10.1145/3469877.3490594.

J. Liu, M. Juuti, Y. Lu, and N. Asokan. 2017. "Oblivious neural network predictions via MiniONN transformations." In: *ACM Conference on Computer and Communications Security (CCS)*, 619–631.

L. Liu, J. Zhang, S. Song, and K. B. Letaief. 2023. "Binary Federated Learning with Client-Level Differential Privacy." In: *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, 3849–3854. DOI: 10.1109/GLOBECOM54140.2023.10437593.

P. Liu and W. Zhang. 2022. "Towards practical privacy-preserving solution for outsourced neural network inference." In: *IEEE International Conference on Cloud Computing (CLOUD)*. IEEE, 357–362.

T.-L. Liu, Y.-T. Ku, M.-C. Ho, F.-H. Liu, M.-C. Chang, C.-F. Hsu, W.-C. Chen, and S.-H. Hung. 2023. "An Efficient CKKS-FHEW/TFHE Hybrid Encrypted Inference Framework." In: *European Symposium on Research in Computer Security*. Springer, 535–551.

X.-Y. Liu, R. Zhu, D. Zha, J. Gao, S. Zhong, M. White, and M. Qiu. 2025. "Differentially private low-rank adaptation of large language model using federated learning." *ACM Transactions on Management Information Systems*, 16, 2, 1–24.

X. Liu, H. Li, Q. Qian, and H. Ren. 2023. "Efficient Homomorphic Convolution for Secure Deep Learning Inference." In: *2023 20th Annual International Conference on Privacy, Security and Trust (PST)*, 1–6. DOI: 10.1109/PST58708.2023.10320171.

X. Liu, L. Xie, Y. Wang, J. Zou, J. Xiong, Z. Ying, and A. V. Vasilakos. 2021. "Privacy and Security Issues in Deep Learning: A Survey." *IEEE Access*, 9, 4566–4593. DOI: 10.1109/ACCESS.2020.3045078.

X. Liu, Z. Liu, Q. Li, K. Xu, and M. Xu. 2024. "Pencil: Private and Extensible Collaborative Learning without the Non-Colluding Assumption." In: *Symposium on Network and Distributed System Security (NDSS)*.

Y. Liu and Q. Su. 2024. "PPTIF: Privacy-Preserving Transformer Inference Framework for Language Translation." *IEEE Access*, 12, 48881–48897. DOI: 10.1109/ACCESS.2024.3384268.

Q. Lou, B. Feng, G. Charles Fox, and L. Jiang. 2020. "Glyph: Fast and accurately training deep neural networks on encrypted data." *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 9193–9202.

Q. Lou and L. Jiang. 2021. "HEMET: A homomorphic-encryption-friendly privacy-preserving mobile neural network architecture." In: *International Conference on Machine Learning (ICML)*. PMLR, 7102–7110.

W.-j. Lu, Z. Huang, Z. Gu, J. Li, J. Liu, C. Hong, K. Ren, T. Wei, and W. Chen. Feb. 2025. "BumbleBee: Secure Two-party Inference Framework for Large Transformers." In: *Proceedings of the Network and Distributed System Security Symposium (NDSS)*. (Feb. 2025). ISBN: 979-8-9894372-8-3. DOI: 10.14722/ndss.2025.230057.

J. Luo et al.. 2025. "Cryptography-based privacy-preserving large language models: a lifecycle survey of frameworks, methods, and future directions." *Artificial Intelligence Review*.

H. Lycklama, L. Burkhalter, A. Viand, N. Küchler, and A. Hithnawi. 2023. "RoFL: Robustness of Secure Federated Learning." In: *IEEE Symposium on Security and Privacy (SP)*. IEEE, 453–476. DOI: 10.1109/SP46215.2023.10179400.

L. Lyu. 2020. "Lightweight crypto-assisted distributed differential privacy for privacy-preserving distributed learning." In: *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

L. Lyu and C.-H. Chen. 2020. "Differentially private knowledge distillation for mobile analytics." In: *International ACM Conference on Research and Development in Information Retrieval*, 1809–1812.

V. Lyubashevsky, C. Peikert, and O. Regev. 2010. "On ideal lattices and learning with errors over rings." In: *Advances in Cryptology – EUROCRYPT 2010*. Springer, 1–23.

