Data Privacy in Enterprise AI

Neil G. Marchant[0000−0001−5713−4235],
Ying Zhao[0000−0002−9563−7519],
Benjamin I. P. Rubinstein[0000−0002−2947−6980] and
Olga Ohrimenko[0000−0002−9735−0538]

Abstract Enterprise AI systems often process vast amounts of personal and sensitive data, making them vulnerable to breaches and misuse. Consider, for example, a financial institution that trains a predictive model on sensitive customer data that is to be shared with partner organizations. If the model is shared without proper precautions, the partner organization or a malicious employee may attempt to breach the privacy of the original training data, by launching an attack to reconstruct this data from model parameters. Such vulnerabilities erode trust in enterprise and AI systems, may fail to comply with privacy legislation, and can lead to financial, intellectual property or reputational loss. Privacy is a cornerstone of ethical and responsible AI deployment. This chapter provides a guide to the tools necessary to navigate the landscape of AI privacy, ultimately facilitating broader and more secure AI adoption in the enterprise.

Key words: privacy attacks, differential privacy, federated learning, machine unlearning, secure multiparty computation, trusted execution environments

1 Introduction

The rapid advancement of artificial intelligence (AI) has ushered in an era of unprecedented data utilization. Modern AI systems, particularly large language models, demonstrate an insatiable appetite for data, with performance improvements closely tied to the scale of training data. This relationship is evidenced by empirical scaling laws observed by researchers (Kaplan et al., 2020), and exemplified by a commercial language model published in 2024, which was trained on a staggering 15 trillion tokens (Llama team, 2024).

N. G. Marchant · Y. Zhao · B. I. P. Rubinstein · O. Ohrimenko
School of Computing and Information Systems, University of Melbourne, Victoria, Australia, e-mail: nmarchant@unimelb.edu.au

While this data-driven approach has led to remarkable achievements, it also presents significant privacy challenges for enterprises adopting AI technologies. The landscape of privacy concerns in AI has evolved rapidly over the past decade. Initially, privacy discussions in enterprise AI focused primarily on data storage and access. However, as AI models have become more sophisticated and widely deployed, the scope of privacy concerns has expanded to include issues such as model inversion attacks, differential privacy in federated learning, and the potential for unintended memorization in large language models. This evolution underscores the need for a comprehensive and adaptive approach to privacy in enterprise AI.

The data used to train or fine-tune AI models often contains sensitive information, ranging from personally identifiable information in customer interactions to proprietary business intelligence and trade secrets. For instance, an e-commerce platform might train its AI on proprietary purchasing metrics (Bawack et al., 2022), or an engineering firm might optimize product designs using historical performance data (Yüksel et al., 2023). In both cases, the training data represents valuable intellectual property that requires protection.

The risk of data leakage in AI systems is multifaceted and can occur at various stages of the AI lifecycle. During the training phase, data or its derivatives may be intercepted, especially when using public cloud infrastructure or decentralized approaches like federated learning. Post-deployment, there's a risk of data extraction from the model itself or its outputs. Even test data, such as prompts shared with large language model services, may be subject to privacy concerns.

These privacy risks extend beyond the immediate threat of data disclosure. Enterprises must also consider secondary risks, including regulatory non-compliance costs, reputational damage, and potential customer abandonment (Wong et al., 2023). As such, privacy considerations are paramount in AI projects, and are made more challenging by the business needs of balancing disclosure risks with achieving business critical model performance.

1.1 Defining Privacy in the Context of Enterprise AI

Before exploring the specifics of privacy risks and mitigations in AI systems, it's crucial to establish what we mean by "privacy" in this context. While there is no universal consensus on the definition of privacy, for the purposes of this chapter, we adopt the risk-based approach outlined in the NIST Privacy Framework (National Institute of Standards and Technology, 2020). This framework conceptualizes privacy as a condition that safeguards values such as human autonomy and dignity, achievable through various means including limiting observation and ensuring individual control over personal information.

In the context of enterprise AI, privacy primarily concerns the appropriate handling and protection of data used in AI systems, including training data, model parameters, and inference data. This encompasses not only personal information of individuals but also proprietary business data and intellectual property. Our focus

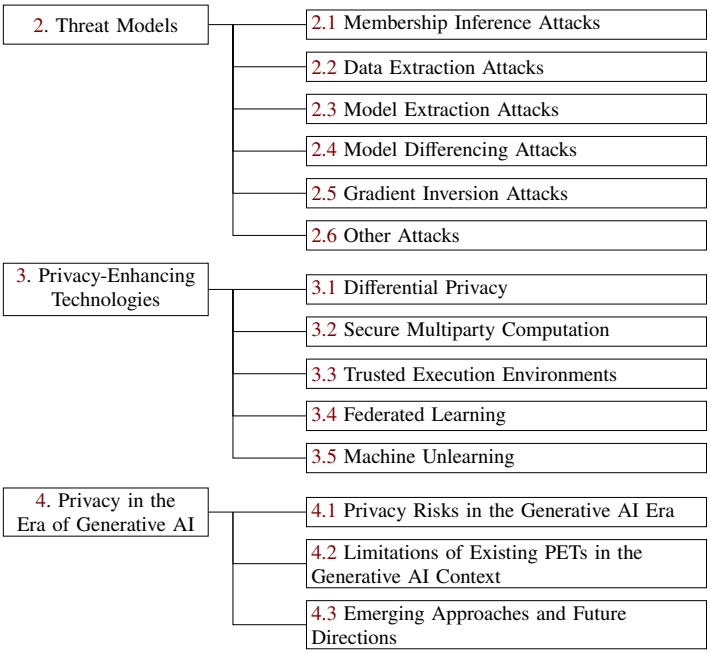


Fig. 1 Chapter Structure Overview (Sections 2–4)

will be on managing privacy risks, which we define as potential adverse effects that individuals or organizations may experience due to data processing activities in AI workflows. This approach allows us to address privacy concerns in a practical, risk-oriented manner that aligns with enterprise needs and regulatory requirements.

1.2 Chapter Overview

This chapter aims to provide a comprehensive overview of privacy in the context of enterprise AI, focusing on key risks and potential mitigations specific to AI workflows, as illustrated in Fig 1. In Section 2, we explore various threat models and privacy risks, including membership inference attacks, data extraction attacks, and side-channel attacks. This examination of threat models serves as a foundation for understanding the specific challenges faced in AI privacy. Section 3 presents a range of privacy-enhancing technologies that can be employed to mitigate these risks, such as differential privacy, secure multiparty computation, and federated learning. Finally, Section 4 discusses the evolving landscape of privacy in the era of generative AI, addressing new challenges and potential solutions.

It's important to note that while this chapter focuses on AI-specific privacy concerns, it does not cover broader privacy topics such as general data protection regula-

tions, data management practices, or privacy impact assessments. For information on privacy legislation like GDPR or CCPA, readers are directed to specialized resources (Lorè et al., 2023; Bukaty, 2019; IT Governance Privacy Team, 2020). Similarly, for general data management practices and privacy governance, other comprehensive guides are available (Plotkin, 2013; Janssen et al., 2020; Mahanti, 2021).

By providing this focused examination of privacy in AI, we aim to equip enterprise decision-makers, data scientists, and AI practitioners with the knowledge needed to navigate the complex intersection of AI capabilities and privacy requirements. As we move into our discussion of threat models in the next section, which, while comprehensive, is not exhaustive given the rapidly evolving nature of the field, we will see how understanding these specific risks is crucial for developing effective privacy strategies in enterprise AI deployments.

2 Threat Models

Understanding and addressing privacy risks in AI systems requires a structured approach that goes beyond general data protection measures. This section introduces the concept of threat modeling as a powerful tool for analyzing and mitigating privacy risks specific to AI contexts.

Threat modeling, a practice commonly used in cybersecurity, involves systematically identifying potential threats to a system by considering an adversary's capabilities, incentives, prior knowledge, and constraints. In this chapter, we specify each threat model by describing three key components:

- *Adversary's Capabilities:* What actions the attacker can perform, such as querying the model or accessing certain data.
- *Adversary's Knowledge:* What information the attacker possesses about the system, model, or data.
- *Adversary's Goal:* What the attacker aims to achieve through the attack.

This structured approach allows us to systematically analyze each privacy risk and develop appropriate mitigation strategies.

While this approach may be unfamiliar to readers without a security background, it provides invaluable insights for privacy protection in AI. By viewing privacy risks through the lens of threat models, we can better understand the nature and scope of potential vulnerabilities, assess their likelihood and impact, and develop more effective mitigation strategies. Without this structured approach, organizations risk overlooking critical vulnerabilities or implementing ineffective safeguards—essentially flying blind in their privacy protection efforts.

The NIST Privacy Framework outlines four primary ways organizations can handle privacy risks: mitigation, transfer, avoidance, or acceptance (National Institute of Standards and Technology, 2020). In this chapter, we focus primarily on risks and threats that arise specifically in AI contexts, presenting technical measures for mitigation in Section 3. However, it's important to acknowledge that risk transfer also

plays a significant role in modern AI ecosystems, particularly as many organizations outsource AI services to third-party providers.

To maintain a focused discussion, we limit our scope to threats that are unique to or exacerbated by AI systems. We do not cover general data security threats such as data breaches, social engineering, or malware, as these are not specific to AI contexts. For the purposes of this chapter, we define AI-specific privacy risks as those that directly relate to the AI components of a system, such as the model, its training process, or its outputs. This includes risks that arise from the way AI models learn, store, and process information, as well as how they generate outputs based on their training.

These AI-specific privacy risks include: membership inference attacks, which attempt to determine if specific data was used in training; data extraction attacks, which aim to reconstruct training data; model extraction attacks, which try to replicate the functionality of a model; model differencing attacks, which exploit changes in model behavior; gradient inversion attacks, which reconstruct data from model updates; and side channel attacks, which leverage unintended information leakage during AI operations.

2.1 Membership Inference Attacks

Membership inference (MI) attacks aim to predict whether an individual's data was used to train a model, posing a foundational privacy risk in AI (Shokri et al., 2017; Carlini et al., 2022; Hu et al., 2022). These attacks identify patterns in a model's behavior that suggest memorization or overfitting to specific training data.

In most settings, MI attacks have a low direct impact on privacy, since disclosing whether an individual's data was used to train a model is not typically sensitive. However, they can be sensitive in certain contexts—e.g., inferring criminal records from recidivism prediction models. More importantly, MI attacks serve as a foundation for stronger data extraction attacks (Carlini et al., 2021) and privacy auditing (Jagielski et al., 2020b; Nasr et al., 2021).

2.1.1 Threat Model

The attack involves a victim who deploys a model and an adversary attempting to determine if a specific instance was in the training set. Conventionally, the model is assumed to be a classifier that outputs confidence scores for all classes (Shokri et al., 2017; Carlini et al., 2022), however models that hide confidence scores may still be vulnerable (Choquette-Choo et al., 2021). More recently, MI attacks have been extended to generative models (Ren et al., 2025) and explainable models (Liu et al., 2024a).

Adversary's Capabilities The adversary has black-box query access to the model and sometimes to the training data distribution. Query access to the training data dis-

tribution is used to gather data for training "shadow models" in some attacks (Shokri et al., 2017; Carlini et al., 2022).

Adversary's Knowledge The adversary must know the target instance (input and output) but not the model parameters or full training data. Knowledge of the model architecture and training algorithm used by the victim can give the adversary an advantage, but is not essential (Carlini et al., 2022).

Adversary's Goal The adversary seeks to *accurately* predict whether the target instance was included in the victim's training data. Carlini et al. (2022) argue that attacks should maximize the true-positive rate for a fixed low false-positive rate (e.g., $\leq 0.1\%$).

2.1.2 Attack Methodology

A MI attack requires an adversary to distinguish between two possible scenarios: one where the target instance *was* in the training set, and one where it *was not* in the training set. It is therefore natural to cast a MI attack as a statistical hypothesis test, where the null hypothesis H_0 and alternative hypothesis H_1 are as follows:

$$H_0: \quad \text{"the target instance } x \text{ was not in the training set of model } M\text{"}$$
$$H_1: \quad \text{"the target instance } x \text{ was in the training set of model } M\text{"}.$$

The adversary must then design a test to decide whether there is sufficient evidence to reject H_0, and conclude that the target instance was used for training. There is considerable variation in the design of the test and the formal interpretation of H_0 and H_1 across the literature.

Generally speaking, the adversary must define a test statistic $T(M,x)$ that depends on the target instance x and the model M. Ideally, $T(M,x)$ should be sensitive to the inclusion or exclusion of x in the training set for M. For example, the adversary could define $T(M,x)$ to be the model's loss on x, which can be computed given query access to the model's confidence scores. Recent attacks use a likelihood ratio for the test statistic, which is the optimal test for a fixed false-positive rate (Carlini et al., 2022; Ye et al., 2022).

To complete the design of the test, the adversary must choose the rejection region S—the set of values of the test statistic for which the null hypothesis H_0 is rejected. For the previous example, where $T(M,x)$ is the model's loss, the rejection region S can be defined as the set of loss values below some threshold τ. This amounts to rejecting H_0 when the model's loss is small, which may indicate memorization of x. Another approach is to learn the rejection region from data by training a classifier (Shokri et al., 2017).

To meet the adversary's goal, the rejection region should be chosen to achieve a small false-positive rate. However, estimating the false-positive rate is a significant challenge in designing an effective MI attack. Mathematically, the false-positive rate (FPR) is defined as the probability that the test statistic is in the rejection region,

given that H_0 is true:

$$\text{FPR} = \Pr[T(M, x) \in S | H_0].$$

To estimate this probability, the adversary must model the distribution of the test statistic $T(M, x)$ in a setting where H_0 is true.

If the model's training algorithm is randomized and the training data is fixed under H_0, the distribution of the test statistic can be estimated by training multiple "shadow models" and computing $T(M, x)$ for each one. However, this is not feasible for the adversary, as they do not have access to the victim's training data, and they may not have access to the model architecture or training algorithm either. To get around this problem, the adversary can make further approximations. For instance, the victim's training data can be approximated by data drawn from the same distribution—if query access to the training data distribution is available—or a similar distribution. The adversary can try to make an informed guess about the victim's model architecture and training algorithm, however research has shown that strong attacks are still possible even when there is a mismatch (Carlini et al., 2022).

For large foundation models, reliably estimating FPR is challenging, limiting the reliability of MI attacks as evidence of data usage. Alternative approaches include using canary data or data extraction attacks (Section 2.2) (Zhang et al., 2024c).

2.1.3 Defenses

Defenses against MI attacks can be categorized into two types: those based on empirical approaches and those that come with provable guarantees. Empirical defenses, employ various techniques like adversarial training (Nasr et al., 2018), confidence score manipulation (Jia et al., 2019; Li et al., 2024), and model distillation Tang et al. (2022) to mitigate vulnerabilities. However, these methods lack formal privacy guarantees. In contrast, differential privacy (DP; covered in Section 3.1) provides a rigorous framework that offers provable protection against MI attacks. An (ϵ, δ)-DP mechanism bounds the power of any membership inference attack (Dong et al., 2022). Specifically, for small significance levels α[1], the power of the test is upper bounded by $e^{\epsilon}\alpha + \delta$, which is only slightly larger than α when ϵ and δ are small.

2.2 Data Extraction Attacks

As AI models grow in size and complexity, their capacity to memorize training data increases, posing a significant privacy risk. Data extraction attacks exploit this vulnerability, particularly in large generative models, where adversaries can elicit memorized training examples through careful probing (Carlini et al., 2021, 2023a). When evaluated against production language models, these attacks have achieved

[1] We assume here that $0 < \alpha \leq \frac{1-\delta}{1+e^{\epsilon}}$.

success rates of 0.5–3% per query under conservative assumptions (Nasr et al., 2025).

