

Differential Privacy in Machine Learning: From Symbolic AI to LLMs

FRANCISCO AGUILERA-MARTÍNEZ and FERNANDO BERZAL, Department of Computer Science and Artificial Intelligence, University of Granada, Spain

Machine learning models should not reveal particular information that is not otherwise accessible. Differential privacy [64] provides a formal framework to mitigate privacy risks by ensuring that the inclusion or exclusion of any single data point does not significantly alter the output of an algorithm, thus limiting the exposure of private information. This survey paper explores the foundational definitions of differential privacy, reviews its original formulations and tracing its evolution through key research contributions. It then provides an in-depth examination of how DP has been integrated into machine learning models, analyzing existing proposals and methods to preserve privacy when training ML models. Finally, it describes how DP-based ML techniques can be evaluated in practice. By offering a comprehensive overview of differential privacy in machine learning, this work aims to contribute to the ongoing development of secure and responsible AI systems.

CCS Concepts: • **Security and privacy** → **Malware and its mitigation**; • **Applied computing** → *Document management and text processing*; • **Computing methodologies** → **Natural language generation**; **Neural networks**.

Additional Key Words and Phrases: Differential privacy, Machine Learning, Artificial Intelligence, Artificial Neural Networks, Deep Learning, Natural Language Processing, Large Language Models, Security Threats, Defense Mechanisms

1 Introduction

The proliferation of large-scale data collection and analysis has brought unprecedented capabilities but also significant privacy risks. The tension between extracting meaningful insights from data and safeguarding individual privacy has become increasingly pronounced.

Traditional privacy preserving techniques, such as anonymization, often rely on removing personally-identifiable information to protect sensitive data [172]. However, these methods have proven insufficient against sophisticated re-identification attacks, which exploit auxiliary information to link anonymized records back to individuals [173].

The “Fundamental Law of Information Recovery” [68] reveals a trade-off between data utility and privacy: answering a sufficient number of seemingly innocuous queries about a database inevitably reveals sensitive information about it [56]. As a consequence, any useful computation on a dataset necessarily results in some privacy loss.

Heuristic approaches to privacy have been repeatedly shown to fail, so robust mathematical frameworks are desirable to ensure privacy without compromising utility. This need for formal guarantees has catalyzed the development of differential privacy (DP), a mathematically rigorous framework that quantifies and bounds the privacy risk posed by data analysis [64], offering a safeguard against inference attacks [52] [195].

The central tenet of differential privacy is indistinguishability. A randomized algorithm (or mechanism) is considered differentially private if its output behavior remains statistically similar regardless of whether any single individual data is included in or excluded from the input dataset. The intuition is that if an observer cannot reliably determine if a particular person’s data was used in the computation, then the computation reveals little specific information about that person. This provides a strong guarantee: the privacy risk incurred by participating in a differentially-private analysis is minimal, roughly equivalent to the risk if one’s data had not been included at all. This concept provides a form of plausible deniability for participants.

Authors’ Contact Information: Francisco Aguilera-Martínez, faguileramartinez@acm.org; Fernando Berzal, berzal@acm.org, Department of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain.

Differential privacy is a property of the algorithm or mechanism performing the analysis, not a property of the dataset itself. To achieve this indistinguishability, DP mechanisms must be randomized. Deterministic algorithms cannot satisfy the definition as their output is fixed for a given input. This inherent randomization introduces carefully calibrated noise or uncertainty into the computation, masking the contribution of any single individual while preserving the statistical utility of the overall result. The guarantee provided by DP is relative and probabilistic, bounding the change in output probabilities rather than promising absolute secrecy, acknowledging the fundamental trade-off between utility and privacy.

Differential privacy [64] defines privacy through the lens of a worst-case guarantee: the probability of any output from a data analysis algorithm changes by at most a factor of e^ϵ when a single individual’s data is added or removed, with a small failure probability δ . This ϵ, δ -DP framework provides a quantifiable measure of privacy loss, enabling precise trade-offs between privacy and utility. Its strength lies in its resilience to arbitrary auxiliary information, making it robust against attacks such as membership inference, where adversaries attempt to determine whether an individual’s data was used in training [195]. Beyond its theoretical elegance, DP has spurred a rich ecosystem of algorithmic innovations tailored to machine learning, addressing challenges like high-dimensional data, iterative computations, and distributed systems [68].

The concept of tracker [53], an adversary that could learn the confidential contents of a database by creating a series of targeted queries and remembering their results, led to the conclusion that the privacy properties of a database can only be preserved by considering each new query in light of (possibly all) previous queries. This led to the inclusion of privacy accountants in DP-preserving ML algorithms, which is used to track and manage the cumulative privacy loss, so that the total privacy budget is not exceeded over multiple data queries.

Subsequent research advancements have refined DP’s theoretical foundations and practical utility. Methods have been proposed to optimize the privacy-utility trade-off, particularly in settings requiring iterative data access, such as the algorithms used to train ML models.

Common use cases of DP algorithms include mobile devices, healthcare, and financial data. The first deployment of DP was made by the U.S. Census Bureau, for displaying the commuting patterns of the population of the United States [154]. The first widespread use of local differential privacy was made by RAPPOR [72] for collecting Google Chrome browser telemetry data. Microsoft also collects telemetry data privately in Windows [55]. Predictive text or voice recognition models trained across user devices use privacy-preserving federated learning (e.g. Google’s Gboard uses DP to improve keyboard suggestions without compromising user privacy [228]). Hospitals collaboratively train models on patient data without sharing sensitive records. Banks train fraud detection models without exposing customer data. DP adoption in industry applications, from Google’s [72] or Uber’s [119] private query analysis to Apple’s emoji usage modeling [201], underscores its practical relevance [68].

The integration of differential privacy into machine learning has attracted the attention of researchers and practitioners for years. Our survey on the use of DP in ML is structured as follows. First, we introduce DP theoretical foundations, including its original formal definition, the role of sensitivity, multiple variants that have been proposed over the years, and their formal properties, with an emphasis on their composability, their robustness to post-processing, and their potential degradation in the presence of correlated data. Next, we examine how DP has been applied in different kinds of ML models, from symbolic and probabilistic AI to deep learning and model-agnostic techniques, such as federated learning and PATE [178]. Finally, we discuss how DP effectiveness can be evaluated in practice and the privacy-utility trade-off common to DP deployments. Through this exploration, we seek to contribute to the ongoing research on privacy-preserving technologies in an increasingly data-centric world.

2 Differential privacy

Differential privacy (DP) provides a mathematical definition for the privacy loss associated with any data release drawn from a database ¹. DP establishes a mathematically rigorous framework to ensure that the presence or absence of a single individual in a dataset does not significantly affect the outcome of a computation. DP protects the privacy of individual data examples by injecting carefully calibrated noise into statistical computations. Statistical information from the dataset can still be used (e.g. aggregate patterns) while provably limiting what can be inferred about any individual in the dataset.

2.1 Formal definition

Intuitively, a model M exhibits differential privacy if its answers are indistinguishable when trained over datasets that differ in just one individual data example.

2.1.1 ϵ -differential privacy: Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith introduced ϵ -differential privacy in 2006 [64]. Differential privacy measures privacy risk by a parameter ϵ that bounds the log-likelihood ratio of the output of a (private) algorithm under two databases differing in a single individual's data.

Formally, a non-interactive mechanism M (e.g. a machine learning model) satisfies ϵ -differential privacy if, for all neighboring datasets x, x' (i.e., datasets differing in at most one example) and for all measurable subsets of outputs S , the following non-strict inequality holds:

$$\left| \log \left(\frac{P[M(x) \in S]}{P[M(x') \in S]} \right) \right| \leq \epsilon$$

which is typically rewritten as

$$P[M(x) \in S] \leq e^\epsilon P[M(x') \in S]$$

The parameter $\epsilon > 0$, known as privacy loss or leakage, controls the level of privacy. Smaller values of ϵ provide stronger privacy guarantees, ensuring that the distributions induced by $M(x)$ and $M(x')$ remain nearly indistinguishable.