X. Ma, F. Zhang, X. Chen, and J. Shen. 2018. "Privacy preserving multi-party computation delegation for deep learning in cloud computing." *Information Sciences*, 459, 103–116.

B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. 2017. "Communication-efficient learning of deep networks from decentralized data." In: *International Conference on Artificial Intelligence and Statistics*. PMLR, 1273–1282.

L. Mehner, S. N. von Voigt, and F. Tschorsch. 2021. "Towards explaining epsilon: A worst-case study of differential privacy risks." In: *2021 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 328–331.

L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov. 2019. "Exploiting unintended feature leakage in collaborative learning." In: *IEEE Symposium on Security and Privacy (SP)*. IEEE, 691–706.

M. Miranda, E. S. Ruzzetti, A. Santilli, F. M. Zanzotto, S. Bratières, and E. Rodolà. 2025. "Preserving Privacy in Large Language Models: A Survey on Current Threats and Solutions." *Transactions on Machine Learning Research*. https://openreview.net/forum?id=Ss9MTTN7OL.

P. Mishra, R. Lehmkuhl, A. Srinivasan, W. Zheng, and R. A. Popa. 2020. "DELPHI: A Cryptographic Inference Service for Neural Networks." In: *29th USENIX Security Symposium (USENIX Security 20)*, 2505–2522. ISBN: 978-1-939133-17-5. https://www.usenix.org/conference/usenixsecurity20/presentation/mishra.

P. Mohassel and P. Rindal. 2018. "ABY3: A mixed protocol framework for machine learning." In: *ACM Conference on Computer and Communications Security (CCS)*, 35–52.

P. Mohassel and Y. Zhang. 2017. "SecureML: A system for scalable privacy-preserving machine learning." In: *IEEE Symposium on Security and Privacy (SP)*. IEEE, 19–38.

T. T. Mueller, J. C. Paetzold, C. Prabhakar, D. Usynin, D. Rueckert, and G. Kaissis. June 2023. "Differentially Private Graph Neural Networks for Whole-Graph Classification." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45, 6, (June 2023), 7308–7318. DOI: 10.1109/TPAMI.2022.3228315.

P. Nanayakkara, M. A. Smart, R. Cummings, G. Kaptchuk, and E. M. Redmiles. 2023. "What are the chances? explaining the epsilon parameter in differential privacy." In: *32nd USENIX Security Symposium (USENIX Security 23)*, 1613–1630.

K. Nandakumar, N. Ratha, S. Pankanti, and S. Halevi. 2019. "Towards deep neural network training on encrypted data." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 40–48.

M. Nasr, S. Mahloujifar, X. Tang, P. Mittal, and A. Houmansadr. July 2023. "Effectively Using Public Data in Privacy Preserving Machine Learning." In: *Proceedings of the 40th International Conference on Machine Learning* (Proceedings of Machine Learning Research). Vol. 202. PMLR, (July 2023), 25718–25732. https://proceedings.mlr.press/v202/nasr23a.html.

D. Natarajan, A. Loveless, W. Dai, and R. Dreslinski. 2023. "CHEX-MIX: Combining Homomorphic Encryption with Trusted Execution Environments for Oblivious Inference in the Cloud." In: *IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 73–91.

Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng, et al.. 2011. "Reading digits in natural images with unsupervised feature learning." In: *NIPS workshop on deep learning and unsupervised feature learning* 5. Vol. 2011. Granada, 7.

L. K. Ng and S. S. Chow. 2021. "GForce: GPU-Friendly Oblivious and Rapid Neural Network Inference." In: *30th USENIX Security Symposium (USENIX Security 21)*, 2147–2164.

L. K. Ng and S. S. Chow. 2023. "SoK: Cryptographic Neural-Network Computation." In: *IEEE Symposium on Security and Privacy (SP)*. IEEE, 497–514.

M. Nocker, D. Drexel, M. Rader, A. Montuoro, and P. Schöttle. 2023. "HE-MAN – Homomorphically Encrypted MAchine Learning with ONnx Models." In: *International Conference on Machine Learning Technologies*, 35–45. ISBN: 9781450398329. DOI: 10.1145/3589883.3589889.