2.2.1 Threat Model

Data extraction attacks involve a victim who trains a model on private data and an adversary attempting to extract specific training examples. Research to date has focused on controlled settings to measure worst-case privacy risks, assuming the adversary has access to training examples or likely sources (Carlini et al., 2021, 2023a; Nasr et al., 2025).Further research is needed to understand risks under more realistic threat models.

Adversary's Capabilities The adversary may have black-box or white-box access to the target model. Black-box access typically allows the adversary to submit queries and receive responses, without controlling generation hyperparameters or the internal source of randomness. White-box access allows the adversary to observe the model weights and architecture, permitting control of randomness and computation of loss functions and gradients. Access to multiple models trained on the same data (e.g., checkpoints or size variants) can improve extraction success (More et al., 2024).

Adversary's Knowledge For targeted attacks, the adversary must have some knowledge about the training example they are trying to extract. It is common to assume knowledge of part of the training example — e.g., the caption of an image (Carlini et al., 2023a) or the prefix of a text snippet (Carlini et al., 2023b). Untargeted attacks, on the other hand, do not require knowledge of the training data, but may benefit from access to likely training sources.

Adversary's Goals The adversary aims to extract training examples, either indiscriminately (untargeted) or specifically (targeted). Success criteria vary, with current research setting high bars for extraction fidelity (Carlini et al., 2023a; Nasr et al., 2025), however lower-fidelity extraction still presents a risk (More et al., 2024).

2.2.2 Attack Methodology

Data extraction attacks against generative models typically involve two steps: eliciting memorized responses and checking for memorization.

Eliciting Memorized Responses This step is relatively straightforward for models that have only undergone pre-training, as their sole objective is to mimic the training data distribution conditioned on the prompt. As a result, the model is likely to emit memorized training examples if the prompt perfectly matches part of a training example. For instance, Carlini et al. (2023a) demonstrate that an image of a person can be extracted from a diffusion model simply by prompting with the person's name.

Eliciting memorized responses may be more challenging for instruction-tuned or aligned models designed to avoid leaking sensitive data. In such cases, adversaries

may circumvent alignment procedures by designing adversarial prompts that trick the model into reverting to its pre-training objective. For example, Nasr et al. (2025) successfully extracted sensitive information from a proprietary aligned language model by instructing it to repeat a single word indefinitely.

Regardless of the querying strategy, submitting each query multiple times can be beneficial. Due to model randomness, collecting multiple responses increases the likelihood that at least one will contain a training example.

Checking for Memorization After obtaining responses from the model, the adversary must verify whether they contain training examples or not. This can be done through direct comparison if training data is accessible, or via membership inference attacks. The latter can be adapted depending on the adversary's capabilities—e.g., Carlini et al. (2023a) describe weaker membership inference attacks against diffusion models that only require black-box access, as well as stronger attacks that use white-box access.

Auditing Data extraction attacks can play a crucial role in privacy auditing of generative models (Zhang et al., 2024c). Since the adversary is the same party as the victim when auditing, they may have stronger capabilities and knowledge than a realistic adversary, allowing them to design more powerful attacks.For instance, if the adversary controls the training data, they can insert out-of-distribution "canary" examples that are easier to extract than in-distribution examples (Carlini et al., 2019).

2.2.3 Defenses

Defending against data extraction attacks is an ongoing challenge. While several empirical strategies have shown promise, most lack rigorous privacy guarantees and do not fully eliminate the risk.

- *Data sanitization*: Detects and removes sensitive information before training begins. While it prevents the model from memorizing information it has never seen, data sanitization is rarely perfect. Sensitive information may be difficult to detect or embedded in ways that make removal challenging without compromising data quality.
- *Training data deduplication*: This strategy emerged after the discovery that models are much more likely to emit examples that are duplicated many times in the training set (Carlini et al., 2023b; Kandpal et al., 2022). While removing exact duplicates is straightforward, near-duplicates are more difficult to catch and can also lead to memorization.
- *Alignment techniques*: These methods involve fine-tuning models on guardrail-specific datasets that include scenarios like leakage of sensitive information. The goal is to encourage the model to operate within defined privacy boundaries. However, alignment may not completely mitigate the risk, as sophisticated adversaries can potentially bypass guardrails through prompt injection attacks (Nasr et al., 2025).

An exception to these empirical defenses is *differentially private training*, which comes with formal guarantees. It ensures the output model is statistically insensitive to the presence or absence of a single training example, thereby bounding the risk of extraction attacks. While differentially private training can significantly impact model performance, it has been applied successfully to language models for writing prediction, where it prevented extraction attacks with only a marginal impact on utility (Carlini et al., 2019).

2.3 Model Extraction Attacks

Model extraction attacks involve adversaries systematically probing a model's inputs and outputs to create an unauthorized copy, without access to the original training data, model parameters, or training algorithm (Tramèr et al., 2016; Jagielski et al., 2020a). This threatens the intellectual property of organizations who have invested substantially in model development. Beyond economic motivations, adversaries may attempt to extract a model to conduct downstream security or privacy attacks, essentially upgrading their access from black-box to white-box. This dual threat of intellectual property theft and security risk makes model extraction a serious concern for deployed AI systems.

2.3.1 Threat Model

The victim develops a model they wish to make available for benign usage without disclosing internal details, exposing it through an API. The adversary attempts to create an unauthorized copy using this API.

Adversary's Capabilities The adversary is assumed to have black-box query access to the victim's model, with responses varying depending on the model type (e.g., class labels, confidence scores, or raw logits for classification models). They may additionally have access to a dataset of unlabeled model inputs that can be used to generate queries, though this isn't always essential (Orekondy et al., 2019).

Adversary's Knowledge The adversary's knowledge about the victim's model may range from minimal to substantial, potentially including information about the model architecture or pre-trained components.

Adversary's Goals The adversary seeks to create a copy of the victim's model, either for economic incentives or to gain white-box access for downstream attacks (Jagielski et al., 2020a). For economic motivations, an approximate replication of functionality may suffice. For enabling white-box attacks, a functionally equivalent copy is desired. The adversary may also aim to minimize the number of queries to reduce costs and improve stealthiness.

2.3.2 Attack Methodology

There are two broad approaches to model extraction: *learning-based attacks* are suitable when the adversary seeks an approximate copy of the model, whereas *high-fidelity attacks* are designed to produce a functionally equivalent copy.

Learning-Based Attacks These attacks yield an approximate copy by training a new model using the victim's model as a labeling oracle. Typically the adversary has access to a training set of unlabeled examples. A simple approach is to select a subset of examples from this set, query labels from the oracle, and train the duplicate model from scratch using fully supervised learning (Tramèr et al., 2016). However, this may be inefficient in terms of the number of queries required. Various approaches have been proposed to improve query efficiency, including adaptive querying strategies (Orekondy et al., 2019), semi-supervised learning (Jagielski et al., 2020a), active learning (Chandrasekaran et al., 2020), and building on top of pre-trained models (Orekondy et al., 2019).

If the adversary does not have access to an unlabeled training set, they can use data-free extraction attacks. Generating random inputs from a surrogate distribution has been shown to work in some cases (Orekondy et al., 2019), but extraction accuracy and efficiency improves as the surrogate distribution approaches the victim's training distribution. To this end, adversaries may train a generative model to synthesize query examples that maximize disagreement between the victim and duplicated model (Truong et al., 2021).

High-Fidelity Attacks These attacks typically require knowledge about the victim model's architecture, as well as high-precision query access to model outputs (e.g., logits). That knowledge is then used to develop algorithms to recover the model parameters. For example, attacks have been developed that exploit the piecewise linear nature of neural networks with ReLU activations (Jagielski et al., 2020b; Rolnick and Kording, 2020). More recently, Carlini et al. (2024) demonstrated an extraction attack against a proprietary language model, that is able to recover the embedding projection layer. While this attack only recovers a small fraction of the parameters, it may prove useful for future attacks, and implicitly reveals information about the model's size, which may otherwise have been confidential.

2.3.3 Defenses

While various defensive strategies have been proposed, no current method offers comprehensive protection without potential compromises to model performance or functionality (Oliynyk et al., 2023). Defenses fall into two categories:

Proactive Defenses These aim to prevent model extraction by modifying the model's fundamental characteristics or limiting information exposed in its outputs. A common strategy is to randomly perturb the model's inputs or outputs, to make the model's behavior less certain to the adversary, without compromising model perfor-

mance. This can be done using information theoretic principles tailored for model privacy (Wang et al., 2021) or differential privacy (Zheng et al., 2019; Yan et al., 2022). To limit information exposed by the model, the victim can censor confidence scores or logits in the output, or reduce their precision, however this may limit the utility for benign users (Tramèr et al., 2016; Carlini et al., 2024).

Reactive Defenses These focus on detecting ongoing attacks or providing evidence of model ownership. The victim may be able to detect an ongoing extraction attack by monitoring the distribution of incoming queries (Juuti et al., 2019), though this can be vulnerable to attack (Feng et al., 2023). Even if the victim is unable to prevent model extraction, they may be able to gather evidence to pursue legal action against a suspected attacker. Proof-of-learning requires the victim to demonstrate unique characteristics of the training process that would be difficult for an adversary to reproduce (Jia et al., 2021b). Dataset inference can identify distinctive behavior in a model that is only likely to present when trained on the victim's training data (Maini et al., 2021). Finally, watermarking techniques can be used to embed imperceptible signals in the model that only the owner knows how to extract (Jia et al., 2021a).

2.4 Model Differencing Attacks

The release of multiple versions of a model, through updates or fine-tuning, amplifies privacy risks by enabling model differencing attacks. These attacks analyze changes between model versions to infer information about training data that was added, removed, or modified (Salem et al., 2020; Zanella-Béguelin et al., 2020; Chen et al., 2021). This expands the attack surface and can amplify other privacy risks, such as membership inference or reconstruction attacks (Jagielski et al., 2023). The risk is particularly acute when data changes between releases are small and targeted, such as removing an individual's data in response to a deletion request (Chen et al., 2021).

2.4.1 Threat Model

In a model differencing attack, an adversary observes two or more sequentially released model versions with modified training data. The models typically share the same architecture but may have different parameter values. The victim could be the organization releasing the models or specific individuals whose data was modified between releases.

Adversary's Capabilities Access to model versions can range from black-box to white-box. An adversary with black-box access can query the models and compare outputs. They may also exploit access to intermediate model states or gradients if these are exposed through APIs or collaborative learning settings. An adversary with white-box access can analyze changes in model parameters directly, potentially revealing more precise information about the training data modifications.

Adversary's Knowledge The adversary may have knowledge about when training data was modified, the number of examples changed, or partial information about the modified examples. In some cases, the adversary might even know which specific examples were modified (e.g., through public announcements of data deletion requests), and use this information to focus their attack on reconstructing the unknown features of those examples.

Adversary's Goals The adversary aims to infer information about training examples that change between model releases. The scope and granularity of the inferred information may vary: from determining whether a specific example was added or removed to reconstructing features of modified examples.

2.4.2 Attack Methodology

Research has primarily focused on two categories of model differencing attacks: (1) specialized membership inference attacks, where the aim is to determine whether a particular training example was added or removed between model releases; and (2) specialized data extraction attacks, where the aim is to extract all or part of the modified training data. These attacks are related to their counterparts for standalone models discussed in Sections 2.1 and 2.2. Here we discuss specializations that exploit differences between successive model releases.

Membership Inference Attacks In the model differencing setting, the adversary's goal is to infer whether a given training example was added or removed between two model releases. This can be achieved by running a standalone membership inference (MI) attack on each model, however stronger attacks have been proposed that use the model releases *jointly* for inference. Chen et al. (2021) use shadow models to train a MI classifier that uses the outputs from all models to make a prediction. On the other hand, Jagielski et al. (2023) combine information from standalone MI attacks against each model, achieving a higher attack accuracy using a smaller number of shadow models.

Data Extraction Attacks These attacks operate differently from the standalone setting, in that they undertake a differential analysis of model behavior. For structured data, techniques include majority voting on disagreeing examples (Gao et al., 2022), predicting updated attributes based on model confidence changes (Hui et al., 2023), and exact solutions for simple models like linear regression (Bertran et al., 2024). For unstructured data, approaches include gradient inversion attacks (Hu et al., 2024), encoder training using shadow models (Salem et al., 2020), and beam search for extracting text snippets with large perplexity differences (Zanella-Béguelin et al., 2020).

2.4.3 Defenses

Defending against model differencing attacks presents a significant challenge, as it requires balancing privacy protection with the practical needs of model updates and performance improvements. Several strategies have been proposed, each with its own trade-offs:

- *Reducing update frequency and batch updates*: Modifying a larger number of training examples between releases makes individual changes harder to isolate. However, this may conflict with operational needs for frequent updates.
- *Limiting output information*: For scenarios where models are exposed via query APIs (black-box settings), withholding confidence scores (Chen et al., 2021; Salem et al., 2020) or providing only top-k predictions (Zanella-Béguelin et al., 2020) can hamper attack efficacy while preserving much of the model's utility.
- *Differential privacy*: Applying differential privacy (DP) to model releases or outputs provides formal guarantees about individual training example protection. However, DP inherently limits performance improvements in updated models, highlighting the tension between privacy protection and utility (Zanella-Béguelin et al., 2020).

As model updates remain crucial for many AI applications, developing robust defenses against differencing attacks that preserve both privacy and utility continues to be an important challenge for the field.

2.5 Gradient Inversion Attacks

Gradient inversion attacks exploit access to gradients to reconstruct private training examples, posing a significant privacy risk in machine learning systems (Wang et al., 2019; Zhu et al., 2019; Boenisch et al., 2023). This vulnerability is particularly critical in distributed learning paradigms like federated learning (FL), where participants exchange model updates instead of raw data. Despite FL's promotion as privacy-preserving, these attacks demonstrate that a curious server or other participants can potentially recover individual training examples from shared gradients.

2.5.1 Threat Model

Gradient inversion attacks are predominantly studied within federated learning systems, where a participants share model updates with each other or a central server. The most common threat model casts the central server as the adversary and the participants as victims.

Adversary's Capabilities Two settings are typically considered: an honest-but-curious server that passively observes updates (Zhu et al., 2019; Geiping et al., 2020), or a dishonest server that may modify the model architecture (Fowl et al.,

2022) or deviate from the established protocol (Fowl et al., 2023; Boenisch et al., 2023) to make data reconstruction easier.

Adversary's Knowledge Since the central server is typically managed by the model developer, it is usually assumed to have full knowledge of the model architecture, loss function, and training algorithm, and may have background knowledge about the data distribution. For instance, in the vision domain, attacks may leverage prior information about the distribution of images to steer the attack towards more realistic reconstructions (Yin et al., 2021; Jeon et al., 2021).

Adversary's Goal The primary goal is to recover information about the participant's private training data, ranging from exact reconstruction of entire batches (Dimitrov et al., 2024) to recovering small portions of highly sensitive data (Chu et al., 2023).

2.5.2 Attack Methodology

Gradient inversion attacks vary depending on whether the central server is honest-but-curious or dishonest. We provide an overview of attacks in both these settings, covering approaches for both real-valued and discrete data, including images and text.

Honest-but-Curious Server Many attacks are based on gradient matching (Zhu et al., 2019), where the server produces an estimate $\hat{x}$ for the private data x by minimizing the distance between the received gradient and the gradient evaluated at $\hat{x}$. Regularization terms can improve reconstruction quality for large batch sizes and high-dimensional data (Geiping et al., 2020; Jeon et al., 2021). Exact reconstruction is possible in certain scenarios (Zhu and Blaschko, 2021; Dimitrov et al., 2024).

Discrete data, like text, poses a challenge for gradient inversion attacks, as gradient-based optimizers cannot be used directly to solve the gradient matching problem. Approximate reconstruction of multiple text sequence has been demonstrated by incorporating auxiliary language models and alternating between discrete and continuous optimization steps (Balunovic et al., 2022). Recent work has demonstrated exact recovery of long sequences for transformer models (Petrov et al., 2024).