When ϵ is small, $\log(1 + \epsilon) \approx \epsilon$, so the initial definition of ϵ -indistinguishability, or ϵ -differential privacy, is roughly equivalent to

$$\frac{P[M(x) \in S]}{P[M(x') \in S]} \approx 1 \pm \epsilon$$

The definition of ϵ -differential privacy requires that a change to one example in the training data only creates a small change in the model output probability distribution, as seen by a potential attacker. Since deterministic mechanisms cannot satisfy this definition except in trivial cases, DP inherently requires randomization in the output generation process.

2.1.2 (ϵ, δ) -differential privacy: While ϵ -differential privacy provides strong guarantees, there are scenarios where it may be too restrictive. In practice, a relaxation known as (ϵ, δ) -differential privacy [62] allows for a small probability δ of the privacy guarantee being violated, making it more suitable for real-world applications.

¹DP was originally described in terms of statistical databases [64], sets of data that are collected under the pledge of confidentiality for the purpose of producing statistics that, by their production, do not compromise the privacy of those individuals who provided the data. In other words, since a person's privacy cannot be compromised by a statistical release if their data are not in the database, in differential privacy, each individual is given roughly the same privacy that would result from having their data removed. That is, the statistical functions run on the database should not be substantially affected by the removal, addition, or change of any individual in the data.

Formally, a randomized mechanism M with domain $N^{|X|}$ satisfies (ϵ, δ) -differential privacy if, for all neighboring datasets x, x' such that $\|x - x'\|_1 \leq 1$ and for all measurable output subsets S , the following property holds [68]:

$$P[M(x) \in S] \leq e^\epsilon P[M(x') \in S] + \delta.$$

When $\delta = 0$, (ϵ, δ) -differential privacy reduces to pure ϵ -differential privacy.

In (ϵ, δ) -differential privacy, also known as approximate differential privacy, δ represents the probability that the privacy guarantee may be broken, which is typically required to be negligible, often less than the inverse of any polynomial in the database size $|x|$.

Differential privacy can be interpreted as bounding the error rates of a hypothesis test [122]. Given a random output y of the mechanism M , we choose the null hypothesis H_0 , when y came from a database without a particular individual example, and the alternative hypothesis H_1 , when y came from a database with the particular example. We have two error rates: the false positive rate (FPR) or probability of a false alarm (i.e. type I error, the adversary guesses H_1 when H_0 is true) and the false negative rate (FNR) or probability of a missed detection (i.e. type II error, the adversary guesses H_0 when H_1 is true). The differential privacy condition on a mechanism M is equivalent to the following set of constraints on the probability of false alarm and missed detection for a fixed (ϵ, δ) :

$$P_{\text{FP}} + e^\epsilon P_{\text{FN}} \geq 1 - \delta$$

$$e^\epsilon P_{\text{FP}} + P_{\text{FN}} \geq 1 - \delta$$

In other words, it is impossible to get both small false positive rates and false negative rates from data obtained via a differentially private mechanism [122]. Actually, this trade-off is common in many ML scenarios. Reducing false positives often leads to more false negatives, and vice versa. If the criteria for identifying a positive case are very strict (to reduce false positives), we may miss some actual positive cases (leading to false negatives). Conversely, if the criteria are very lenient (to reduce false negatives), we may incorrectly identify negative cases as positive (leading to false positives).

2.2 Sensitivity

How much does any individual contribute to the result of a database query or a training example to the model output? As the query is made on fewer and fewer examples, more noise needs to be added to the query result to produce the same amount of privacy.

A key concept in the implementation of differential privacy is sensitivity, which quantifies the maximum possible change in the output of a function when a single example in the dataset is modified. Specifically, the sensitivity of a function f is defined as [64]:

$$\Delta f = \max_{x, x': d(x, x')=1} \|f(x) - f(x')\|_1$$

where x and x' differ by a single example. Since this sensitivity is determined by all possible pairs (x, x') , it is also known as global sensitivity [174].

Adding noise proportional to the global sensitivity of a function can obscure individual contributions while preserving aggregate statistics, as demonstrated by the SuLQ framework [21].

It should be noted that the noise magnitude depends on the global sensitivity and the privacy parameter ϵ , but not on the particular instance x . This noise might be higher than needed, since it does not reflect the function's typical

insensitivity to individual inputs. Hence a similar local sensitivity [174] can be defined for a particular instance x :

$$\Delta_x f = \max_{x': d(x, x')=1} \|f(x) - f(x')\|_1$$

Sensitivity is essential for determining how much noise needs to be added to the output to ensure privacy [104]. For many functions, the global sensitivity framework yields unacceptably high noise levels, whereas adding noise proportional to a smooth upper bound to the local sensitivity might still be safe (the smooth sensitivity framework [174]).

To achieve ϵ -differential privacy, noise is typically introduced into the results of the differential privacy mechanism. In other words, differential privacy relies on adding carefully calibrated random noise to the function that we want to compute. Common distributions for noise generation include the continuous Laplace and Gaussian distributions for functions that output real-valued numbers, as well as the geometric distribution for discrete outputs. Other mechanisms, such as the exponential mechanism and posterior sampling, sample from a problem-dependent family of distributions instead.

- The *Laplace mechanism* adds noise sampled from a Laplace distribution with scale $\Delta f / \epsilon$ and mean 0. Given a query function f , the private response mechanism is:

$$\mathcal{M}_{\text{Laplace}}(x) = f(x) + \text{Laplace}\left(\mu = 0, \lambda = \frac{\Delta f}{\epsilon}\right)$$

where $\text{Laplace}\left(\mu = 0, \lambda = \frac{\Delta f}{\epsilon}\right)$ represents the Laplace noise that provides the necessary randomness to ensure that the privacy requirement is met [64].

- The *Gaussian mechanism* adds noise from the Gaussian distribution (which requires the L2 norm in the definition of sensitivity) instead of the Laplace distribution. The variance of the Gaussian distribution is calibrated according to the sensitivity and privacy parameters to provide (ϵ, δ) -differential privacy [68]:

$$\mathcal{M}_{\text{Gauss}}(x) = f(x) + \mathcal{N}\left(\mu = 0, \sigma^2 = \frac{2 \ln(1.25/\delta) \cdot (\Delta f)^2}{\epsilon^2}\right)$$

- A discrete variant of the Laplace mechanism, called the *geometric mechanism*, is universally utility-maximizing [93] [94] [95], optimizing expected loss across various priors and symmetric, monotone univariate loss functions through a data-independent post-processing step. For any prior (such as auxiliary information or beliefs about data distributions) and any symmetric and monotone univariate loss function, the expected loss of any differentially private mechanism can be matched or improved by running the geometric mechanism followed by a data-independent post-processing transformation. The result also holds for minimax (risk-averse) consumers [101]. However, no such universal mechanism exists for multi-variate loss functions [27] [28].
- The *exponential mechanism* [161] comes with guarantees about the quality of the output, even for functions that are not robust to additive noise, and those whose output may not even permit perturbation. The mechanism $\mathcal{E}(x)$ chooses an output y with probability proportional to $e^{\epsilon q(x, y)} \times \mu(y)$, where x is the input, q is a scoring function that reflects the appeal of the (x, y) pair and $\mu(y)$ is a base measure. This mechanism skews a base measure to the largest degree possible while ensuring differential privacy, focusing probability on the outputs of highest value. The mechanism $\mathcal{E}(x)$ provides $(2\epsilon\Delta q)$ -differential privacy, where Δq measures the change in q due to a single change in x .

- A *posterior sampling mechanism* [54] is also differentially private and provides finite sample bounds for distinguishability-based privacy under a strong adversarial model.

When adding noise to ensure privacy may not be appropriate, such as when the function being analyzed is not continuous or involves discrete structures like strings or trees, posterior sampling models, such as the exponential mechanism, are required for privacy preservation.