M. Nocker, L. Henke, and P. Schöttle. 2025. "FHE ML Tuxedo: A Tailored Wrapper Architecture for Homomorphic Encryption in Machine Learning." In: *2025 55th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. IEEE, 266–273.

M. Nocker, F. Merkle, P. Schöttle, and M. Janetschek. 2026. "Protecting Privacy in IoT-Based Deep Learning: State-of-the-Art Methods and Challenges." In: *Applied Cryptography and Network Security Workshops*. Ed. by M. Manulis. Springer Nature Switzerland, Cham, 163–184. ISBN: 978-3-032-01799-4.

E. Oyallon and S. Mallat. 2015. "Deep roto-translation scattering for object classification." In: *IEEE Conference on Computer Vision and Pattern Recognition*, 2865–2873.

P. Paillier. 1999. "Public-key cryptosystems based on composite degree residuosity classes." In: *Advances in Cryptology – EUROCRYPT '99*. Springer, 223–238.

Q. Pang, J. Zhu, H. Möllering, W. Zheng, and T. Schneider. 2024. "Bolt: Privacy-preserving, accurate and efficient inference for transformers." In: *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE, 4753–4771.

A. Pankova and P. Laud. 2022. "Interpreting epsilon of differential privacy in terms of advantage in guessing or approximating sensitive attributes." In: *2022 IEEE 35th Computer Security Foundations Symposium (CSF)*. IEEE, 96–111.

N. Papernot, M. Abadi, Ú. Erlingsson, I. Goodfellow, and K. Talwar. 2017. "Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data." In: *International Conference on Learning Representations*. https://openreview.net/forum?id=HkwoSDPgg.

N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. 2017. "Practical black-box attacks against machine learning." In: *ACM ASIA Conference on Computer and Communications Security (ASIACCS)*, 506–519.

N. Papernot, P. McDaniel, A. Sinha, and M. P. Wellman. 2018. "SoK: Security and Privacy in Machine Learning." In: *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, 399–414. DOI: 10.1109/EuroSP.2018.00035.

N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and U. Erlingsson. 2018. "Scalable Private Learning with PATE." In: *International Conference on Learning Representations*. https://openreview.net/forum?id=rkZB1XbRZ.

N. Papernot, A. Thakurta, S. Song, S. Chien, and Ú. Erlingsson. 2021. "Tempered sigmoid activations for deep learning with differential privacy." In: *AAAI Conference on Artificial Intelligence* 10. Vol. 35, 9312–9321.

J. Park, H. Kim, Y. Choi, and J. Lee. July 2023. "Differentially Private Sharpness-Aware Training." In: *Proceedings of the 40th International Conference on Machine Learning* (Proceedings of Machine Learning Research). Vol. 202. PMLR, (July 2023), 27204–27224. https://proceedings.mlr.press/v202/park23g.html.

A. Patra, T. Schneider, A. Suresh, and H. Yalame. 2021. "ABY2.0: Improved Mixed-Protocol Secure Two-Party Computation." In: *30th USENIX Security Symposium (USENIX Security 21)*, 2165–2182.

A. Patra and A. Suresh. 2020. "BLAZE: blazing fast privacy-preserving machine learning." In: *Symposium on Network and Distributed System Security (NDSS)*.

F. Piccialli, V. Di Somma, F. Giampaolo, S. Cuomo, and G. Fortino. 2021. "A survey on deep learning in medicine: Why, how and when?" *Information Fusion*, 66, 111–137.

J. Qian, P. Zhang, H. Zhu, M. Liu, J. Wang, and X. Ma. 2023. "LHDNN: Maintaining High Precision and Low Latency Inference of Deep Neural Networks on Encrypted Data." *Applied Sciences*, 13, 8, 4815.

H. Qin, D. He, Q. Feng, M. K. Khan, M. Luo, and K.-K. R. Choo. 2024. "Cryptographic Primitives in Privacy-Preserving Machine Learning: A Survey." *IEEE Transactions on Knowledge and Data Engineering*, 36, 5, 1919–1934. DOI: 10.1109/TKDE.2023.3321803.