Dishonest Server Attacks in this setting provide participants with a malicious model to aid in the recovery of data from gradients. One approach is to select a malicious model architecture, before the model is fixed and compiled for production. For example, Fowl et al. (2022) propose to insert a special "imprint" layer, which is designed with a sparse activation pattern so that each data point only activates one or a few neurons. This allows the attacker to isolate individual data points in the gradient.

Recent research has explored attacks against transformer-based language models, by modifying the weights in the attention layers to achieve malicious behaviors. Chu et al. (2023) achieve targeted reconstruction of short sensitive sequences (e.g., social security numbers) by modifying weights in attention layers to tag relevant sequences and altering weights in linear layers to filter and capture only the tagged data of

interest. Similar ideas have been used to reconstruct long sequences by cleverly encoding token information within the gradient of the model parameters (Fowl et al., 2023).

2.5.3 Defenses

Reasonably effective defenses are available against gradient inversion attacks, albeit at the cost of increased computation and communication overhead. Two primary defense strategies have been proposed:

- *Secure aggregation*: Cryptographic protocols that aim to ensure only aggregated updates are observed by the server (Bonawitz et al., 2017). While they are relatively efficient, recovery from aggregated updates is possible for some protocols (Kariyappa et al., 2023) and they typically require synchronous training (Zhang et al., 2020).
- *Homomorphic encryption*: Allows computations on encrypted data, providing strong privacy guarantees (Phong et al., 2018; Truex et al., 2019). While there is typically a substantial computational and communication overhead, recent work has made progress in addressing efficiency concerns in a federated learning context (Zhang et al., 2020).

Local differential privacy, while conceptually relevant, is generally not regarded as a practical defense against gradient inversion attacks in federated learning settings. It tends to cause severe degradation in model utility due to the amount of noise that must be added to achieve reasonable privacy guarantees.

2.6 Other Attacks

2.6.1 Linkage Attacks

Linkage attacks combine information from multiple data sources to re-identify or infer sensitive information about individuals (Narayanan et al., 2011). While these attacks don't directly exploit AI systems, they represent a significant privacy risk in the context of AI due to the potential for automation and enhancement using AI capabilities.

Linkage attacks involve an adversary with access to auxiliary information that can be correlated with supposedly "anonymized" data or model outputs. This auxiliary information may be publicly available or obtained by mounting other attacks. The adversary must have some knowledge about potentially linkable information types. For example, an adversary might recognize that combining a anonymized medical dataset with public voting records could link individuals based on common attributes like age, gender, and zip code.

In an AI context, the risk of linkage attacks is exacerbated by several factors:

- *Automation*: AI can automate the process of finding correlations between datasets, making linkage attacks more efficient and scalable.
- *Data aggregation*: AI systems often require large amounts of data from various sources, potentially creating more opportunities for linkage attacks if not properly managed.
- *Integration with AI-specific attacks*: Linkage attacks become more powerful when combined with AI-specific attacks like membership inference or model inversion. The auxiliary information can provide the context needed to make these attacks more effective or to interpret their results in ways that reveal more sensitive information.

Defending against linkage attacks requires implementing formal privacy techniques such as differential privacy and data minimization, alongside systematic risk assessment processes.

2.6.2 Side-Channel Attacks

Side-channel attacks against AI systems exploit indirect information leakage through observable physical or operational characteristics, presenting unique privacy risks beyond traditional security concerns. These attacks are particularly relevant in enterprise AI deployments, especially when using shared infrastructure or cloud services.

In the context of AI, side-channel attacks typically aim to extract sensitive information about the model, its parameters, or the training/inference data. The adversary may have limited direct access to the system, but can observe characteristics such as timing, power consumption, or memory access patterns during training or inference.

Key types of side-channel attacks relevant to AI privacy include:

Timing Attacks By measuring execution time of specific operations, adversaries can infer information about the model or input data. Notably, these attacks can compromise differential privacy guarantees in AI systems. Jin et al. (2022) demonstrated success rates up to 99.65% for end-to-end timing attacks on private sums computed with discrete Laplace noise, undermining the intended privacy guarantees.

Memory and Cache Attacks Adversaries can exploit patterns of memory access, including cache behavior, to infer sensitive information about AI models or their data. By monitoring which memory locations or cache lines are accessed during model execution, attackers can potentially reconstruct parts of the model's architecture, parameters or even characteristics of training or inference data. This is particularly concerning for enterprises using multi-tenant cloud services for AI workloads, where memory and cache resources are often shared between different users or processes.

Power Analysis Attacks By measuring power consumption of hardware during model execution, adversaries can infer information about operations being performed. For example, Wolf et al. (2021) use collected power traces to distinguish between models with small architectural differences with over 90% accuracy, while Wei et al. (2018) recover inputs to convolutional networks running on FPGAs.

Mitigations for side-channel attacks in AI systems may involve:

- Considering the use of specialized hardware, such as TEEs, designed to resist side-channel attacks for highly sensitive AI workloads.
- Exploring the use of oblivious algorithms to mask data and memory access patterns.
- Employing side-channel aware differential privacy frameworks that incorporate potential information leakage into privacy calculations.
- Carefully managing resource allocation in cloud environments to minimize shared resources between different tenants.

3 Privacy-Enhancing Technologies

Having explored the landscape of privacy threats in enterprise AI, we now turn our attention to privacy-enhancing technologies (PETs) designed to mitigate these risks. These technologies offer a range of defenses against the attacks we've discussed, from membership inference to gradient inversion.

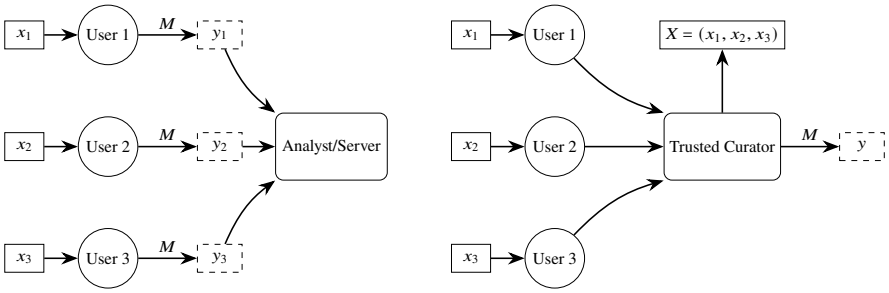
This section examines these technologies, their principles, and their applications in addressing privacy risks. Importantly, privacy in enterprise AI requires nuanced solutions tailored to specific contexts and threat models. We will explore various approaches, each offering distinct strengths and trade-offs in protecting sensitive information in AI systems.

Our discussion will cover several mainstream privacy-enhancing technologies: differential privacy, secure multiparty computation, trusted execution environments, federated learning, and machine unlearning. Through this overview, we aim to provide a comprehensive understanding of the tools available to enterprises for enhancing privacy in their AI deployments, enabling informed decisions about which technologies best suit specific use cases.

3.1 Differential Privacy

In the era of AI and machine learning, differential privacy (DP) has emerged as a cornerstone of responsible data use and model development. As noted in Sections 2.1 and 2.2, AI models trained on vast amounts of potentially sensitive data risk inadvertently memorizing and revealing private information. This is particularly concerning in enterprise settings, where AI systems may process confidential customer data, proprietary business information, or other sensitive records.

DP provides a rigorous framework for bounding privacy leakage in AI systems, enabling organizations to quantify and control the privacy-utility trade-off. It can be applied at various stages of the machine learning pipeline, from data collection and preprocessing to model training and deployment. At its core, DP is not a single



Local Model: Each user applies a randomized mechanism M locally before sending data to the server.

Central Model: Users send raw data to a trusted curator. The curator applies the privacy mechanism M on the collected dataset.

Fig. 2 Illustration of the local (left) and central (right) deployment models of differential privacy.

algorithm but a comprehensive mathematical framework encompassing a variety of techniques and approaches.

In this section, we will explore the core concepts of DP, its different models of deployment, and key mechanisms used to achieve DP in machine learning. We will also discuss the challenges and limitations of DP, provide a forward-looking outlook for enterprises considering the adoption of differential privacy in their AI systems.

3.1.1 Deployment Models for Differential Privacy

Differential privacy (DP) can be implemented in various ways, depending on where the trust boundary lies in the data processing pipeline. The two primary models are the central and local model, as illustrated in Fig. 2.

Central Model In this model, a trusted curator collects and holds raw data from individuals, applying DP mechanisms before releasing any information. This model is often the most practical and efficient when the data is already collected and stored in a central location and there is a clear authority responsible for data governance. However, it raises concerns about the concentration of raw data in one location and requires careful implementation of security measures to guard against data breaches.

Local Model This model removes the assumption of a trusted curator. Each individual privatizes their own data before sharing it with a data processor (Duchi et al., 2013). While this offers stronger privacy guarantees by reducing the risk of large-scale data breaches, it typically requires much stronger noise injection, often leading to significant utility loss. This can make complex analyses or accurate model training challenging, limiting its practical applications.

The choice between deployment models depends on various factors including the specific application, legal and regulatory requirements, trust assumptions, and the desired balance between privacy and utility. In enterprise AI settings, the central

model might be more common when dealing with internal data, while the local model could be preferred for collecting data from external sources. Regardless of the deployment model, the fundamental principles of DP remain consistent. For ease of exposition, we will focus on the central model in subsequent discussions.

3.1.2 Formalizing Differential Privacy

Differential privacy (DP) provides a formal framework for quantifying and guaranteeing privacy in data analysis. The key idea is to make it difficult to distinguish between two possible worlds: one where an individual contributes their data, and one where they do not. This indistinguishability is achieved by introducing controlled randomness into the data processing (Dwork and Roth, 2014).

To formalize this concept, consider a dataset $D \in \mathcal{D}$ containing sensitive records from individuals, and a randomized algorithm M that processes this data to produce an output $S \in \mathcal{R}$. We define neighboring datasets D and D' as those differing in exactly one record—these correspond to the two "world" that should be difficult to distinguish (Dwork and Roth, 2014). The formal definition of (ϵ, δ)-differential privacy is as follows:

Definition 1 A randomized algorithm M is (ϵ, δ)-differentially private if for all neighboring datasets $D, D' \in \mathcal{D}$, and for all subsets of outputs $S \subseteq \mathcal{R}$:

$$\Pr[M(D) \in S] \leq e^{\epsilon} \cdot \Pr[M(D') \in S] + \delta.$$

Here, $\epsilon \geq 0$ and $0 \leq \delta < 1$ quantify the privacy guarantee, with smaller values indicating stronger privacy. When $\delta = 0$, the inequality ensures that the probability of observing any particular event is at most e^{ϵ} times as likely under D as under D'. For small ϵ, this means the presence or absence of an individual changes probabilities by at most about ϵ. The case The case where $\delta > 0$ allows for a small probability of violating this bound. This relaxation can allow for greater utility in some cases, particularly when dealing with unlikely events.

Selecting the Privacy Budget The parameters ϵ and δ constitute the "privacy budget" of the algorithm, representing a trade-off between privacy protection and output utility. Researchers at Google characterize the level of privacy into three tiers based on the choice of ϵ (Ponomareva et al., 2023):

- Tier 1: $\epsilon \leq 1$. Provides a strong privacy guarantee at the expense of utility.
- Tier 2: $1 < \epsilon \leq 10$. Provides a reasonable level of privacy for many applications.
- Tier 3: $\epsilon > 10$. Provides weak to no formal guarantees, however still represents an improvement over no privacy protection at all.

The parameter δ should be chosen to be cryptographically small, typically scaling inversely with the size of the dataset. A common rule of thumb is to set $\delta \ll \frac{1}{n}$, where n is the number of records in the dataset, which ensures the probability of a significant privacy breach remains negligible even for large datasets.

Despite the mathematical rigor of DP, explaining its guarantees to non-experts presents significant challenges. A study by Nanayakkara et al. (2022) found that even with carefully designed visualizations, many participants struggled to fully grasp the implications of different epsilon values.

Useful Properties of Differential Privacy DP's widespread adoption is partly due to its practical properties:

Composition DP satisfies composition theorems that allow for accounting of overall privacy loss when applying a sequence of DP mechanisms. Advanced composition techniques allow the total ϵ to grow sublinearly with a small degradation in δ, provided the individual privacy parameters are fixed in advance (Dwork et al., 2010). Further improvements are possible by exploiting properties of specific mechanisms or using variants like zero concentrated DP (Bun and Steinke, 2016) or Gaussian DP (Dong et al., 2022). It is also possible to compose mechanisms while adapting privacy parameters on-the-fly using fully adaptive composition (Whitehouse et al., 2023).

Post-processing DP is invariant under post-processing, meaning any arbitrary transformation of a DP output remains DP without degradation in the privacy guarantee. This property provides robustness against unplanned analyses and simplifies the design of privacy-preserving systems by separating privacy considerations from downstream data uses. This flexibility is particularly valuable in evolving research or business environments where future data needs may not be fully known at the time of data release.

3.1.3 Differentially Private Mechanisms for AI

We now present two prominent differentially private mechanisms for training AI models: Differentially Private Stochastic Gradient Descent (DP-SGD) and Differentially Private Follow The Regularized Leader (DP-FTRL), with a primary focus on DP-SGD due to its widespread use and general applicability.

Differentially Private Stochastic Gradient Descent DP-SGD, proposed by Abadi et al. (2016), is a differentially private adaptation of stochastic gradient descent (SGD). It is described in pseudocode in Algorithm 1.

Comparison with SGD DP-SGD diverges from ordinary SGD in several key ways:

- *Batching.* Batches are assembled by drawing examples with/out replacement (line 4). Deviating from this specification may compromise the privacy guarantee.
- *Gradient Computation.* Gradients are computed *per example* (line 6) unlike SGD, where they are typically computed *per batch*.
- *Gradient Clipping.* Individual gradients are clipped to bound their ℓ_2-norm (line 6). This bounds the sensitivity of the computation.
- *Noise Addition.* Calibrated Gaussian noise is added to the averaged clipped gradients (line 9). This step is crucial for providing the DP guarantee.

Algorithm 1 Differentially Private SGD

Input: Training examples $D = \{x_1, \ldots, x_N\}$, loss function L, learning rate η_t, noise scale σ, gradient norm bound C
Initialize model parameters θ_0 randomly
for $t = 0, \ldots, T-1$ **do**
    Select a batch B_t by either (1) sampling b examples from D without replacement or (2) sampling examples from D with replacement with probability $q = b/N$
    for $x_i \in B_t$ **do**
        Compute gradient $g_i \leftarrow \nabla_\theta L(\theta_t, x_i)$
        Clip gradient $\bar{g}_i \leftarrow g_i / \max(1, \|g_i\|_2 / C)$
    end for
    Add noise $\tilde{g} \leftarrow \frac{1}{|B_t|}\left(\sum_i \bar{g}_i + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$
    Update model parameters $\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{g}$
end for
Output: θ_T

- *Privacy Accounting*. The privacy loss is tracked during training to ensure it does not exceed the prespecified budget.

Practical Considerations Implementing DP-SGD involves balancing privacy and utility. Higher privacy levels typically require more noise, potentially reducing model accuracy. DP-SGD also incurs increased computational costs due to per-example gradient computations and the added noise slowing convergence. The performance can be sensitive to data distribution, with skewed or outlier-heavy datasets potentially requiring more noise and leading to greater utility loss. As a result, careful consideration of the data distribution and potential preprocessing may be necessary when applying DP-SGD to real-world datasets.

Implementations DP-SGD has been implemented in several machine learning frameworks, including TensorFlow Privacy Abadi et al. (2016) and Opacus for PyTorch (Yousefpour et al., 2022).