Challenges also arise when sensitivity is difficult to assess, yet both Laplace and exponential mechanisms remain useful in these cases. In iterative algorithms, such as clustering methods (e.g. k-means and DBSCAN density-based clustering), privacy can still be maintained by applying some algorithmic techniques. Such algorithmic techniques include releasing approximate counts for each cluster instead of exposing individual data points (using the geometric mechanism, the optimal and universal discrete extension of the Laplace mechanism), implementing differentially-private data structures (e.g. a differentially private region quadtree [108] for protecting GPS locations while keeping a reduced local sensitivity), or generating synthetic data, where the synthetic dataset mimics the statistical properties of the original dataset.

It should also be noted, however, that a vulnerability is present in many implementations of differentially private mechanisms. The vulnerability [166] is based on irregularities of floating-point implementations of the privacy-preserving Laplacian mechanism. Unlike its mathematical abstraction, the textbook sampling procedure results in a porous distribution over double-precision numbers that allows one to breach differential privacy with just a few queries into the implementation of the differentially-private mechanism.

Pan-privacy extends differential privacy [61]: A pan-private algorithm is private “inside and out,” remaining differentially private even if its internal state becomes visible to an adversary [67]. Not only the outputs, but also the distributions on the internal states of the algorithm should be similar for slightly different input data.

Beyond its privacy-preserving benefits, differential privacy also imposes an inherent limitation on the influence of any single individual data on the final output. This characteristic has profound implications beyond privacy, particularly in strategic settings where individuals may have incentives to manipulate outcomes. Differentially-private mechanisms naturally exhibit approximate strategy-proofness. In mechanism design, a strategyproof mechanism is a game form in which each player has a weakly-dominant strategy, so that no player can gain by spying over the other players to know what they are going to play. In other words, individual participants or coalitions have limited ability to alter the outcome in their favor. This insight extends to machine learning applications, where training models with differential privacy ensures that no single data point disproportionately affects the model parameters, thus improving the model robustness and generalization capability [161].

2.3 Extensions and variants

In the landscape of privacy-preserving data analysis, the foundational framework of differential privacy was initially introduced with a strict worst-case guarantee parameterized by ϵ -sets as the stage for subsequent refinements.

A set of dimensions [181] capture how the original DP definition can be changed using different ideas, so that each variant or extension can be viewed as a point in a multidimensional space:

- Quantification of privacy loss (how is the privacy loss quantified across outputs?) for averaging risk and obtaining better composition properties.
- Neighborhood definition (which properties are protected from the attacker?) for protecting specific values or multiple individuals.

Definition	Description (idea)	Reference
ϵ -DP	Original idea (ϵ -indistinguishability)	[222] [64]
(ϵ, δ) -DP	Approximate DP (δ failure probability)	[62]
(ϵ, δ) -PDP	Probabilistic DP (probabilistic relaxation)	[164]
ϵ -KLP	Kullback–Leibler privacy (mutual information)	[48]
(α, ϵ) -RDP	Rényi differential privacy (Rényi divergence)	[167]
(μ, τ) -mCDP	Mean-concentrated DP (tail distribution)	[69]
(ξ, ρ) -zCDP	Zero-concentrated DP (tail distribution)	[32] [33]
(f, ϵ) -DDP	Divergence DP (divergence function)	[8]
(P, ϵ) -OSDP	One-sided DP (sensitive records)	[133]
(R, c, ϵ) -DDP	Dependent differential privacy (data correlation)	[145]
ϵ -FLP	Free-lunch privacy (strongest variant, no utility)	[129]
(\mathcal{R}, ϵ) -GDP	Generic DP (generic neighboring relation)	[129]
$(G, \mathcal{I}_Q, \epsilon)$ -BP	Blowfish privacy (generic DP inspired by the Pufferfish framework)	[107]
f -DP	f -differential privacy (hypothesis testing interpretation)	[57]
GDP	Gaussian differential privacy (hypothesis testing interpretation)	[57]
(D, ϵ) -IDP	Individual DP (for a particular dataset), a.k.a. conditioned DP	[200] [38]
LDP	Local differential privacy (for single data examples)	[125]

Table 1. Some variants and extensions of the original DP definition.

- Variation of privacy loss (can the privacy loss vary across inputs?) for modeling users with different privacy requirements.
- Background Knowledge (how much prior knowledge does the attacker have?) for using mechanisms that add less or no noise to data.
- Formalism change (which formalism is used to describe the attacker’s knowledge gain?) for exploring other intuitive notions of privacy.
- Relativization of knowledge gain (what is the knowledge gain relative to?) for guaranteeing privacy for correlated data.
- Computational power (how much computational power can the attacker use?) for combining cryptography techniques with DP.

Table 1 provides an overview of some variants and extensions of differential privacy that have been proposed in the literature. In the following sections, we examine the most relevant among them, beginning with concentrated DP, Rényi DP, and Gaussian DP, and concluding with local DP.

2.3.1 Concentrated Differential Privacy (CDP). Concentrated Differential Privacy (CDP) [69] is a refinement of the traditional (approximate) differential privacy, i.e. (ϵ, δ) -DP, designed to provide stronger privacy guarantees while improving accuracy in data analysis, which makes it particularly useful in machine learning.

In essence, CDP fine-tunes the privacy mechanism to provide better results without compromising security. Instead of applying a uniform level of randomness, it allows for a tighter, more efficient balance between privacy and accuracy. This balance ensures that the privacy loss remains small most of the time, making it easier to get meaningful insights while still protecting individual identities.

Its name derives from shifting from a uniform bound on privacy loss to a probabilistic concentration approach, which offers enhanced flexibility and tighter composition bounds for iterative data queries. CDP introduces a sophisticated

relaxation of the classical model, shifting the focus from absolute bounds on privacy loss to a probabilistic characterization that leverages concentration properties, an approach that merits rigorous examination for its theoretical and practical implications.

Formally, CDP defines privacy in terms of a bound on the moment generating function of the privacy loss random variable, rather than using a strict probabilistic constraint as in traditional differential privacy. Given a mechanism \mathcal{M} that operates on a dataset, CDP ensures that for all possible outputs S ,

$$E[e^{\lambda L}] \leq e^{\frac{\lambda^2 \sigma^2}{2}}$$

where L is the privacy loss random variable, σ^2 is a privacy parameter controlling variance (a.k.a. a variance proxy), and λ is any real number.

The privacy loss random variable measures how much information an adversary can gain about an individual's data when a privacy-preserving mechanism is applied. Given a randomized mechanism \mathcal{M} and any pair of neighboring datasets x and x' differing in a single record, the privacy loss random variable is defined as

$$L(\mathcal{M}, D, D') = \log \frac{P[\mathcal{M}(x) \in S]}{P[\mathcal{M}(x') \in S]}$$

for the possible outputs S of \mathcal{M} , where $P[\mathcal{M}(D) \in S]$ and $P[\mathcal{M}(x') \in S]$ represent the probabilities of obtaining S when the mechanism is applied to x and x' , respectively.

Formally, a mechanism \mathcal{M} satisfies (μ, σ^2) -mCDP, mean-concentrated differential privacy [69], if

$$E[e^{\lambda(L-\mu)}] \leq e^{\frac{\lambda^2 \sigma^2}{2}} \text{ for all } \lambda \in \mathbb{R},$$

thereby ensuring that the moment-generating function of the centered privacy loss is dominated by that of a Gaussian distribution.

We say that CDP exhibits sub-Gaussian tail behavior [121], with mean bounded by μ ($E[L] \leq \mu$) and variance proxy σ^2 , a.k.a. τ ($Var[L] \leq \sigma^2$) [30] [187], when we refer to the way its privacy loss (the randomness in how much an individual's data can influence the outcome) is controlled. In CDP, the privacy loss random variable satisfies a concentration property that limits the likelihood of extreme privacy breaches (i.e. large deviations from expected privacy loss are rare)².

