Review

Attacks and Defenses in Differentially Private Deep Learning: New Security Risks in New Era

Kaiyan Zhao , Zhe Sun , Lihua Yin * , Tianqing Zhu *

Posted Date: 16 March 2026

doi: 10.20944/preprints202603.1179.v1

Keywords: differential privacy; DP-DL systems; privacy risks; attack and defense

 doi:10.20944/preprints202603.1179.v1

Review

Attacks and Defenses in Differentially Private Deep Learning: New Security Risks in New Era †

Kaiyan Zhao [1], Zhe Sun [1], Lihua Yin [1,*] and Tianqing Zhu [2,*]

[1] Cyberspace Institute of Advanced Technology, Guangzhou University, China
[2] Faculty of Data Science, City University of Macau, China
* Correspondence: yinlh@gzhu.edu.cn (L.Y.); tqzhu@cityu.edu.mo (T.Z.)
† This research was supported in part by the National Natural Science Foundation of China (No. 62472114), in part by the Joint Funding Special Project for Guangdong-Hong Kong Science and Technology Innovation (No. 2024A0505040027), in part by the Guangdong Basic and Applied Basic Research Foundation (No. 2026A1515011322).

Abstract

With the rapid advancement of deep learning, differential privacy has become a key technique for protecting sensitive data with a formal guarantee of privacy. By injecting noise and enforcing privacy budgets, differentially private deep learning (DP-DL) systems are able to protect individual data points yet still maintain a model's utility. However, recent studies reveal that DP-DL systems can be vulnerable to different types of attacks throughout their lifecycle. Naturally, this has attracted the attention of both academia and industry. Critically, these risks are not the same as those associated with traditional deep learning. This is because the differential privacy mechanism itself introduces new attack surfaces that adversaries can exploit. Our work focuses on the distinct vulnerabilities that can arise at the data, algorithm, and architecture levels. By analyzing representative attacks and corresponding defenses, this survey highlights emerging challenges and outlines promising research directions. Overall, our aim is to make differential privacy more robust and deployable in real-world deep learning systems.

Keywords: differential privacy; DP-DL systems; privacy risks; attack and defense

1. Introduction

Given its many benefits, differential privacy (DP) is increasingly integrated into deep learning. As such, differentially private deep learning (DP-DL) systems have been deployed across a wide range of sensitive scenarios, including recommendation systems [1], medical systems [2], and natural language processing tasks [3–5]. Traditional deep learning models without differential privacy are known to leak sensitive information, either through a model's outputs, its internal representations, or when sharing parameters. However, what is less known is that DP-DL systems are vulnerable to a unique set of their own privacy risks arising from the interactions between the differential privacy mechanisms and the models. For example, to maintain a model's utility, one typically limits the amount of noise added to the dataset and/or increases the privacy budget. But this could introduce the risk of a repeated-query attack [6,7] or a floating-point attack [8]. Importantly, these attacks do not suggest that differential privacy itself is unsafe; rather, they highlight that, in deploying real-world DP-DL systems, privacy risks primarily arise from practical design choices, implementation details, and the trade-offs between privacy guarantees and model utility.

As differential privacy (DP) techniques transition from theoretical constructs into practical deployments in deep learning systems, it has become increasingly evident that discrepancies may arise between the theoretical privacy guarantees of DP and the actual security provided by real-world DP–DL systems [9]. Theoretical guarantees are typically derived under idealized assumptions and worst-case analyses, whereas the practical privacy protection of deployed systems is influenced by many factors, including semantically correlated data, algorithmic implementations, complex model

 doi:10.20944/preprints202603.1179.v1

architectures, and varying attacker capabilities. These factors may introduce new vulnerabilities that are not fully captured by existing theoretical analyses.

Understanding these potential risks is therefore essential for the responsible and effective use of differential privacy in deep learning. First, it is crucial to systematically identify the security and privacy risks inherent in DP–DL systems, enabling practitioners and researchers to apply DP mechanisms more cautiously and avoid overestimating their practical protection capabilities. Second, beyond risk identification, it is equally important to analyze existing mitigation strategies and potential defense mechanisms that can reduce these vulnerabilities and enhance the robustness of DP-enabled learning systems. Finally, by comprehensively examining both risks and countermeasures, we aim to provide insights that can guide future research toward resolving these limitations and improving the design and deployment of differential privacy mechanisms in deep learning. In this review, we categorize DP–DL systems into three levels and discuss the distinct security risks particular to each level.

- Data-level risks: Differential privacy is designed to preserve statistical utility under a bounded privacy loss; however, the strong representational capacity of deep learning models can glean stable statistical and semantic correlations from real data. When combined with certain prior knowledge (e.g., population-level statistics or correlated attributes), attackers might be able to exploit such learned correlations to infer sensitive information [10,11]. Significantly, such inference risks do not contradict the formal guarantees of DP. Rather, they reflect the limitations that arise from distributional properties and adversarial assumptions.
- Algorithm-level risks: Typically, a differential privacy mechanism is fully integrated into a deep learning system's training pipeline. Here, gradient clipping and noise injection are usually used to control sensitivity. However, implementation-level deviations in real-world systems (e.g., finite-precision computation [12]) can undermine the effectiveness of the noise, weakening any guarantees of privacy or even invalidating them completely. Notably, such risks do not stem from the inherent flaws of the differential privacy mechanism. Rather, they arise from disparities between how a system has actually been implemented versus the idealized assumptions inherent to the differential privacy algorithm.
- Architecture-level risks: Each model architecture, even within the realm of deep learning, will have substantially different numbers of parameters, representational capacity, and optimization dynamics. Additionally, the way each system memorizes training data will be different. Prior studies [13,14] have shown that these empirical differences can affect a model's sensitivity to specific attacks, leading to varying degrees of privacy leaks even given the same privacy budget. Crucially, these variations do not reflect any breakdown of the theoretical guarantees differential privacy affords. Instead, they reflect the fact that the worst-case privacy bounds provided by differential privacy are inherently tolerant of diverse model behaviors. Nonetheless, from a system design and deployment standpoint, failing to account for architecture-dependent differences in empirical leaks may result in an incomplete assessment of practical privacy risks.

In practice, all these risks undermine the ability of DP-DL systems to guarantee that their training data are safe from prying eyes. Such challenges to security and trustworthiness not only compromise large-scale deployment in privacy-sensitive scenarios but could also result in violations of data protection regulations. In this context, systematically analyzing privacy risks at each level of architecture is essential if we are to build reliable and secure DP-DL systems.

The Contributions of This Survey. This survey systematically examines attacks and defenses on DP-DL systems from three perspectives: data level, algorithm level, and system architecture level (as shown in Figure 1). Its goal is to illuminate the potential risks in DP-DL deployments, while providing a structured framework to guide future research. The main contributions are as follows:

1. We propose an analytical framework for assessing DP-DL security across the data, algorithm, and architecture levels. This framework comprehensively reviews DP-specific risks, constructing a complete threat landscape.

 doi:10.20944/preprints202603.1179.v1

2. We synthesize and categorize existing defense methods according to their corresponding risk sources, providing a structured "risk-attack-defense" view that unifies previously scattered insights.
3. We analyze the gap between the theoretical privacy guarantee of differential privacy and actual system security, emphasizing the disparities between empirical privacy ε_{emp} and a system's theoretical budget ε.
4. We outline key future research directions to guide the development of more robust and secure DP-DL systems.

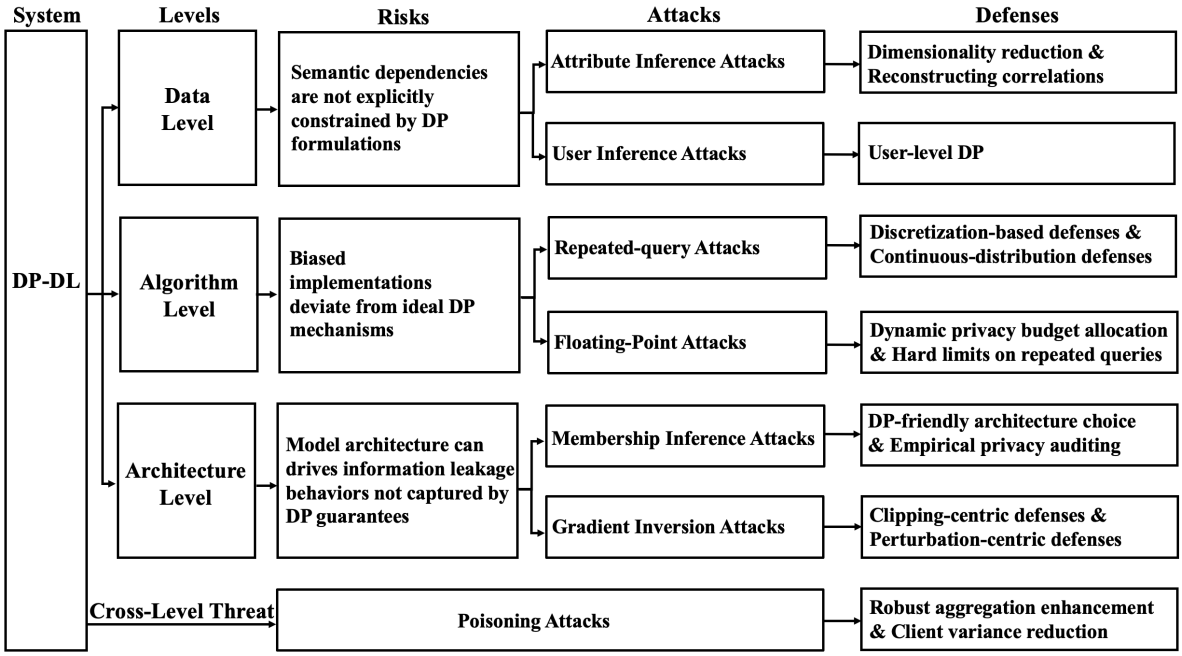

Figure 1. The structure of the survey on threats and defenses in DP-DL. The framework is organized into three levels: data, algorithm, and architecture. For each level, the first column summarizes the associated risks, the second column lists representative attacks exploiting these risks, and the third column presents corresponding defense strategies. Cross-level threats, such as poisoning attacks, and their mitigations are also included.

2. Preliminaries

This section introduces the fundamental concepts and frameworks of differentially private deep learning (DP-DL). Section 2.1 defines differential privacy and formulates DP-DL by modeling the deep learning training process as a randomized mechanism with (ε, δ)-DP guarantees, ensuring a bounded influence of individual training samples on released models. Section 2.2 presents a multi-level framework of DP-DL, classifying existing methods into data-level, algorithm-level, and architecture-level techniques according to where privacy mechanisms are integrated in the deep learning models.

2.1. The Definition of DP-DL

Differentially private deep learning (DP-DL) provides formal privacy guarantees for training a deep neural network by modeling the learning procedure as a randomized mechanism that satisfies DP [15–19]. Within this framework, the influence of individual training records on the released model or inference interface is provably bounded.

2.1.1. Differential Privacy [20]

Let $\mathcal{X}$ denote the data domain and let $\mathcal{D} = \mathcal{X}^n$ denote the space of datasets of size n. Two datasets $D, D' \in \mathcal{D}$ are considered adjacent, denoted by $D \sim D'$, if they differ in at most one data record. A randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ satisfies (ε, δ)-differential privacy if, for all adjacent datasets $D \sim D'$ and all measurable subsets $S \subseteq \mathcal{R}$,

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

 doi:10.20944/preprints202603.1179.v1

Here, $\varepsilon > 0$ controls the magnitude of privacy loss associated with modifying a single record, while $\delta \in [0,1)$ bounds the probability of deviations beyond this limit and is typically chosen to be negligible. A closely related notion is the sensitivity of a function f, which characterizes the maximum influence of a single record and determines the scale of noise required for differential privacy: $\Delta_2(f) = \max_{D \sim D'} \|f(D) - f(D')\|_2$ [117]. Differential privacy is commonly divided into two types: central differential privacy (CDP), which assumes a trusted data curator, and local differential privacy (LDP), which does not rely on such trust. Common CDP mechanisms include the Laplace [21], Gaussian [22], and exponential mechanisms [23]. Common LDP mechanisms include randomized response and its variants (e.g., generalized randomized response and unary encoding) [24], RAPPOR [25], and Hadamard response [26].

2.1.2. Differentially private deep learning (DP-DL)

Building upon the notion of differential privacy, DP-DL systems model the deep learning training process as a randomized mechanism acting on the training data. More specifically, consider a training algorithm $\mathsf{Train} : \mathcal{D} \times \mathcal{H} \to \mathcal{O}$ that maps a training dataset $D \in \mathcal{D}$ together with a fixed set of hyperparameters $h \in \mathcal{H}$ to a releasable output in the space $\mathcal{O}$. The hyperparameters h include the model's architecture, the optimization method, the learning rate schedule, the batch size, and the number of training iterations. Meanwhile, the output space $\mathcal{O}$ is comprised of the trained model parameters and intermediate checkpoints, or a model accessible through an inference interface. Given D and h, the training procedure produces an output $O \leftarrow \mathsf{Train}(D;h)$, where randomness arises from stochastic optimization and explicit noise injection.

The training algorithm Train satisfies (ε, δ)-differentially private deep learning if, for any fixed choice of hyperparameters h, the induced mapping from the training datasets to the outputs complies with differential privacy. Formally, for all adjacent datasets $D \sim D'$ and all measurable subsets $S \subseteq \mathcal{O}$,

$$\Pr[\mathsf{Train}(D;h) \in S] \leq e^{\varepsilon} \Pr[\mathsf{Train}(D';h) \in S] + \delta. \tag{2}$$

This formulation (2) ensures that all released information satisfies the same privacy constraints. Post processing preserves the privacy guarantee, while repeated executions incur a cumulative privacy loss according to standard composition principles.

2.2. *Multi-Level DP-DL Frameworks*

DP-DLs are not comprised of a single, isolated privacy mechanism. Rather, privacy guarantees are made possible by systematically introducing privacy-preserving constraints at different stages of a model's training process. Hence, differential privacy mechanisms can be divided into three categories based on where and how the privacy mechanism has been implemented within the training process, i.e., at the data-level, the algorithm-level, or the architecture-level. As such, DP-DL systems are typically unified multi-level frameworks, as shown in Figure 2.

(1) *Data-level DP-DL*: In data-level DP-DL, privacy protection is achieved by randomizing the training data prior to the learning process. Instead of operating directly on the original dataset, the model is trained on a privatized version produced by a data randomization mechanism. Under this paradigm, the training algorithm depends on the original dataset only through this transformed dataset, ensuring that the learning process never directly accesses the raw data.

Several forms of data perturbation can be employed to construct such a randomized dataset. Raw data perturbation [27] introduces randomness directly at the input level by independently modifying each sample through noise injection. Feature perturbation [28] instead operates in a learned representation space: the input samples are first mapped to feature embeddings, after which noise is injected into the resulting features before training proceeds. Label perturbation [29,30] modifies the supervision signal by replacing the original labels with randomized versions according to a predefined randomization rule. In addition to these direct perturbation strategies, the training data may also be replaced with synthetic data generated by a differentially private data generator [31–33]. In this

 doi:10.20944/preprints202603.1179.v1

case, a new dataset is sampled from a generator trained under differential privacy constraints, and the downstream learning algorithm operates exclusively on the generated samples.

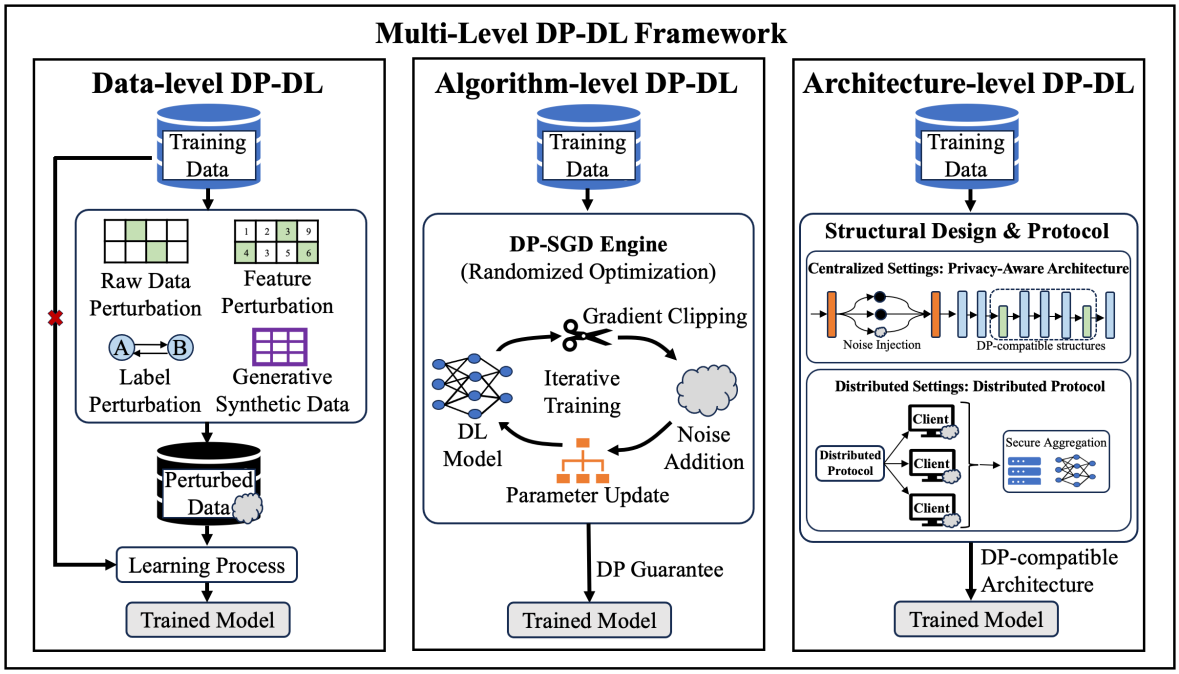

Figure 2. Applied differential privacy in deep learning. Differential privacy can be integrated into deep learning at three levels: data-level, where training data are randomized through perturbation or synthetic generation; algorithm-level, where DP-SGD enforces privacy via gradient clipping and noise during optimization; and architecture-level, where privacy-aware model designs or distributed protocols embed DP constraints into the training system.