Differentially Private Follow-the-Reguarlized-Leader DP-FTRL (Kairouz et al., 2021) is an alternative mechanism particularly well-suited for federated learning (see Section 3.4. It operates on aggregated model updates rather than per-example gradients, making it easier to implement in environments where access to individual data points is restricted. In DP-FTRL, clients train local models using the FTRL algorithm, send model updates to a central server, which then adds noise to ensure differential privacy before refining the global model. While DP-FTRL offers advantages in distributed settings, DP-SGD generally provides stronger privacy guarantees but requires more granular data access.

3.1.4 Summary and Enterprise Outlook

Differential privacy has emerged as a powerful framework for protecting individual privacy in machine learning applications. Mechanisms like DP-SGD and DP-FTRL

have made it possible to train and release models with strong privacy guarantees, albeit at the cost of reduced utility or increased computational burden.

For enterprises considering the adoption of DP in their AI systems, several key considerations and future trends are worth noting:

- *Regulatory Compliance:* As privacy regulations evolve, DP may become increasingly important for demonstrating compliance. Enterprises may consider monitoring regulatory developments and examining how DP could fit into their compliance strategies.
- *Privacy-Utility Trade-offs:* Organizations need to carefully balance privacy guarantees with model performance. This may involve sector-specific assessments of acceptable privacy levels and performance thresholds.
- *Explainability and Trust:* DP can be a powerful tool for building trust with customers and stakeholders, however communicating its benefits and limitations remains challenging. Enterprises may consider developing strategies for effectively explaining their privacy-preserving measures.

As the field progresses, we can expect differential privacy to become an increasingly integral part of the enterprise AI landscape, particularly in domains dealing with sensitive data. Organizations that proactively engage with DP technologies and build relevant expertise will be better positioned to navigate the evolving privacy landscape while leveraging the potential of their data assets.

3.2 Secure Multiparty Computation

Secure multiparty computation (SMPC) mitigates privacy risks when multiple parties wish to cooperate by sharing data and compute, for example, when jointly learning a model from a distributed dataset. This section examines two fundamental approaches to SMPC that have demonstrated particular promise for AI applications: secret sharing and homomorphic encryption.

3.2.1 Secret Sharing

Secret sharing splits data into seemingly random "shares" distributed among multiple parties, where individual shares reveal nothing about the original data, yet computations performed on these shares yield meaningful results when combined. It allows organizations to collaborate on data analysis, model training, and inference tasks without exposing their underlying data assets.

For AI applications requiring both collaboration and data protection, several secret sharing approaches form the foundation of secure computation:

- *Shamir's Threshold Secret Sharing* (Shamir, 1979) implements a (t, n)-threshold scheme using polynomial interpolation where any t out of n parties can reconstruct the secret, providing information-theoretic security.

- *Replicated Secret Sharing* as used in CryptGPU (Tan et al., 2021) distributes redundant shares so each party in their 2-out-of-3 scheme holds two shares, enabling faster computation with lower communication overhead.
- *Additive Secret Sharing* splits a secret s into shares that sum to the original value ($s = s_1 + s_2 + ... + s_n$), enabling efficient addition and forming the basis of frameworks like SecureML (Mohassel and Zhang, 2017).
- *Boolean Secret Sharing* operates on bits rather than field elements, enabling efficient implementation of non-linear operations such as ReLU activation functions and comparison-based operations (Patra and Suresh, 2020).

These primitives are used in a complete SMPC system as follows:

1. *Input Sharing*: Data owners distribute their private inputs using one of the secret sharing schemes described above.
2. *Secure Evaluation*: Parties perform computations directly on shares without revealing the underlying data.
   - Addition operations can typically be performed locally without interaction
   - Multiplication operations require carefully designed protocols with communication between parties
   - Complex operations like neural network activations require specialized protocols
3. *Output Reconstruction*: Only the final results are revealed, while intermediate values remain protected.

The choice of sharing scheme directly impacts the performance-security tradeoff. For example, additive sharing enables fast addition operations and serves as a key component in matrix multiplications, while boolean sharing excels at non-linear functions like ReLU and sigmoid.

AI workloads present unique challenges for SMPC due to their computational intensity. Several frameworks have emerged to address these requirements:

SecureML Mohassel and Zhang (2017) implements privacy-preserving machine learning between two non-colluding parties using several secret sharing primitives. It achieves efficiency through optimized fixed-point arithmetic, vectorized operations, and SMPC-friendly activation functions. On a dataset of size 60,000 with 784 features, their implementation runs logistic regression in less than 30 seconds and trains a 3-layer neural network within 6 hours. The Python code below demonstrates linear regression, accomplished with a few lines of code.

Linear Regression using SecureML

```
# 1. Forward propagation
y_pred = secure_multiply(X_shared, w_shared)

# 2. Compute difference from true labels
diff = y_pred - y_shared
```

```
# 3. Compute gradient
gradient = secure_multiply(X_shared.T, diff)

# 4. Update weights
w_shared = w_shared - learning_rate * gradient
```

CrypTen Knott et al. (2021) bridges theoretical SMPC protocols and practical AI development with a PyTorch-like API, enabling seamless transition to SMPC. This is demonstrated below where the code appears like idiomatic PyTorch code.

SGD step in PyTorch and CryptTen

```
# Given encrypted tensors x, y, w
# Secure SGD update in CrypTen
grad = (x.matmul(w) - y).matmul(x.transpose(0, 1))
w = w - alpha * grad
```

CryptGPU (Tan et al., 2021) addresses SMPC computational bottlenecks through GPU acceleration. Using replicated secret sharing in a three-party honest-majority setting, it achieves 150× faster convolutions, 10× faster ReLU activations, and 2–36× end-to-end acceleration for training and inference by mapping secret-shared operations to GPU-optimized primitives.

Recent advances have expanded secret sharing-based SMPC beyond traditional neural networks to transformer architectures. SIGMA (Gupta et al., 2024) enables secure GPT inference using function secret sharing, processing 13 billion parameter models in 33 seconds. SHAFT (Kei and Chow, 2025), utilizing arithmetic secret sharing, further reduces communication overhead by ~25–41%, while maintaining running times comparable to SIGMA.

3.2.2 Homomorphic Encryption

Homomorphic Encryption (HE) enables direct computation on encrypted data without decryption. HE schemes preserve algebraic structure between plaintext and ciphertext domains, with two fundamental properties:

- $E(x) \oplus E(y) = E(x + y)$ [Additive homomorphism]
- $E(x) \otimes E(y) = E(x \times y)$ [Multiplicative homomorphism]

These properties allow operations on encrypted data to yield results that, when decrypted, match the operations performed on the original plaintext. HE schemes can be categorized by their computational capabilities and efficiency trade-offs, ranging

Type	Scheme	Operations	Data Type	Applications
PHE	RSA	Multiplication only	Integers	Digital signatures, data aggregation
	Paillier	Addition only	Integers	Statistical analysis, electronic voting
	ElGamal	Multiplication only	Integers	Privacy systems
SWHE	BGN	Limited addition & 1 multiplication	Integers	Quadratic functions
	BGV	Limited*	Integers	Integer arithmetic
	BFV	Limited*	Integers	Boolean circuits
	CKKS	Limited*	Real/complex	Machine Learning, statistics
FHE	GSW	Unlimited operations	Boolean	Circuit evaluation
	BGV	Unlimited†	Integers	Integer arithmetic
	BFV	Unlimited†	Integers	Boolean circuits
	CKKS	Unlimited†	Real/complex	Machine learning, signal processing
	TFHE	Unlimited†	Boolean	Circuit evaluation, logic operations
	FHEW	Unlimited†	Boolean	Boolean circuits
	MKHE	Varies by base scheme	Base dependent	Cross-organization computation

Table 1 Comparison of HE schemes by type: partially homomorphic encryption (PHE), somewhat homomorphic encryption (SWHE) and fully homomorphic encryption (FHE). Here † denotes "with bootstrapping" and * denotes "without bootstrapping".

from *partially homomorphic* which supports single operations, to *fully homomorphic* which supports arbitrary computations, as detailed in Table 1.

Among recent schemes, CKKS (Cheon et al., 2017) advances AI-compatible HE by enabling approximate arithmetic on real/complex numbers. By encoding vectors directly into polynomial rings, it eliminates integer conversion overhead; this is a critical optimization for privacy-preserving neural networks and ML workloads.

Despite promising theoretical foundations, Homomorphic Encryption faces substantial practical challenges for AI deployment. FHE operations remain 3–6 orders of magnitude slower than their plaintext equivalents, creating a significant performance gap that impacts real-world applications (Bergerat et al., 2023). Memory efficiency presents another critical concern, as ciphertexts typically expand in size by 2–3 orders of magnitude compared to plaintexts. This expansion is particularly problematic for deep learning models with millions of parameters, where encrypted inference can require gigabytes of memory even for relatively small networks. Additionally, optimal HE parameter selection introduces complexity through necessary trade-offs between security levels, numerical precision, and circuit depth, requiring careful consideration during implementation.

Recent engineering advances are rapidly closing implementation gaps in homomorphic encryption for AI applications. *Hardware acceleration* has achieved remarkable breakthroughs, with FPGA implementations demonstrating speedups ranging

from 13× to over 3200× (Sinha Roy et al., 2019; Yang et al., 2025). *Domain-specific architectures* have emerged as critical enablers, with systems like HEAX (Riazi et al., 2020), F1 (Samardzic et al., 2021), HAAC (Mo et al., 2023), and Cinnamon (Jayashankar et al., 2025) leveraging parallel computation across different scales. Particularly noteworthy is OLA (Yang et al., 2025), which implements scale-out architectures that distribute computational workloads across multiple chips to maximize throughput for complex neural network operations.

Algorithmic optimizations have simultaneously addressed memory and communication bottlenecks through advanced data encoding techniques. Frameworks such as TenSEAL (Benaissa et al., 2021) and Gazelle (Juvekar et al., 2018) have dramatically reduced resource requirements, with Gazelle achieving a remarkably compact 0.5MB memory footprint. These optimizations are complemented by *compiler-level innovations* from systems like EVA (Dathathri et al., 2020) and Zama's Concrete-ML (Zama, 2022), which automate parameter selection, optimize circuit depth, and approximate non-linear functions with polynomials, enabling efficient encrypted computation without requiring deep cryptographic expertise from developers.

The integration of these advancements has enabled the development of practical privacy-preserving AI systems. Hybrid cryptographic frameworks like Cheetah (Huang et al., 2022) strategically combine homomorphic encryption for linear operations (convolution, batch normalization, fully-connected layers) with oblivious transfer protocols for non-linear activations (ReLU, truncation), to reduce latency and communication overhead.

Production-ready libraries have further bridged research-practice gaps, with Microsoft SEAL providing general-purpose HE capabilities that form the foundation for many higher-level implementations. OpenMined's TenSEAL, built upon SEAL, extends these capabilities with specialized support for tensor operations including addition, subtraction, and multiplication, making it particularly suitable for machine learning workloads. NVIDIA has integrated HE into its Clara Train SDK to enable privacy-preserving federated learning, while IBM's HElib (Halevi and Shoup, 2020) offers highly optimized implementations of multiple HE schemes. Intel's HE-Transformer completes this ecosystem by enabling efficient tensor operations on encrypted data through hardware-optimized primitives.

3.2.3 Summary and Enterprise Outlook

Secret Sharing distributes data fragments across multiple parties who must coordinate computation, providing post-quantum resistance and robustness against certain corruptions, but requiring significant communication overhead. Homomorphic Encryption allows computing directly on encrypted data without decryption, offering a stronger threat model but suffering from extreme computational costs and vulnerability to side-channel attacks. Both approaches remain orders of magnitude away from efficiently supporting large AI models.

As hardware acceleration continues to evolve through specialized FPGA implementations and domain-specific architectures, and as algorithmic innovations further

reduce computational and communication overhead, we can expect wider adoption of SMPC techniques across industries where data privacy is paramount.

Organizations implementing SMPC should first evaluate which approach best matches their specific privacy needs and technical constraints. Enterprises should leverage production-ready frameworks like TenSEAL, Microsoft SEAL, or CrypTen that offer streamlined development through high-level APIs and machine learning framework integrations. Hybrid approaches can optimize privacy-utility trade-offs by applying SMPC only to the most sensitive computations. For sustained privacy requirements, investing in specialized hardware accelerators (now available through many cloud providers) can substantially improve performance. While SMPC is not yet viable for all AI workloads, organizations should begin exploring these technologies now, as privacy-preserving computation is becoming both a regulatory necessity and competitive advantage.

3.3 Trusted Execution Environments

Trusted Execution Environments (TEEs) can provide protection to AI workloads via isolation—allowing such workloads to run in an untrusted computing environment. TEEs provide hardware-enforced isolation for sensitive computations, creating protected enclaves where code and data remain confidential even from privileged software like operating systems. This architecture (McKeen et al., 2013) establishes processor-level security features that create distinct trusted and untrusted execution domains, enabling secure execution of confidential workloads in otherwise untrusted cloud environments.

Fig. 3 illustrates one approach to hardware-enforced isolation between trusted and untrusted execution environments (Sierra-Arriaga et al., 2020). This figure demonstrates a clear boundary between a non-secure world with untrusted applications on a conventional operating system (OS) and a secure world where trusted applications operate on a specialized trusted OS. This security model ensures only trusted applications can access sensitive processor functions, protected memory regions, and dedicated peripherals. This security model provides confidentiality and integrity protection that persists even when the cloud provider presents risks, though it's worth noting that sophisticated attackers with physical access may still potentially bypass these protections through side-channel methods.

However, it's important to note that not all TEEs operate according to this exact model. For instance, Intel Software Guard Extensions (SGX) implements a different approach that doesn't require a separate trusted OS counterpart, instead creating secure enclaves directly within the untrusted operating system environment (McKeen et al., 2013). These enclaves utilize hardware-based memory encryption and integrity verification to protect sensitive data and computations from privileged software. While Fig. 3 shows a clear physical separation between environments, the underlying mechanisms for establishing and maintaining isolation vary considerably across TEE technologies. Some implementations rely on processor modes and hardware

security extensions, while others use memory encryption and attestation protocols. This architectural diversity enables TEEs to be deployed across various computing environments—from resource-constrained mobile devices to high-performance cloud infrastructure—while maintaining strong security guarantees appropriate to each context's specific requirements and threat models.

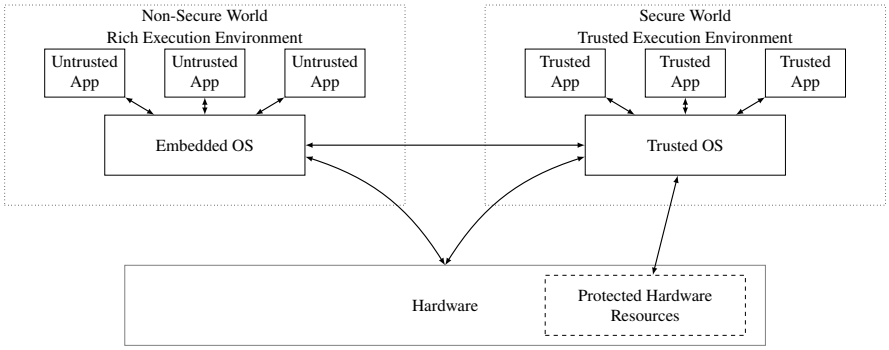


Fig. 3 Hardware-enforced isolation: trusted vs. untrusted execution environments.

3.3.1 Commercial TEE Hardware Architectures

The landscape of Trusted Execution Environment (TEE) technologies has evolved significantly, with numerous commercial implementations now available across diverse computing platforms. These technologies can be broadly categorized into hardware-centric TEEs that embed security directly into processors and chipsets, and cloud service TEEs that provide confidential computing capabilities as managed offerings. As shown in Table 2, which represents our compilation of information from vendor documentation and product websites as of May 2025, each architecture employs distinct approaches to achieving the fundamental TEE goals of isolation, encryption, and attestation. Hardware-centric solutions range from Intel's enclave-based SGX to ARM's dual-world TrustZone architecture, while cloud providers have integrated these hardware technologies into their service offerings. Understanding the security features and market availability of these TEE architectures is essential for organizations seeking to implement robust security measures for their most sensitive computational workloads.