Zero-concentrated differential privacy (zCDP) refines CDP by defining the privacy loss using a differential privacy-like bound in terms of a quadratic constraint on the Rényi divergence between distributions [186][212]. This allows for stronger composition properties (when multiple privacy-preserving operations are applied, zCDP ensures that the accumulated privacy loss remains well-controlled) and improved accuracy (tighter bounds on noise addition, leading to better utility in machine learning and statistical analysis).

A mechanism \mathcal{M} satisfies ρ -zCDP, zero-concentrated differential privacy [32], if, for all neighboring datasets x and x' , $D_\alpha(\mathcal{M}(x) \parallel \mathcal{M}(x')) \leq \rho(\alpha - 1)$ where D_α represents the Rényi divergence of order α (for any order $\alpha > 1$). Larger values of α correspond to tighter privacy bounds, tuning sensitivity to rare events, whereas ρ controls the strength of the privacy guarantees (a privacy budget that controls how much information can leak when analyzing data, so that smaller ρ leads to stronger privacy).

²A random variable has sub-Gaussian tail behavior if its probability of taking extreme values (very large deviations from the mean) drops exponentially, rather than following a heavy-tailed distribution.

The Rényi divergence is a way to measure how different two probability distributions are. The Rényi divergence of positive order ($\alpha \neq 1$) of a probability distribution P from another probability distribution Q is

$$D_\alpha(P \parallel Q) = \frac{1}{1-\alpha} \log \sum_x p(x)^\alpha q(x)^{1-\alpha}$$

where, for $\alpha > 1$, $p(x)^\alpha q(x)^{1-\alpha} = p(x)^\alpha / q(x)^{\alpha-1}$. This definition generalizes to continuous spaces by replacing the probabilities by densities and the sum by an integral. When the order $\alpha \rightarrow 1$, the Rényi divergence reduces to the Kullback-Leibler (KL) divergence [135], commonly used in information theory: $D_1(P \parallel Q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$. When $\alpha > 1$, the Rényi divergence gives more weight to larger differences between the distributions, making it more sensitive to rare events. As $\alpha \rightarrow \infty$, the divergence focuses only on the largest discrepancy between the two distributions, which is related to min-entropy (i.e. a measurement of the worst-case uncertainty, $H_\infty(X) = -\log \max_x p(x)$, how much uncertainty remains if we assume the most likely event will happen). The max-divergence is the worst-case relative difference between distributions: $D_\infty(P \parallel Q) = \log \max_x \frac{p(x)}{q(x)}$.

In ρ -zCDP, $\mathbb{E} \left[e^{(\alpha-1)L} \right] \leq e^{\rho\alpha}$ and $\mathbb{E} \left[e^{\lambda L} \right] \leq e^{\lambda^2 \rho / 2}$. The privacy loss random variable, whose moments are now bounded,³ is sub-Gaussian with mean approximately ρ and variance bounded by 2ρ . Its definition implies (ϵ, δ) -DP with $\epsilon = \rho + 2\sqrt{\rho \ln(1/\delta)}$ [32].

In general, in (ξ, ρ) -zCDP, $\mathbb{E} \left[e^{\lambda L} \right] \leq e^{\lambda \xi + \lambda^2 \rho / 2}$, where ξ represents the expected privacy loss (mean privacy leakage) and ρ controls the spread (variance) of the privacy loss, preventing extreme deviations. Its definition implies (ϵ, δ) -DP with $\epsilon = \xi + \rho + 2\sqrt{\rho \ln(1/\delta)}$ [32].

The formulation of ρ -zCDP, i.e. $(0, \rho)$ -zCDP, leverages the Rényi divergence with adjustable sensitivity through the parameter α to establish a flexible and analytically robust privacy guarantee. The linear constraint $\rho\alpha$ on the divergence ensures tight concentration of the privacy loss, setting zCDP apart from pure differential privacy, which depends on a worst-case exponential bound, and approximate differential privacy, which tolerates a failure probability.

Four more variants of concentrated differential privacy exist. Approximate zCDP [32] relaxes zCDP by only taking the Rényi divergence on events with high probability instead of on the full distribution, so that δ -approximate $(\epsilon, 0)$ -zCDP is equivalent to (ϵ, δ) -DP. Bounded CDP [32] relaxes zCDP by changing the inequality in the zCDP definition so that $D_\alpha(\mathcal{M}(x) \parallel \mathcal{M}(x')) \leq \xi + \rho\alpha$ for $\alpha \in \{1, m\}$ (i.e. the inequality holds only for the first m orders of the Rényi divergence). Finally, two definitions share the truncated CDP name, where the first [46] relaxes $(0, \rho)$ -zCDP the same way that bounded CDP and the second [31] requires the Rényi divergence to be smaller than $\min\{\xi, \alpha\rho\}$ for all $\alpha \geq 1$.

2.3.2 Rényi Differential Privacy (RDP). RDP is another variant that also uses moment generating functions to analyze privacy loss. Like CDP, it provides stronger composition properties, making it easier to track cumulative privacy loss in iterative computations (e.g., machine learning).

RDP is closely related to zCDP is closely related but differs in how it constrains the privacy loss, offering a more concentrated bound. A randomized mechanism \mathcal{M} is said to satisfy Rényi Differential Privacy (RDP) of order $\alpha > 1$ with privacy parameter ϵ if for all neighboring datasets x, x' that differ in at most one element, the following holds [167]:

$$D_\alpha(\mathcal{M}(x) \parallel \mathcal{M}(x')) \leq \epsilon$$

³For a random variable X , the moment-generating function is defined as $M_X(t) = E[e^{tX}]$, where e^{tX} transforms the probability distribution into a function that encodes all its moments. The MGF provides a systematic way to control and compute moments. The first derivative at $t = 0$ gives the first moment, expected value, or mean: $E[X] = M'_X(0)$. The second derivative helps compute variance: $\text{Var}[X] = M''_X(0) - M'_X(0)^2$. The n -th derivative gives the n -th moment: $E[X^n] = M^{(n)}_X(0)$. In zCDF, the privacy loss random variable satisfies $\mathbb{E} \left[e^{(\alpha-1)L} \right] \leq e^{\rho\alpha}$ for all $\alpha > 1$, ensuring that privacy leakage is bounded across all moments. This leads to predictable privacy guarantees and strong composition properties.

where D_α is the Rényi divergence [186] of order α . The constraint $\leq \rho(\alpha - 1)$ in zCDP is replaced just by $\leq \epsilon$ in RDP.

As zCDP, RDP extends the classical notion of differential privacy (DP) by leveraging Rényi divergence, a generalization of the the Kullback-Leibler (KL) divergence [135] that ranges from the Kullback-Leibler (KL) divergence itself when $\alpha \rightarrow 1$ to the max-divergence at $\alpha \rightarrow \infty$. From an information-theoretic perspective, RDP quantifies the privacy leakage by bounding the divergence between perturbed probability measures, thereby providing a principled and mathematically rigorous foundation for privacy-preserving algorithms in modern machine learning and statistical inference [167].

RDP offers superior composition properties, which allow for a tighter and more analytically tractable characterization of the cumulative privacy loss in iterative processes such as differentially private stochastic gradient descent (DP-SGD).

RDP provides a more flexible framework for privacy accounting, particularly in high-dimensional settings where privacy amplification via subsampling plays a fundamental role.

Furthermore, RDP retains key invariance properties under post-processing and can be converted to (ϵ, δ) -DP guarantees, thereby establishing a bridge between privacy notions that are crucial in practical deployments. Given an (α, ϵ_{RDP}) -RDP mechanism, we can derive an equivalent (ϵ_{DP}, δ) -DP mechanism using the following relationship [167]:

$$\epsilon_{DP} = \epsilon_{RDP} + \frac{\log(1/\delta)}{\alpha - 1}$$

where α is the Rényi divergence order, ϵ_{RDP} is the RDP privacy parameter, δ is the failure probability parameter in DP (indicating rare privacy breaches), and ϵ_{DP} is the traditional DP privacy loss parameter. The stricter the privacy requirement (lower δ), the more privacy loss we account for in DP (higher ϵ_{DP}). Larger Rényi divergence orders allows for better privacy composition over multiple steps, tightening DP bounds.