Under this framework, the overall training procedure satisfies (ε, δ)-differential privacy provided that the data randomization mechanism limits the influence of any individual record in the original dataset. Because the subsequent training process operates solely on the privatized dataset, it constitutes post-processing of the privacy mechanism and therefore does not weaken the differential privacy guarantee.

(2) *Algorithm-level DP-DL*: At the algorithm level, differential privacy is enforced within the training procedure by randomizing any intermediate computations. In this setting, Algorithm-level DP-DL satisfies the DP-DL condition in Equation (2) and provides an end-to-end differential privacy guarantee. A representative method at this level is differentially private stochastic gradient descent (DP-SGD) [34,35].

In DP-SGD, the model parameters are updated iteratively during training using stochastic gradient descent with additional privacy-preserving steps. At each iteration, a mini-batch of training samples is first randomly selected. For each sample in the mini-batch, the gradient of the loss function with respect to the model parameters is computed. To control the influence of any single data sample, the norm of each per-sample gradient is clipped so that it does not exceed a predefined bound. The clipped gradients from all samples in the mini-batch are then averaged to form the batch gradient.

To ensure differential privacy, random Gaussian noise is added to the averaged gradient before updating the model parameters [17,36]. The amount of noise is controlled by a noise multiplier, which determines the variance of the Gaussian distribution relative to the gradient clipping bound. Finally, the model parameters are updated using the learning rate and the noisy gradient estimate.

(3) *Architecture-level DP-DL*: At the architecture level, differential privacy is achieved through a structural design that embeds sensitivity control into the training procedure. This works in contrast to algorithm-level DP-DL, which relies on randomized optimization.

 doi:10.20944/preprints202603.1179.v1

In centralized settings, architecture-level DP-DL is typically implemented through a training procedure built around a privacy-aware model architecture. Such architectures regulate how information flows through the network during training. This is achieved through several design strategies, including DP-compatible structures [37,38] such as bottlenecked or simplified modules that limit information propagation, normalization mechanisms that remove cross-sample dependencies, and architectural components specifically designed to support noise injection during private training. Given a training dataset and a set of hyperparameters, the training algorithm processes each input by passing it through a sequence of architectural modules. These modules transform the input representations layer by layer while structural components constrain how much information can propagate through the network. During this process, a noise mechanism is typically applied to intermediate or learned representations to limit the influence of any individual data sample. The architecture therefore defines not only the model structure but also the way the training process incorporates privacy-preserving transformations. The final model output is obtained through the overall training procedure determined by this privacy-aware architecture.

In distributed settings, architecture-level DP-DL is realized through a system architecture that determines how training is decomposed across multiple participants. In this case, the architecture specifies a distributed protocol that combines local model training, client-side privacy protection, and secure aggregation [39]. Each participant computes model updates locally using its own data. Before these updates are shared, they are clipped to ensure that the magnitude of each client's contribution is bounded, thereby controlling sensitivity. Random noise may then be added to the clipped updates to provide differential privacy. The privatized updates from multiple participants are subsequently aggregated using secure aggregation mechanisms. This ensures that the central server can only observe the aggregated result rather than any individual participant's update. After receiving the aggregated information, the server performs further processing, such as updating the global model parameters. Importantly, the server's computation is considered post-processing, which does not weaken the privacy guarantees.

Overall, the resulting mapping from the training data to the outputs satisfies the same DP-DL definition as per Equation (2). Moreover, DP-DL at the architecture-level includes both centralized and distributed approaches that enforce differential privacy through structural designs. In this way, architecture-level DP-DL complements algorithm-level methods that rely on randomization at the optimization stage.

3. Risks in DP-DL

For a long time, differential privacy has been widely regarded as a reliable and robust privacy protection mechanism in the field of deep learning. The common view holds that introducing noise during the training process can effectively prevent models from leaking sensitive information from the original data. However, as research deepens, this traditional perception is gradually being broken. More and more evidence is indicating that DP-DL systems are not absolutely safe. Although differential privacy theoretically provides a strong guarantee of privacy, especially in deep learning systems, practically speaking, almost every model is vulnerable to certain risks. To illustrate this issue, we have analyzed most of these risks at three levels – data, algorithm, and architecture – as shown in Figure 3. Data-level risks arise from semantic and statistical correlations in real-world data, which deep models can memorize and use to infer sensitive information even when such information is not explicitly exposed. Algorithm-level risks originate from the disparity between how an approach works in theory versus how it works given its practical implementation. Examples of algorithm-level risk include weakened randomness or engineering approximations that lessen the effectiveness of injected noise – both of which undermine guarantees of privacy. Architecture-level risks reflect the opacity and complexity of deep models, where different architectures, training dynamics, and distributed systems may introduce leakage pathways that fall outside the scope of formal differential privacy guarantees, ultimately degrading real-world protection.

 doi:10.20944/preprints202603.1179.v1

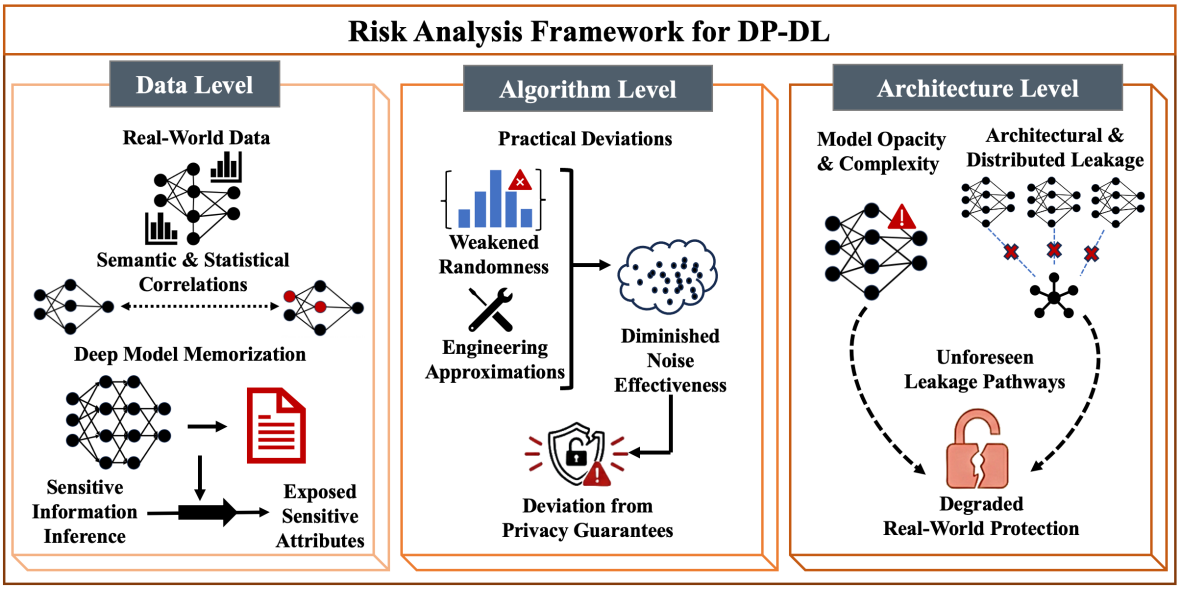

Figure 3. Risk analysis framework for DP-DL. The framework categorizes privacy risks in differentially private deep learning across three levels: data, algorithm, and architecture. Data-level risks arise from semantic and statistical correlations and model memorization, which may enable sensitive attribute inference. Algorithm-level risks result from practical deviations from ideal DP mechanisms, such as weakened randomness and engineering approximations, reducing noise effectiveness. Architecture-level risks stem from the opacity and complexity of deep models and distributed systems, which may introduce unforeseen leakage pathways and weaken real-world privacy protection.

3.1. Data-Level Risks

At the data level, the traditional view holds that perturbing explicit sensitive fields under formal guarantees of differential privacy should be sufficient to prevent leaks. However, DP-DL systems inherently memorize statistically and semantically informative patterns in the data, enabling attackers to infer sensitive information from correlated or contextual signals without direct access. The main reason for this risk is that **semantic dependencies are not explicitly constrained by differential privacy formulations**. Differential privacy is designed to limit information leaks at the individual level by perturbing data or injecting noise. Consequently, the underlying semantic dependencies (e.g., statistical structures, semantic associations, and causal relations) tend to remain substantially preserved when attempting to maintain model utility.

1. *Sensitive attribute inference from non-sensitive attributes.* Standard differential privacy analyses bound information leakage by perturbing individual attributes but do not explicitly account for inference enabled through correlated attributes. In real data scenarios, behavioral, demographic, and contextual attributes often have statistical dependencies [40–42], which causes sensitive attributes to be implicitly encoded along with non-sensitive attributes in the learned representations. Therefore, although the formal differential privacy guarantees remain unchanged, the attacker's ability to infer sensitive attributes may increase, especially for individuals with strong correlations between attributes [10,43].
2. *User-level privacy leakage under record-level DP.* Differential privacy guarantees are defined with respect to a chosen privacy unit, which in many DP-DL systems is implicitly set at the record level [4,44–46]. In deep learning models, multiple records contributed by the same user can be jointly encoded during training, allowing user-level patterns to be captured in model representations. When users contribute multiple records, record-level privacy therefore fails to bound the cumulative information learned about an entire user. This misalignment between the privacy unit and the actual threat model leads to a degradation of user-level protection [11]. Removing or perturbing individual records does not approximate the removal of all information associated

 doi:10.20944/preprints202603.1179.v1

with a user, allowing user-level signals to remain detectable. Consequently, adversaries can infer user participation or attributes even when record-level differential privacy is enforced.

3.2. Algorithm-Level Risks

At the algorithm level, the traditional view holds that as long as noise is rigorously injected into the training process through differentially private mechanisms, all data should be indistinguishable. However, in real DP-DL implementations, using a different engineering approximation and/or the implementation preferences of the developer can significantly weaken or even invalidate the guarantee of privacy. The key is inconsistency; **biased implementations deviate from ideal differential privacy mechanisms**. For example, theory assumes ideal randomness, infinite precision, and strict adjacency relations, while engineering systems must contend with efficiency, reproducibility, and hardware constraints, which inevitably weaken or violate these theoretical conditions.

1. *Repeated-Query Handling Without Proper Privacy Accounting.* In theory, differential privacy protect sensitive data elegantly through well-defined privacy accounting, even given repeated queries.In practice, however, implementation choices and deployment constraints may introduce systematic biases that deviate from this perfect model. For example, repeated or semantically equivalent queries are sometimes handled as independent interactions, negating any accounting or access control. As such, an adversary can collect multiple noisy outputs corresponding to the same underlying value and then reduce or remove any statistical noise through averaging or by inferring the data's distribution [6,7,47]. Notably, this behavior does not contradict the formal definition of differential privacy. On the contrary, it reflects a common flaw in how many algorithms have been implemented.
2. *Finite-Precision Noise Implementation in DP-SGD.* DP-SGD relies on adding noise drawn from continuous distributions (e.g., Laplace or Gaussian), to guarantee differential privacy under bounded sensitivity [48–51]. In practice, however, noise sampling and accumulation in DP-SGD are implemented using finite-precision floating-point arithmetic. This discretization can introduce systematic irregularities in how the noise is distributed, including unreachable values ("holes") or a distorted probability mass. Such artifacts may render the noisy updates statistically distinguishable across neighboring datasets, violating the assumptions of indistinguishability underlying DP-SGD's privacy analysis [8,12,52]. As a result, even when DP-SGD is implemented according to its formal specification, finite-precision effects can undermine privacy guarantees at the algorithmic level.

3.3. Architecture-Level Risks

At the architectural level, the traditional view holds that differential privacy guarantees, once set by ε, apply uniformly across models regardless of how they internally represent data; however, in real DP-DL practice, different neural architectures encode, store, and expose information in highly uneven ways, causing models with identical ε budgets to exhibit divergent empirical leakage behaviors. The main reason behind this risk is architectural variability: **Model architecture can drives information leakage behaviors not captured by DP guarantees.** For example, theoretical DP analyses assume architecture-agnostic noise injection and bounded privacy loss, while actual deep models differ in depth, inductive biases, memorization tendencies and extraction difficulty, thereby weakening the assumption of uniform privacy protection across architectures.

1. *Different model architectures leak privacy differently.* The nominal ε used in DP-DL systems provides a worst-case theoretical guarantee rather than a direct measure of a model's empirical information leakage, denoted as ε_{emp}. Here, the empirical privacy risk ε_{emp} is quantified via DP auditing as the smallest value consistent with the observed success of an optimal membership inference test. Concretely, given false positive and false negative rates (β, α) measured from repeated attacks on neighboring datasets, ε_{emp} is defined as the minimal ε such that $1 - \beta \leq e^{\varepsilon}(1 - \alpha) + \delta$ holds for a fixed δ. Under this formulation, the nominal ε from privacy accounting serves as an architecture-

 doi:10.20944/preprints202603.1179.v1

agnostic upper bound, while ε_{emp} provides an architecture- and training-dependent lower bound on the actual privacy loss realized by the model [13,14,53,54]. This discrepancy underscores the risk that formally private models may still expose non-negligible, model-dependent information in practice.

2. *Gradient consistency enhances the inversion risk of the model.* Gradient updates serve as the primary conduit through which local training data influence the global model in distributed DP-DL. While differential privacy aims to limit information leakage by perturbing gradients, most theoretical analyses treat each update as an independent DP mechanism and rely on adaptive composition to bound the cumulative privacy loss. In practical deep learning systems, however, model architecture, optimization dynamics, and task semantics may induce statistical or geometric regularities across gradient updates that arise from the learning process itself and are not fully captured by the DP mechanism. Such regularities are difficult to characterize formally due to the complex and non-interpretable nature of deep models, yet they may give rise to data-dependent patterns that persist even under DP noise. Prior studies [55–59] provide partial empirical evidence for this phenomenon by exploiting local linear dependencies between gradients and input features, enabling the manipulation or analysis of gradient update equations to recover embedded input signals. Although these observations do not violate formal DP guarantees, they suggest a potential gap between worst-case DP analyses and the empirically observable privacy behaviors of modern deep learning systems.

Poisoning as a Cross-level Threat. Beyond passive privacy risks, poisoning represents an active adversarial strategy whose effectiveness relies on cross-level propagation rather than a single-level instantiation in DP-DL systems [9,60]. While specific poisoning techniques may originate at an individual level, poisoning attacks intentionally exploit and amplify mismatches between differential privacy assumptions and deep learning behavior across the data, algorithm, and architecture levels.

Poisoning attacks are shown separately to emphasize their cross-level propagation rather than their point of instantiation (see Figure 4). At the data level, adversaries may inject malicious samples to bias empirical data statistics and enhance inferential signals; at the algorithm level, such poisoned inputs or updates (such as malicious gradients) can distort gradient statistics, interacting with clipping and noise mechanisms to exacerbate sensitivity amplification; at the architecture level, attackers further leverage architectural opacity and aggregation logic to enable stealthy model manipulation that persists through training and remains difficult to detect.

From this perspective, poisoning is categorized as a cross-level threat because its attack chain spans multiple stages of the DP-DL pipeline, allowing both untargeted poisoning (which degrades overall model utility) and targeted poisoning (which induces precise adversary-controlled behaviors) to be realized without violating nominal DP guarantees.

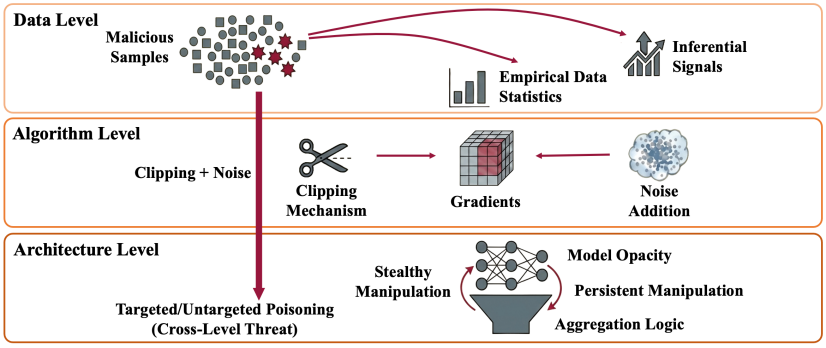

Figure 4. Cross-level propagation of poisoning attacks in a DP-DL pipeline. Poisoning attacks act as a cross-level threat spanning the data, algorithm, and architecture levels. At the data level, adversaries inject malicious samples to bias empirical statistics and amplify inferential signals. At the algorithm level, poisoned inputs or gradients interact with clipping and noise mechanisms, distorting gradient statistics. At the architecture level, attackers exploit model opacity and aggregation logic to enable stealthy and persistent manipulation.

 doi:10.20944/preprints202603.1179.v1

4. Attacks on DP-DL Systems

Although differential privacy is widely regarded in the field of deep learning as a robust privacy protection mechanism, it comes with numerous practical risks that can expose a range of attack surfaces in DP-DL systems (as shown in Table 1). At the data level, residual semantic and statistical structures may still exist after injecting noise that allow adversaries to infer some of the data's attributes. Alternatively, user-level inference attacks can exploit correlations to reveal sensitive attributes or a participant's identity. At the algorithmic level, inconsistencies between differential privacy mechanisms and their implementations can introduce systematic vulnerabilities. For instance, repeated querying can dilute randomness, while floating-point sampling can partially reverse the blanket of noise. These holes create opportunities for successful repeated-query and floating-point attacks. At the architectural level, the complexity of deep models and distributed systems limits the ability of noise to prevent leaks. This not only increases the risk of an adversary mounting a membership inference attack but also a gradient inversion attack. Poisoning further acts as a cross-level threat that simultaneously targets data, algorithms, and architectures to degrade utility. Overall, these risks form a coherent attack chain spanning inference and reconstruction, and degrading the system's integrity. For all these reasons, real-world DP-DL deployments are still susceptible to privacy threats despite the guarantees offered by differential privacy.