R. Ran, X. Luo, W. Wang, T. Liu, G. Quan, X. Xu, C. Ding, and W. Wen. July 2023. "SpENCNN: Orchestrating Encoding and Sparsity for Fast Homomorphically Encrypted Neural Network Inference." In: *Proceedings of the 40th International Conference on Machine Learning* (Proceedings of Machine Learning Research). Ed. by A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett. Vol. 202. PMLR, (July 2023), 28718–28728. https://proceedings.mlr.press/v202/ran23b.html.

K. Ranaweera, D. Smith, D. C. Nguyen, P. N. Pathirana, M. Ding, T. Rakotoarivelo, and A. Seneviratne. 2023. "Improving the Utility of Differentially Private SGD by Employing Wavelet Transforms." In: *2023 IEEE International Conference on Big Data (BigData)*, 1352–1359. DOI: 10.1109/BigData59044.2023.10386897.

D. Rathee, M. Rathee, R. K. K. Goli, D. Gupta, R. Sharma, N. Chandran, and A. Rastogi. 2021. "SiRnn: A math library for secure RNN inference." In: *IEEE Symposium on Security and Privacy (SP)*. IEEE, 1003–1020.

D. Rathee, M. Rathee, N. Kumar, N. Chandran, D. Gupta, A. Rastogi, and R. Sharma. 2020. "CrypTFlow2: Practical 2-party secure inference." In: *ACM Conference on Computer and Communications Security (CCS)*, 325–342.

H. Ren, J. Deng, X. Xie, X. Ma, and Y. Wang. 2024. "FedBoosting: Federated learning with gradient protected boosting for text recognition." *Neurocomputing*, 569, 127126.

X. Ren et al.. 2023. "CHAM: A Customized Homomorphic Encryption Accelerator for Fast Matrix-Vector Product." In: *2023 60th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 1–6.

M. S. Riazi, M. Samragh, H. Chen, K. Laine, K. Lauter, and F. Koushanfar. 2019. "XONN: XNOR-based oblivious deep neural network inference." In: *28th USENIX Security Symposium (USENIX Security 19)*, 1501–1518.

M. S. Riazi, C. Weinert, O. Tkachenko, E. M. Songhori, T. Schneider, and F. Koushanfar. 2018. "Chameleon: A hybrid secure computation framework for machine learning applications." In: *ACM ASIA Conference on Computer and Communications Security (ASIACCS)*, 707–721.

M. Rigaki and S. Garcia. Nov. 2023. "A Survey of Privacy Attacks in Machine Learning." *ACM Comput. Surv.*, 56, 4, Article 101, (Nov. 2023), 34 pages. DOI: 10.1145/3624010.

L. Rosenblatt, X. Liu, S. Pouyanfar, E. de Leon, A. Desai, and J. Allen. 2020. *Differentially Private Synthetic Data: Applied Evaluations and Enhancements*. (2020). https://arxiv.org/abs/2011.05537 arXiv: 2011.05537 (cs.LG).

B. D. Rouhani, M. S. Riazi, and F. Koushanfar. 2018. "DeepSecure: Scalable provably-secure deep learning." In: *Annual Design Automation Conference (DAC)*, 1–6.

W. Ruan, M. Xu, W. Fang, L. Wang, L. Wang, and W. Han. 2023. "Private, efficient, and accurate: Protecting models trained by multi-party learning with differential privacy." In: *IEEE Symposium on Security and Privacy (SP)*. IEEE, 1926–1943.

S. Sajadmanesh and D. Gatica-Perez. 2024. "ProGAP: Progressive Graph Neural Networks with Differential Privacy Guarantees." In: *Proceedings of the 17th ACM International Conference on Web Search and Data Mining* (WSDM '24). Association for Computing Machinery, Merida, Mexico, 596–605. ISBN: 9798400703713. DOI: 10.1145/3616855.3635761.

A. Salem, G. Cherubin, D. Evans, B. Köpf, A. Paverd, A. Suri, S. Tople, and S. Zanella-Béguelin. 2023. "SoK: Let the Privacy Games Begin! A Unified Treatment of Data Inference Privacy in Machine Learning." In: *2023 IEEE Symposium on Security and Privacy (SP)*, 327–345. DOI: 10.1109/SP46215.2023.10179281.