When a mechanism is (α, ϵ_{RDP}) -RDP, it is also (ϵ_{DP}, δ) -DP for any $\epsilon_{DP} > \epsilon_{RDP}$ and $\delta = e^{-(\alpha-1)(\epsilon_{DP}-\epsilon_{RDP})}$. This relationship is used, for instance, in the moments accountant (MA) [1], a privacy analysis technique for ML algorithms. However, the approximation is loose and does not hold for all $\epsilon_{DP} \geq 0$.

Given an (α, ϵ_{RDP}) -RDP mechanism, what are the smallest ϵ_{DP} and δ so that the mechanism is also (ϵ_{DP}, δ) -DP? [5] relates approximate DP to RDP based on the joint range of two f-divergences that underlie the approximate DP and RDP. This enables us to derive the optimal approximate DP parameters of a mechanism that satisfies a given level of RDP. An (α, ϵ_{RDP}) -RDP mechanism provides $(0, \delta)$ -DP if $\epsilon_{RDP} < \log \frac{\alpha}{\alpha-1}$ and $\delta \in [\zeta_\alpha e^{(\alpha-1)\epsilon_{RDP}}, \frac{1}{\alpha}]$, where $\zeta_\alpha = \frac{1}{\alpha} \left(1 - \frac{1}{\alpha}\right)^{\alpha-1}$.

Given an (α, ϵ_{RDP}) -RDP mechanism, what is the characterization of its privacy region? A relationship between RDP and the hypothesis test interpretation of DP allows us to translate the RDP constraint into a tradeoff between type I and type II error probabilities of a certain binary hypothesis test [5].

Finally, it should also be noted that the Gaussian mechanism with variance σ^2 is (α, ϵ_{RDP}) -RDP for $\epsilon_{RDP} = \rho\alpha$ with $\rho = \frac{1}{2\sigma^2}$. Given that, for Gaussian noise, the Rényi divergence of order α is $D_\alpha = \alpha/2\sigma^2$ and RDP requires the Rényi divergence to be bounded by $\epsilon_{RDP} = \rho\alpha$, $\rho = \frac{1}{2\sigma^2}$.

2.3.3 Gaussian Differential Privacy (GDP). Instead of using the Rényi divergence or the standard DP framework, GDP specifically models privacy loss as a Gaussian distributions [57]. This allows privacy guarantees to be expressed using a trade-off function, ensuring optimal trade-offs between privacy and utility. While GDP has strong theoretical properties, CDP provides a more flexible framework and is applicable to a broader range of algorithms.

Given a mechanism \mathcal{M} and two neighboring datasets x and x' , the trade-off function $T(\mathcal{M}(x), \mathcal{M}(x'))$ describes how well an adversary can distinguish between them. More specifically, the trade-off function quantifies the best statistical test an adversary can use to distinguish outputs from two distributions.

Mathematically, for any acceptance threshold τ , which represents the probability of accepting an event under two distributions, the trade-off function is

$$T(P, Q)(\tau) = \sup_{\text{test } A} [P(A)] \text{ subject to } Q(A) \leq \tau$$

where P and Q are the probability distributions over the outputs of $\mathcal{M}(x)$ and $\mathcal{M}(x')$, while A represents any decision rule (statistical test) that aims to distinguish between the distributions.

The trade-off function is always non-negative and monotonically decreasing as τ increases (as we allow more false positives in distinguishing the two datasets, the probability of correctly identifying P also increases). Swapping P and Q results in a reflected trade-off function $T(Q, P)(\tau) = 1 - T(P, Q)(1 - \tau)$, which ensures that privacy analysis accounts for adversaries testing either dataset as the true distribution.

Gaussian Differential Privacy (GDP) is articulated as a specialized instance of f -differential privacy (f -DP), a generalization of differential privacy that frames privacy as a hypothesis testing problem.

A randomized mechanism \mathcal{M} satisfies μ -GDP if, for all neighboring datasets x and x' differing by a single element, the trade-off function $T(\mathcal{M}(x), \mathcal{M}(x'))$ is bounded by a Gaussian trade-off function with mean μ :

$$T(\mathcal{M}(x), \mathcal{M}(x')) \leq T(N(0, \mu))$$

where $N(0, \mu)$ represents a Gaussian mechanism centered at zero with privacy noise controlled by μ (larger μ indicates weaker privacy).

In GDP, privacy loss can be formalized using hypothesis testing, particularly in terms of Type I and Type II errors in distinguishing between two neighboring datasets. GDP characterizes privacy in terms of the best achievable trade-off between Type I and Type II errors in distinguishing between two distributions:

$$\inf_A \{\beta_A(\alpha)\} \geq T(N(0, \mu))(\alpha)$$

where A is a rejection rule (a statistical decision test used to differentiate between the probability distributions over the outputs of a mechanism applied to two neighboring datasets), α is the Type I error (the probability of falsely rejecting the null hypothesis, i.e. false positive rate), $\beta_A(\alpha)$ is the Type II error (the probability of failing to reject the null hypothesis when it's false, i.e. the false negative rate), the infimum is taken over all possible rejection rules A , ensuring the best possible trade-off, and $T(N(0, \mu))(\alpha)$ is the Gaussian trade-off function, ensuring that the privacy loss follows a Gaussian distribution with mean μ .

GDP guarantees that no adversary can achieve a better trade-off between false positives and false negatives than what is dictated by a Gaussian mechanism. Since GDP expresses privacy in terms of distinguishability between neighboring datasets using the best achievable trade-off between Type I and Type II errors and, for the Gaussian mechanism, this trade-off is bounded by a Gaussian trade-off function with mean $\mu = \Delta f / \sigma$, where Δf is the maximum possible change in the function output when one data point is modified and σ determines how much noise is added, adding Gaussian noise inherently satisfies μ -GDP, with privacy loss following a Gaussian distribution.

GDP aligns privacy loss with the normal distribution, making composition analysis much smoother in large-scale applications, the key advantage of GDP. Privacy loss accumulates like a sum of Gaussian variables, following the central limit theorem: when applying multiple privacy-preserving steps, the cumulative privacy loss follows a Gaussian

distribution. If a mechanism satisfies μ -GDP, then the trade-off function under composition approximates a normal (Gaussian) curve.

The GDP guarantee that privacy loss follows a normal (Gaussian) distribution with mean μ can be converted into the traditional DP framework by relating the GDP privacy parameter μ to the (ϵ, δ) -DP parameters. GDP can be converted to (ϵ, δ) -DP using the following relationship:

$$\delta(\epsilon) = \Phi\left(-\frac{\epsilon}{\mu} + \frac{\mu}{2}\right) - e^{-\epsilon} \cdot \Phi\left(-\frac{\epsilon}{\mu} - \frac{\mu}{2}\right)$$

where Φ is the standard normal CDF. This relationship gives a tight bound on δ for given ϵ and μ . You can also numerically invert it to find ϵ for a given δ . For any desired δ , the corresponding ϵ is given by $\epsilon = \mu + \sqrt{2 \log(1/\delta)}$.

For stronger privacy, lower δ means fewer permitted violations and leads to higher effective ϵ . Higher μ means that the privacy loss is greater and leads to weaker DP guarantees (higher ϵ). The privacy parameter in GDP plays a similar role to ϵ in DP, representing how much information an adversary can gain.