Table 1. A summary of different attacks.

Attack type	Literature	Visibility	Evaluation Metrics
Attribute inference attacks	[43]	▨	Accuracy, PPV
	[61]	□	
User inference attacks	[62]	■	ROC, AUROC, TPR@FPR
Repeated-query attacks	[6,7]	▨	Error rate, Precision, R_{test}, R_{unif}, success probability
	[63]	■	
Floating-point attacks	[8,52]	▨	ASR, Attack rate, accuracy
Membership inference attacks	[53,64,65]	■	TPR@FPR, PPV, Precision, AUC
	[53,65,66]	□	
Gradient inversion attacks	[57,58,67,68]	□	MSE, PSNR, CW-SSIM, LPIPS, SSIM, FSIM
Poisoning attacks	[69]	■	ASR, Accuracy, DCR, Error Rate
	[9,70–73]	▨	

■: Blackbox; ▨: Graybox; □: Whitebox.

4.1. Data-Level Attacks

Data-level attacks exploit semantic correlations and structural patterns in training data to infer sensitive information from DP-DL systems, such as attribute inference attacks and user-level inference attacks. Attribute inference attacks leverage feature correlations learned by the model to infer sensitive attributes, while user-level inference attacks determine whether a specific user participated in model training by aggregating the model's behavior over multiple correlated samples. These attacks highlight that DP-DL systems must properly address semantic-level risks by constraining or protecting sensitive correlations.

4.1.1. Attribute Inference Attacks on DP-DL Systems

In an attribute inference attack, the adversary exploits the correlation(s) between high-dimensional attributes. These attacks can be successful in both white- and gray-box cases, as shown in Figure 5.

Jayaraman et al. [61] investigated white-box attacks against DP-MLP models. Their focus is on attacks that exploit the correlation between the model's internal neuron activation values and any sensitive attributes. This correlation exists because, even given differentially private training, the model still learns the statistical distribution of the training data, which includes inter-attribute correlations. Jayaraman and colleagues discuss three different stages of the attack. (1) In the preparation stage, the

 doi:10.20944/preprints202603.1179.v1

attacker prepares an auxiliary dataset and uniformly flips the sensitive attribute values to the desired target value. (2) In the probing stage, the attacker identifies a small set of the most critical neurons by analyzing the correlation between the model's internal neuron activations and the target sensitive attributes. (3) In the inference stage, the attacker infers whether a candidate record contains the target sensitive attribute by observing how intensely key neurons activate.

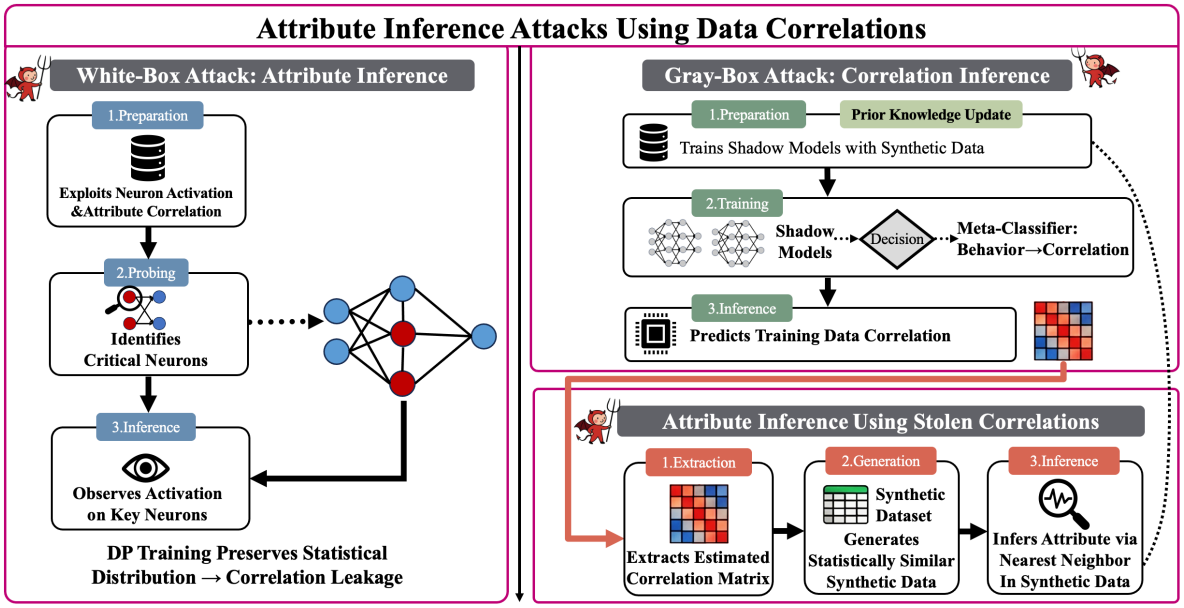

Figure 5. Attribute inference attacks on DP-DL systems. In the white-box setting, attackers exploit correlations between neuron activations and sensitive attributes to infer hidden attributes. In the gray-box setting, attackers infer training data correlations from model outputs using shadow models and generate synthetic data to perform attribute inference.

By contrast, Creţu et al. [43] studied gray-box attacks, again with DP-MLP models, proving that differential privacy cannot effectively prevent correlations from leaking. Moreover, attackers do not need direct access to the model's parameters or training data; they can simply infer the data structures stored in the model's internal memory by observing its external behaviors (e.g., confidence scores). The attacker's first step is to perform a correlation inference attack as follows. (1) In the preparation stage, the adversary use prior knowledge, such as marginal distributions or partial correlation constraints, to generate synthetic datasets with different known correlations. The attacker then trains a set of corresponding shadow models. (2) In the training stage, the adversary trains a meta-classifier that maps model behaviors to data correlations by analyzing the confidence outputs of the shadow models on a fixed query set. (3) In the inference stage, the attacker feeds the confidence outputs of the target model on the same query set into the meta-classifier to predict the correlation interval of its training data.

The next step is to leverage the correlation matrix stolen from the target model to launch an attribute inference attack. Again, this attack has three stages. (1) In the extraction stage, the attacker first extracts the estimated correlations from all the input variables in the target model. (2) In the generation stage, the attacker generates a synthetic dataset that closely matches the statistical characteristics of the original based on the stolen correlation structure and any known marginal distributions. (3) In the inference stage, the attacker searches the synthetic dataset for entries that partially match the target record and uses the average value of their sensitive attributes as the final inference result.

4.1.2. User Inference Attacks on DP-DL Systems

While record-level differential privacy is designed to obscure the contribution of any single data record, it fails to address the collective behavioral patterns of users who contribute multiple correlated records over time. This creates a fundamental mismatch between the privacy guarantee and real-world data, leading to a distinct class of threats called user-level privacy inference. In these

 doi:10.20944/preprints202603.1179.v1

attacks, an adversary aims to determine whether a specific user participated in model training, rather than inferring whether any one individual record was part of the training set[62].

User inference attacks typically progress through three stages, as shown in Figure 6. (1) Behavioral distribution probing: An adversary collects one or more samples from a target user, such as emails, posts, or text snippets that were not used during training, then queries the model and assembles the generated outputs, such as likelihood scores and log-probabilities. Notably, while record-level differential privacy injects noise into each training update, it does not fundamentally change the distributional affinity between the model and the users whose data contributed to fine-tuning. (2) Cross-model reference comparison: The adversary queries a reference model that was not trained on the target user's data (e.g., a pre-trained base model) and compares its outputs with those of the fine-tuned model. If the fine-tuned model consistently assigns higher likelihoods or lower perplexities to the user's samples, it indicates that the records were included in the training process. (3) Statistical aggregation and decision: by aggregating multiple query outputs, the adversary constructs a decision statistic indicating whether the target user's distribution aligns with the fine-tuned model. Across many correlated samples, the record-level noise becomes statistically insignificant compared to the user-level signal formed by repeated contributions. This attack vector highlights a key limitation: record-level differential privacy does not fully protect against inferences drawn from the aggregated footprint of a user's data across multiple training instances.

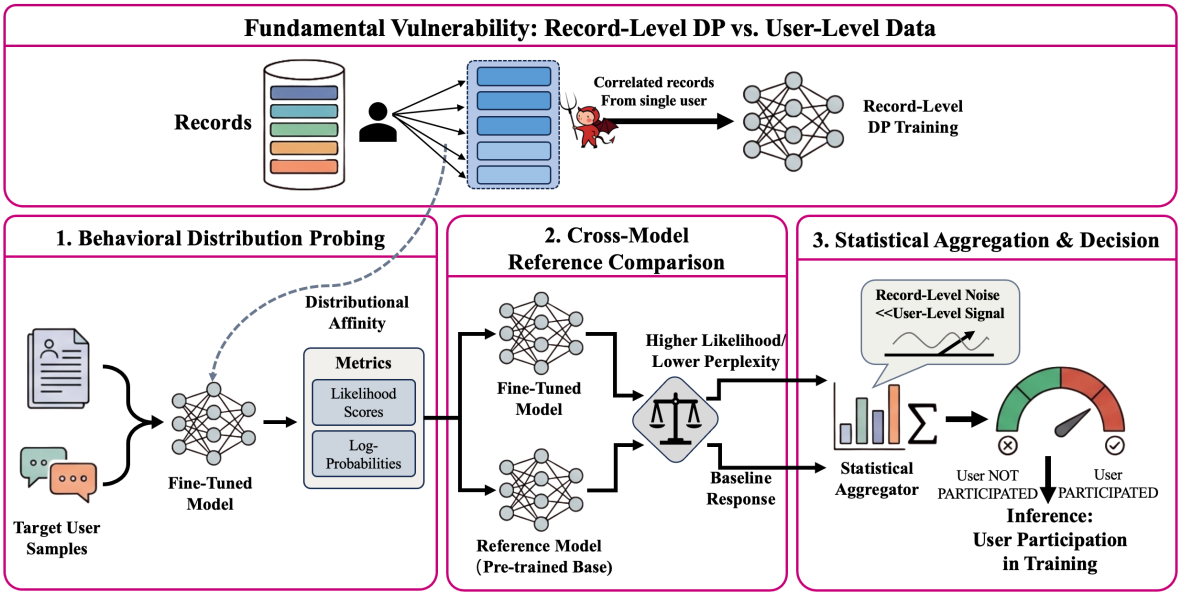

Figure 6. User inference attacks on DP-DL systems. Attackers probe the model with target user samples, compare outputs with a reference model, and aggregate likelihood signals to infer whether the user participated in training despite record-level DP protection.

4.2. Algorithm-Level Attacks

Algorithm-level attacks exploit mismatches between the idealized assumptions of differential privacy mechanisms and their concrete algorithmic implementations. By exploiting techniques such as repeated queries or floating-point attacks, an attacker can systematically weaken or eliminate the intended randomness. These attacks highlight that the practical protection of differential privacy not only depends on formal privacy guarantees, but also on careful algorithmic design and robust implementation details.

4.2.1. Repeated-Query Attacks on DP-DL Systems

When repeated or semantically equivalent queries are treated as independent interactions without proper privacy accounting, an adversary can accumulate multiple noisy observations of the same

 doi:10.20944/preprints202603.1179.v1

underlying quantity. This creates an opportunity for adversaries to mount a repeated-query attack through statistical aggregation or distributional inference. Figure 7 shows how.

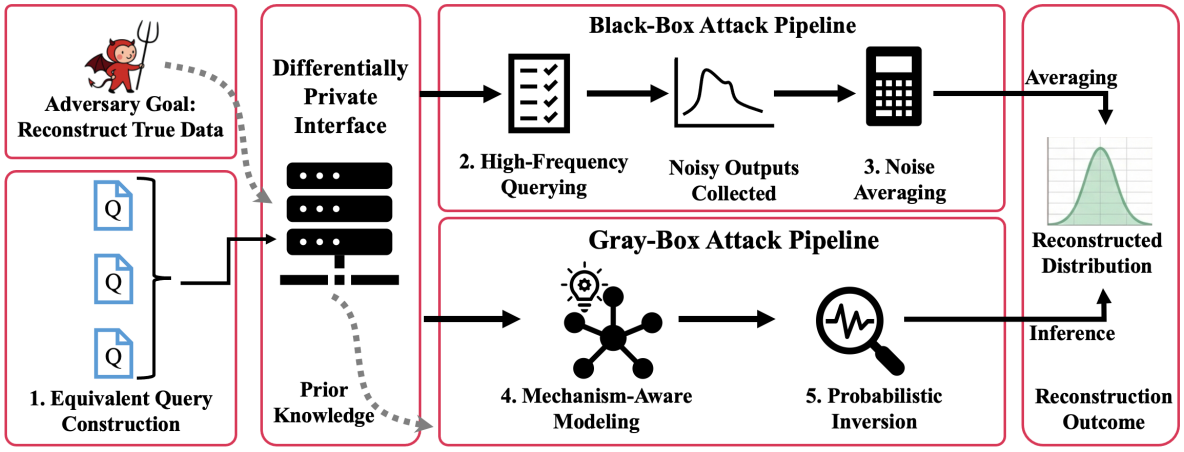

Figure 7. Repeated-query attacks on DP-DL systems. Attackers issue multiple equivalent queries to a DP interface and collect noisy outputs. In the black-box setting, statistical averaging over repeated queries reduces noise and reveals the underlying data distribution. In the gray-box setting, partial knowledge of the DP mechanism enables mechanism-aware modeling and probabilistic inversion for more precise reconstruction.

Adversaries typically execute such attacks in a black-box setting through the following sequence of steps [63]. (1) Constructing equivalent queries: The adversary designs multiple queries that either share identical structures, involve identical sets of data contributors, or have equivalent logical semantics. This ensures each query targets the same true underlying value; thus, only the random noise component will differ for each response. (2) Repeat the query as frequently as possible: Within the limits of the system's static privacy budget allocation, the adversary repeatedly submits these equivalent queries and collects a large set of noisy outputs or probability distributions. (3) Averaging and reconstructing the noise: The adversary then applies statistical processing, such as simple averaging or more sophisticated distribution fitting, to the collected results to reconstruct the true data distribution.

In gray-box settings, the adversary can further exploit partial knowledge of the underlying mechanism through the following additional steps [6,7]. (4) Mechanism-aware output modeling: Leveraging prior knowledge of the aggregation rule, noise distribution, or privacy parameters, the adversary explicitly models the stochastic response mechanism and characterizes the conditional output distribution induced by a fixed latent quantity. (5) Probabilistic inversion and parameter inference: Based on the modeled response distribution, the adversary applies analytical inversion or optimization-based estimation techniques to infer the latent internal state (e.g., vote histograms or model parameters) with higher precision than what can be achieved through black-box averaging.

This attack exploits a fundamental structural weakness in systems with static privacy budget allocation: the number of queries can grow indefinitely, while the privacy budget does not scale accordingly, creating a universal threat to such DP-DL systems.

4.2.2. Floating-Point Attacks on DP-DL Systems

Floating-point attacks exploit the discrepancy between the ideal noise distributions assumed in differential privacy theory and their finite-precision implementation in hardware. In practice, Laplace and Gaussian noise are generated by sampling uniformly random bits, mapping them to discretely representable floating-point values, and applying non-linear transformations such as logarithms and square roots. Finite mantissa precision and rounding cause these transformations to collapse many inputs onto the same representable values, so the resulting noise distribution is supported on a discrete subset of the real line rather than being continuous. When this noise is added to a query result, the support of the released output distribution is obtained by translating this discrete set by the true query value. For neighboring datasets with different true outputs, these translated supports may partially or

 doi:10.20944/preprints202603.1179.v1

entirely fail to overlap. Consequently, there exist output events that occur with nonzero probability under one dataset but with zero probability under a neighboring dataset, which violates the defining condition of (ε, δ)-DP unless the resulting probability mass is bounded by δ [52].

The adversary's objective is to determine which of two neighboring datasets was used to produce an observed differentially private output. The attacker is assumed to know the query function, the privacy parameters, and the specific floating-point implementation of the noise generator, but not its internal randomness [8]. Given this knowledge, the attack proceeds as shown in Figure 8. (1) Candidate hypothesis enumeration: The adversary enumerates candidate neighboring datasets and their corresponding true query values as hypotheses. (2) Noise feasibility analysis: For each candidate, the adversary subtracts the hypothesized true value from the observed output to derive an implied noise sample. This sample is then checked for feasibility against the constraints of the concrete floating-point sampler; for example, verifying whether it could plausibly result from a floating-point implementation of the Laplace mechanism's logarithmic transform. (3) Dataset inference: If the implied noise is infeasible for one candidate hypothesis but remains feasible for another, the adversary can definitively rule out the former. This allows for certain inference of the true dataset, thereby breaking the intended differential privacy guarantee in practical computational environments.

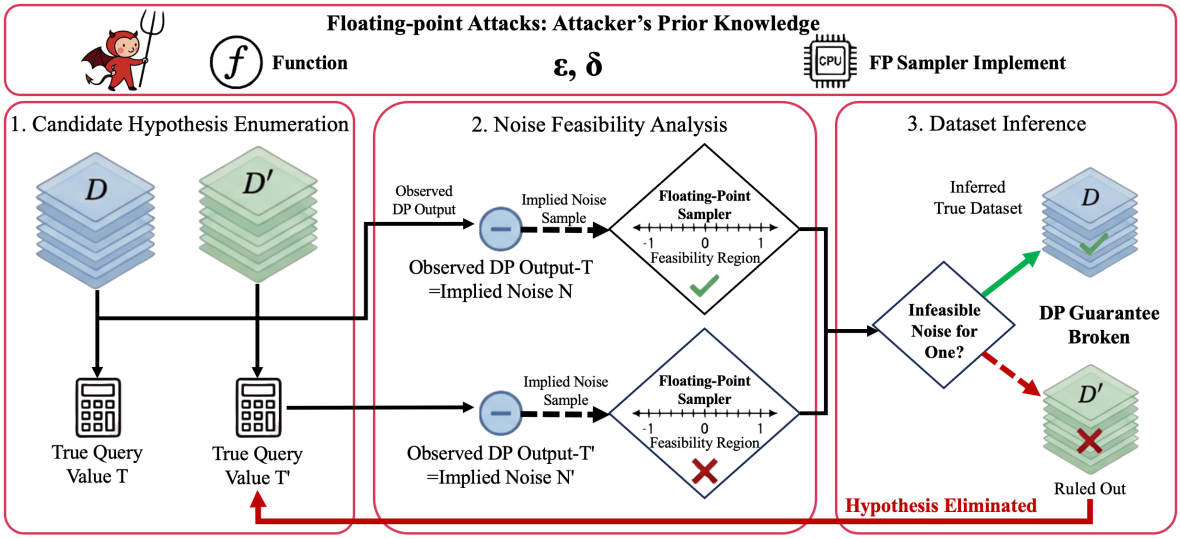

Figure 8. Floating-point attacks on DP-DL systems. Attackers infer the true dataset by testing whether the noise implied by DP outputs falls within the feasible range of the floating-point noise sampler. Hypotheses that produce infeasible noise values are eliminated, allowing the true dataset to be identified.