3.3.2 CPU-GPU TEE Integration

In the trusted execution landscape, CPU-based and GPU-based TEEs have developed along separate technological paths due to fundamental differences in hardware

Table 2 Commercial Trusted Execution Environment (TEE) Architectures

TEE Technology	Key Security Features	Market Availability
Hardware-Centric TEEs		
Intel SGX	Application-level enclaves with memory encryption, attestation, and isolated execution	Widespread in server market, reduced in consumer chips
ARM TrustZone	Secure/non-secure world separation with hardware-enforced isolation	Billions of devices across mobile and embedded markets
NVIDIA Confidential Computing	GPU workload protection, secure memory, and protected execution paths	Enterprise GPU offerings for AI/ML workloads
AMD SEV	VM memory encryption, nested page protection, and secure VM launch	Widely deployed in EPYC servers for cloud and enterprise data centers
IBM Secure Service Container	Encrypted containers, tamper protection, and automatic data encryption	IBM Z mainframes and LinuxONE systems for enterprise market
Apple Secure Enclave	Isolated security processor, encrypted memory, and secure boot chain	All modern Apple devices (iPhones, iPads, Macs)
Samsung Knox	Isolated workspace, secure boot, and real-time kernel protection	Samsung enterprise and consumer mobile devices
Cloud Service TEEs		
Google Confidential Computing	Isolated VMs, confidential containers, and AMD SEV-based protection	Google Cloud Platform commercial service
Microsoft Azure Confidential Computing	SGX VMs, confidential containers, and attestation services	Azure cloud service offerings
Alibaba Cloud Confidential Computing	SGX and SEV support, data isolation, and encryption in use	Alibaba Cloud commercial offering

architecture, instruction sets, and execution models (Volos et al., 2018). CPU TEEs (like Intel SGX, AMD SEV, and ARM TrustZone) evolved earlier and focus on general-purpose computing security, while GPU TEEs (such as NVIDIA's Confidential Computing) emerged more recently to address the specific needs of parallel computation in AI workloads. Despite this separate development, modern enterprise AI deployments increasingly demand seamless integration between these distinct security technologies to provide comprehensive protection across heterogeneous computing resources.

Modern enterprise AI security systems now use hybrid approaches that combine these complementary TEE technologies into unified security zones. These integrated systems create end-to-end secure computing paths where data remains protected throughout its lifecycle, from ingest and preprocessing (secured by CPU TEEs) through model training and inference (protected by GPU TEEs). This security coverage spans the entire AI pipeline, creating a continuous chain of trust across different computing resources (Tramer and Boneh, 2019). In typical production environments, Intel SGX or AMD SEV secures the data preparation and management stages, while NVIDIA's confidential computing technologies protect the compute-intensive model execution.

Recent benchmarks on NVIDIA H100 GPUs reveal that TEE security features add minimal performance overhead for AI workloads. Zhu et al. (2024) demonstrated that enabling TEE mode on H100 GPUs incurs less than 7% average performance penalty for LLM inference tasks, with this overhead approaching zero for larger models. Significantly, the performance impact stems primarily from encrypted data transfers between CPU and GPU TEEs across the PCIe bus, not from computations within the GPU itself. These findings are particularly relevant for enterprise AI deployments,

confirming that TEE protection becomes more efficient precisely for the large-scale, computation-intensive workloads typical in production environments.

The research community has also made significant progress in developing trusted execution environments for heterogeneous computing systems, with substantial efforts focusing on the integration of CPUs and GPUs to address their unique security challenges while minimizing performance overhead. Early prototypes like Graviton established secure channels between untrusted hosts and GPUs, while HIX advanced this approach by placing key GPU control functionalities within a CPU-based enclave (Wang and Oswald, 2024). HETEE proposed a data-center level GPU TEE supporting confidential computing without requiring chip-level modifications. ARM-based TEE research has evolved with StrongBox, the first ARM-based GPU TEE for secure computation without hardware modifications, and CRONUS, which developed a framework supporting spatial sharing within accelerators (Wang and Oswald, 2024). Most recently, PORTAL (Sang et al., 2025) addresses the challenges of integrating diverse co-processors and peripherals within mobile ARM SoCs by achieving secure device I/O through strict memory isolation without memory encryption, specifically targeting performance, scalability, and power efficiency challenges that have hindered wider adoption of confidential computing architectures.

3.3.3 Security Vulnerabilities and Mitigation Strategies

Vulnerabilities While integrated TEE technologies offer promising security benefits across CPU-GPU environments, they remain vulnerable to a range of sophisticated attacks that must be addressed to ensure truly secure heterogeneous computing. Side-channel attacks represent a critical threat to heterogeneous TEEs, especially in CPU-GPU systems. Traditional TEE protections focus primarily on preventing direct memory access from untrusted software but often fail to address the subtle information leakage that occurs through execution patterns, memory access timing, and resource contention. TEEs offer inadequate protection against sophisticated side-channel attacks exploiting cache timing or page faults, as demonstrated by successful attacks against Intel SGX enclaves (Xiao et al., 2017). Similarly, speculative execution attacks like Spectre and Meltdown, as well as physical attacks including cold boot and power analysis techniques, continue to pose significant threats to TEE security guarantees (Van Schaik et al., 2024).

These side-channels become even more problematic in heterogeneous computing environments where the architectural complexity introduces additional attack surfaces. Researchers have demonstrated multiple attack vectors unique to GPU environments, including microarchitectural attacks exploiting shared resources (such as L2 caches across GPUs connected via NVLink), timing attacks based on GPU-specific execution patterns, and physical access vulnerabilities targeting PCIe traffic. Academic studies (Wang and Oswald, 2024; Dutta et al., 2023) have highlighted specific vulnerabilities in various GPU architectures, showing that even with TEE protection, challenges remain in securing the entire computational pipeline. The complexity of CPU-GPU interactions creates unique side-channel vulnerabilities

that are difficult to address through hardware isolation alone, as TEEs alone cannot prevent side-channel leakage through memory access patterns.

Defenses Frameworks like STACCO provide sophisticated differential analysis techniques to identify subtle side-channel vulnerabilities in TEE implementations (Xiao et al., 2017). Ohrimenko et al. (2016) demonstrated how CPU TEEs can be leveraged constructively by implementing data-oblivious machine learning algorithms within Intel SGX enclaves. By ensuring "uniform" memory access patterns regardless of input data, these techniques effectively shield sensitive information from memory side-channel observation. This approach has been further advanced by TLBlur (Vanoverloop et al., 2025), which automatically tracks and prefetches page accesses to conceal execution patterns from privileged adversaries. The complementary nature of TEEs and oblivious algorithms represents a crucial research direction for privacy-preserving AI applications, combining hardware isolation against direct attacks with algorithmic protections that can eliminate the subtle information leakage traditionally undermining TEE security guarantees. By combining hardware TEE protections with algorithmic techniques, the system can provide strong security guarantees: hardware isolation to prevent direct access and algorithmic solutions to mitigate some of the indirect observations through side-channels. These findings inform ongoing research in hardening TEE implementations to address attack surfaces through techniques like resource partitioning, cache isolation, and encrypted memory systems.

3.3.4 Summary and Enterprise Outlook

Trusted Execution Environments provide hardware-enforced isolation for sensitive AI workloads, with modern implementations now effectively spanning heterogeneous CPU-GPU architectures. Despite significant advances in integration that minimize performance penalties for large models (less than 7% for LLMs on NVIDIA H100 GPUs), there are several aspects that enterprises may consider before deploying these technologies. For example, for effective TEE implementation, AI teams should identify critical pipeline components requiring protection, establish continuous attestation chains across processing stages, and develop clear governance frameworks for confidential computing deployments.

In parallel with implementation advancements, academic-industry collaborations continue to standardize secure communication between heterogeneous TEEs and unify attestation protocols (Confidential Computing Consortium, 2023). These standardization efforts complement technical implementations (Volos et al., 2018; Tramer and Boneh, 2019) and enhance the viability of secure AI deployments across diverse hardware environments. However, even with standardized protocols, security remains contextual to specific deployments. When deploying TEE technologies, organizations should conduct thorough threat modeling end-to-end, including the supplier of the TEEs and the environment where TEEs are deployed. For example, if memory side-channels are a threat, then susceptibility of the code to information

leakage should be evaluated and methods for hardening AI workloads should be considered.

3.4 Federated Learning

The principle of data minimization is fundamental to privacy preservation, but it presents a significant challenge for traditional machine learning approaches that rely on centralized data collection. Federated learning (FL) addresses this challenge by enabling model training across multiple clients without sharing raw data. In FL, clients collaboratively train a model by exchanging model weights or gradients, either among themselves or with a centralized server.

However, as discussed in Sections 2.2 and 2.5, these model updates are not intrinsically privacy-preserving and can potentially leak information about the underlying training data. While basic FL schemes remain vulnerable to such leakage, it is possible to mitigate these risks by integrating additional privacy-enhancing technologies, such as secure multiparty computation, secure enclaves, and differential privacy covered in previous sections.

This section will examine the key principles of federated learning, analyze its privacy benefits and challenges, and explore how it can be combined with other PETs to enhance privacy. We will also discuss practical considerations for implementing FL in enterprise settings.

3.4.1 Federated Learning Deployment Modes

Federated learning can be deployed in two primary modes, each with distinct privacy implications:

Centralized FL A central server orchestrates the learning process, with clients downloading the current model, training locally, and sending updates back for aggregation. While straightforward to implement and manage, this mode introduces a single point of failure and may raise trust issues regarding the central server, potentially allowing for inference attacks (Nasr et al., 2019).

Decentralized FL Clients communicate directly with each other, sharing updates in a peer-to-peer fashion using technologies such as gossip protocols (Tang et al., 2023) or blockchain (Bao et al., 2019). This mode is more resilient to failures and can offer stronger privacy guarantees by eliminating a central point of visibility. However, it's more complex to implement and may be more vulnerable to inference attacks from malicious clients (Lyu et al., 2020).

The choice of deployment mode depends on factors including privacy requirements, network topology, and computational resources. For example, a consortium of banks might prefer decentralized FL for a fraud detection model to ensure no single entity has complete visibility of all updates, while a company improving

product recommendations across branches might choose centralized FL for easier management, with appropriate security measures at the central server.

3.4.2 Federated Averaging: A Canonical FL Algorithm

To grasp both the privacy benefits and challenges of federated learning, it's helpful to understand the process. Federated Averaging (FedAvg) (McMahan et al., 2017), which extends stochastic gradient descent to the federated setting, is one of the most widely used algorithms for centralized FL and has served as a foundation for more advanced algorithms.

The FedAvg process, as detailed in Algorithm 2, proceeds as follows:

1. **Initialization:** A central server chooses a model architecture and initializes global weights w_0 (line 2).
2. **Client Selection:** For each round, the server selects a subset of clients to participate (line 4).
3. **Local Training:** Selected clients receive the current global model weights and perform local training (line 6). This step is crucial for privacy preservation, as all data handling occurs locally.
4. **Aggregation:** The server aggregates the updates by averaging model parameters from all participating clients (line 8).
5. **Iteration:** Steps 2-4 are repeated for multiple rounds until the model converges or a predefined number of rounds is reached.

While clients never share raw data, which is a key privacy-preserving aspect of FL, these model updates can still potentially leak information, necessitating additional privacy measures discussed next.

3.4.3 Additional Privacy Measures

While federated learning offers inherent privacy advantages by keeping raw data localized, it's not immune to privacy vulnerabilities. As discussed in Sections 2.5 and 2.2, gradients or model weights can leak information about the training data. This risk exists in both centralized and decentralized FL settings, and even the final global model can pose privacy risks similar to those in standard centralized learning.

To address these vulnerabilities, FL is often combined with other privacy-enhancing technologies:

Differential Privacy (DP) Both local and central DP can be applied to FL. With local DP, clients add noise to their weight updates before sending them to untrusted parties (Truex et al., 2020), however it typically results in poor utility for a reasonable level of privacy (Kim et al., 2021). A more practical approach is central DP, which assumes a trusted central server while preventing privacy leakage in the final model. For example, DP-FedAvg (Geyer et al., 2018) adapts the FedAvg algorithm

Algorithm 2 Federated Averaging

procedure FedAvg ▷ Run on server
  Initialize global weights w_0
  for each round $t = 1, 2, \ldots$ **do**
    Select a subset S_t of K clients
    for client $k \in S_t$ **in parallel do**
      $w_t^k \leftarrow \text{ClientUpdate}(k, w_{t-1})$
    end for
    Global weight update: $w_t \leftarrow \frac{1}{K} \sum_{k \in S_t} w_t^k$
  end for
  return Global weights w_t
end procedure

procedure ClientUpdate(k, w) ▷ Run on client k
  Randomly partition local data into batches $\mathcal{B}$
  for epoch $i = 1, \ldots, E$ **do**
    for batch $x \in \mathcal{B}$ **do**
      Local weight update: $w \leftarrow w - \eta \nabla L(w; x)$
    end for
  end for
  return Local weights w to server
end procedure

(Algorithm 2) to limit an attacker's ability to determine whether a specific client participated in the training process. This is distinct from protecting individual data points and works by sub-sampling clients and adding noise during the aggregation step, effectively hiding a single client's contribution.

Secure Multiparty Computation (SMPC) SMPC techniques, particularly secure aggregation and homomorphic encryption, can be integrated into FL to enhance privacy during the training process. Secure aggregation protocols efficiently protect individual updates during aggregation, but leave the aggregated update exposed to the central server and require synchronous training (Bonawitz et al., 2017). Homomorphic encryption offers stronger protection by securing both individual and aggregated updates, but at a higher computational cost. These approaches have different threat model assumptions: secure aggregation typically assumes an honest-but-curious central server, while homomorphic encryption provides stronger guarantees against a malicious server.

Trusted Execution Environments (TEEs) TEEs offer hardware-based isolation for sensitive computations in FL, primarily used on the server side to protect the aggregation process (Mo et al., 2021). In a typical setup, clients encrypt model updates before transmission, and the server's TEE decrypts and aggregates the updates in a protected environment. While TEEs offer strong privacy guarantees, they assume trustworthiness of the hardware manufacturer and TEE implementation, and can be vulnerable to side-channel attacks (Bulck et al., 2018).

By combining these technologies, FL can provide robust privacy guarantees against a wide range of threat models. However, each technology comes with its own

trade-offs in terms of computational overhead, communication costs, and potential impact on model utility.

3.4.4 Summary and Enterprise Outlook

Federated learning represents a significant advancement in privacy-preserving AI, allowing enterprises to collaborate on model training while respecting data privacy and ownership. By keeping raw data localized and sharing only model updates, FL provides a foundation for overcoming data silos and enabling collaboration in scenarios where direct data sharing is infeasible. The integration of additional privacy-enhancing technologies, such as secure aggregation protocols and Trusted Execution Environments (TEEs), has largely addressed privacy concerns in the FL training process.

Looking forward, enterprises considering FL implementation should:

- Assess their specific use case and privacy requirements to determine the most appropriate FL deployment mode and additional privacy measures needed.
- Invest in building technical expertise or partnering with specialized providers to navigate the complexity of implementing secure FL systems.
- Stay informed about advancements in differentially private machine learning, as these will be crucial for protecting the privacy of final aggregated models, especially if they are to be released or shared.

As privacy-preserving FL becomes more accessible, enterprises that proactively develop capabilities in this area will be well-positioned to leverage collaborative learning opportunities while maintaining robust privacy protections. This could lead to competitive advantages through improved models and expanded collaboration possibilities, all while adhering to stringent privacy standards.