S. Sav, A. Pyrgelis, J. R. Troncoso-Pastoriza, D. Froelicher, J.-P. Bossuat, J. S. Sousa, and J.-P. Hubaux. 2021. "POSEIDON: Privacy-preserving federated neural network learning." In: *Symposium on Network and Distributed System Security (NDSS)*.

A. Schlögl and R. Böhme. 2020. "eNNclave: offline inference with model confidentiality." In: *ACM Workshop on Artificial Intelligence and Security*, 93–104.

J. Schmidhuber. 2015. "Deep learning in neural networks: An overview." *Neural Networks*, 61, 85–117. DOI: https://doi.org/10.1016/j.neunet.2014.09.003.

N. Senanayake, R. Podschwadt, D. Takabi, V. D. Calhoun, and S. M. Plis. 2022. "NeuroCrypt: Machine learning over encrypted distributed neuroimaging data." *Neuroinformatics*, 20, 1, 91–108.

A. S. Shamsabadi and N. Papernot. 2023. "Losing Less: A Loss for Differentially Private Deep Learning." *Proceedings on Privacy Enhancing Technologies*, 2023, 3, 307–320.

W. Shi, R. Shea, S. Chen, C. Zhang, R. Jia, and Z. Yu. 2022. "Just fine-tune twice: Selective differential privacy for large language models." In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 6327–6340.

R. Shokri and V. Shmatikov. 2015. "Privacy-preserving deep learning." In: *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 1310–1321.

R. Shokri, M. Stronati, C. Song, and V. Shmatikov. 2017. "Membership inference attacks against machine learning models." In: *IEEE Symposium on Security and Privacy (SP)*. IEEE, 3–18.

K. Simonyan and A. Zisserman. 2015. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. (2015). https://arxiv.org/abs/1409.1556 arXiv: 1409.1556 (cs.CV).

S. Sinha, S. Saha, M. Alam, V. Agarwal, A. Chatterjee, A. Mishra, D. Khazanchi, and D. Mukhopadhyay. 2022. "Exploring bitslicing architectures for enabling fhe-assisted machine learning." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41, 11, 4004–4015.

Y. Son, K. Han, Y. S. Lee, J. Yu, Y.-H. Im, and S.-Y. Shin. Dec. 2021. "Privacy-preserving breast cancer recurrence prediction based on homomorphic encryption and secure two party computation." *PLOS ONE*, 16, 12, (Dec. 2021), e0260681. DOI: 10.1371/journal.pone.0260681.

J. Spanier, K. B. Oldham, and R. H. Romer. 1988. *An atlas of functions*. (1988).

M. Steiner, G. Tsudik, and M. Waidner. 1996. "Diffie-Hellman key distribution extended to group communication." In: *Proceedings of the 3rd ACM Conference on Computer and Communications Security*, 31–37. https://dl.acm.org/doi/pdf/10.1145/238168.238182.

V. Stephanie, I. Khalil, M. S. Rahman, and M. Atiquzzaman. 2022. "Privacy-Preserving Ensemble Infused Enhanced Deep Neural Network Framework for Edge Cloud Convergence." *IEEE Internet of Things Journal*, 10, 5, 3763–3773.

P. Subramani, N. Vadivelu, and G. Kamath. 2021. "Enabling fast differentially private SGD via just-in-time compilation and vectorization." *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 26409–26421.

Y. Sugizaki, H. Tsuchida, T. Hayashi, K. Nuida, A. Nakashima, T. Isshiki, and K. Mori. 2023. "Threshold Fully Homomorphic Encryption Over the Torus." In: *European Symposium on Research in Computer Security*. Springer, 45–65.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. 2014. "Intriguing properties of neural networks." In: *International Conference on Learning Representations (ICLR)*.

S. Tan, B. Knott, Y. Tian, and D. J. Wu. 2021. "CRYPTGPU: Fast privacy-preserving machine learning on the GPU." In: *IEEE Symposium on Security and Privacy (SP)*. IEEE, 1021–1038.