4.3. Architecture-Level Attacks

Architecture-level attacks exploit properties of model architectures and training pipelines to recover sensitive information despite any guarantee of differential privacy. More specifically, membership inference attacks leverage residual statistical differences in model outputs or losses to determine whether a sample was included in the training data, while gradient attacks reconstruct the original training data by exploiting the fact that federated learning settings typically share or clip the gradients. Critically, these attacks highlight that practical privacy risks in DP-DL systems not only depend on the privacy budget but also on design choices made at the architectural level.

4.3.1. Membership Inference Attacks on DP-DL Systems

Theoretically, differential privacy provides a worst-case (ε, δ)-DP guarantee that bounds the distinguishability of neighboring datasets. However, in the real world, information leaks are highly dependent on the specific scenario. They can be influenced by a myriad of factors, including the model's architecture, the particular training procedures, an adversary's prior knowledge, and so on. Consequently, if the privacy budget is not sufficiently tight, membership inference attacks will likely be successful [74] (see Figure 9).

 doi:10.20944/preprints202603.1179.v1

In black-box settings, the attacker can only observe the model's prediction outputs, but residual statistical differences in the output distribution can still be inferred. Jayaraman et al. [53], for example, used the framework based on shadow models from [64] to train an attack classifier by comparing the output-vector distributions of "member/non-member" samples. This allowed them to perform black-box membership inference on a DP-DL system. Likewise, Riaz et al. [65] proposed an even weaker black-box variant in the DP-BCD setting. Here, they did not need to construct a shadow model. Instead, the attack features were derived directly from the output predictions, which still delivered an inference success rate better than random guessing.

Under white-box conditions, attackers can access the model's parameters or its training statistics, permitting the use of more fine-grained signals to improve inference accuracy. In this vein, Yeom et al. [66] inferred membership by checking whether a sample's loss was lower than the average training loss. If significantly lower, they inferred the sample to be in the training set; otherwise, they treated it as a non-member. Jayaraman et al. [53] also used this method – this time to evaluate multiple differential privacy training mechanisms. Their tests show that DP-DL systems do not typically meet their theoretical privacy guarantees. Lastly, Riaz et al. [65] proposed a white-box attack designed to penetrate DP-BCD models, where attackers could access the full model architecture and therefore construct shadow models. This, of course, further bolstered the success of the membership inference attacks they launched.

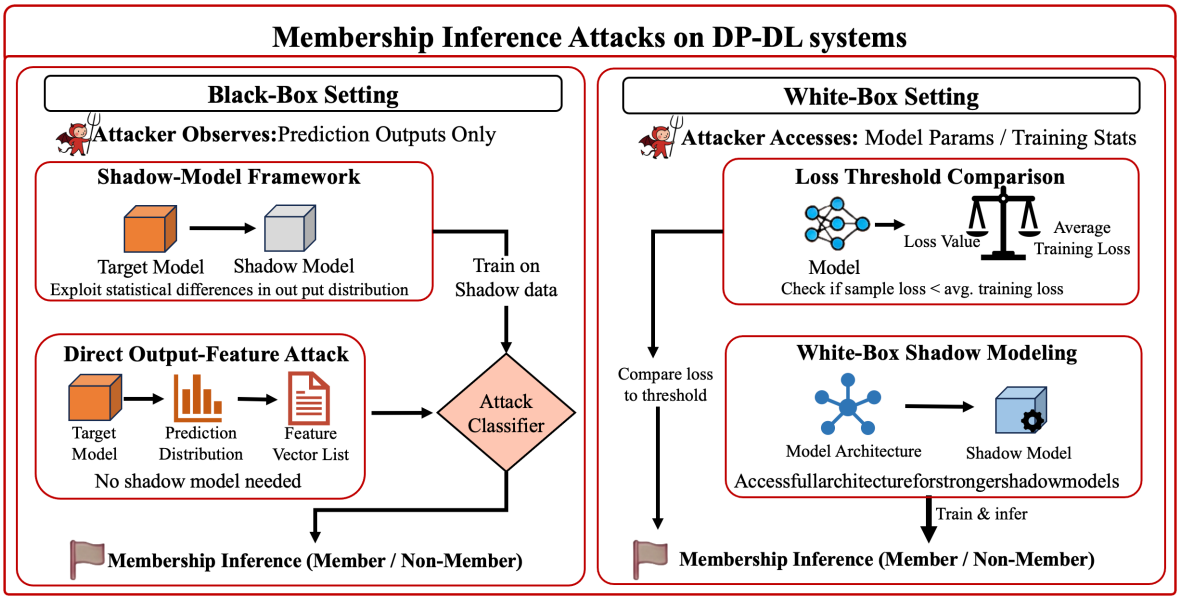

Figure 9. Membership inference attacks on DP-DL systems. Attackers determine whether a sample was used in training. In the black-box setting, prediction outputs are analyzed using shadow models or output features. In the white-box setting, access to model parameters enables loss-based or shadow-model attacks for membership inference.

4.3.2. Gradient Inversion Attacks on DP-DL Systems

In federated learning, deep models often need to loosen the clipping parameters and/or reduce noise to maintain reasonable accuracy. However, this tends to have the negative effect of increasing the quality with which an adversary can reconstruct the model's gradient. Gradient inversion attacks, such as DLG [75], InvertGrad [76], and IG [77] are white-box attacks. The adversary analyzes the gradients shared during training and uses techniques like gradient inversion to reconstruct the original training samples (as shown in Figure 10). Studies have shown that clipping strategies, such as layer-wise clipping and dynamic thresholds, often play a greater role than noise in determining whether a gradient inversion attack is effective [58].

 doi:10.20944/preprints202603.1179.v1

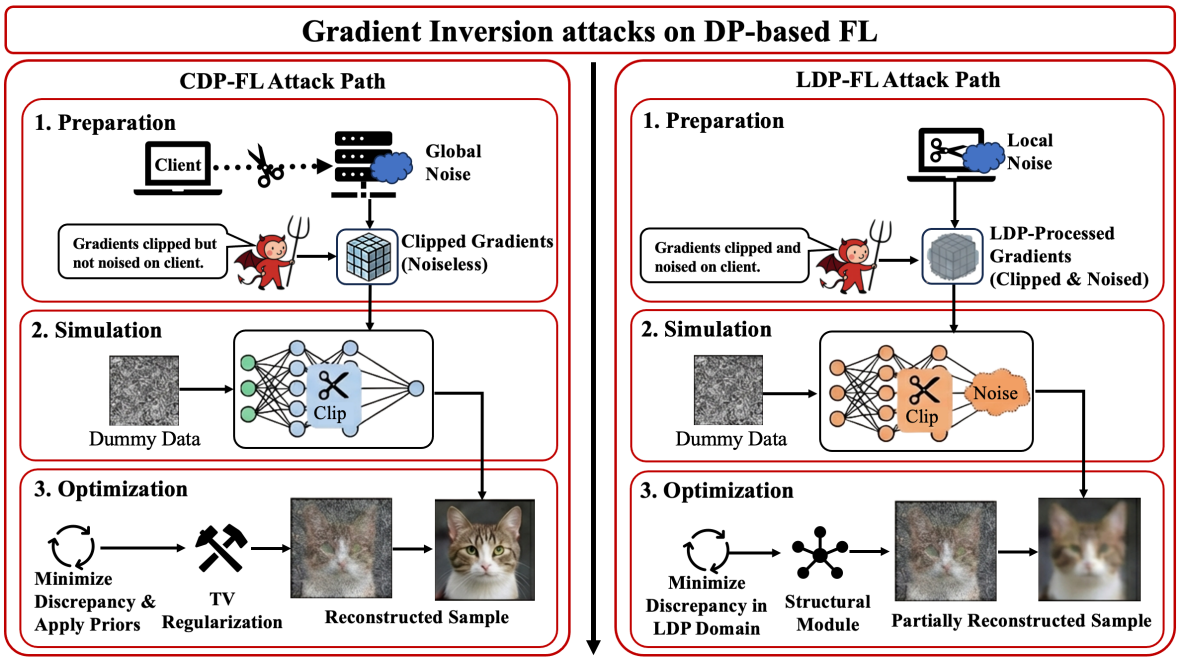

Figure 10. Gradient inversion attacks on DP-based federated learning systems. Attackers reconstruct training data from shared gradients. In CDP-FL, clipped but unnoised gradients enable accurate reconstruction through simulation and optimization. In LDP-FL, attackers model the local clipping and noise process to iteratively reconstruct partial data from perturbed gradients.

CDP-based federated learning systems provide privacy protection through gradient clipping and adding noise server-side. The core idea is to inject global noise during aggregation to limit the influence of any single client update on the global model. However, attackers can directly obtain the unadulterated gradient and then perform a gradient inversion attack. Attacks on CDP-based FL systems generally consist of three key stages [58,67]. (1) In the preparation stage, the attacker intercepts the clipped gradient updates uploaded by the client. These clipped gradients become the main signals that leak private information. (2) In the simulation stage, the attacker performs clipping-aware gradient computations on randomly initialized dummy inputs. This means strictly simulating the client's clipping mechanism during both the forward and backward passes to ensure that the simulated gradients lie in the same clipping domain as the real gradients. (3) In the optimization stage, the attacker minimizes the difference between the simulated gradients and the real gradients within this clipping domain. The adversary then further applies image priors or structural enhancement techniques, like TV regularization, to optimize the reconstruction. This process gradually yields input samples that closely resemble the real data.

LDP-based FL provides protection by clipping gradients and adding noise locally on the client side. The idea is to perturb each client's update before uploading, which stops the server from directly accessing the original gradients. However, clipping and noise cannot fundamentally eliminate the structural correlations between the gradients and the data. This means attackers can still reconstruct some of the original samples using gradient inversion techniques. Again, these attacks comprise three primary stages [57,58,68]. (1) In the preparation stage, the attacker intercepts the LDP gradient updates uploaded by the target client, which have already been clipped and perturbed with noise. (2) In the simulation stage, the attacker performs gradient computations on randomly initialized dummy inputs, strictly reproducing the client's LDP processing pipeline, including the same gradient clipping scheme and noise perturbation model. This ensures that the simulated gradients and the real gradients lie in the same clipping domain. (3) In the optimization stage, the attacker minimizes the differences between the simulated and real gradients within the LDP-constrained gradient domain, while incorporating image priors, structured reconstruction modules, and regularization strategies. As this process continues, the dummy inputs gradually begin to match the client's real data.

 doi:10.20944/preprints202603.1179.v1

4.4. Cross-Level Attacks on DP-DL Systems: Poisoning Attacks

In recent years, poisoning attacks against DP-DL systems have attracted significant attention, with most studies investigating these attacks from two perspectives: data poisoning and model poisoning (see Figure 11). In general, they systematically analyze how adversaries operating under different knowledge assumptions can degrade model utility or manipulate model behavior without noticeably disrupting the training process [70]. Additionally, many of these articles show that, although differential privacy mechanisms introduce noise and gradient clipping to protect privacy, they may also create opportunities for stealthy and effective poisoning attacks. Table 2 summarizes poisoning attacks, along with their key mechanisms and descriptions.

4.4.1. Data Poisoning Attacks on DP-DL Systems

As an example, Jagielski et al. [9] demonstrate a gray-box data poisoning attack that audits the privacy of DP-SGD in practical terms. The attacker is assumed to know the training algorithm, the model's architecture, and its hyperparameters, but learning can only be influenced through carefully crafted training samples. By inserting a small number of poisoning points aligned with low-variance gradient directions, the attack remains effective despite clipping gradients and injecting noise, yielding empirical lower bounds on the privacy parameter ε. Adversaries can therefore use statistics to distinguish between neighboring datasets.

Alternatively, Peng et al. [69] propose a black-box attack based on label manipulation. The attacker strictly follows the training protocol and does not directly forge gradients nor disrupt the differential privacy mechanism. Instead, the adversary alters local data labels to change the direction of the loss-function gradient. The study involves two concrete techniques: static label flipping, which maps the original labels to predetermined incorrect classes, and dynamic label flipping, which selects the least likely class under the current global model to maximize gradient perturbation. Both methods influence the aggregation results by altering the direction of the loss gradient without relying on any internal information from the model.

4.4.2. Model Poisoning Attacks on DP-DL Systems

These attacks typically leverage the attacker's knowledge of the model's architecture, its training process, and its aggregation rules. Combined with the increased update dispersion caused by differentially private noise, attackers can craft malicious gradients that are difficult to detect. In gray-box settings, Wang et al. [71] proposed three different methods to achieve fine-grained control over uploaded gradients: LLRA, TMMA, and AdvPA. LLRA modifies local gradients through reverse optimization so as to maximize the training loss. TMMA is derived via analytical updates based on the number of clients and the aggregation formula, thereby steering the global model toward attacker-specified targets. AdvPA further introduces distributional similarity constraints, ensuring that malicious gradients can still be accepted by robust aggregators, such as Multi Krum and Trimmed Mean. By contrast, Zheng et al. [72] employed genetic algorithms to search for noise that satisfies differential privacy distribution constraints, making malicious updates statistically consistent with benign ones. This enabled them to bypass robust aggregation and conduct a stealthy poisoning attack. Yang et al. [73] disrupted or degraded the global model across multiple training rounds by applying position permutation and scale transformation to the model's parameters.

 doi:10.20944/preprints202603.1179.v1

Table 2. A summary of poisoning attacks.

Attack Type	Literature	Key Mechanism	Description
Data Poisoning	Jagielski et al. [9]	Low-variance gradient poisoning	Insert poisoning samples along low-variance gradients
	Peng et al. [69]	Static/Dynamic label flipping	Manipulate labels to perturb loss gradients
Model Poisoning	Wang et al. [71]	LLRA / TMMA / AdvPA gradient control	Craft malicious gradients to steer the global model
	Zheng et al. [72]	Consistent malicious updates	Generate DP-consistent noise via genetic algorithms for stealthy malicious updates
	Yang et al. [73]	Parameter permutation and scaling	Apply parameter permutation and scaling transformations to degrade the global model

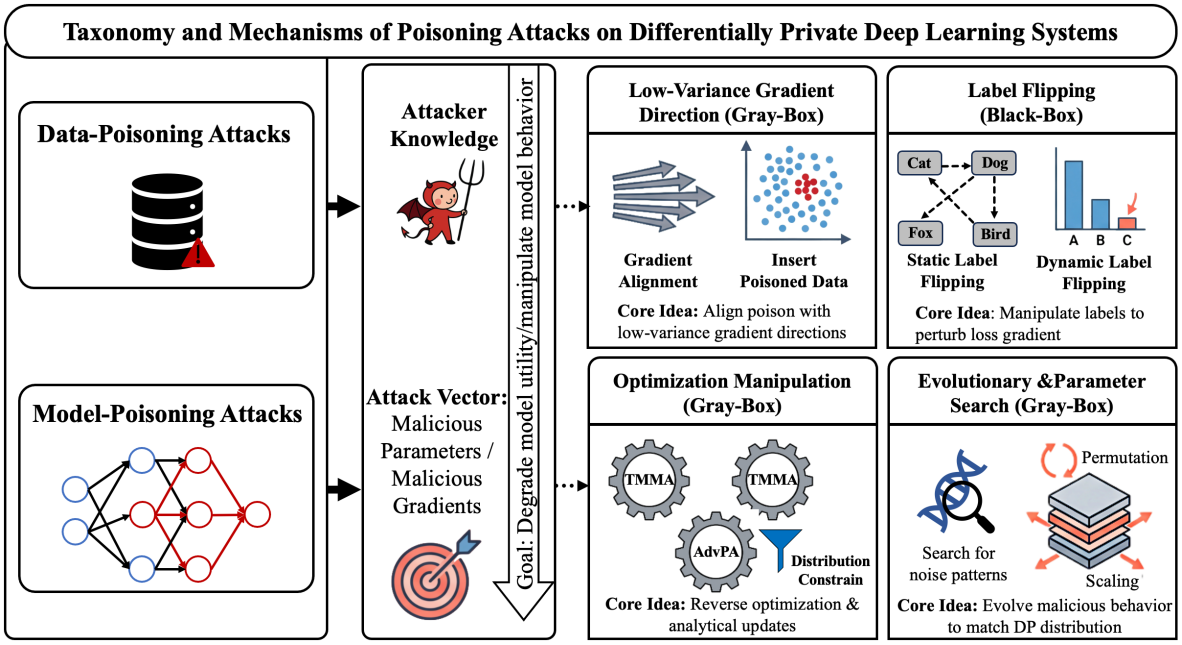

Figure 11. Poisoning attacks on DP-DL systems. Poisoning attacks include data poisoning and model poisoning. Data poisoning alters training samples or labels to perturb gradient directions. Model poisoning injects malicious gradients or parameters through optimization or search strategies to manipulate model behavior under DP noise.