The GDP formulation marks a profound shift in differential privacy [64] by reinterpreting privacy through the lens of trade-off functions, offering a more granular and exact alternative to traditional metrics like Rényi divergence [212] or the (ϵ, δ) -DP bounds, which often rely on loose approximations under composition. Parameterization by μ encapsulates privacy as the effective separation in a Gaussian distinguishability test, providing an intuitive yet mathematically precise measure that simplifies analysis across diverse applications. Unlike prior frameworks, GDP reliance on the Gaussian trade-off function yields closed-form expressions for composition, eliminating the need for numerical approximations or conservative bounds. Its asymptotic convergence property, which is rooted in a central limit theorem, positions it as a universal endpoint for privacy loss accumulation.

GDP is particularly useful in iterative algorithms, where multiple computations are performed sequentially, such as in stochastic optimization, machine learning training, and data analysis workflows. The key advantage of GDP in these scenarios is its ability to provide accurate privacy composition over many iterations, while maintaining strong privacy guarantees. For instance, in stochastic gradient descent [189], tight privacy accounting is critical.

GDP is also useful in subsampling scenarios, where a mechanism operates on a randomly chosen subset of a dataset rather than the full dataset. Subsampling improves privacy by reducing an adversary's ability to infer individual participation, and GDP provides a natural way to quantify this effect. When a mechanism selects a random subset of data instead of the full dataset, the sensitivity of individual data points decreases. Since privacy loss accumulates like a Gaussian distribution in GDP, subsampling reduces the effective privacy parameter μ . Given a mechanism that satisfies μ -GDP, subsampling reduces the effective privacy loss according to $\mu_{\text{subsampled}} = q \cdot \mu$, where q is the subsampling rate. When only a fraction q of the data is used, the privacy guarantees scale proportionally, leading to better privacy protection. In traditional DP, subsampling amplifies privacy guarantees, meaning the effective (ϵ, δ) -DP parameters improve. DP naturally captures this amplification effect through its Gaussian trade-off function. In DP-SGD [1], which subsamples mini-batches of data at each step, GDP provides a direct way to measure privacy loss over multiple subsampling rounds, making it ideal for deep learning applications.

2.3.4 Local differential privacy. Differential privacy can be described as a constraint on the algorithms used to publish aggregate information about a statistical database which limits the disclosure of private information of records in the database. An algorithm is differentially private if an observer seeing its output cannot tell whether a particular individual's information was used in the computation. [65] In local differential privacy [59], true data is kept hidden

from even the data scientist working on it, by applying differentially private mechanisms to the data, before any statistics are computed.

Differentially private algorithms work in two main modalities: curator and local [9]. The curator model assumes a trusted centralized curator that collects all the personal information and then analyzes it. In contrast, the local model enables privacy without a trusted curator. The local model does not involve a central repository and is suitable for distributed systems. Each piece of personal information is randomized by its provider to protect privacy even if all information provided to the analysis is revealed. A disadvantage of the local model is that it requires introducing noise at a significantly higher level than what is required in the curator model [9].

Local differential privacy (LDP) [125] [124] is designed for scenarios where each user perturbs their own data before sharing it, making it more robust against direct attacks. Centralized DP (CDP), on the other hand, is generally used in centralized settings where privacy guarantees are applied at an aggregate level.

A randomized algorithm \mathcal{M} that takes a user's private data as input is said to provide ϵ -local differential privacy if, for all pairs of users' possible private data (x and x') and for all outputs S :

$$\frac{P[\mathcal{M}(x) \in S]}{P[\mathcal{M}(x') \in S]} \leq e^\epsilon$$

The difference between the definition of local differential privacy and the definition of standard (global) differential privacy is that the local differentially private algorithm takes a single user's data, whereas the standard definition of differential privacy considers pairs of neighboring datasets. A randomized response mechanism (e.g., flipping a bit with probability dependent on ϵ) satisfies ϵ -DP in the local model, where users privatize their data before sharing.

The randomized response survey technique proposed by Stanley L. Warner in 1965 is frequently cited as an example of local differential privacy [222]. Warner's innovation was the introduction of what could now be called the "untrusted curator" model, where the entity collecting the data may not be trustworthy. Before users' responses are sent to the curator, the answers are randomized in a controlled manner, guaranteeing differential privacy while still allowing valid population-wide statistical inferences. In 2003, [75] gave a definition equivalent to local differential privacy. In 2008, [125] first used the term 'local private learning' and showed it to be equivalent to randomized response.

In LDP, noise is added before data collection (at the user level) instead of after collection. In traditional DP approaches, trust is required in the data processor, whereas no trust is needed in the data processor when using LDP, since individual data is protected at its source. This makes LDP suitable for decentralized data collection (e.g. Google's Chrome privacy statistics or Apple's data anonymization), at the cost of lower accuracy, since noise is added early (in contrast to traditional approaches, where raw data is processed).

2.4 Formal properties of differential privacy mechanisms

The relaxation in (ϵ, δ) -differential privacy provides additional flexibility in designing differentially private mechanisms by enabling a trade-off between privacy and utility. In particular, mechanisms that satisfy (ϵ, δ) -differential privacy, including zCDP, RDP, f-DP, and GDP, still inherit key properties of pure differential privacy, including composability, robustness to post-processing, and graceful degradation in the presence of correlated data, which ensure that privacy guarantees are preserved under transformations and repeated interactions with private data [68].

2.4.1 Composability. Composability refers to the fact that the joint distribution of the outputs of (possibly adaptively chosen) differentially private mechanisms satisfies differential privacy.

DP mechanism	Composition bound	Growth rate
ϵ -DP	$k \cdot \epsilon$	Linear
(ϵ, δ) -DP	$\sqrt{2k \log(1/\delta')} \cdot \epsilon + k \cdot \epsilon(e^\epsilon - 1)$	Sublinear ϵ , linear δ
(μ, τ^2) -mCDP	$\mu_{total} = k \cdot \mu \quad \tau_{total}^2 = k \cdot \tau^2$	Linear
ρ -zCDP	$k \cdot \rho + 2\sqrt{k \cdot \rho \cdot \ln(1/\delta)}$	Linear
(α, ϵ) -RDP	$k \cdot \epsilon$	Linear
μ -GDP	$\mu_{total} = \sqrt{k} \cdot \mu$	Sublinear
ϵ -LDP	$k \cdot \epsilon$	Linear

Table 2. Sequential composition privacy bounds for different DP frameworks: How privacy degrades when multiple DP mechanisms are applied sequentially on the same dataset.

In sequential composition, if we query an ϵ -differential privacy mechanism k times, and the randomization of the mechanism is independent for each query, then the result would be $k\epsilon$ -differentially private. In general, with k different ϵ_i -differential privacy mechanisms, any function of them is $(\sum_{i=1}^k \epsilon_i)$ -differentially private [162]. When the mechanisms are computed on disjoint subsets of the private database, in parallel composition, then the function would be $\max_i \epsilon_i$ -differentially private instead [162].

A common reason for adopting (ϵ, δ) -differential privacy arises in the context of k -fold adaptive composition [70], a differentially private analogue of the “left or right” notion of security for encryption schemes [14]. In k -fold composition, a process is repeated multiple times (k times), each time using a different slice of the data. The process is adaptive in the sense that, each time we ask a question, we might change what we ask based on the answers we got before. An (ϵ, δ) -differentially private mechanism ensures that the composite mechanism satisfies (ϵ', δ') -differential privacy, where $\epsilon' = \sqrt{2k \ln(1/\delta')} + k\epsilon(e^\epsilon - 1)$ and $\delta' = k\delta + \delta_0$ for some $\delta_0 > 0$ [70]. This theoretical result highlights the feasibility of producing accurate query responses across extensive query sets while preserving individual privacy through adaptive boosting techniques.

If a mechanism satisfies (ϵ, δ) -DP, then, after k sequential queries, the total privacy parameters are bounded as follows: $\epsilon_{total} = \sqrt{k}\epsilon$ and $\delta_{total} = k\delta$. ϵ grows sublinearly (\sqrt{k} scaling), meaning privacy loss accumulates more slowly than in pure DP, while δ grows linearly, meaning the probability of privacy failure increases proportionally with the number of queries.