T. Tang, Z. Han, Z. Cai, S. Yu, X. Zhou, T. Oseni, and S. K. Das. Mar. 2024. "Personalized Federated Graph Learning on Non-IID Electronic Health Records." *IEEE transactions on neural networks and learning systems*, PP, (Mar. 2024). DOI: 10.1109/tnnls.2024.3370297.

J. Tchaye-Kondi, Y. Zhai, J. Shen, and L. Zhu. 2023. "Privacy-preserving offloading in edge intelligence systems with inductive learning and local differential privacy." *IEEE Transactions on Network and Service Management*, 20, 4, 5026–5037.

Y. Tian, X. Ke, Z. Tao, S. Ding, F. Xu, Q. Li, H. Han, S. Zhong, and X. Fu. 2022. "Privacy-Preserving and Robust Federated Deep Metric Learning." In: *IEEE/ACM International Symposium on Quality of Service (IWQoS)*. IEEE, 1–11.

H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou. 2021. "Training data-efficient image transformers & distillation through attention." In: *International conference on machine learning*. PMLR, 10347–10357.

F. Tramèr and D. Boneh. 2021. "Differentially Private Learning Needs Better Features (or Much More Data)." In: *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=YTWGvpFOQD-.

F. Tramèr and D. Boneh. 2019. "Slalom: Fast, Verifiable and Private Execution of Neural Networks in Trusted Hardware." In: *7th International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=rJVorjCcKQ.

F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart. 2016. "Stealing machine learning models via prediction APIs." In: *25th USENIX Security Symposium (USENIX Security 16)*, 601–618.

A. Treiber, A. Molina, C. Weinert, T. Schneider, and K. Kersting. 2020. *CryptoSPN: Privacy-preserving Sum-Product Network Inference*. (2020). arXiv: 2002.00801 (cs.AI).

L. Trieu Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai. 2017. "Privacy-preserving deep learning via additively homomorphic encryption." *IEEE Transactions on Information Forensics and Security*, 13, 5, 1333–1345.

S. Truex, L. Liu, K.-H. Chow, M. E. Gursoy, and W. Wei. 2020. "LDP-Fed: Federated learning with local differential privacy." In: *ACM International Workshop on Edge Systems, Analytics and Networking*, 61–66.

T. Tung-Lin and W. Pei-Yuan. 2023. "SepMM: A General Matrix Multiplication Optimization Approach for Privacy-Preserving Machine Learning." In: *2023 IEEE Conference on Dependable and Secure Computing (DSC)*, 1–10. DOI: 10.1109/DSC61021.2023.10354193.

United States Congress. 1996. *Health Insurance Portability and Accountability Act of 1996*. U.S. Code. Accessed: 2026-07-17. (1996). https://www.govinfo.gov/content/pkg/PLAW-104publ191/pdf/PLAW-104publ191.pdf.

A. Viand, P. Jattke, M. Haller, and A. Hithnawi. 2023. "HECO: Fully Homomorphic Encryption Compiler." In: *32nd USENIX Security Symposium (USENIX Security 23)*, 4715–4732.

A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis. 2018. "Deep learning for computer vision: A brief review." *Computational intelligence and neuroscience*, 2018, 1, 7068349.

S. Wagh, D. Gupta, and N. Chandran. 2019. "SecureNN: 3-Party Secure Computation for Neural Network Training." *Proceedings on Privacy Enhancing Technologies*, 2019, 3, 26–49.

S. Wagh, S. Tople, F. Benhamouda, E. Kushilevitz, P. Mittal, and T. Rabin. 2021. "Falcon: Honest-Majority Maliciously Secure Framework for Private Deep Learning." *Proceedings on Privacy Enhancing Technologies*, 2021, 1, 188–208.

J. Wang, J. Zhang, W. Bao, X. Zhu, B. Cao, and P. S. Yu. 2018. "Not just privacy: Improving performance of private deep learning in mobile cloud." In: *ACM International Conference on Knowledge Discovery & Data Mining*, 2407–2416.

J. Wang, G. Xie, Y. Huang, J. Lyu, F. Zheng, Y. Zheng, and Y. Jin. 2023. "FedMed-GAN: Federated domain translation on unsupervised cross-modality brain image synthesis." *Neurocomputing*, 546, 126282.