3.5 Machine Unlearning

The "right to erasure" enshrined in privacy regulations such as the European Union's General Data Protection Regulation (GDPR) has sparked growing interest in data deletion practices. While these regulations don't explicitly extend to trained models, research demonstrating that models can memorize training data (Carlini et al., 2019) has raised privacy concerns and potential regulatory implications. In response, the field of machine unlearning has emerged, aiming to develop methods to remove the influence of specific data points from trained models efficiently, without the need for complete retraining.

This section explores the landscape of machine unlearning techniques, categorizing them into certified and heuristic approaches. We discuss their privacy implications and challenges, and provide an outlook for this emerging field with

recommendations for organizations considering unlearning as a privacy-enhancing technology.

3.5.1 Unlearning Approaches

Machine unlearning approaches can be broadly categorized into certified and heuristic methods.

Certified Approaches Certified unlearning approaches provide guarantees about the extent to which data has been "forgotten". Most operate by performing an approximate update to the model (such as a Newton or gradient step) followed by the addition of noise (Guo et al., 2020; Neel et al., 2021). This can produce a model statistically indistinguishable from one retrained without the data to forget. The notion of statistical indistinguishability can be formalized in various ways. One common approach due to Guo et al. (2020), is inspired by (ϵ, δ)-differential privacy. It provides more effective unlearning guarantees as ϵ and δ approach zero.

Definition 2 Let A be a training algorithm that produces a model $\theta = A(D)$ given training data D. An unlearning algorithm U is an (ϵ, δ)-unlearner for A if, for any subset $Z \subset D$ to be forgotten, the distributions of $A(D \setminus Z)$ and $U(A(D), D, Z)$ are (ϵ, δ)-indistinguishable.

While offering formal guarantees, certified methods have limitations. Many rely on assumptions such as strong convexity, which are not satisfied by popular models like neural networks (Guo et al., 2020; Neel et al., 2021), although researchers are trying to lessen these assumptions (Mu and Klabjan, 2024; Zhang et al., 2024a). Moreover, they often struggle with multiple successive erasure requests and can negatively impact model performance due to noise addition.

Notably, there is a strong connection between certified unlearning and differential privacy, with the latter providing strong unlearning guarantees even for adaptive deletion requests that depend on previously published models (Gupta et al., 2021).

Heuristic Approaches Heuristic approaches approximate the effect of retraining without erased data, often using techniques such as influence functions (Koh and Liang, 2017) or fine-tuning on the retained data. While more practical for complex models, they lack formal guarantees. Popular heuristic methods include Amnesiac (Graves et al., 2021), which selectively reverses parameter updates from batches containing data to be erased; SISA (Bourtoule et al., 2021), which trains multiple models on disjoint data subsets so that only some of the models need to be retrained; and SCRUB (Kurmanji et al., 2023), which uses a teacher-student formulation to unlearn.

Researchers typically evaluate these methods empirically, using metrics such as the similarity to a fully retrained model or resistance to membership inference attacks on erased data. However, the lack of formal guarantees means that these methods may not fully remove all traces of erased data from the model.

3.5.2 Privacy Considerations

While unlearning methods aim to enhance privacy by removing the influence of specific data points, they also introduce new privacy considerations and potential vulnerabilities.

Longitudinal Privacy Risks The release of multiple versions of unlearned models over time introduces significant privacy challenges. Chen et al. (2021) demonstrated that differencing attacks can potentially reconstruct erased data by comparing successive model versions. This vulnerability is particularly concerning when successive unlearned models are released as data removal requests are processed. To mitigate these risks, organizations may need to limit the frequency of model releases, or employ techniques that add noise to model updates to obscure the effects of individual unlearning operations.

Interaction with other PETs Unlearning methods do not exist in isolation but rather complement and interact with other privacy-enhancing technologies (PETs). For instance, combining unlearning with differential privacy (DP) can provide stronger privacy guarantees (Gupta et al., 2021; Huang and Canonne, 2023). DP can be applied to the unlearning process itself, ensuring that the act of removing data doesn't leak information about the removed data. Unlearning can also be integrated into federated learning systems for local data removal without requiring access to the central model (Liu et al., 2024b). However, these combinations may introduce new challenges, such as increased computational complexity or potential degradation in model utility, requiring careful balancing of privacy and performance objectives.

Adversarial Unlearning Requests The ability to request data removal opens up new avenues for attacks. Malicious actors could potentially exploit unlearning mechanisms to degrade model performance by strategically requesting the removal of critical training examples (Huang et al., 2025) or use carefully crafted sequences of unlearning requests to extract information about the training data or the model. Defending against such attacks requires robust verification mechanisms to ensure the legitimacy of unlearning requests, as well as techniques to maintain model integrity. Organizations must consider not only the privacy implications for data subjects but also the security of the machine learning system as a whole when implementing unlearning.

3.5.3 Summary and Enterprise Outlook

While machine unlearning shows promise in addressing privacy concerns and potential future regulatory requirements, it remains primarily a research topic rather than a technology ready for deployment. As we've demonstrated, current approaches face significant challenges, including limited scalability, potential impacts on model performance, and vulnerability to longitudinal privacy risks.

Although unlearning is not yet ready for enterprise deployment, forward-thinking organizations could:

- Monitor developments in unlearning research, particularly advancements in efficiency, scalability, and privacy guarantees.
- Assess the potential impact of unlearning on existing machine learning pipelines and data management practices.
- Engage with legal and compliance teams to evaluate how unlearning might fit into broader data protection strategies.

As the field matures, unlearning may become an essential tool in the privacy-preserving machine learning toolkit. Organizations that proactively engage with unlearning research will be well positioned to leverage its potential benefits.

4 Privacy in the Era of Generative AI

The rise of large foundation models and generative AI has fundamentally altered the privacy landscape. While many traditional privacy risks and protection mechanisms still apply, the unique characteristics of these models introduce novel challenges and exacerbate existing ones. This section explores how the privacy landscape has shifted in the era of generative AI, focusing on new and amplified risks, the limitations of existing privacy-enhancing technologies (PETs), and emerging approaches to privacy protection.

4.1 Privacy Risks in the Generative AI Era

Generative AI models, particularly large language models (LLMs), have not only amplified existing privacy risks but also introduced new concerns. One of the most significant issues is the unprecedented capacity for memorization exhibited by these models. As they scale up, LLMs have been shown to memorize and potentially leak more information from their training data, amplifying the risk of data extraction attacks (Carlini et al., 2021). Interestingly, recent research has found that LLMs exhibit higher memorization of data presented earlier and later in training, suggesting that placing sensitive data in the middle of training might be beneficial for privacy protection (Leybzon and Kervadec, 2024).

The outputs of generative models present another challenge, as they are typically much more information-dense than those of traditional classification and regression models. This increased information density results in a higher risk of information leakage, not just from training data but also from sensitive information in input prompts (Wu et al., 2024b). The risk is further compounded by the advent of multimodal models like CLIP (Radford et al., 2021), which introduce new privacy vulnerabilities from the interaction between different data modalities. These models are

susceptible to membership inference attacks that exploit the relationship between text and image data, potentially revealing more information than single-modality models (Ko et al., 2023).

LLMs also face unique security challenges that have privacy implications. They are vulnerable to "jailbreaking" attacks, where carefully crafted prompts cause the model to bypass its safety constraints. Such attacks have been shown to successfully extract private information from commercial models such as ChatGPT (Li et al., 2023). Moreover, the reasoning capabilities of LLMs enable them to perform linkage attacks given access to external data provided in prompts or via retrieval-augmented generation (OpenAI, 2024). This is particularly concerning as it allows malicious actors to automate and scale attacks that previously may have been limited by human labor constraints.

Lastly, the massive datasets required to train foundation models often involve complex, opaque data supply chains. This opacity introduces new privacy risks related to unauthorized data use, lack of consent, and potential violations of data protection regulations. Zhang et al. (2024b) argue that consent management is needed throughout the data lifecycle, and training information and metadata must be preserved to enable side-effect mitigation, such as through unlearning techniques.

4.2 Limitations of Existing PETs in the Generative AI Context

Traditional privacy-enhancing technologies (PETs) face new challenges when applied to large foundation models and generative AI.

Differential privacy (DP), a cornerstone of privacy protection, encounters significant hurdles in this new context. While DP remains a powerful tool, its application to generative models can result in substantial utility loss (Yu et al., 2022). The scale of data required for pre-training makes DP seem infeasible at that stage, although it has shown promise in fine-tuning scenarios (Yu et al., 2021). In the domain of computer vision, applying DP to the training of diffusion models typically leads to divergence for small-scale datasets, while its feasibility for larger datasets remains an open question (Chen et al., 2024).

Federated learning (FL), another popular PET, also faces challenges in the context of generative AI. While FL can help protect raw data by keeping it decentralized, the aggregated updates in FL settings can still leak significant amounts of private information, particularly in the context of language models (Fowl et al., 2023). This leakage undermines the privacy guarantees that FL aims to provide, necessitating additional protective measures.

Secure computation techniques, such as secure multiparty computation (SMPC) and trusted execution environments (TEEs), encounter scalability issues when applied to foundation models. The immense computational requirements of these models make traditional SMPC and CPU-based TEEs impractical. However, recent advancements in hardware, such as Nvidia's Hopper and Blackwell GPUs with built-in TEEs, offer promising solutions. Zhu et al. (2024) found that enabling TEEs on

Nvidia Hopper GPUs for large-scale LLM inference incurs an overhead of less than 7%, while Nvidia claims near-identical performance for their Blackwell architecture.

4.3 Emerging Approaches and Future Directions

To address the new and amplified risks posed by generative AI, researchers are exploring novel approaches to privacy protection. One such approach involves using LLMs themselves to remove sensitive information from training data, a process known as scrubbing. However, current commercial models like GPT-4 have shown limitations in this task, failing to detect all pieces of personal information in benchmark datasets (Bubeck et al., 2023; Mireshghallah et al., 2024). This suggests that while AI-powered scrubbing holds promise, it requires further refinement to match or exceed human performance in identifying and protecting sensitive information.

Alignment and self-censoring techniques represent another avenue for privacy protection in generative AI. LLMs can be trained to reject requests that could violate privacy, such as performing linkage attacks (OpenAI, 2024). In the image domain, researchers have proposed methods to steer image generation in diffusion models away from memorized data (Chen et al., 2024). However, the effectiveness of these alignment efforts is challenged by automated red-teaming techniques that can generate adversarial prompts to circumvent such protections (Nie et al., 2024).

Privacy-preserving fine-tuning has emerged as a promising direction, particularly given the challenges of applying differential privacy to pre-training. Several studies have demonstrated the feasibility of fine-tuning LLMs using DP analogues of standard training algorithms, achieving reasonable privacy-utility trade-offs (Yu et al., 2021; Li et al., 2022; Wu et al., 2024a). While these approaches protect against privacy leakage in the fine-tuned model parameters, they may still expose training data to human annotators. To address this issue, Yu et al. (2024) propose replacing user prompts with synthetic prompts generated by a privately fine-tuned model, further enhancing privacy protection throughout the training process.

In-context learning presents an alternative paradigm that allows off-the-shelf LLMs to be used without fine-tuning by providing training examples in the prompt. However, this approach introduces its own privacy challenges, as sensitive information in the training examples could be leaked in the LLM's response, particularly in adversarial settings. To mitigate this risk, researchers have applied differential privacy through randomized aggregation of LLM responses using disjoint exemplar sets (Wu et al., 2024b; Duan et al., 2023).

In conclusion, while generative AI presents significant privacy challenges, it also offers opportunities for innovative solutions. As the field evolves, developing privacy-preserving techniques that can keep pace with the rapid advancements in model capabilities and applications will be crucial. The interplay between new risks and emerging protective measures underscores the dynamic nature of privacy in the AI era, emphasizing the need for continued research and adaptation of privacy-enhancing technologies.

5 Conclusion and Future Outlook

This chapter has explored the complex landscape of privacy in enterprise AI, examining key privacy risks, mitigation strategies, and the evolving challenges posed by generative AI. As we conclude, it is crucial to reflect on the main insights and consider the path forward for organizations navigating this dynamic field.

5.1 Key Takeaways

Our exploration of privacy risks and threat models underscores the importance of adopting a threat-specific approach to privacy. By viewing privacy through the lens of specific threats, organizations can develop targeted and effective mitigation strategies, rather than implementing generic solutions that may leave critical vulnerabilities unaddressed. This approach is particularly crucial given the fast-paced nature of AI development, where new privacy risks can emerge rapidly.

The diverse toolkit of privacy-enhancing technologies (PETs) available for AI applications offers promising solutions, but also presents complex trade-offs. Organizations must carefully balance privacy protections with model utility, computational demands, and system complexity. The intricacies of these trade-offs highlight the need for specialized expertise in privacy engineering, rather than treating privacy as an afterthought in AI development.

Our examination of privacy in the era of generative AI demonstrates how recent advancements have significantly altered the privacy landscape. These powerful models have not only exacerbated existing privacy risks but also introduced novel challenges. While research in this area is progressing rapidly, there remains a substantial gap between current capabilities and robust privacy-preserving generative AI systems. This gap underscores the need for ongoing vigilance and adaptation in privacy strategies.

The dynamic nature of AI privacy means that what constitutes state-of-the-art today may be outdated within months. This rapid change presents both opportunities and challenges for enterprises. While new techniques and tools are constantly emerging to enhance privacy protections, the pace of change makes it challenging to establish stable, long-term privacy strategies.

Perhaps the most critical takeaway is the need for organizations to remain adaptable and committed to continuous learning. Enterprises must regularly reassess their privacy measures, staying informed about emerging threats, novel protection mechanisms, and evolving best practices. This ongoing process of learning and adaptation is essential for navigating the complex and ever-changing intersection of AI and privacy.

5.2 Future Outlook for Privacy and AI in the Enterprise

Looking ahead, several key areas will likely shape the future of privacy in enterprise AI. The balance between AI innovation and privacy will become increasingly critical as AI capabilities expand. Organizations will face growing pressure to innovate their AI capabilities while maintaining robust privacy protections, requiring careful consideration of privacy implications at every stage of AI development and deployment.

Data governance for AI systems will need to evolve to match the growing complexity of AI models, particularly in the realm of generative AI. Organizations will need to develop comprehensive strategies for managing data throughout its lifecycle, from collection and processing to model training and output generation.

Regulatory compliance will present ongoing challenges as privacy regulations evolve and potentially diverge across different jurisdictions. Enterprises may need to adapt their AI systems to comply with a patchwork of requirements, potentially leading to the development of more flexible, privacy-preserving AI architectures that can be easily adjusted to meet varying regulatory standards.

Emerging technologies will continue to introduce new privacy challenges, likely outpacing current privacy protections. Enterprises must remain proactive in identifying and mitigating these emerging risks to stay ahead of potential vulnerabilities.

5.3 The Role of Enterprises and Call to Action

Enterprises play a crucial role in shaping the future of AI privacy. By prioritizing privacy in their AI initiatives, organizations can not only protect stakeholders but also contribute to the development of more responsible AI practices industry-wide.

We urge readers to:

- Invest in privacy expertise and integrate privacy considerations into all stages of AI development and deployment.
- Stay informed about the latest developments in AI privacy research and emerging technologies.
- Contribute to the broader conversation on AI privacy by sharing experiences and best practices with the community.
- Advocate for privacy-preserving AI within their organizations and industries, aligned with articulated, specific, threat models.

In conclusion, while the challenges of maintaining privacy in AI systems are significant, they are not insurmountable. By approaching these challenges with a combination of technical rigor, ethical consideration, and grounding in legal regulation and governance, enterprises can harness the power of AI while respecting and protecting individual privacy. As the field continues to evolve, maintaining this balance will be crucial for the responsible and sustainable development of AI technologies.

Acknowledgements N. Marchant, B. I. P. Rubinstein and Y. Zhao acknowledge support from Australian Research Council Discovery Project DP220102269.

Competing Interests The authors have no conflicts of interest to declare that are relevant to the content of this chapter.