Concentrated DP sub-Gaussian bound enables stronger composition properties, meaning that when multiple privacy-preserving mechanisms are applied successively, the accumulated privacy loss grows in a well-controlled manner. Since a sub-Gaussian distribution has tight concentration around its mean, the tail probabilities (i.e., the chance of large privacy loss) drop off exponentially. Even if you apply many mechanisms, the expected privacy loss grows slowly and predictably. Since privacy risks don’t escalate uncontrollably, CDP provides a more efficient and scalable approach compared to traditional DP, making it particularly useful in repeated queries.

The mean-concentrated DP (mCDP) formulation departs from the stringent ϵ -DP guarantee of pure differential privacy by emphasizing the concentration of privacy loss rather than its worst-case extremum. Under k -fold composition, the privacy loss accumulates as $O(\sqrt{k} \cdot \mu + k \cdot \sigma)$, a sublinear improvement over the linear $k \cdot \epsilon$ growth of the classical model. This enables more efficient noise calibration, enhancing data utility in iterative or adaptive analyses while preserving robust privacy guarantees.

The zero-concentrated DP (zCDP) has a more elegant privacy composition. If a mechanism satisfies ρ -zCDP with privacy parameter ρ , then after k sequential queries, the total privacy parameter scales additively: $\rho_{total} = k \cdot \rho$. When converted into ϵ -DP, $\epsilon_{total} = k \cdot \rho + 2\sqrt{k \cdot \rho \cdot \ln(1/\delta)}$. While the total ϵ does grow with k , the dominant privacy risk (the part that matters most in practice) grows much more slowly than in pure DP, where ϵ grows strictly linearly, whereas the second term in zCDP scales with \sqrt{k} . If you fix ϵ , ϵ grows roughly with \sqrt{k} and k grows like ϵ^2 . Given its quadratic accumulation of privacy loss, the number of queries you can make scales quadratically with the privacy budget. In other words, zCDP allows many more queries before hitting the same privacy budget as in traditional DP. In traditional DP, you get linear scaling (double the queries, double the privacy loss). In zCDP, you get quadratic scaling (double the privacy budget, four times the number of queries).

In (α, ϵ) -RDP, after k sequential queries, the total privacy parameter also scales linearly $\epsilon_{total} = k\epsilon$. One of the key strengths of RDP is its tight and clean composition rule: if you compose k (α, ϵ) -RDP mechanisms, the composition satisfies $(\alpha, k \cdot \epsilon)$ -RDP. It is still linear in ϵ but, crucially, it is independent of δ , unlike approximate (ϵ, δ) -DP.

In μ -GDP, privacy loss accumulates sublinearly, $\mu_{total} = \sqrt{k}\mu$. The probability of large privacy loss decays like a Gaussian tail, providing a tight and symmetric privacy guarantee. GDP provides stronger composition guarantees (privacy loss accumulates more smoothly over multiple queries) since traditional DP relies on worst-case bounds, which can lead to more extreme privacy loss fluctuations (its composition bounds can be looser, making privacy harder to manage over multiple queries).

In summary, pure DP, local DP, zCDP, RDP, and mCDP all show linear growth in privacy loss. Approximate DP grows faster due to the additional exponential term in its advanced composition. f-DP composition is not discussed here because its composition depends on the specific f-divergence used and often requires numerical methods. In practice Gaussian DP and zCDP grow sub-linearly in terms of ϵ -equivalent privacy loss (\sqrt{k} scaling), making them much more efficient for many compositions. This property makes them valuable in iterative processes, including training deep learning models.

Each framework provides different trade-offs between privacy strength and composition efficiency. If you're working with iterative algorithms, zCDP and GDP tend to offer better composition guarantees than traditional DP.⁴

2.4.2 Robustness to post-processing. The robustness to post-processing is a fundamental property of differential privacy (DP) frameworks. It ensures that once data has been privatized, any further computation or transformation cannot weaken the privacy guarantee. In fact, robustness to post-processing is a key property for the modular use of differential privacy. While post-processing could affect fairness or utility [240], it should never compromise the privacy guarantees.

This robustness to post-processing property is defined as follows: for any deterministic or randomized function F defined over the image of the mechanism \mathcal{M} , if \mathcal{M} satisfies ϵ -differential privacy, so does $F(\mathcal{M})$ [64].

Pure DP is robust to post-processing because the definition of ϵ -DP is preserved under any data-independent transformation. Approximate DP is robust to post-processing because the (ϵ, δ) bounds remain valid even after arbitrary post-processing (including randomized post-processing, as long as it doesn't access the original data). zCDP and RDP are defined via a Rényi divergence, which is non-increasing under post-processing, so that any function applied to the output of a zCDP mechanism preserves its ρ bound in zCDP and you can safely apply further analysis or transformations without increasing privacy loss in RDP. Like zCDP, mCDP relies on properties of the privacy loss random variable (its moments), which are preserved under post-processing. GDP is based on the privacy loss being sub-Gaussian, and

⁴In more realistic scenarios, the concurrent composition properties of interactive mechanisms, whereby an adversary can arbitrarily interleave its queries to the different DP mechanisms, have also been studied [207] [152] [208]. In fact, computing the optimal bound for composing k arbitrary DP algorithms is, in general, #P-complete [171].

this property is preserved under post-processing. In general, f -divergences are contractive under post-processing (i.e., they don't increase), so the f -DP guarantee remains intact no matter how the output is used. Finally, in LDP, each user perturbs their data before sharing them, so any transformation after that does not affect the privacy guarantee.

Composition permits modular construction and analysis of differentially private mechanisms and motivates the concept of privacy loss budget. When all elements of a complex mechanism that access sensitive data are separately differentially private, so will be their combination, followed by arbitrary post-processing. The DP guarantee is composable and repeating invocations of differentially private algorithms lead to a graceful degradation of privacy.

2.4.3 Graceful degradation in the presence of correlated data. Differential privacy guarantees can be weak in some real-world scenarios. A no-free-lunch theorem [129] shows that it is not possible to provide privacy and utility without making assumptions about how the data are generated. Privacy of an individual is preserved when it is possible to limit the inference of an attacker about the participation of the individual in the data generating process.

Group privacy extends privacy guarantees to protect small groups, not just individuals. Obviously, an ϵ -DP mechanism satisfies $k\epsilon$ -DP for groups of size k , i.e. datasets differing in k records [64] [68, Section 2.3].

The concept of graceful degradation in differential privacy refers to how well a privacy framework maintains its guarantees when the assumption of independent data entries is violated, i.e., when data is correlated. eps

Different DP frameworks handle such situations with disparate degrees of success:

- Pure DP, ϵ -DP, the privacy guarantee holds regardless of correlations, because it's defined over neighboring datasets (differing in one individual). However, when individuals' data are highly correlated (e.g., family members), removing one person may still reveal information about others. So the guarantee is still valid, but less meaningful in practice.
- Approximate DP, (ϵ, δ) -DP, is similar to pure DP. The core guarantee still applies to neighboring datasets and the δ term allows for a small probability of failure. However, correlations can make the δ failure events more likely to leak meaningful information.
- Mean-concentrated DP, mCDP, tracks the mean and variance of privacy loss, which can be skewed by correlations. Therefore, the guarantees degrade more gracefully than in pure DP, but still suffer under strong dependencies.
- Zero-concentrated DP, zCDP, assumes a concentration of privacy loss, which may not hold under strong correlations. The theoretical guarantees remain, but the interpretation of ρ becomes less tight.
- Rényi DP, RDP, like zCDP, relies on divergence measures that assume independence for tightness. The bound still holds, but actual leakage may be higher than expected.
- Gaussian DP, GDP, assumes the privacy loss is sub-Gaussian, which may not hold under correlation. In other words, the Gaussian tail bounds may underestimate the true risk. In general, the degradation of f -DP mechanisms depend on the specific divergence used, since some f -divergences are more sensitive to correlations than others.
- Local DP, LDP, is inherently robust to correlations, making it ideal for decentralized settings. Since each user perturbs their data before sharing, correlations in the raw data don't affect the privacy guarantees.