5. Defenses for DP-DL Systems

DP-DL systems face a broad spectrum of privacy and security threats that arise from data correlations, algorithmic interactions, numerical implementations, and model architectures. To counter these risks, existing defenses operate at different layers of the learning pipeline, ranging from data preprocessing and privacy mechanism design to architectural and optimization choices (see Table 3). This section provides a systematic taxonomy of defense strategies for DP-DL systems, organized again by level. In each category, we analyze how the available defenses target specific attack vectors, their underlying mechanisms, and the practical trade-offs they introduce between privacy protection, model utility, robustness, and system efficiency.

 doi:10.20944/preprints202603.1179.v1

Table 3. Comparison of defense techniques.

Defense Techniques	Literature	Targeted Threats	Protection Capability
Dimensionality reduction	[78,79]	Attribute inference attacks	This method removes some identifiable features, but attackers can still infer attributes from the remaining information.
Reconstructing correlations	[80]		This method weakens the correlation between sensitive attributes and gradients or representations.
User-level DP	[81–83]	User inference attacks	A correctly implemented user-level DP mechanism provides strong theoretical privacy guarantees.
Dynamic privacy budget allocation	[7]	Repeated-query attacks	Dynamic allocation improves robustness, but poor design may waste the privacy budget.
Hard limits on repeated queries	[47,84]		Hard limits block frequent probing queries but reduce usability and performance.
Discretization-based defenses	[8,52]	Floating-point attacks	Discretization reduces floating-point side-channel leakage but cannot eliminate all leakage paths.
Continuous-distribution defenses	[12,85]		This method preserves distribution properties and suppresses floating-point exploitation more effectively.
DP-friendly architecture choice	[37,86,87]	Membership inference attacks	DP-friendly architectures reduce overfitting and memorization but do not directly guarantee privacy.
Empirical privacy auditing	[13,14,54]		Auditing identifies privacy risks but does not directly provide protection.
Clipping-centric defenses	[88–91]	Gradient inversion attacks	Gradient clipping limits gradient magnitude but cannot fully prevent information recovery.
Perturbation-centric defenses	[92–95]		Noise injection reduces the recoverability of sensitive information.
Robust aggregation enhancement	[96–98]	Poisoning attacks	Robust aggregation rules reduces the impact of malicious updates.
Client variance reduction	[99,100]		Variance reduction stabilizes updates but may not stop strategic poisoning.

5.1. Data-Level Defenses

Data-level defenses aim to mitigate privacy leaks by intervening directly in the data representation or generation process before a model is trained. By modifying correlations, distributions, or granularity at the data level, these methods seek to reduce the exploitable signals available to inference attackers while preserving task-relevant utility. Such defenses are particularly effective against attacks that rely on the statistical dependencies inherent in raw or preprocessed datasets.

5.1.1. Defenses Against Attribute Inference Attacks

Attribute inference attacks on high-dimensional data exploit correlations between attributes, so defenders can intervene at the data preprocessing or generation stage to handle such correlations (see Figure 12).

 doi:10.20944/preprints202603.1179.v1

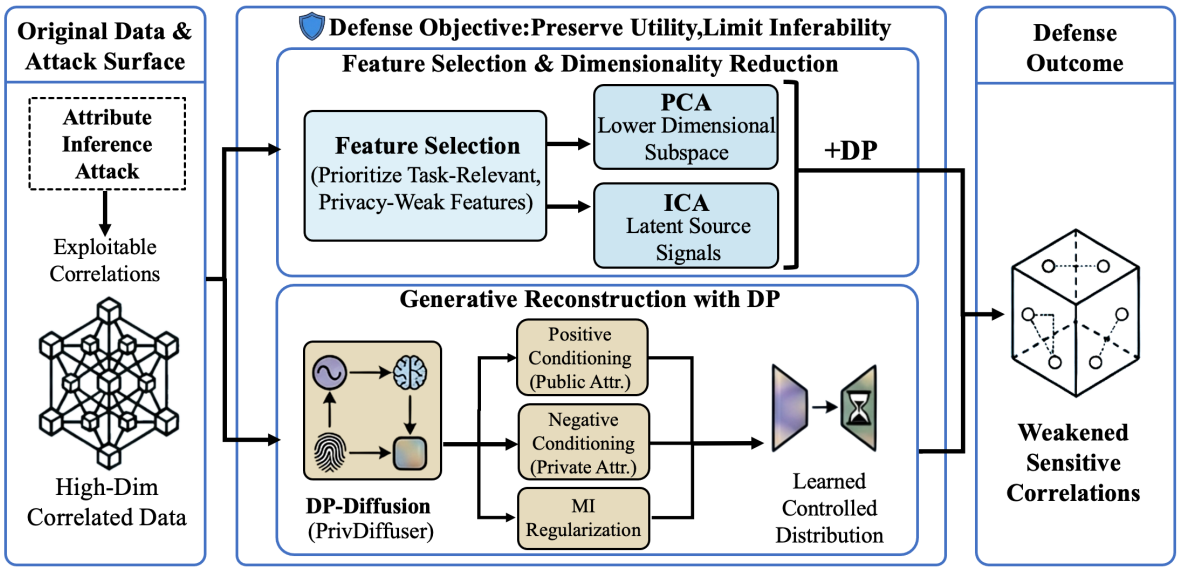

Figure 12. Defenses against attribute inference in DP-DL. Defenses weaken exploitable correlations in high-dimensional data through feature selection and dimensionality reduction, or through DP-based generative models that learn controlled distributions with reduced sensitive correlations.

Dimensionality reduction restricts correlation. During preprocessing, dimensionality-reduction techniques like principal component analysis (PCA) [78] or independent component analysis (ICA) [79] can map data into a lower-dimensional subspace, preserving information relevant to the main task while weakening sensitive correlations. To better exploit these techniques, feature selection [101,102] further optimizes the correlation structure by prioritizing features that are strongly correlated with the main task label but weakly correlated with sensitive attributes using measures such as linear regression or mutual information. PCA identifies major variance directions through the covariance structure, achieving preliminary decorrelation but only providing linear decorrelation for non-Gaussian data. On the other hand, ICA uses higher-order statistics to pursue statistical independence and further separate latent source signals, making it suitable for more complex correlations. Applying differential privacy mechanisms in the lower-dimensional subspace can better prevent attribute inference attacks while preserving model utility. However, it is important to emphasize that decorrelation does not completely eliminate feature dependencies. Rather, it selectively weakens the prominent correlations between non-sensitive and sensitive attributes. Notably, these are most exploitable by attackers.

Reconstructing correlations via generative models. Generative models protected with differential privacy, such as DP-GAN [103], DP-VAE [104], and diffusion models [80], can be used as a preprocessing step to learn a controlled distribution from the original data. Hence, one can generate synthetic data to strike a balance between preserving correlations and protecting privacy. Additionally, as the data are being generated, the joint distribution between sensitive attributes and other features can be directly modulated, thereby actively limiting the correlations that attackers can exploit. Along these lines, PrivDiffuser [80] introduces positive conditioning for public attributes and negative conditioning for private ones during diffusion. This, combined with mutual-information regularization, decouples the two attribute types. This approach preserves the public attribute information but actively decreases how easy it is to infer the data's private attributes. This approach both preserves any public attributes, while also actively making it more difficult to infer private attributes from the generated data. Experiments show that diffusion models (e.g., TabDDPM) outperform GANs/VAEs in capturing complex correlations.

In summary, dimensionality reduction and differentially private generative reconstruction mitigate attribute inference attacks by weakening exploitable correlations. However, they also introduce trade-offs in utility, robustness, and deployability. Modifying correlation structures can reduce representational capacity when sensitive attributes are strongly linked to the main task, so performance degrades as a result. Researchers also see weaker robustness under distribution shifts and potential

 doi:10.20944/preprints202603.1179.v1

distortions in fairness due to proxy features or information loss. Moreover, DP-based generative models tend to have unstable training processes and a higher computational overhead, while synthetic data can suffer from distributional distortions and compliance concerns. All this highlights the need to balance privacy protection with the system's overall reliability and efficiency.

5.1.2. Defenses Against User Inference Attacks

Record-level privacy attacks exploit fine-grained signals leaked by models, suggesting that differential privacy at the example level may not offer a sufficient guarantee of protection. Naturally, this has motivated stronger user-level protection against inference on correlated records [83] as a more stringent and practically-relevant defense paradigm. User-level differential privacy guarantees that replacing all the records contributed by a single user will not significantly change the model's output [81]. Mechanistically, user-level differential privacy treats a user's entire contribution as the unit of privacy loss accounting. Thus, noise is amortized across all correlated records, which blocks record-level inferences formed from aggregating multiple examples along with other behavioral correlations. See Figure 13 for an illustration.

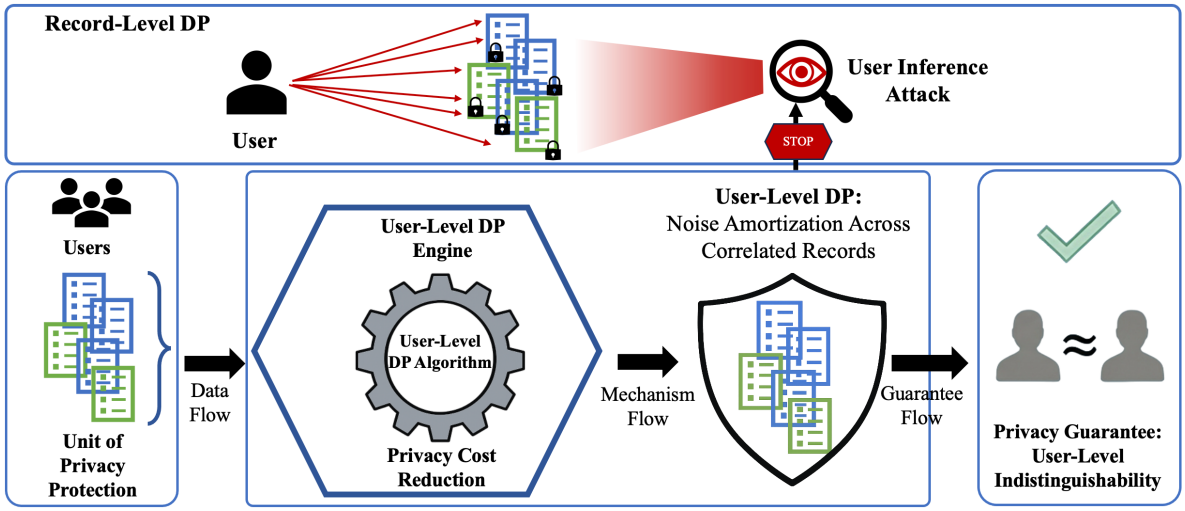

Figure 13. Defenses against user inference attacks in DP-DL. User-level DP treats all records of a user as a single privacy unit and distributes noise across them, mitigating inference from correlated records.

Levy et al. [81] formalize user-level differentially private learning algorithms in which each user contributes m samples, where m denotes the number of records associated with a single user. They show that the privacy cost decreases as $1/\sqrt{m}$ as the number of samples per user increases, establishing that the privacy overhead shrinks with intra-user redundancy and directly weakens record-level leakage signals. Ghazi et al. [82] further provide generic transformations that convert item-level differential privacy algorithms into user-level ones, achieving multiplicative savings of $\tilde{O}(\sqrt{m})$ in required users for the same utility, which makes user-level protection attainable even when per-user samples are scarce. Finally, recent LLM fine-tuning work empirically validates that user-level differential privacy mitigates user-level membership inference and the risk of data leaks that persist under example-level DP. These techniques have great practical relevance in real-world deployments involving correlated per-user examples [83]. Functionally, these defenses convert 'record-level' privacy threats into guarantees of 'user-level indistinguishability', closing gaps where adversaries have exploited behavioral, linguistic, or distributional consistency across records. Collectively, they demonstrate that upgrading the granularity of differential privacy from items to users can form an effective and principled defense line against record-level privacy inferences, while retaining statistical utility for learning tasks.

In summary, user-level differential privacy provides one of the strongest principled defenses against record-level inference when correctly implemented, offering robust protection given correlated

 doi:10.20944/preprints202603.1179.v1

user data. By aligning privacy guarantees with user-level data generation and enabling privacy amplification across records, it forms a practical foundation for protecting fine-grained privacy in deployed systems.

5.2. Algorithm-Level Defenses

Algorithm-level defenses aim to mitigate privacy attacks by modifying the design and execution of differential privacy mechanisms themselves. By redesigning interaction protocols or noise mechanisms, these methods address vulnerabilities that arise during model execution and deployment. Such defenses are particularly effective against adaptive adversaries who exploit repeated interactions or numerical artifacts.

5.2.1. Defenses Against Repeated-Query Attacks

The essence of repeated-query attacks lies in the attacker's ability to issue the same or highly correlated queries multiple times and statistically average the responses perturbed with identically distributed noise. This approach has the effect of cancelling out the noise and approximating the sample's true value. To counter this, a dynamic privacy budget checks the query history and progressively increases the amount of noise introduced, breaking the assumption that repeated responses follow the same distribution. By contrast, hard limits on repeated queries track cumulative privacy loss and cap the number of allowable queries, directly preventing noise averaging and the adversary from accumulating excessive information (see Figure 14).

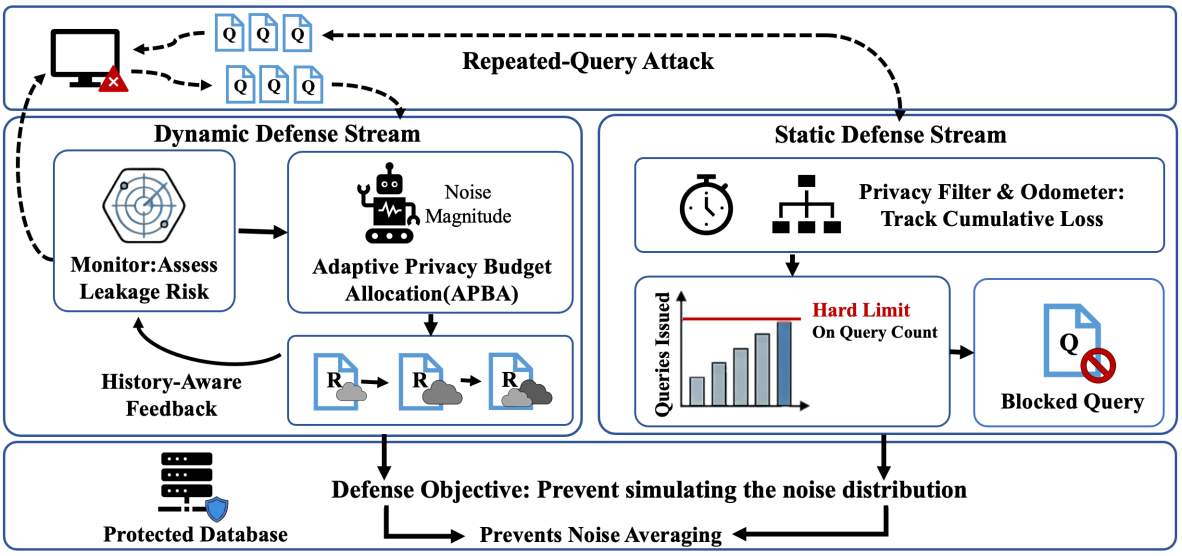

Figure 14. Defenses against repeated-query attacks in DP-DL. The framework prevents noise averaging from repeated or correlated queries. Dynamic defenses monitor query history and adaptively increase noise through privacy budget allocation. Static defenses track cumulative privacy loss using privacy filters or odometers and enforce hard limits on query counts.

Dynamic privacy budget allocation. To defend against model extraction attacks driven by query-flooding, Yan et al. [7] proposed a Monitoring-based Differential Privacy (MDP) framework together with Adaptive Privacy Budget Allocation (APBA). As mentioned above, this strategy takes a history-aware approach to managing privacy budgets. The key vulnerability targeted by this design is that repeated or semantically equivalent queries are treated as independent interactions, resulting in identically distributed noise that can be statistically averaged out by an adversary. MDP addresses this issue by explicitly incorporating query history into the privacy mechanism. A monitor module evaluates the potential for a privacy leak with each incoming query in real time from an information-theoretic perspective. Hence, the system learns to recognize when similar or increasingly risky queries are being issued. Based on this assessment, APBA adaptively adjusts the response behavior by

 doi:10.20944/preprints202603.1179.v1

reducing the allocated budget and increasing the amount of injected noise as the perceived risk of a leak grows. Crucially, this process breaks the assumption that repeated responses are drawn from the same noise distribution. By escalating the strength of noise for repeated or correlated queries, the system prevents attackers from acquiring multiple low-noise observations of the same underlying quantity. Consequently, statistical averaging and variance reduction techniques become useless.

Hard limits on repeated queries. Some of the earliest and most general runtime budget-control mechanisms for interactive differential privacy were introduced by Rogers et al. [84], who proposed privacy filters and privacy odometers. These mechanisms limit the frequency with which a user can query a model by tracking the cumulative privacy loss. A privacy filter then halts service once the consumed budget approaches a predetermined threshold, preventing further high-frequency or repeated queries. A privacy odometer records the privacy cost consumed so far, allowing the system to dynamically assess the remaining budget. By bounding the cumulative privacy loss, these mechanisms directly restrict noise-averaging subterfuge, while providing a system-level safeguard against flood-style attacks. Building on this system perspective, Li et al. [47] further analyzed how much information repeated linear queries can reveal and constructed the "most aggressive linear query" to characterize the worst-case leak. Their analysis yields a strict upper bound on the number of safe repeated queries: once this limit is exceeded, even differentially private noise cannot prevent attackers from exploiting the concentration of repeated outputs to approximate the true value. In effect, this provides a theoretical foundation for enforcing hard limits on repeated queries in practical scenarios.

In practice, defenses against repeated-query attacks introduce trade-offs between privacy protection, response accuracy, and system availability. Dynamic budget allocation improves resilience to noise averaging but requires a mechanism that can detect similar queries in real-time, as well as stateful budget tracking. Each increases both complexity and latency. Similarly, hard query limits offer strong and predictable protection with low runtime overhead, but may reduce usability for benign high-frequency users, highlighting the need to balance robustness with efficiency and service quality.