References

Abadi M, Chu A, Goodfellow I, McMahan HB, Mironov I, Talwar K, Zhang L (2016) Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, CCS '16, pp 308–318

Balunovic M, Dimitrov D, Jovanović N, Vechev M (2022) LAMP: Extracting text from gradients with language model priors. In: Advances in Neural Information Processing Systems, vol 35, pp 7641–7654

Bao X, Su C, Xiong Y, Huang W, Hu Y (2019) FLChain: A blockchain for auditable federated learning with trust and incentive. In: 2019 5th International Conference on Big Data Computing and Communications (BIGCOM), pp 151–159

Bawack RE, Wamba SF, Carillo KDA, Akter S (2022) Artificial intelligence in E-commerce: a bibliometric study and literature review. Electronic Markets 32(1):297–338

Benaissa A, Retiat B, Cebere B, Belfedhal AE (2021) TenSEAL: A library for encrypted tensor operations using homomorphic encryption. arXiv:2104.03152 [cs.CR]

Bergerat L, Boudi A, Bourgerie Q, Chillotti I, Ligier D, Orfila JB, Tap S (2023) Parameter optimization and larger precision for (T)FHE. Journal of Cryptology 36(3):28

Bertran MA, Tang S, Kearns M, Morgenstern JH, Roth A, Wu S (2024) Reconstruction attacks on machine unlearning: Simple models are vulnerable. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems

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

Bonawitz K, Ivanov V, Kreuter B, Marcedone A, McMahan HB, Patel S, Ramage D, Segal A, Seth K (2017) Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS '17, pp 1175–1191

Bourtoule L, Chandrasekaran V, Choquette-Choo CA, Jia H, Travers A, Zhang B, Lie D, Papernot N (2021) Machine unlearning. In: 2021 IEEE Symposium on Security and Privacy (SP), pp 141–159

Bubeck S, Chandrasekaran V, Eldan R, Gehrke J, Horvitz E, Kamar E, Lee P, Lee YT, Li Y, Lundberg S, Nori H, Palangi H, Ribeiro MT, Zhang Y (2023) Sparks of artificial general intelligence: Early experiments with GPT-4. arXiv:2303.12712 [cs.CL]

Bukaty P (2019) The California Consumer Privacy Act (CCPA): An implementation guide. IT Governance Publishing

Bulck JV, Minkin M, Weisse O, Genkin D, Kasikci B, Piessens F, Silberstein M, Wenisch TF, Yarom Y, Strackx R (2018) Foreshadow: Extracting the keys to the intel SGX kingdom with transient Out-of-Order execution. In: 27th USENIX Security Symposium (USENIX Security 18), p 991–1008

Bun M, Steinke T (2016) Concentrated differential privacy: Simplifications, extensions, and lower bounds. In: Proceedings, Part I, of the 14th International Conference on Theory of Cryptography - Volume 9985, Springer-Verlag, pp 635–658

Carlini N, Liu C, Erlingsson Ú, Kos J, Song D (2019) The Secret Sharer: Evaluating and testing unintended memorization in neural networks. In: 28th USENIX Security Symposium (USENIX Security 19), pp 267–284

Carlini N, Tramèr F, Wallace E, Jagielski M, Herbert-Voss A, Lee K, Roberts A, Brown T, Song D, Erlingsson Ú, Oprea A, Raffel C (2021) Extracting training data from large language models. In: 30th USENIX Security Symposium (USENIX Security 21), pp 2633–2650

Carlini N, Chien S, Nasr M, Song S, Terzis A, Tramèr F (2022) Membership inference attacks from first principles. In: 2022 IEEE Symposium on Security and Privacy (SP), pp 1897–1914

Carlini N, Hayes J, Nasr M, Jagielski M, Sehwag V, Tramèr F, Balle B, Ippolito D, Wallace E (2023a) Extracting training data from diffusion models. In: 32nd USENIX Security Symposium (USENIX Security 23), pp 5253–5270

Carlini N, Ippolito D, Jagielski M, Lee K, Tramer F, Zhang C (2023b) Quantifying memorization across neural language models. In: The Eleventh International Conference on Learning Representations

Carlini N, Paleka D, Dvijotham KD, Steinke T, Hayase J, Cooper AF, Lee K, Jagielski M, Nasr M, Conmy A, Wallace E, Rolnick D, Tramèr F (2024) Stealing part of a production language model. In: Proceedings of the 41st International Conference on Machine Learning, vol 235, pp 5680–5705

Chandrasekaran V, Chaudhuri K, Giacomelli I, Jha S, Yan S (2020) Exploring connections between active learning and model extraction. In: 29th USENIX Security Symposium (USENIX Security 20), pp 1309–1326

Chen C, Liu D, Xu C (2024) Towards memorization-free diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 8425–8434

Chen M, Zhang Z, Wang T, Backes M, Humbert M, Zhang Y (2021) When machine unlearning jeopardizes privacy. In: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, CCS '21, pp 896–911

Cheon JH, Kim A, Kim M, Song Y (2017) Homomorphic encryption for arithmetic of approximate numbers. In: Advances in Cryptology – ASIACRYPT 2017, Springer International Publishing, pp 409–437

Choquette-Choo CA, Tramer F, Carlini N, Papernot N (2021) Label-only membership inference attacks. In: Proceedings of the 38th International Conference on Machine Learning, vol 139, pp 1964–1974

Chu HM, Geiping J, Fowl LH, Goldblum M, Goldstein T (2023) Panning for gold in federated learning: Targeted text extraction under arbitrarily large-scale aggregation. In: The Eleventh International Conference on Learning Representations

Confidential Computing Consortium (2023) Confidential computing: Hardware-based trusted execution for applications and data. Tech. rep., Confidential Computing Consortium, URL https://confidentialcomputing.io/wp-content/uploads/sites/10/2023/03/CCC_outreach_whitepaper_updated_November_2022.pdf, accessed: 2025-05-14

Dathathri R, Kostova B, Saarikivi O, Dai W, Laine K, Musuvathi M (2020) Eva: an encrypted vector arithmetic language and compiler for efficient homomorphic computation. In: Proceedings of the 41st ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI 2020, p 546–561

Dimitrov DI, Baader M, Müller MN, Vechev M (2024) Spear: Exact gradient inversion of batches in federated learning. In: Advances in Neural Information Processing Systems, vol 37, pp 106768–106799

Dong J, Roth A, Su WJ (2022) Gaussian differential privacy. Journal of the Royal Statistical Society Series B: Statistical Methodology 84(1):3–37

Duan H, Dziedzic A, Papernot N, Boenisch F (2023) Flocks of stochastic parrots: Differentially private prompt learning for large language models. In: Advances in Neural Information Processing Systems, vol 36, pp 76852–76871

Duchi JC, Jordan MI, Wainwright MJ (2013) Local privacy and statistical minimax rates. In: 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, pp 429–438

Dutta SB, Naghibijouybari H, Gupta A, Abu-Ghazaleh N, Marquez A, Barker K (2023) Spy in the GPU-box: Covert and side channel attacks on multi-GPU systems. In: Proceedings of the 50th Annual International Symposium on Computer Architecture, ISCA '23

Dwork C, Roth A (2014) The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science 9(3–4):211–407, DOI 10.1561/0400000042

Dwork C, Rothblum GN, Vadhan S (2010) Boosting and differential privacy. In: 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, pp 51–60

Feng R, Hooda A, Mangaokar N, Fawaz K, Jha S, Prakash A (2023) Stateful defenses for machine learning models are not yet secure against black-box attacks. In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, CCS '23, pp 786–800

Fowl LH, Geiping J, Czaja W, Goldblum M, Goldstein T (2022) Robbing the Fed: Directly obtaining private data in federated learning with modified models. In: International Conference on Learning Representations

Fowl LH, Geiping J, Reich S, Wen Y, Czaja W, Goldblum M, Goldstein T (2023) Decepticons: Corrupted transformers breach privacy in federated learning for language models. In: The Eleventh International Conference on Learning Representations

Gao J, Garg S, Mahmoody M, Vasudevan PN (2022) Deletion inference, reconstruction, and compliance in machine (un)learning. In: Proceedings on Privacy Enhancing Technologies Symposium (PoPETs), vol 2022, pp 415–436

Geiping J, Bauermeister H, Dröge H, Moeller M (2020) Inverting gradients – how easy is it to break privacy in federated learning? In: Advances in Neural Information Processing Systems, vol 33, pp 16937–16947

Geyer RC, Klein T, Nabi M (2018) Differentially private federated learning: A client level perspective. arXiv:1712.07557 [cs.CR]

Graves L, Nagisetty V, Ganesh V (2021) Amnesiac machine learning. Proceedings of the AAAI Conference on Artificial Intelligence 35(13):11516–11524

Guo C, Goldstein T, Hannun A, Van Der Maaten L (2020) Certified data removal from machine learning models. In: Proceedings of the 37th International Conference on Machine Learning, vol 119, pp 3832–3842

Gupta K, Jawalkar N, Mukherjee A, Chandran N, Gupta D, Panwar A, Sharma R (2024) SIGMA: Secure GPT inference with function secret sharing. Proceedings on Privacy Enhancing Technologies 2024(4):61–79

Gupta V, Jung C, Neel S, Roth A, Sharifi-Malvajerdi S, Waites C (2021) Adaptive machine unlearning. In: Advances in Neural Information Processing Systems, vol 34, pp 16319–16330

Halevi S, Shoup V (2020) Design and implementation of HElib: a homomorphic encryption library. Cryptology ePrint Archive, Paper 2020/1481

Hu H, Salcic Z, Sun L, Dobbie G, Yu PS, Zhang X (2022) Membership inference attacks on machine learning: A survey. ACM Comput Surv 54(11s)

Hu H, Wang S, Dong T, Xue M (2024) Learn what you want to unlearn: Unlearning inversion attacks against machine unlearning. In: 2024 IEEE Symposium on Security and Privacy (SP), IEEE Computer Society, pp 3257–3275

Huang Y, Canonne CL (2023) Tight bounds for machine unlearning via differential privacy. arXiv:2309.00886 [cs.LG]

Huang Y, Liu D, Chua L, Ghazi B, Kamath P, Kumar R, Manurangsi P, Nasr M, Sinha A, Zhang C (2025) Unlearn and burn: Adversarial machine unlearning requests destroy model accuracy. In: The Thirteenth International Conference on Learning Representations

Huang Z, jie Lu W, Hong C, Ding J (2022) Cheetah: Lean and fast secure Two-Party deep neural network inference. In: 31st USENIX Security Symposium (USENIX Security 22), pp 809–826

Hui T, Farokhi F, Ohrimenko O (2023) Information leakage from data updates in machine learning models. In: Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security, AISec '23, pp 35–41

IT Governance Privacy Team (2020) EU General Data Protection Regulation (GDPR) – An implementation and compliance guide, 4th edn. IT Governance Publishing

Jagielski M, Carlini N, Berthelot D, Kurakin A, Papernot N (2020a) High accuracy and high fidelity extraction of neural networks. In: 29th USENIX Security Symposium (USENIX Security 20), pp 1345–1362

Jagielski M, Ullman J, Oprea A (2020b) Auditing differentially private machine learning: How private is private SGD? In: Advances in Neural Information Processing Systems, vol 33, pp 22205–22216

Jagielski M, Wu S, Oprea A, Ullman J, Geambasu R (2023) How to combine membership-inference attacks on multiple updated machine learning models. In: Proceedings on Privacy Enhancing Technologies Symposium (PoPETs), vol 2023, pp 211–232

Janssen M, Brous P, Estevez E, Barbosa LS, Janowski T (2020) Data governance: Organizing data for trustworthy artificial intelligence. Government Information Quarterly 37(3):101493

Jayashankar S, Chen E, Tang T, Zheng W, Skarlatos D (2025) Cinnamon: A framework for scale-out encrypted ai. In: Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1, ASPLOS '25, p 133–150

Jeon J, Kim J, Lee K, Oh S, Ok J (2021) Gradient inversion with generative image prior. In: Advances in Neural Information Processing Systems, vol 34, pp 29898–29908

Jia H, Choquette-Choo CA, Chandrasekaran V, Papernot N (2021a) Entangled watermarks as a defense against model extraction. In: 30th USENIX Security Symposium (USENIX Security 21), pp 1937–1954

Jia H, Yaghini M, Choquette-Choo CA, Dullerud N, Thudi A, Chandrasekaran V, Papernot N (2021b) Proof-of-learning: Definitions and practice. In: 2021 IEEE Symposium on Security and Privacy (SP), pp 1039–1056

Jia J, Salem A, Backes M, Zhang Y, Gong NZ (2019) MemGuard: Defending against black-box membership inference attacks via adversarial examples. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS '19, pp 259–274

Jin J, McMurtry E, Rubinstein BIP, Ohrimenko O (2022) Are we there yet? Timing and floating-point attacks on differential privacy systems. In: 2022 IEEE Symposium on Security and Privacy (SP), pp 473–488

Juuti M, Szyller S, Marchal S, Asokan N (2019) PRADA: Protecting against DNN model stealing attacks. In: 2019 IEEE European Symposium on Security and Privacy (EuroS&P), pp 512–527

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

Kairouz P, Mcmahan B, Song S, Thakkar O, Thakurta A, Xu Z (2021) Practical and private (deep) learning without sampling or shuffling. In: Proceedings of the 38th International Conference on Machine Learning, PMLR, Proceedings of Machine Learning Research, vol 139, pp 5213–5225

Kandpal N, Wallace E, Raffel C (2022) Deduplicating training data mitigates privacy risks in language models. In: Proceedings of the 39th International Conference on Machine Learning, vol 162, pp 10697–10707

Kaplan J, McCandlish S, Henighan T, Brown TB, Chess B, Child R, Gray S, Radford A, Wu J, Amodei D (2020) Scaling laws for neural language models. arXiv:2001.08361 [cs.LG]

Kariyappa S, Guo C, Maeng K, Xiong W, Suh GE, Qureshi MK, Lee HHS (2023) Cocktail party attack: Breaking aggregation-based privacy in federated learning using independent component analysis. In: Proceedings of the 40th International Conference on Machine Learning, vol 202, pp 15884–15899

Kei AYL, Chow SSM (2025) Shaft: Secure, handy, accurate and fast transformer inference. In: Proceedings 2025 Network and Distributed System Security Symposium, Internet Society, NDSS 2025

Kim M, Günlü O, Schaefer RF (2021) Federated learning with local differential privacy: Trade-offs between privacy, utility, and communication. In: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp 2650–2654

Knott B, Venkataraman S, Hannun A, Sengupta S, Ibrahim M, van der Maaten L (2021) CrypTen: Secure multi-party computation meets machine learning. In: Advances in Neural Information Processing Systems, vol 34, pp 4961–4973

Ko M, Jin M, Wang C, Jia R (2023) Practical membership inference attacks against large-scale multi-modal models: A pilot study. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp 4871–4881

Koh PW, Liang P (2017) Understanding black-box predictions via influence functions. In: Proceedings of the 34th International Conference on Machine Learning, vol 70, pp 1885–1894

Kurmanji M, Triantafillou P, Hayes J, Triantafillou E (2023) Towards unbounded machine unlearning. In: Advances in Neural Information Processing Systems, vol 36, pp 1957–1987

Leybzon DD, Kervadec C (2024) Learning, forgetting, remembering: Insights from tracking LLM memorization during training. In: Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP, Association for Computational Linguistics, pp 43–57

Li H, Guo D, Fan W, Xu M, Huang J, Meng F, Song Y (2023) Multi-step jailbreaking privacy attacks on ChatGPT. In: Findings of the Association for Computational Linguistics: EMNLP 2023, Association for Computational Linguistics, pp 4138–4153

Li J, Li N, Ribeiro B (2024) MIST: Defending against membership inference attacks through Membership-Invariant Subspace Training. In: 33rd USENIX Security Symposium (USENIX Security 24), pp 2387–2404