Q. Wang, W. Ma, and G. Liu. 2021. "SieveNet: Decoupling activation function neural network for privacy-preserving deep learning." *Information Sciences*, 573, 262–278.

R. Wang, M. Luo, Q. Feng, C. Peng, and D. He. Feb. 2024. "Multi-Party Privacy-Preserving Faster R-CNN Framework for Object Detection." *IEEE Transactions on Emerging Topics in Computational Intelligence*, 8, 1, (Feb. 2024), 956–967. DOI: 10.1109/TETCI.2023.3296502.

W. Wang, T. Wang, L. Wang, N. Luo, P. Zhou, D. Song, and R. Jia. 2021. "DPlis: Boosting Utility of Differentially Private Deep Learning via Randomized Smoothing." *Proceedings on Privacy Enhancing Technologies*, 2021, 4, 163–183.

Y. Wang, S. Adams, P. Beling, S. Greenspan, S. Rajagopalan, M. Velez-Rojas, S. Mankovski, S. Boker, and D. Brown. 2018. "Privacy preserving distributed deep learning and its application in credit card fraud detection." In: *IEEE International Conference On Trust, Security And Privacy In Computing And Communications/IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE, 1070–1078.

J.-L. Watson, S. Wagh, and R. A. Popa. 2022. "Piranha: A GPU platform for secure computation." In: *31st USENIX Security Symposium (USENIX Security 22)*, 827–844.

J. Webster and R. T. Watson. 2002. "Analyzing the Past to Prepare for the Future: Writing a Literature Review." *MIS Quarterly*, 26, 2, xiii–xxiii. Retrieved Feb. 9, 2024 from http://www.jstor.org/stable/4132319.

L. Wu, H. Shi, S. Fu, Y. Luo, and M. Xu. 2023. "p2Detect: Electricity Theft Detection With Privacy Preservation for Both Data and Model in Smart Grid." *IEEE Transactions on Smart Grid*, 14, 3, 2301–2312. DOI: 10.1109/TSG.2022.3214194.

H. Xiao, K. Rasul, and R. Vollgraf. Aug. 28, 2017. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. (Aug. 28, 2017). arXiv: cs.LG/1708.07747 (cs.LG).

H. Xiao, J. Wan, and S. Devadas. 2023. "Geometry of Sensitivity: Twice Sampling and Hybrid Clipping in Differential Privacy with Optimal Gaussian Noise and Application to Deep Learning." In: *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications*

Security (CCS '23). Association for Computing Machinery, Copenhagen, Denmark, 2636–2650. ISBN: 9798400700507. DOI: 10.1145/3576915.3623142.

H. Xiao, B. Biggio, G. Brown, G. Fumera, C. Eckert, and F. Roli. 2015. "Is feature selection secure against training data poisoning?" In: *International Conference on Machine Learning (ICML)*. PMLR, 1689–1698.

H. Xiao, Q. Zhang, Q. Pei, and W. Shi. 2021. "Privacy-preserving neural network inference framework via homomorphic encryption and SGX." In: *IEEE International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 751–761.

C. Xu, J. Ren, D. Zhang, Y. Zhang, Z. Qin, and K. Ren. 2019. "GANobfuscator: Mitigating information leakage under GAN via differential privacy." *IEEE Transactions on Information Forensics and Security*, 14, 9, 2358–2371. DOI: 10.1109/TIFS.2019.2897874.

W. Xu, B. Shi, J. Zhang, Z. Feng, T. Pan, and B. Dong. 2023. "MDP: Privacy-Preserving GNN Based on Matrix Decomposition and Differential Privacy." In: *2023 IEEE International Conference on Joint Cloud Computing (JCC)*, 38–45. DOI: 10.1109/JCC59055.2023.00011.

Z. Xu, Y. Zhang, G. Andrew, C. Choquette, P. Kairouz, B. Mcmahan, J. Rosenstock, and Y. Zhang. 2023. "Federated learning of gboard language models with differential privacy." In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, 629–639.

A. C.-C. Yao. 1986. "How to generate and exchange secrets." In: *Symposium on Foundations of Computer Science*. IEEE, 162–167.