5.2.2. Defenses Against Floating-Point Attacks

Floating-point attacks exploit the discrepancy between the continuous noise distributions assumed in differential privacy theory and their finite-precision implementation, which can create "support gaps" that violate indistinguishability guarantees. Defenses against such attacks address this opportunity by modifying the noise generation process itself. Table 4 summarizes representative defenses and their key mechanisms.

Table 4. A summary of defenses against floating-point attacks.

Defense Type	Literature	Key Mechanism	Description
Discretization-based	Mironov et al. [52]	Snapping mechanism	Discretize outputs via lattice rounding and clipping
	Jin et al. [8]	Discrete Laplace/Gaussian	Generate integer-valued noise distributions
Continuous-based	Haney et al. [12]	Interval refining	Refine sampling intervals for consistent float rounding
	Holohan et al. [85]	Mantissa Bit Manipulation	Modify mantissa bits to remove leakage patterns

Discretization-based defenses. Mironov et al. [52] proposed the snapping mechanism, which rounds noisy outputs to a predetermined discrete lattice and clips them to a bounded interval. This ensures that adjacent inputs share the same discrete support, guaranteeing (ε, δ)-DP under floating-point arithmetic. Similarly, Jin et al. [8] studied discrete Laplace and discrete Gaussian mechanisms that operate directly over integer-valued noise distributions. These eliminate the need for floating-point inverse-CDF sampling and yield well-defined probability mass functions over representable domains.

Continuous-distribution defenses. Other approaches aim to approximate continuous noise distributions within floating-point space without fully discretizing them. Haney et al. [12] introduced

 doi:10.20944/preprints202603.1179.v1

interval refining, an inverse-transform sampling algorithm that iteratively refines probability intervals until all mapped values round to the same floating-point number. This makes the final output statistically equivalent to sampling from a continuous distribution followed by rounding, thereby eliminating support holes. Holohan et al. [85] proposed Mantissa Bit Manipulation (MBM), which post-processes floating-point noise samples by selectively modifying mantissa bits. This suppresses location-dependent leakage while preserving the overall statistical shape of the target distribution.

In summary, defenses against floating-point attacks trade numerical fidelity and computational efficiency for stronger implementation-level privacy guarantees. Discretization-based mechanisms provide clear and robust DP guarantees under finite precision but may introduce quantization error and reduced output granularity. Continuous-distribution defenses reduce floating-point leakage but introduce additional computational complexity and implementation overhead, making them harder to deploy efficiently in practice. These considerations highlight that securing DP implementations against floating-point leakage requires balancing theoretical correctness, numerical accuracy, and system efficiency in real-world deployments.

5.3. Architecture-Level Defenses

Architecture-level defenses aim to mitigate privacy leaks by modifying the structural properties of learning systems rather than relying solely on post-hoc protections. By shaping gradient behavior, optimization dynamics, and information flow within models, these approaches reduce the attack surface exposed to adversaries.

5.3.1. Defenses Against Membership Inference Attacks

To defend against membership inference attacks, existing studies have developed two main lines of techniques: one improves the training mechanism itself by refining DP-SGD or optimizing the model's architecture to reduce per-step sensitivity; the other relies on empirical auditing, using attack simulations to estimate real privacy loss and reveal implementation errors or latent risks, as shown in Figure 15.

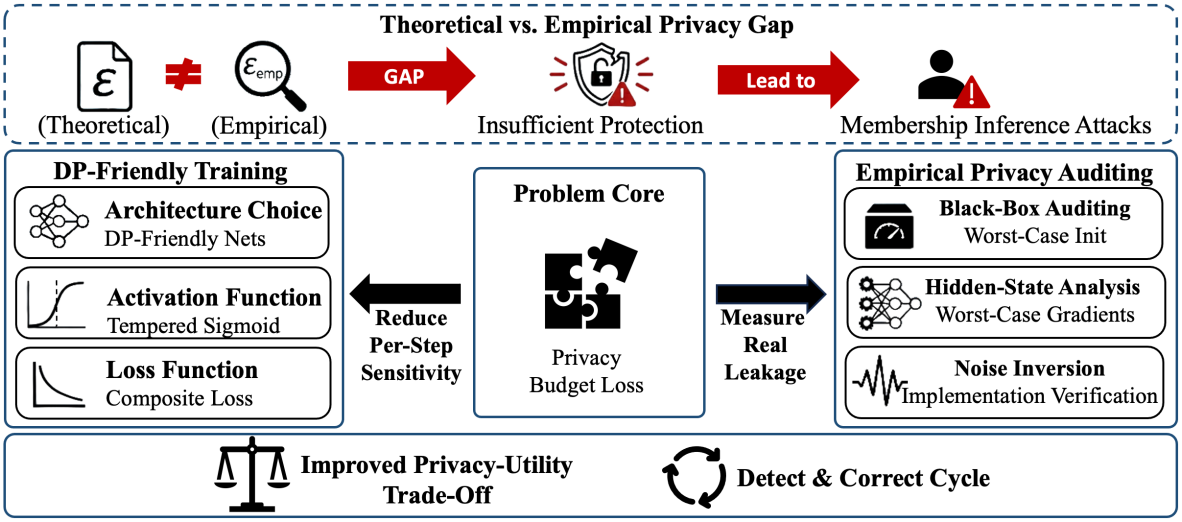

Figure 15. Defenses against membership inference in DP-DL. Defenses address the gap between theoretical and empirical privacy. DP-friendly training reduces per-step sensitivity through architecture, activation, and loss design. Empirical privacy auditing evaluates real leakage using attack simulations and implementation checks.

DP-friendly architecture choice. In DP-SGD, gradient clipping controls the per-step sensitivity and the amount of noise injected is calibrated to this sensitivity. Therefore, reducing the sensitivity per-step decreases the amount of injected noise required, which in turn serves as a key approach to alleviating any overall loss of privacy budget [105]. Cheng et al. [37] systematically demonstrate that different network structures markedly affect gradient distributions and clipping behavior, showing that "DP-

 doi:10.20944/preprints202603.1179.v1

friendly architectures" can substantially reduce the effective noise required and thereby improve the privacy-utility trade-off. Papernot et al.[86] further observe that standard activation functions lead to unstable gradient distributions and high clipping costs under DP-SGD. They also introduce a tempered sigmoid activation, which stabilizes unclipped gradient norms and reduces information loss from clipping. Notably, this approach significantly outperforms ReLU across multiple benchmarks. From an optimization perspective, Shamsabadi et al. [87] show that cross-entropy loss causes weight and activation growth that amplifies clipping effects. Hence, they propose a DP-SGD-oriented composite loss that reduces sensitivity and accelerates convergence, thereby decreasing budget loss at its source. Overall, this line of work reduces per-step privacy consumption by constraining gradient magnitude and improving training stability, bringing actual privacy loss closer to theoretical guarantees.

Empirical privacy auditing. Due to the complexity of deep models, theoretical differential privacy guarantees are typically derived from worst-case analyses and may be overly loose in practice. Empirical privacy auditing is therefore crucial for assessing the practical tightness of differential privacy guarantees. Annamalai et al. [54] introduce a tighter black-box auditing method that selects worst-case initial parameters to approximate the maximum ε_{emp}, achieving far stronger lower bounds across datasets. Going further, Cebere et al. [14] study the "hidden-state" setting and replace canary insertion with the construction of worst-case gradient sequences, showing that hiding intermediate models does not necessarily improve privacy, thereby prompting a reevaluation of DP-SGD's theoretical upper bounds. Additionally, Huang et al. [13] propose a noise-inversion-based general auditing framework that yields highly accurate budget estimates in the small-ε regime. It can also detect whether an implementation truly satisfies its claimed privacy guarantees. Collectively, these works strengthen the reliability of differential privacy training by quantifying real leaks and revealing hidden sources of budget loss. In effect, they form a "detect-and-correct" cycle that complements theoretical analysis.

In practice, DP-friendly architectures and training choices can substantially reduce overfitting and memorization, thereby lowering the empirical risk of an adversary inferring a sample's membership. However, they do not guarantee privacy protection by themselves. Empirical privacy auditing complements these approaches by revealing hidden leaks and enabling regression testing across implementations and updates, yet it functions as an evaluation and diagnostic tool rather than a preventive defense. Together, these tools highlight that robust protection against membership inference requires both careful model and training design and continuous empirical validation to ensure privacy claims hold in deployed systems.

5.3.2. Defenses Against Gradient Inversion Attacks

Gradient inversion attacks can be very good at reconstructing training data by simply exploiting the stable structure of gradients. Therefore, effective defense mechanisms usually aim to weaken the reversible association between the gradients and the input data. Specifically, restricting the feasible region of the gradient or altering its statistical consistency can significantly reduce the possibility of attackers acquiring reliable reconstruction clues through an inversion optimization. There are two main streams of defense methods based on this idea: clipping-centric defenses and perturbation-centric defenses.

Clipping-centric defenses. In gradient inversion attacks on CDP-DL systems, attackers directly observe unadulterated gradients, making clipping the primary mechanism for limiting any recoverable signal available to the adversary. High-fidelity reconstructions arise from loose gradient-norm constraints, where the structural relationship between gradients and inputs remains difficult to conceal even in the presence of substantial noise. Therefore, clipping-centric defenses works by restricting the feasible gradient space. Adaptive clipping by Andrew et al. [88] is a representative method. It uses quantile statistics to adaptively set clipping thresholds so that gradients stay within a smooth and controlled normal range over time. DP-FedACN [89] further adjusts thresholds based on gradient-norm trends and clipping loss, allowing clipping to automatically adapt to different training stages. Although Adap DP-FL [90] and ADPFL [91] also incorporate adaptive noise, their core mechanism still lies in dynamically setting clipping scales based on the norms of previous-round or layer-level importance to stabilize gradient distributions.

 doi:10.20944/preprints202603.1179.v1

Perturbation-centric defenses. In gradient inversionattacks under LDP-DL, attackers exploit clipped and noisy gradients. However, when good task accuracy is necessary, and gradients cannot be overly restricted, perturbation-centric methods are often still able to suppress such attacks by altering the statistical structure of the gradients visible to the attacker. The key idea is that even if gradient norms stay large, sufficiently dynamic, structured, or non-stationary perturbations will prevent the attacker's inversion optimization from converging to interpretable inputs. For instance, Fed-αCDP [92] injects noise into per-example gradients at each client step and applies a round-wise decay mechanism, causing early-stage gradients to be strongly perturbed. This fundamentally harms the reconstruction conditions required to initiate an attack. ANP [93] introduces a two-phase noise strategy separating critical and non-critical periods, using even-odd round noise cancellation to maintain accuracy. AGP and ADPFL [94] both manipulate the noise structurally. Using Grad-CAM or layer-importance analysis, they apply strong noise only to non-critical gradient dimensions, making it difficult for attackers to recover input information from effective channels. OUTPOST [95] introduces attack-aware perturbations, where noise injection is dynamically determined by diffusing the model-weights or through local training steps. Either disrupt the temporal stability needed for an attack.

Table 5 shows representative defense methods against gradient inversion attacks and their descriptions. In summary, clipping-centric defenses are effective at constraining the gradient space and reducing leaks, but they are often insufficient on their own, as strong attackers can still exploit the remaining structured gradients to recover input information. Perturbation-centric defenses, especially when combined with clipping and proper privacy accounting, more effectively disrupt gradient consistency and significantly lower reconstruction quality given an adaptive attack. These observations suggest that robustly protecting the gradient information relies on combining both gradient restriction with carefully designed noise mechanisms.

Table 5. A summary of defenses against gradient inversion attacks.

Defense Type	Literature	Description
Clipping-centric	Adaptive clipping [88]	Tracks gradient norm quantiles to adapt clipping bounds.
	DP-FedACN [89]	Adjusts clipping thresholds using gradient trends and clipping loss.
	Adap DP-FL [90]	Tunes clipping bounds round-by-round from gradient statistics.
	ADPFL [91]	Co-adapts clipping thresholds and privacy budgets across layers and rounds.
Perturbation-centric	Fed-αCDP [92]	Injects adaptive per-example noise into local gradients with decaying variance.
	ANP [93]	Applies staged noise injection with strong early perturbation and later noise cancellation.
	AGP and ADPFL [94]	Perturbs low-importance gradient dimensions via Grad-CAM analysis.
	OUTPOST [95]	Decides gradient perturbation dynamically via attack risk estimation.

5.4. Defenses Against Poisoning Attacks

Defenses against poisoning attacks in DP-based federated learning systems aim to mitigate malicious model manipulation by redesigning the aggregation strategies and client update dynamics while still imposing privacy constraints. By explicitly accounting for DP-induced noise and heterogeneous effects, these methods reduce the ambiguity between benign and adversarial updates during training (as shown in Figure 16). Such defenses are particularly important in high-noise regimes, where conventional assumptions of robustness break down and lightweight poisoning attacks become harder to detect.

Robust aggregation under differential privacy noise. In a DP federated learning system, local noise injection not only protects client privacy but also magnifies gradient discrepancies caused by data heterogeneity. As a result, conventional robust aggregation methods that rely on geometric distances or statistical consistency struggle to distinguish malicious updates from benign ones. To address this issue, recent studies have focused on designing aggregators that remain robust under DP-induced

 doi:10.20944/preprints202603.1179.v1

noise while preserving their ability to detect and mitigate model poisoning attacks. As an example, BVDFed [96] evaluates client updates by measuring loss changes on a public dataset and introduces a noise compensation mechanism to derive DP-independent loss scores, thereby avoiding the degradation of distance-based metrics under high noise. DP-BREM [97] improves update consistency through client momentum and centralized clipping, while confining differentially private noise to low-sensitivity aggregated momentum. This design preserves Byzantine robustness even in highly noisy settings. Robust-HDP [98] adopts a noise modeling perspective: by applying robust PCA to decompose the update matrix, it estimates client-specific noise levels and performs noise-aware weighted aggregation, naturally suppressing poisoning updates that exhibit sparse shifts in high-dimensional spaces. Collectively, these methods enhance the aggregator's ability to distinguish a malicious update from a benign one, thereby providing an effective defense against poisoning attacks.

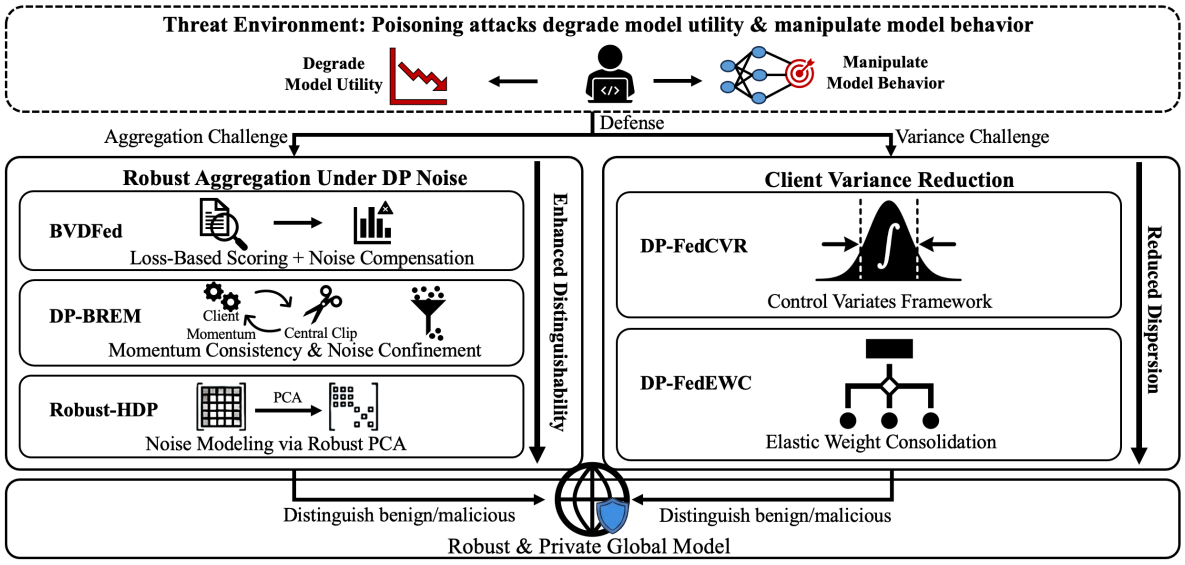

Figure 16. Defenses against poisoning attacks in DP-based federated learning. Defenses improve robustness by strengthening aggregation under DP noise and reducing client update variance. Noise-aware robust aggregation distinguishes malicious updates, while variance reduction concentrates benign updates to improve detection of poisoning behavior.

Client variance reduction. Differentially private noise exacerbates client drift caused by heterogeneous data, leading benign client updates to exhibit highly dispersed statistical patterns under high noise. This dispersion allows lightweight data poisoning attacks, such as label flipping, to be easily concealed by the noise. Consequently, reducing variance among benign clients and re-concentrating the update distribution is critical for improving robustness. DP-FedCVR [99] addresses drift jointly induced by heterogeneous data and differentially private noise via a control variates framework – an approach that theoretically guarantees stable convergence under any privacy budget. Enhancing the statistical consistency of benign updates makes anomalous updates more distinguishable. DP-FedEwc [100] restricts changes to critical parameters using elastic weight consolidation, ensuring that personalized models remain within a shared subspace across training rounds. It further introduces an adaptive noise perturbation mechanism that satisfies privacy constraints while preserving the accuracy of parameter importance estimation, thereby limiting additional heterogeneity introduced by differential privacy noise. Overall, these methods reduce the dispersion of benign updates, substantially improving the detectability of data poisoning attacks in high-noise regimes and providing a more stable foundation for robust aggregation and anomaly detection.

In practice, defenses against poisoning attacks in federated scenarios balance robustness against adversarial manipulation with efficient training processes, low communication costs, and high system complexity. Robust aggregation and variance-reduction techniques increase the victim's ability to detect an attack in high-noise regimes. However, often, success relies on auxiliary information,

 doi:10.20944/preprints202603.1179.v1

additional state tracking, or public data, which may not be available across all deployments. These trade-offs underscore that effectively protecting against a poisoning attack in a federated setting requires carefully integrated robustness mechanisms, privacy constraints, and system-level feasibility considerations.