Li X, Tramer F, Liang P, Hashimoto T (2022) Large language models can be strong differentially private learners. In: International Conference on Learning Representations

Liu H, Wu Y, Yu Z, Zhang N (2024a) Please tell me more: Privacy impact of explainability through the lens of membership inference attack. In: 2024 IEEE Symposium on Security and Privacy (SP), pp 4791–4809

Liu Z, Jiang Y, Shen J, Peng M, Lam KY, Yuan X, Liu X (2024b) A survey on federated unlearning: Challenges, methods, and future directions. ACM Comput Surv 57(1)

Llama team (2024) The Llama 3 herd of models. arXiv:2407.21783 [cs.AI]

Lorè F, Basile P, Appice A, de Gemmis M, Malerba D, Semeraro G (2023) An AI framework to support decisions on GDPR compliance. J Intell Inf Syst 61(2):541–568

Lyu L, Yu H, Zhao J, Yang Q (2020) Threats to Federated Learning, Springer International Publishing, pp 3–16

Mahanti R (2021) Data Governance and Data Management: Contextualizing Data Governance Drivers, Technologies, and Tools. Springer Singapore

Maini P, Yaghini M, Papernot N (2021) Dataset inference: Ownership resolution in machine learning. In: International Conference on Learning Representations

McKeen F, Alexandrovich I, Berenzon A, Rozas CV, Shafi H, Shanbhogue V, Savagaonkar UR (2013) Innovative instructions and software model for isolated execution. In: Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy, HASP '13

McMahan B, Moore E, Ramage D, Hampson S, Arcas BAy (2017) Communication-Efficient Learning of Deep Networks from Decentralized Data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, vol 54, pp 1273–1282

Mireshghallah N, Kim H, Zhou X, Tsvetkov Y, Sap M, Shokri R, Choi Y (2024) Can LLMs keep a secret? Testing privacy implications of language models via contextual integrity theory. In: The Twelfth International Conference on Learning Representations

Mo F, Haddadi H, Katevas K, Marin E, Perino D, Kourtellis N (2021) PPFL: privacy-preserving federated learning with trusted execution environments. In: Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys '21, pp 94–108

Mo J, Gopinath J, Reagen B (2023) HAAC: A hardware-software co-design to accelerate garbled circuits. In: Proceedings of the 50th Annual International Symposium on Computer Architecture, ISCA 2023, Orlando, FL, USA, June 17-21, 2023, ACM, pp 10:1–10:13

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

More Y, Ganesh P, Farnadi G (2024) Towards more realistic extraction attacks: An adversarial perspective. arXiv:2407.02596 [cs.CR]

Mu S, Klabjan D (2024) Rewind-to-Delete: Certified machine unlearning for nonconvex functions. arXiv:2409.09778 [cs.LG]

Nanayakkara P, Bater J, He X, Hullman J, Rogers J (2022) Visualizing privacy-utility trade-offs in differentially private data releases. Proceedings on Privacy Enhancing Technologies 2022(2):601–618

Narayanan A, Shi E, Rubinstein BIP (2011) Link prediction by de-anonymization: How we won the Kaggle Social Network Challenge. In: The 2011 International Joint Conference on Neural Networks, pp 1825–1834

Nasr M, Shokri R, Houmansadr A (2018) Machine learning with membership privacy using adversarial regularization. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS '18, pp 634–646

Nasr M, Shokri R, Houmansadr A (2019) Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In: 2019 IEEE Symposium on Security and Privacy (SP), pp 739–753

Nasr M, Songi S, Thakurta A, Papernot N, Carlin N (2021) Adversary instantiation: Lower bounds for differentially private machine learning. In: 2021 IEEE Symposium on Security and Privacy (SP), pp 866–882

Nasr M, Rando J, Carlini N, Hayase J, Jagielski M, Cooper AF, Ippolito D, Choquette-Choo CA, Tramèr F, Lee K (2025) Scalable extraction of training data from aligned, production language models. In: The Thirteenth International Conference on Learning Representations

National Institute of Standards and Technology (2020) NIST privacy framework: A tool for improving privacy through enterprise risk management, Version 1.0

Neel S, Roth A, Sharifi-Malvajerdi S (2021) Descent-to-Delete: Gradient-based methods for machine unlearning. In: Proceedings of the 32nd International Conference on Algorithmic Learning Theory, vol 132, pp 931–962

Nie Y, Wang Z, Yu Y, Wu X, Zhao X, Guo W, Song D (2024) PrivAgent: Agentic-based red-teaming for LLM privacy leakage. arXiv:2412.05734 [cs.CR]

Ohrimenko O, Schuster F, Fournet C, Mehta A, Nowozin S, Vaswani K, Costa M (2016) Oblivious Multi-Party machine learning on trusted processors. In: 25th USENIX Security Symposium (USENIX Security 16), pp 619–636

Oliynyk D, Mayer R, Rauber A (2023) I know what you trained last summer: A survey on stealing machine learning models and defences. ACM Comput Surv 55(14s)

OpenAI (2024) GPT-4 technical report. arXiv:2303.08774 [cs.CL]

Orekondy T, Schiele B, Fritz M (2019) Knockoff nets: Stealing functionality of black-box models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

Patra A, Suresh A (2020) BLAZE: Blazing fast privacy-preserving machine learning. In: Proceedings 2020 Network and Distributed System Security Symposium, NDSS 2020

Petrov I, Dimitrov DI, Baader M, Mueller MN, Vechev M (2024) DAGER: Exact gradient inversion for large language models. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems

Phong LT, Aono Y, Hayashi T, Wang L, Moriai S (2018) Privacy-preserving deep learning via additively homomorphic encryption. IEEE Transactions on Information Forensics and Security 13(5):1333–1345

Plotkin D (2013) Data Stewardship: An Actionable Guide to Effective Data Management and Data Governance. Morgan Kaufmann Publishers Inc.

Ponomareva N, Hazimeh H, Kurakin A, Xu Z, Denison C, McMahan HB, Vassilvitskii S, Chien S, Thakurta AG (2023) How to DP-fy ML: A practical guide to machine learning with differential privacy. Journal of Artificial Intelligence Research 77:1113–1201

Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, Sastry G, Askell A, Mishkin P, Clark J, Krueger G, Sutskever I (2021) Learning transferable visual models from natural language supervision. In: Proceedings of the 38th International Conference on Machine Learning, vol 139, pp 8748–8763

Ren J, Chen K, Chen C, Sehwag V, Xing Y, Tang J, Lyu L (2025) Self-comparison for dataset-level membership inference in large (vision-)language model. In: The Web Conference 2025

Riazi MS, Laine K, Pelton B, Dai W (2020) Heax: An architecture for computing on encrypted data. In: Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS '20, pp 1295–1309

Rolnick D, Kording K (2020) Reverse-engineering deep ReLU networks. In: Proceedings of the 37th International Conference on Machine Learning, vol 119, pp 8178–8187

Salem A, Bhattacharya A, Backes M, Fritz M, Zhang Y (2020) Updates-Leak: Data set inference and reconstruction attacks in online learning. In: 29th USENIX Security Symposium (USENIX Security 20), pp 1291–1308

Samardzic N, Feldmann A, Krastev A, Devadas S, Dreslinski R, Peikert C, Sanchez D (2021) F1: A fast and programmable accelerator for fully homomorphic encryption. In: MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture, MICRO '21, pp 238–252

Sang F, Lee J, Zhang X, Kim T (2025) PORTAL: Fast and Secure Device Access with Arm CCA for Modern Arm Mobile System-on-Chips (SoCs). In: 2025 IEEE Symposium on Security and Privacy (SP), pp 4099–4116

Shamir A (1979) How to share a secret. Commun ACM 22(11):612–613

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

Sierra-Arriaga F, Branco R, Lee B (2020) Security issues and challenges for virtualization technologies. ACM Comput Surv 53(2)

Sinha Roy S, Turan F, Jarvinen K, Vercauteren F, Verbauwhede I (2019) FPGA-based high-performance parallel architecture for homomorphic computing on encrypted data. In: 2019 IEEE International Symposium on High Performance Computer Architecture (HPCA), pp 387–398

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

Tang X, Mahloujifar S, Song L, Shejwalkar V, Nasr M, Houmansadr A, Mittal P (2022) Mitigating membership inference attacks by Self-Distillation through a novel ensemble architecture. In: 31st USENIX Security Symposium (USENIX Security 22), pp 1433–1450

Tang Z, Shi S, Li B, Chu X (2023) GossipFL: A decentralized federated learning framework with sparsified and adaptive communication. IEEE Transactions on Parallel and Distributed Systems 34(3):909–922

Tramer F, Boneh D (2019) Slalom: Fast, verifiable and private execution of neural networks in trusted hardware. In: International Conference on Learning Representations

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

Truex S, Baracaldo N, Anwar A, Steinke T, Ludwig H, Zhang R, Zhou Y (2019) A hybrid approach to privacy-preserving federated learning. In: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, AISec'19, pp 1–11

Truex S, Liu L, Chow KH, Gursoy ME, Wei W (2020) Ldp-fed: federated learning with local differential privacy. In: Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking, EdgeSys '20, pp 61–66

Truong JB, Maini P, Walls RJ, Papernot N (2021) Data-free model extraction. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 4769–4778

Van Schaik S, Seto A, Yurek T, Batori A, AlBassam B, Genkin D, Miller A, Ronen E, Yarom Y, Garman C (2024) SoK: SGX.Fail: How stuff gets exposed. In: 2024 IEEE Symposium on Security and Privacy (SP), pp 4143–4162

Vanoverloop D, Sanchez A, Toffalini F, Piessens F, Payer M, Van Bulck J (2025) TLBlur: Compiler-assisted automated hardening against controlled channels on off-the-shelf Intel SGX platforms. In: 34th USENIX Security Symposium (USENIX Security 25)

Volos S, Vaswani K, Bruno R (2018) Graviton: Trusted execution environments on GPUs. In: 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18), USENIX Association, pp 681–696

Wang Q, Oswald D (2024) Confidential computing on heterogeneous CPU-GPU systems: Survey and future directions. arXiv:2408.11601 [cs.CR]

Wang X, Xiang Y, Gao J, Ding J (2021) Information laundering for model privacy. In: International Conference on Learning Representations

Wang Z, Song M, Zhang Z, Song Y, Wang Q, Qi H (2019) Beyond inferring class representatives: User-level privacy leakage from federated learning. In: IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, IEEE Press, pp 2512–2520

Wei L, Luo B, Li Y, Liu Y, Xu Q (2018) I know what you see: Power side-channel attack on convolutional neural network accelerators. In: Proceedings of the 34th Annual Computer Security Applications Conference, Association for Computing Machinery, ACSAC '18, pp 393–406

Whitehouse J, Ramdas A, Rogers R, Wu S (2023) Fully-adaptive composition in differential privacy. In: Proceedings of the 40th International Conference on Machine Learning, vol 202, pp 36990–37007

Wolf S, Hu H, Cooley R, Borowczak M (2021) Stealing machine learning parameters via side channel power attacks. In: 2021 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), pp 242–247

Wong RY, Chong A, Aspegren RC (2023) Privacy legislation as business risks: How GDPR and CCPA are represented in technology companies' investment risk disclosures. Proc ACM Hum-Comput Interact 7(CSCW1)

Wu F, Inan HA, Backurs A, Chandrasekaran V, Kulkarni J, Sim R (2024a) Privately aligning language models with reinforcement learning. In: The Twelfth International Conference on Learning Representations

Wu T, Panda A, Wang JT, Mittal P (2024b) Privacy-preserving in-context learning for large language models. In: The Twelfth International Conference on Learning Representations

Xiao Y, Li M, Chen S, Zhang Y (2017) STACCO: Differentially analyzing side-channel traces for detecting SSL/TLS vulnerabilities in secure enclaves. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS '17, pp 859–874

Yan H, Li X, Li H, Li J, Sun W, Li F (2022) Monitoring-based differential privacy mechanism against query flooding-based model extraction attack. IEEE Transactions on Dependable and Secure Computing 19(4):2680–2694

Yang Y, Kannan R, Prasanna VK (2025) OLA: An FPGA-based overlay accelerator for privacy preserving machine learning with homomorphic encryption. In: Proceedings of the 2025 ACM/SIGDA International Symposium on Field Programmable Gate Arrays, FPGA '25, p 127–138

Ye J, Maddi A, Murakonda SK, Bindschaedler V, Shokri R (2022) Enhanced membership inference attacks against machine learning models. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS '22, pp 3093–3106

Yin H, Mallya A, Vahdat A, Alvarez JM, Kautz J, Molchanov P (2021) See through gradients: Image batch recovery via GradInversion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 16337–16346

Yousefpour A, Shilov I, Sablayrolles A, Testuggine D, Prasad K, Malek M, Nguyen J, Ghosh S, Bharadwaj A, Zhao J, Cormode G, Mironov I (2022) Opacus: User-friendly differential privacy library in pytorch. arXiv:2109.12298 [cs.LG]

Yu D, Zhang H, Chen W, Yin J, Liu TY (2021) Large scale private learning via low-rank reparametrization. In: Proceedings of the 38th International Conference on Machine Learning, vol 139, pp 12208–12218

Yu D, Naik S, Backurs A, Gopi S, Inan HA, Kamath G, Kulkarni J, Lee YT, Manoel A, Wutschitz L, Yekhanin S, Zhang H (2022) Differentially private fine-tuning of language models. In: International Conference on Learning Representations

Yu D, Kairouz P, Oh S, Xu Z (2024) Privacy-preserving instructions for aligning large language models. In: Proceedings of the 41st International Conference on Machine Learning, vol 235, pp 57480–57506

Yüksel N, Börklü HR, Sezer HK, Canyurt OE (2023) Review of artificial intelligence applications in engineering design perspective. Engineering Applications of Artificial Intelligence 118:105697

Zama (2022) Concrete ML: a privacy-preserving machine learning library using fully homomorphic encryption for data scientists. https://github.com/zama-ai/concrete-ml

Zanella-Béguelin S, Wutschitz L, Tople S, Rühle V, Paverd A, Ohrimenko O, Köpf B, Brockschmidt M (2020) Analyzing information leakage of updates to natural language models. In: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, CCS '20, pp 363–375

Zhang B, Dong Y, Wang T, Li J (2024a) Towards certified unlearning for deep neural networks. In: Proceedings of the 41st International Conference on Machine Learning, vol 235, pp 58800–58818

Zhang C, Li S, Xia J, Wang W, Yan F, Liu Y (2020) BatchCrypt: Efficient homomorphic encryption for Cross-Silo federated learning. In: 2020 USENIX Annual Technical Conference (USENIX ATC 20), pp 493–506

Zhang D, Xia B, Liu Y, Xu X, Hoang T, Xing Z, Staples M, Lu Q, Zhu L (2024b) Privacy and copyright protection in generative AI: A lifecycle perspective. In: Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering - Software Engineering for AI, CAIN '24, pp 92–97

Zhang J, Das D, Kamath G, Tramèr F (2024c) Membership inference attacks cannot prove that a model was trained on your data. arXiv:2409.19798 [cs.LG]

Zheng H, Ye Q, Hu H, Fang C, Shi J (2019) Bdpl: A boundary differentially private layer against machine learning model extraction attacks. In: Sako K, Schneider S, Ryan PYA (eds) Computer Security – ESORICS 2019, Springer International Publishing, pp 66–83

Zhu J, Blaschko MB (2021) R-GAP: Recursive gradient attack on privacy. In: International Conference on Learning Representations

Zhu J, Yin H, Deng P, Almeida A, Zhou S (2024) Confidential computing on NVIDIA Hopper GPUs: A performance benchmark study. arXiv:2409.03992 [cs.DC]

Zhu L, Liu Z, Han S (2019) Deep leakage from gradients. In: Advances in Neural Information Processing Systems, vol 32