C. Yu, S. Shen, S. Wang, K. Zhang, and H. Zhao. 2025. "Communication-Efficient Hybrid Federated Learning for E-Health With Horizontal and Vertical Data Partitioning." *IEEE Transactions on Neural Networks and Learning Systems*, 36, 3, 5614–5628. DOI: 10.1109/TNNLS.2024.3383748.

D. Yu, H. Zhang, W. Chen, and T.-Y. Liu. 2021. "Do not Let Privacy Overbill Utility: Gradient Embedding Perturbation for Private Learning." In: *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=7aogOj_VYO0.

B. Yuan, S. Zhai, Y. Zhang, N. Ding, D. Gu, and S.-F. Sun. Aug. 2024. "MD-ML: Super Fast Privacy-Preserving Machine Learning for Malicious Security with a Dishonest Majority." In: *33rd USENIX Security Symposium (USENIX Security 24)*. USENIX Association, Philadelphia, PA, (Aug. 2024), 2227–2244. ISBN: 978-1-939133-44-1. https://www.usenix.org/conference/usenixsecurity24/presentation/yuan.

S. Zagoruyko and N. Komodakis. 2017. *Wide Residual Networks*. (2017). https://arxiv.org/abs/1605.07146 arXiv: 1605.07146 [cs.CV].

M. D. Zeiler and R. Fergus. 2014. "Visualizing and Understanding Convolutional Networks." In: *Computer Vision – ECCV 2014*. Ed. by D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars. Springer International Publishing, Cham, 818–833. ISBN: 978-3-319-10590-1.

W. Zeng, M. Li, W. Xiong, T. Tong, W.-j. Lu, J. Tan, R. Wang, and R. Huang. Oct. 2023. "MPCViT: Searching for Accurate and Efficient MPC-Friendly Vision Transformer with Heterogeneous Attention." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. (Oct. 2023), 5052–5063.

Q. Zhang, J. Ma, J. Lou, L. Xiong, and X. Jiang. 2020. "Towards training robust private aggregation of teacher ensembles under noisy labels." In: *IEEE International Conference on Big Data*. IEEE, 1103–1110.

X. Zhang, Z. Bu, S. Wu, and M. Hong. 2024. "Differentially Private SGD Without Clipping Bias: An Error-Feedback Approach." In: *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=uFbWHyTlPn.

X. Zhang, S. Ji, H. Wang, and T. Wang. 2017. "Private, yet practical, multiparty deep learning." In: *IEEE International Conference on Distributed Computing Systems*. IEEE, 1442–1452.

Z. Zhang, Y. Yang, Y. Dai, Q. Wang, Y. Yu, L. Qu, and Z. Xu. 2023. "Fedpetuning: When federated learning meets the parameter-efficient tuning methods of pre-trained language models." In: *Findings of the Association for Computational Linguistics: ACL 2023*, 9963–9977.

L. Zhao, Q. Wang, Q. Zou, Y. Zhang, and Y. Chen. 2019. "Privacy-preserving collaborative deep learning with unreliable participants." *IEEE Transactions on Information Forensics and Security*, 15, 1486–1500.

J. Zhou, J. Li, E. Panaousis, and K. Liang. 2020. "Deep binarized convolutional neural network inferences over encrypted data." In: *IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*. IEEE, 160–167.

Z. Zhou, Q. Fu, Q. Wei, and Q. Li. 2022. "LEGO: A hybrid toolkit for efficient 2PC-based privacy-preserving machine learning." *Computers & Security*, 120, 102782.

L. Zhu, Z. Liu, and S. Han. 2019. "Deep leakage from gradients." *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 14774–14784.

X. Zhu, V. Y. F. Tan, and X. Xiao. 2023. "Blink: Link Local Differential Privacy in Graph Neural Networks via Bayesian Estimation." In: *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security* (CCS '23). Association for Computing Machinery, Copenhagen, Denmark, 2651–2664. ISBN: 9798400700507. DOI: 10.1145/3576915.3623165.

A. Ziller, D. Usynin, R. Braren, M. Makowski, D. Rueckert, and G. Kaissis. 2021. "Medical imaging deep learning with differential privacy." *Scientific Reports*, 11, 1, 13524.

Received 17 November 2025; accepted 01 October 2026