6. Discussion and Future Directions

This section discusses the limitations of existing differential privacy defenses and outlines promising future research directions at the data, algorithm, and architecture levels. Rather than viewing these levels in isolation, it is important to emphasize that they play complementary roles in addressing privacy leaks. Together, these perspectives point toward a more holistic and system-aware understanding of differential privacy in practice.

6.1. Data-Level Perspective

Limitations of existing defenses. Current data-level defenses mainly attempt to reduce attribute correlation or hide individual records – for example, by reducing dimensionality or generating synthetic data [106,107]. However, such defenses often treat correlations in a localized and task-specific manner and may not fully suppress the semantic or causal dependencies embedded in real datasets. The guarantees of privacy also tend to be limited when a user contributes multiple records to a training set. Under these circumstances, adversaries may be able to infer user-level information even when perturbations have been applied at the record level.

Future directions. Future data-level research might look to incorporate statistical relational modeling, causal inference, and/or semantic analysis into their differential privacy mechanisms to better characterize privacy at the level of structured and interdependent data [108–110]. Combining differential privacy with synthetic data generation and representation learning may also strengthen the privacy guarantees that apply when distributions shift or in cross-domain transfer scenarios [111]. These directions of research highlight the importance of advancing from record-centric protection toward privacy concepts that align more closely with the semantics and structure of modern high-dimensional data.

6.2. Algorithm-Level Perspective

Limitations of existing defenses. Existing algorithm-level defenses typically reinforce DP-SGD through adaptive clipping, modified noise schedules, or runtime query controls [112,113]. While these approaches may mitigate specific attack vectors, they rely on assumptions about randomness, floating-point precision, and interaction patterns that are not always guaranteed in practical deployments. As a result, empirical privacy leaks can diverge from theoretical estimates when system-level imperfections accumulate.

Future directions. Further progress at the algorithm level may benefit from integrating differential privacy with cryptography [114], secure hardware primitives [115], and information-theoretic leakage models [116]. Here, the aim would be to strengthen guarantees beyond purely statistical perturbation. Moreover, combining differential privacy with adaptive privacy accounting and adversarial interaction modeling may give rise to more effective support for long-term and multi-party learning scenarios [117]. Such directions emphasize a multidisciplinary perspective in which differential privacy serves as the core principle. This core can then be bolstered with methods from other disciplines to construct a more comprehensive privacy methodology.

6.3. Architecture-Level Perspective

Limitations of existing defenses. The fundamental limitation of architectural-level defenses is that they tend to reduce the risk of information leaks, rather than explicitly blocking attack signals at the mechanistic level [118]. Such defenses typically mitigate privacy leaks by stabilizing gradients and reducing model sensitivity; however, their effectiveness is often contingent on the specific model architecture, task, and hyperparameter choices, which limits the extent to which their benefits generalize across settings. Consequently, architectural design alone may not be sufficient to reliably defend

 doi:10.20944/preprints202603.1179.v1

against strong or adaptive attacks. Rather, structural solutions are more effective when combined with complementary approaches such as noise-based mechanisms and empirical auditing techniques.

Future directions. Architecture-level research may increasingly examine privacy-aware model and system co-design, where architectural components and deployment settings are jointly optimized with privacy considerations [119]. There is also growing potential in hybrid privacy architectures that combine differential privacy with secure multiparty computation, distributed systems engineering, and robust optimization [120,121]. These developments may support multi-layered privacy infrastructures that operate across algorithmic, architectural, and system dimensions, especially in collaborative, federated, and streaming learning environments.

7. Conclusions

This paper provides a systematic review of the risks, attacks, and defenses in DP-DL systems at the data, algorithm, and architecture levels. The material reviewed reveals a distinct gap between the guarantees differential privacy offers in theory and the practical security one can expect from a real-world deployment. By establishing a unified "risk-attack-defense" framework, we have summarized the representative attack vectors and their corresponding countermeasures. Part of this has involved emphasizing the critical role that implementation details and system design play in real-world privacy protection. Looking ahead, future research should aim to better balance privacy, utility, and engineering feasibility so as to deploy secure and reliable DP-DL systems in practical applications.

References

1. Liu, X.; Chen, Y.; Pang, S. Defending Against Membership Inference Attack for Counterfactual Federated Recommendation With Differentially Private Representation Learning. *IEEE Transactions on Information Forensics and Security* **2024**, *19*, 8037–8051. https://doi.org/10.1109/TIFS.2024.3453031.
2. Orabe, Z.; Vasankari, A.; Pahikkala, T.; Kaisti, M.; Airola, A. Securing deep learning models with differential privacy for cardiovascular disease prediction. *Biomedical Signal Processing and Control* **2026**, *112*, 108502. https://doi.org/https://doi.org/10.1016/j.bspc.2025.108502.
3. Hong, J.; Wang, J.T.; Zhang, C.; LI, Z.; Li, B.; Wang, Z. DP-OPT: Make Large Language Model Your Privacy-Preserving Prompt Engineer. In Proceedings of the The Twelfth International Conference on Learning Representations, 2024.
4. Li, X.; Tramèr, F.; Liang, P.; Hashimoto, T. Large Language Models Can Be Strong Differentially Private Learners. *CoRR* **2021**, *abs/2110.05679*, [2110.05679].
5. Mai, P.; Yan, R.; Huang, Z.; Yang, Y.; Pang, Y. Split-and-Denoise: Protect large language model inference with local differential privacy, 2024, [arXiv:cs.AI/2310.09130].
6. WANG, J.; Schuster, R.; Shumailov, I.; Lie, D.; Papernot, N. In Differential Privacy, There is Truth: on Vote-Histogram Leakage in Ensemble Private Learning. In Proceedings of the Advances in Neural Information Processing Systems, 2022, Vol. 35, pp. 29026–29037.
7. Yan, H.; Li, X.; Li, H.; Li, J.; Sun, W.; Li, F. Monitoring-Based Differential Privacy Mechanism Against Query Flooding-Based Model Extraction Attack. *IEEE Transactions on Dependable and Secure Computing* **2022**, *19*, 2680–2694.
8. Jin, J.; McMurtry, E.; Rubinstein, B.I.P.; Ohrimenko, O. Are We There Yet? Timing and Floating-Point Attacks on Differential Privacy Systems. In Proceedings of the 43rd IEEE Symposium on Security and Privacy, SP 2022, May 22-26. IEEE, 2022, pp. 473–488.
9. Jagielski, M.; Ullman, J.; Oprea, A. Auditing differentially private machine learning: how private is private SGD? In Proceedings of the Proceedings of the 34th International Conference on Neural Information Processing Systems, 2020.
10. Jiang, Y.; Ma, B.; Wang, X.; Yu, G.; Sun, C.; Ni, W.; Liu, R.P. Exploiting attribute correlation for reconstruction attacks on differentially private multi-attributed data. *Journal of Information Security and Applications* **2025**, *94*, 104224.
11. Chua, L.; Ghazi, B.; Huang, Y.; Kamath, P.; Kumar, R.; Liu, D.; Manurangsi, P.; Sinha, A.; Zhang, C. Mind the Privacy Unit! User-Level Differential Privacy for Language Model Fine-Tuning, 2024, [arXiv:cs.CL/2406.14322].

 doi:10.20944/preprints202603.1179.v1

12. Haney, S.; Desfontaines, D.; Hartman, L.; Shrestha, R.; Hay, M. Precision-based attacks and interval refining: how to break, then fix, differential privacy on finite computers, 2022, [arXiv:cs.CR/2207.13793].
13. Huang, W.; Zhang, Z.; Zhao, W.; Peng, J.; Xu, W.; Liao, Y.; Zhou, S.; Wang, Z. Auditing privacy budget of differentially private neural network models. *Neurocomputing* **2025**, *614*, 128756. https://doi.org/https://doi.org/10.1016/j.neucom.2024.128756.
14. Cebere, T.; Bellet, A.; Papernot, N. Tighter Privacy Auditing of DP-SGD in the Hidden State Threat Model, 2025, [arXiv:cs.LG/2405.14457].
15. Zhu, T.; Li, G.; Zhou, W.; Yu, P.S. Differentially Private Data Publishing and Analysis: A Survey. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 1619–1638. https://doi.org/10.1109/TKDE.2017.2697856.
16. Zhu, T.; Ye, D.; Wang, W.; Zhou, W.; Yu, P.S. More Than Privacy: Applying Differential Privacy in Key Areas of Artificial Intelligence. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 2824–2843. https://doi.org/10.1109/TKDE.2020.3014246.
17. Blanco-Justicia, A.; Sánchez, D.; Domingo-Ferrer, J.; Muralidhar, K. A Critical Review on the Use (and Misuse) of Differential Privacy in Machine Learning. *ACM Comput. Surv.* **2022**, *55*. https://doi.org/10.1145/3547139.
18. Demelius, L.; Kern, R.; Trügler, A. Recent Advances of Differential Privacy in Centralized Deep Learning: A Systematic Survey. *ACM Comput. Surv.* **2025**, *57*. https://doi.org/10.1145/3712000.
19. Zhang, X.; Zhang, Q. Defending against attacks in deep learning with differential privacy: a survey. *Artificial Intelligence Review* **2025**, *58*.
20. Dwork, C.; Roth, A. The Algorithmic Foundations of Differential Privacy. *Found. Trends Theor. Comput. Sci.* **2014**, *9*, 211–407. https://doi.org/10.1561/0400000042.
21. Dwork, C.; McSherry, F.; Nissim, K.; Smith, A. Calibrating Noise to Sensitivity in Private Data Analysis. In Proceedings of the Theory of Cryptography, Berlin, Heidelberg, 2006; pp. 265–284.
22. Balle, B.; Wang, Y.X. Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In Proceedings of the International conference on machine learning. PMLR, 2018, pp. 394–403.
23. Zhao, Y.; Du, J.T.; Chen, J. Scenario-based adaptations of differential privacy: A technical survey. ACM New York, NY, 2024, Vol. 56, pp. 1–39.
24. Cormode, G.; Maddock, S.; Maple, C. Frequency Estimation under Local Differential Privacy. *Proc. VLDB Endow.* **2021**, *14*, 2046–2058. https://doi.org/10.14778/3476249.3476261.
25. Erlingsson, Ú.; Pihur, V.; Korolova, A. Rappor: Randomized aggregatable privacy-preserving ordinal response. In Proceedings of the Proceedings of the 2014 ACM SIGSAC conference on computer and communications security, 2014, pp. 1054–1067.
26. Acharya, J.; Sun, Z.; Zhang, H. Hadamard response: Estimating distributions privately, efficiently, and with little communication. In Proceedings of the The 22nd International Conference on Artificial Intelligence and Statistics. PMLR, 2019, pp. 1120–1129.
27. Shen, Z.; Zhong, T.; Sun, H.; Qi, B. RRN: A differential private approach to preserve privacy in image classification. *IET Image Process.* **2023**, *17*, 2192–2203. https://doi.org/10.1049/IPR2.12784.
28. Pittaluga, F.; Zhuang, B. LDP-Feat: Image Features with Local Differential Privacy. In Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023. IEEE, 2023, pp. 17534–17544. https://doi.org/10.1109/ICCV51070.2023.01612.
29. Ghazi, B.; Golowich, N.; Kumar, R.; Manurangsi, P.; Zhang, C. Deep learning with label differential privacy. *Advances in neural information processing systems* **2021**, *34*, 27131–27145.
30. Malek Esmaeili, M.; Mironov, I.; Prasad, K.; Shilov, I.; Tramer, F. Antipodes of label differential privacy: Pate and alibi. *Advances in neural information processing systems* **2021**, *34*, 6934–6945.
31. Matsumoto, T.; Miura, T.; Shibahara, T.; Kii, M.; Iwahana, K.; Saisho, O.; Okamura, S. Differentially Private Sequential Data Synthesis with Structured State Space Models and Diffusion Models. In Proceedings of the Neurips Safe Generative AI Workshop 2024, 2024.
32. Li, K.; Gong, C.; Li, X.; Zhao, Y.; Hou, X.; Wang, T. From Easy to Hard: Building a Shortcut for Differentially Private Image Synthesis. In Proceedings of the IEEE Symposium on Security and Privacy, SP 2025, May 12-15. IEEE, 2025, pp. 3988–4006.
33. Zhang, Z.; Wang, T.; Li, N.; Honorio, J.; Backes, M.; He, S.; Chen, J.; Zhang, Y. PrivSyn: Differentially Private Data Synthesis. In Proceedings of the 30th USENIX Security Symposium, August 11-13, 2021, 2021, pp. 929–946.

 doi:10.20944/preprints202603.1179.v1

34. Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H.B.; Mironov, I.; Talwar, K.; Zhang, L. Deep learning with differential privacy. In Proceedings of the Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, 2016, pp. 308–318.
35. Xiang, L.; Yang, J.; Li, B. Differentially-Private Deep Learning from an optimization Perspective. In Proceedings of the 2019 IEEE Conference on Computer Communications, INFOCOM 2019, Paris, France, April 29 - May 2, 2019. IEEE, 2019, pp. 559–567.
36. Chen, L.; Yue, D.; Ding, X.; Wang, Z.; Choo, K.R.; Jin, H. Differentially Private Deep Learning With Dynamic Privacy Budget Allocation and Adaptive Optimization. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 4422–4435.
37. Cheng, A.; Wang, J.; Zhang, X.S.; Chen, Q.; Wang, P.; Cheng, J. Dpnas: Neural architecture search for deep learning with differential privacy. In Proceedings of the Proceedings of the AAAI conference on artificial intelligence, 2022, Vol. 36, pp. 6358–6366.
38. Kurakin, A.; Chien, S.; Song, S.; Geambasu, R.; Terzis, A.; Thakurta, A. Toward Training at ImageNet Scale with Differential Privacy. *CoRR* **2022**, *abs/2201.12328*, [2201.12328].
39. Kairouz, P.; Liu, Z.; Steinke, T. The distributed discrete gaussian mechanism for federated learning with secure aggregation. In Proceedings of the International conference on machine learning. PMLR, 2021, pp. 5201–5212.
40. Hay, M.; Miklau, G.; Jensen, D.D.; Towsley, D.F.; Li, C. Resisting structural re-identification in anonymized social networks. *VLDB J.* **2010**, *19*, 797–823. https://doi.org/10.1007/S00778-010-0210-X.
41. Jiang, H.; Pei, J.; Yu, D.; Yu, J.; Gong, B.; Cheng, X. Applications of Differential Privacy in Social Network Analysis: A Survey. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 108–127. https://doi.org/10.1109/TKDE.2021.3073062.
42. Jang, H.J.; Cho, K.O. Applications of deep learning for the analysis of medical data. *Archives of pharmacal research* **2019**, *42*, 492–504.
43. Cretu, A.; Guépin, F.; de Montjoye, Y. Dataset correlation inference attacks against machine learning models. *CoRR* **2021**, *abs/2112.08806*, [2112.08806].
44. Yu, D.; Naik, S.; Backurs, A.; Gopi, S.; Inan, H.A.; Kamath, G.; Kulkarni, J.; Lee, Y.T.; Manoel, A.; Wutschitz, L.; et al. Differentially Private Fine-tuning of Language Models. In Proceedings of the The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022, 2022.
45. Yue, X.; Inan, H.; Li, X.; Kumar, G.; McAnallen, J.; Shajari, H.; Sun, H.; Levitan, D.; Sim, R. Synthetic text generation with differential privacy: A simple and practical recipe. In Proceedings of the Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2023, pp. 1321–1342.
46. Tang, X.; Shin, R.; Inan, H.A.; Manoel, A.; Mireshghallah, F.; Lin, Z.; Gopi, S.; Kulkarni, J.; Sim, R. Privacy-Preserving In-Context Learning with Differentially Private Few-Shot Generation. In Proceedings of the The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024.
47. Li, X.; Li, H.; Zhu, H.; Huang, M. The optimal upper bound of the number of queries for Laplace mechanism under differential privacy. *Inf. Sci.* **2019**, *503*, 219–237. https://doi.org/10.1016/J.INS.2019.07.001.
48. Sander, T.; Sylvestre, M.; Durmus, A. Implicit bias in noisy-sgd: With applications to differentially private training. In Proceedings of the International Conference on Artificial Intelligence and Statistics. PMLR, 2024, pp. 3295–3303.
49. Wang, J.; Zhou, Z.H. Differentially private learning with small public data. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2020, Vol. 34, pp. 6219–6226.
50. Hong, J.; Wang, H.; Wang, Z.; Zhou, J. Learning model-based privacy protection under budget constraints. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2021, Vol. 35, pp. 7702–7710.
51. Kulkarni, J.; Lee, Y.T.; Liu, D. Private non-smooth erm and sco in subquadratic steps. *Advances in Neural Information Processing Systems* **2021**, *34*, 4053–4064.
52. Mironov, I. On significance of the least significant bits for differential privacy. In Proceedings of the the ACM Conference on Computer and Communications Security, CCS'12, Raleigh, NC, USA, October 16-18, 2012. ACM, 2012, pp. 650–661. https://doi.org/10.1145/2382196.2382264.
53. Jayaraman, B.; Evans, D. Evaluating Differentially Private Machine Learning in Practice. In Proceedings of the 28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, August 14-16, 2019. USENIX Association, 2019, pp. 1895–1912.
54. Muthu Selva Annamalai, M.S.; De Cristofaro, E. Nearly tight black-box auditing of differentially private machine learning. 2024, Vol. 37, pp. 131482–131502.

 doi:10.20944/preprints202603.1179.v1

55. Li, Z.; Zhang, J.; Liu, L.; Liu, J. Auditing Privacy Defenses in Federated Learning via Generative Gradient Leakage. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, June 18-24. IEEE, 2022, pp. 10122–10132.
56. Yue, K.; Jin, R.; Wong, C.; Baron, D.; Dai, H. Gradient Obfuscation Gives a False Sense of Security in Federated Learning. In Proceedings of the 32nd USENIX Security Symposium, USENIX Security 2023, August 9-11, 2023. USENIX Association, 2023, pp. 6381–6398.
57. You, Z.; Dong, X.; Li, S.; Liu, X.; Ma, S.; Shen, Y. Local Differential Privacy Is Not Enough: A Sample Reconstruction Attack Against Federated Learning With Local Differential Privacy. *IEEE Trans. Inf. Forensics Secur.* **2025**, *20*, 1519–1534.
58. Hu, J.; Du, J.; Wang, Z.; Pang, X.; Zhou, Y.; Sun, P.; Ren, K. Does Differential Privacy Really Protect Federated Learning From Gradient Leakage Attacks? *IEEE Trans. Mob. Comput.* **2024**, *23*, 12635–12649.
59. Boenisch, F.; Dziedzic, A.; Schuster, R.; Shamsabadi, A.S.; Shumailov, I.; Papernot, N. Reconstructing Individual Data Points in Federated Learning Hardened with Differential Privacy and Secure Aggregation. In Proceedings of the 8th IEEE European Symposium on Security and Privacy, EuroS&P 2023, Delft, Netherlands, July 3-7, 2023. IEEE, 2023, pp. 241–257. https://doi.org/10.1109/EUROSP57164.2023.00023.
60. Wang, Z.; Ma, J.; Wang, X.; Hu, J.; Qin, Z.; Ren, K. Threats to Training: A Survey of Poisoning Attacks and Defenses on Machine Learning Systems. *ACM Comput. Surv.* **2023**, *55*, 134:1–134:36. https://doi.org/10.1145/3538707.
61. Jayaraman, B.; Evans, D. Are Attribute Inference Attacks Just Imputation? In Proceedings of the Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022. ACM, 2022, pp. 1569–1582.
62. Kandpal, N.; Pillutla, K.; Oprea, A.; Kairouz, P.; Choquette-Choo, C.A.; Xu, Z. User Inference Attacks on Large Language Models. In Proceedings of the Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024. Association for Computational Linguistics, 2024, pp. 18238–18265.
63. Asghar, H.J.; Kaafar, D. Averaging Attacks on Bounded Noise-based Disclosure Control Algorithms. *Proc. Priv. Enhancing Technol.* **2020**, *2020*, 358–378. https://doi.org/10.2478/POPETS-2020-0031.
64. Shokri, R.; Stronati, M.; Song, C.; Shmatikov, V. Membership Inference Attacks Against Machine Learning Models. In Proceedings of the 2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017. IEEE Computer Society, 2017, pp. 3–18.
65. Riaz, S.; Ali, S.; Wang, G.; Latif, M.A.; Iqbal, M.Z. Membership inference attack on differentially private block coordinate descent. *PeerJ Comput. Sci.* **2023**, *9*, e1616. https://doi.org/10.7717/PEERJ-CS.1616.
66. Yeom, S.; Giacomelli, I.; Fredrikson, M.; Jha, S. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. In Proceedings of the 31st IEEE Computer Security Foundations Symposium, July 9-12, 2018. IEEE Computer Society, 2018, pp. 268–282.
67. Yu, W.; Fang, H.; Chen, B.; Sui, X.; Chen, C.; Wu, H.; Xia, S.; Xu, K. GI-NAS: Boosting Gradient Inversion Attacks Through Adaptive Neural Architecture Search. *IEEE Trans. Inf. Forensics Secur.* **2025**, *20*, 7648–7662.
68. Weng, S.; Gou, Y.; Zhang, L.; Imran, M.A. Evaluating privacy loss in differential privacy based federated learning. *Future Gener. Comput. Syst.* **2025**, *172*, 107848. https://doi.org/10.1016/J.FUTURE.2025.107848.
69. Peng, J.; Li, W.; Vlaski, S.; Ling, Q. Mean aggregator is more robust than robust aggregators under label poisoning attacks on distributed heterogeneous data. *Journal of Machine Learning Research* **2025**, *26*, 1–51.
70. Cheu, A.; Smith, A.D.; Ullman, J.R. Manipulation Attacks in Local Differential Privacy. In Proceedings of the 42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021. IEEE, 2021, pp. 883–900. https://doi.org/10.1109/SP40001.2021.00001.
71. Wang, Z.; Tong, W.; Han, T.; Chen, H.; Zhang, T.; Mao, Y.; Zhong, S. On Evaluating the Poisoning Robustness of Federated Learning under Local Differential Privacy. *CoRR* **2025**, *abs/2509.05265*, [2509.05265].
72. Zheng, H.; Chen, J.; Liu, T.; Cheng, Y.; Wang, Z.; Wang, Y.; Gao, L.; Ji, S.; Zhang, X. DP-Poison: Poisoning Federated Learning under the Cover of Differential Privacy. *ACM Trans. Priv. Secur.* **2025**, *28*, 7:1–7:28. https://doi.org/10.1145/3702325.
73. Yang, M.; Cheng, H.; Chen, F.; Liu, X.; Wang, M.; Li, X. Model poisoning attack in differential privacy-based federated learning. *Inf. Sci.* **2023**, *630*, 158–172. https://doi.org/10.1016/J.INS.2023.02.025.
74. Rahman, M.A.; Rahman, T.; Laganière, R.; Mohammed, N. Membership Inference Attack against Differentially Private Deep Learning Model. *Trans. Data Priv.* **2018**, *11*, 61–79.
75. Zhu, L.; Liu, Z.; Han, S. Deep Leakage from Gradients. In Proceedings of the NeurIPS 2019, December 8-14, 2019, pp. 14747–14756.

 doi:10.20944/preprints202603.1179.v1

76. Geiping, J.; Bauermeister, H.; Dröge, H.; Moeller, M. Inverting Gradients - How easy is it to break privacy in federated learning? In Proceedings of the NeurIPS 2020, December 6-12, 2020, virtual, 2020.
77. Yin, H.; Mallya, A.; Vahdat, A.; Alvarez, J.M.; Kautz, J.; Molchanov, P. See through gradients: Image batch recovery via gradinversion. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 16337–16346.
78. Wang, D.; Xu, J. Principal Component Analysis in the Local Differential Privacy Model. In Proceedings of the Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019. ijcai.org, 2019, pp. 4795–4801.
79. Ablin, P.; Gramfort, A.; Cardoso, J.F.; Bach, F. Stochastic algorithms with descent guarantees for ICA. In Proceedings of the The 22nd International Conference on Artificial Intelligence and Statistics. PMLR, 2019, pp. 1564–1573.
80. Yang, X.; Ardakanian, O. PrivDiffuser: Privacy-Guided Diffusion Model for Data Obfuscation in Sensor Networks. *Proc. Priv. Enhancing Technol.* **2025**, *2025*, 40–55. https://doi.org/10.56553/POPETS-2025-0118.
81. Levy, D.; Sun, Z.; Amin, K.; Kale, S.; Kulesza, A.; Mohri, M.; Suresh, A.T. Learning with user-level privacy. 2021, Vol. 34, pp. 12466–12479.
82. Ghazi, B.; Kamath, P.; Kumar, R.; Manurangsi, P.; Meka, R.; Zhang, C. User-level differential privacy with few examples per user. 2023, Vol. 36, pp. 19263–19290.
83. Charles, Z.; Ganesh, A.; McKenna, R.; McMahan, H.B.; Mitchell, N.; Pillutla, K.; Rush, K. Fine-Tuning Large Language Models with User-Level Differential Privacy. *CoRR* **2024**, *abs/2407.07737*, [2407.07737].
84. Rogers, R.M.; Roth, A.; Ullman, J.; Vadhan, S. Privacy odometers and filters: Pay-as-you-go composition. 2016, Vol. 29.
85. Holohan, N.; Braghin, S.; Suliman, M. Securing Floating-Point Arithmetic for Noise Addition. In Proceedings of the Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, 14-18,. ACM, 2024, pp. 1954–1966.
86. Papernot, N.; Thakurta, A.; Song, S.; Chien, S.; Erlingsson, Ú. Tempered sigmoid activations for deep learning with differential privacy. In Proceedings of the Proceedings of the AAAI conference on artificial intelligence, 2021, Vol. 35, pp. 9312–9321.
87. Shamsabadi, A.S.; Papernot, N. Losing Less: A Loss for Differentially Private Deep Learning. *Proc. Priv. Enhancing Technol.* **2023**, *2023*, 307–320. https://doi.org/10.56553/POPETS-2023-0083.
88. Andrew, G.; Thakkar, O.; McMahan, B.; Ramaswamy, S. Differentially Private Learning with Adaptive Clipping. In Proceedings of the NeurIPS 2021, December 6-14, 2021, virtual, 2021, pp. 17455–17466.
89. Zhang, B.; Mao, Y.; He, X.; Ping, P.; Huang, H.; Wu, J. Exploring the Privacy-Accuracy Trade-Off Using Adaptive Gradient Clipping in Federated Learning. *IEEE Trans. Netw. Sci. Eng.* **2025**, *12*, 2254–2265.
90. Fu, J.; Chen, Z.; Han, X. Adap DP-FL: Differentially Private Federated Learning with Adaptive Noise. In Proceedings of the IEEE International Conference on Trust, Security and Privacy in Computing and Communications, 9-11, 2022. IEEE, 2022, pp. 656–663.
91. Xu, X.; Li, Y.; Lu, L.; Wang, T. ADPFL: Adaptive Differential Privacy-Enhanced Federated Learning. *IEEE Internet Things J.* **2025**, *12*, 50682–50692. https://doi.org/10.1109/JIOT.2025.3611410.
92. Wei, W.; Liu, L.; Zhou, J.; Chow, K.H.; Wu, Y. Securing Distributed SGD Against Gradient Leakage Threats. *IEEE Trans. Parallel Distributed Syst.* **2023**, *34*, 2040–2054. https://doi.org/10.1109/TPDS.2023.3273490.
93. Li, Z.; Chen, H.; Gao, Y.; Ni, Z.; Xue, H.; Shao, H. Staged Noise Perturbation for Privacy-Preserving Federated Learning. *IEEE Trans. Sustain. Comput.* **2024**, *9*, 936–947. https://doi.org/10.1109/TSUSC.2024.3381812.
94. Li, Z.; Chen, H.; Ni, Z.; Gao, Y.; Lou, W. Towards Adaptive Privacy Protection for Interpretable Federated Learning. *IEEE Trans. Mob. Comput.* **2024**, *23*, 14471–14483. https://doi.org/10.1109/TMC.2024.3443862.
95. Wang, F.; Hugh, E.; Li, B. More Than Enough is Too Much: Adaptive Defenses Against Gradient Leakage in Production Federated Learning. *IEEE/ACM Trans. Netw.* **2024**, *32*, 3061–3075. https://doi.org/10.1109/TNET.2024.3377655.
96. Gao, X.; Fu, S.; Liu, L.; Luo, Y. BVDFed: Byzantine-resilient and verifiable aggregation for differentially private federated learning. *Frontiers Comput. Sci.* **2024**, *18*, 185810.
97. Gu, X.; Li, M.; Xiong, L. DP-BREM: Differentially-Private and Byzantine-Robust Federated Learning with Client Momentum. In Proceedings of the 34th USENIX Security Symposium, USENIX Security 2025, August 13-15. USENIX Association, 2025, pp. 3065–3082.
98. Malekmohammadi, S.; Yu, Y.; Cao, Y. Noise-aware algorithm for heterogeneous differentially private federated learning. 2024.

 doi:10.20944/preprints202603.1179.v1

99. Wang, X.; Wang, S.; Li, Y.; Fan, F.; Li, S.; Lin, X. Differentially Private and Heterogeneity-Robust Federated Learning With Theoretical Guarantee. *IEEE Trans. Artif. Intell.* **2024**, *5*, 6369–6384. https://doi.org/10.1109/TAI.2024.3446759.
100. Liang, J.; Su, S. DP-FedEwc: Differentially private federated elastic weight consolidation for model personalization. *Knowl. Based Syst.* **2024**, *303*, 112401. https://doi.org/10.1016/J.KNOSYS.2024.112401.
101. Galhotra, S.; Shanmugam, K.; Sattigeri, P.; Varshney, K.R. Causal Feature Selection for Algorithmic Fairness. In Proceedings of the SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022. ACM, 2022, pp. 276–285.
102. Salazar, R.; Neutatz, F.; Abedjan, Z. Automated Feature Engineering for Algorithmic Fairness. *Proc. VLDB Endow.* **2021**, *14*, 1694–1702. https://doi.org/10.14778/3461535.3463474.
103. Ho, S.; Qu, Y.; Gu, B.; Gao, L.; Li, J.; Xiang, Y. DP-GAN: Differentially private consecutive data publishing using generative adversarial nets. *J. Netw. Comput. Appl.* **2021**, *185*, 103066. https://doi.org/10.1016/J.JNCA.2021.103066.
104. Weggenmann, B.; Rublack, V.; Andrejczuk, M.; Mattern, J.; Kerschbaum, F. DP-VAE: Human-Readable Text Anonymization for Online Reviews with Differentially Private Variational Autoencoders. In Proceedings of the WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022. ACM, 2022, pp. 721–731. https://doi.org/10.1145/3485447.3512232.
105. Béthune, L.; Massena, T.; Boissin, T.; Bellet, A.; Mamalet, F.; Prudent, Y.; Friedrich, C.; Serrurier, M.; Vigouroux, D. DP-SGD Without Clipping: The Lipschitz Neural Network Way. In Proceedings of the The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024. OpenReview.net, 2024.
106. Shen, H.; Li, J.; Wu, G.; Zhang, M. Data release for machine learning via correlated differential privacy. *Inf. Process. Manag.* **2023**, *60*, 103349. https://doi.org/10.1016/J.IPM.2023.103349.
107. Ramesh, K.; Gandhi, N.; Madaan, P.; Bauer, L.; Peris, C.; Field, A. Evaluating differentially private synthetic data generation in high-stakes domains. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024, 2024, pp. 15254–15269.
108. Küchler, N.; Viand, A.; Lycklama, H.; Hithnawi, A. DPolicy: Managing Privacy Risks Across Multiple Releases with Differential Privacy. In Proceedings of the IEEE Symposium on Security and Privacy, SP 2025, May 12-15, 2025. IEEE, 2025, pp. 3950–3968.
109. Jiang, B.; Zhang, W.; Lu, D.; Du, J.; Sharma, S.; Yan, Q. Meeting Utility Constraints in Differential Privacy: A Privacy-Boosting Approach. In Proceedings of the 2025 IEEE Symposium on Security and Privacy (SP). IEEE, 2025, pp. 3931–3949.
110. Farzam, A.; Sapiro, G. Causal inference under differential privacy: Challenges and mitigation strategies. In Proceedings of the NeurIPS 2024 Causal Representation Learning Workshop, 2024.
111. Ponomareva, N.; Xu, Z.; McMahan, H.B.; Kairouz, P.; Rosenblatt, L.; Cohen-Addad, V.; Guzmán, C.; McKenna, R.; Andrew, G.; Bie, A.; et al. How to DP-fy Your Data: A Practical Guide to Generating Synthetic Data With Differential Privacy. *arXiv preprint arXiv:2512.03238* **2025**.
112. Ertan, M.B.; van Dijk, M. Fundamental Limitations of Favorable Privacy-Utility Guarantees for DP-SGD. *CoRR* **2026**, *abs/2601.10237*, [2601.10237]. https://doi.org/10.48550/ARXIV.2601.10237.
113. Lin, G.; Yan, H.; Kou, G.; Huang, T.; Peng, S.; Zhang, Y.; Dong, C. Understanding adaptive gradient clipping in DP-SGD, empirically. *Int. J. Intell. Syst.* **2022**, *37*, 9674–9700. https://doi.org/10.1002/INT.23001.
114. Liu, S.; Cao, Y.; Murakami, T.; Liu, J.; Yoshikawa, M. CARGO: Crypto-Assisted Differentially Private Triangle Counting Without Trusted Servers. In Proceedings of the 40th IEEE International Conference on Data Engineering, ICDE 2024, Utrecht, The Netherlands, May 13-16, 2024. IEEE, 2024, pp. 1671–1684. https://doi.org/10.1109/ICDE60146.2024.00136.
115. Near, J.P.; Darais, D.; Lefkovitz, N.; Howarth, G.S.; et al. *Guidelines for evaluating differential privacy guarantees*; US Department of Commerce, National Institute of Standards and Technology, 2025.
116. Saeidian, S.; Cervia, G.; Oechtering, T.J.; Skoglund, M. Pointwise Maximal Leakage. *IEEE Trans. Inf. Theory* **2023**, *69*, 8054–8080. https://doi.org/10.1109/TIT.2023.3304378.
117. Errounda, F.Z.; Liu, Y. Adaptive differential privacy in vertical federated learning for mobility forecasting. *Future Gener. Comput. Syst.* **2023**, *149*, 531–546. https://doi.org/10.1016/J.FUTURE.2023.07.033.
118. Morsbach, F.; Dehling, T.; Sunyaev, A. Architecture Matters: Investigating the Influence of Differential Privacy on Neural Network Design. *CoRR* **2021**, *abs/2111.14924*, [2111.14924].

 doi:10.20944/preprints202603.1179.v1

119. Zhang, G.; Liu, B.; Tian, H.; Zhu, T.; Ding, M.; Zhou, W. How Does a Deep Learning Model Architecture Impact Its Privacy? A Comprehensive Study of Privacy Attacks on CNNs and Transformers. In Proceedings of the 33rd USENIX Security Symposium, USENIX Security 2024, Philadelphia, PA, USA, August 14-16, 2024.
120. Punniyamoorthy, V.; Parthi, A.G.; Palanigounder, M.; Kodali, R.K.; Kumar, B.; Kannan, K. A Privacy-Preserving Cloud Architecture for Distributed Machine Learning at Scale. *CoRR* **2025**, *abs/2512.10341*, [2512.10341].
121. Rekik, S.; Mehmood, S. Hybrid GNN–LSTM defense with differential privacy and secure multi-party computation for edge-optimized neuromorphic autonomous systems. *Scientific Reports* **2025**, *15*, 43939.