In practice, the use of differential privacy can lead to privacy breaches, and differential privacy does not always adequately limit inference about participation in the data generating process. Data correlation, i.e., the probabilistic dependence relationship among examples in a database can seriously deteriorate the privacy properties of DP, as demonstrated by Bayesian attacks on differentially private mechanisms [145].

DP frameworks might underestimate the privacy risk and, therefore, the amount of noise that must be added to achieve a desired bound on the adversary's ability to make sensitive inferences. When correlation (or dependency)

among data examples would break the desired privacy guarantees, dependent differential privacy (DDP) [145] explicitly incorporates probabilistic dependency constraints among tuples in the privacy formulation:

$$P[\mathcal{M}(D(L, R)) \in S] \leq e^\epsilon P[\mathcal{M}(D'(L, R)) \in S]$$

where D and D' are neighboring databases, when the modification of an example in database $D(L, R)$ causes change in at most $L - 1$ examples in $D'(L, R)$ due to the probabilistic dependence R between examples. L is the dependence size, i.e. when any example in D is dependent on at most $L - 1$ other examples, whereas the relationship R might be due to the data generating process [129].

In order to handle correlation using differential privacy, [44] just multiplied the sensitivity of the query output by the number of correlated records. This naive approach to group privacy protects databases differing in n examples, by setting ϵ to ϵ/n : instead of having each example ϵ -differentially private protected, now every group of n examples is ϵ -differentially private protected (and each example is ϵ/n -differentially private protected). However, excessive noise is then added to the output, severely degrading the utility of the model. DDP provides a finer dependent perturbation mechanism (DPM) to achieve the desired privacy guarantees [145].

The Pufferfish framework [130] [199] generalizes DP by incorporating adversarial belief about existing data relationships using a data generation model maintained as a distribution over all possible database instances. This provides rigorous privacy guarantees against adversaries who may have access to any auxiliary background information and side information of the database. Both DDP [145] and Blowfish [107], which only considers correlations due to deterministic constraints, are subclasses of the Pufferfish framework.

In common DP frameworks, privacy guarantees typically depend on the subject’s ability to control their data. However, in some situations, privacy must persist even when the subject no longer has control over whether their data is revealed. For instance, we might be interested in keeping something a secret, even when the subject of the secret no longer has the ability to keep it a secret (i.e. withholding their data doesn’t maintain the secret). This requires stronger guarantees than those offered by DP frameworks, which do not protect from public leaks or external correlation attacks. Computational differential privacy [168] frameworks go beyond standard DP by considering adversaries with computational constraints, meaning privacy can be maintained even if adversaries gain partial knowledge. Information-theoretic privacy [4][20] frameworks provide privacy guarantees that hold independent of any adversary’s computational power: even if all the subject’s data is exposed, critical secrets remain secure under predefined conditions.⁵ The real challenge arises when a secret becomes exposed due to external factors, meaning a stronger notion of “future-proof” privacy is required. Some privacy-preserving frameworks like “indistinguishability obfuscation” [7][114] or “traceable anonymization” [85][175][156] attempt to achieve persistent secrecy even under unexpected conditions.

3 Differential privacy in Machine Learning models

Differential privacy (DP) lets us release statistical information about datasets while protecting the privacy of individual data examples. Therefore, its use in Machine Learning enables the use of aggregate patterns in the data while limiting the amount of information that is leaked about specific individuals [60]. By injecting carefully calibrated noise into statistical computations, the utility of the statistic is preserved while provably limiting what can be inferred about any individual in the dataset.

⁵Zero-Knowledge Proofs (ZKPs) [97][98] are cryptographic techniques that allow the verification of a statement without revealing the underlying data. While not strictly a DP framework, ZKPs could contribute to situations where someone needs to prove a claim without exposing private information. Privacy-preserving cryptographic methods, such as secure multiparty computation (MPC) [231][96] and homomorphic encryption [188][90], help in ensuring privacy even when data is processed externally, reducing reliance on a subject’s ability to control their own data.

The incorporation of differential privacy into Machine Learning models relies on two key properties:

- The composition properties of DP mechanisms enable the concept of a privacy budget: a total allowable privacy loss that can be allocated and spent across various computations performed on a dataset. Careful management and accounting of this budget using appropriate composition rules are essential for practical DP applications.
- A cornerstone property of differential privacy is its robustness to post-processing. Once data has been released via a DP mechanism, analysts can perform any kind of subsequent analysis on the released data without weakening the original privacy guarantee. This makes DP highly modular and practical, as the privacy analysis only needs to focus on the initial mechanism interacting with the sensitive data.

Several approaches can be used to train a privacy-preserving ML model [118]. Several general strategies have been developed, with specific adaptations required for different classes of ML algorithms:

- *Output perturbation*: First, learn a model on clean data, i.e. train a ML model M using standard, non-private methods on the sensitive dataset to obtain its parameters θ . Then, use a DP mechanism (e.g. exponential or Laplacian) to generate a noisy model ($\theta_{DP} = \theta + \text{noise}$), i.e. noise is added directly to the trained model parameters before releasing $M_{\theta_{DP}}$ ($\hat{y} = M_{\theta_{DP}}(x)$). Alternatively, noise can be added to the predictions made by the non-private model ($\hat{y} = M_{\theta}(x) + \text{noise}$). For approaches that have many iterations or multiple steps, the DP mechanism can be applied to the output parameters of each iteration/step. While conceptually simple, calculating the sensitivity of the entire training process for complex models like deep neural networks is often intractable, leading to overly large noise and poor utility. This method is more feasible for simpler models with easily bounded sensitivity.
- *Objective perturbation*: Instead of perturbing the output, this method perturbs the objective function (e.g., loss function plus regularization term) that the ML algorithm seeks to minimize. The algorithm then finds the parameters θ_{DP} that minimize this noisy objective: $\theta_{DP} = \arg \min_{\theta} (L(\theta) + \text{noise})$. This approach is commonly used for convex optimization problems found in algorithms like SVM and logistic regression, as the sensitivity of the objective function can often be bounded more easily than the sensitivity of the final parameters.
- *Training mechanism perturbation*: Within iterative optimization algorithms like Stochastic Gradient Descent (SGD), this strategy is known as gradient perturbation. In each iteration, the gradient of the loss function is computed, perturbed with noise, and then used to update the model parameters. This is the dominant approach for training deep neural networks with DP, primarily through the DP-SGD algorithm [1]. It allows privacy to be incorporated incrementally throughout training.
- *Sample and aggregate*: First, split the dataset into many small subsets. Next, combine the results from all subsets to estimate a model, adding noise in this aggregation step. Typically, models are trained on disjoint data subsets and the training results (either model predictions or model parameters) are privately aggregated. In PATE (Private Aggregation of Teacher Ensembles) [178], an ensemble of teacher models are trained on disjoint private data partitions to provide noisy, aggregated labels for a public dataset, which is then used to train a final student model.

In ML, input perturbation is uncommon, even though it can be used successfully in signal processing; noise addition at the input may incur too much perturbation for learning to be possible [193].

The choice of strategy depends significantly on the particular ML algorithm to be used. Below we describe how DP has been incorporated into different ML techniques.

3.1 Differential privacy in Symbolic AI

Decision tree learning is a prime example of symbolic AI in data mining. The most common strategy for learning decision trees from data is a greedy algorithm that, starting from the root of the decision tree, recursively partitions the training examples, a process known as top-down induction of decision trees (TDIDT) and used by ML algorithms such as ID3, C4.5, and CART. Alternatively, decision trees can also be built in a completely random way (RDT, random decision trees [76]).

When taking differential privacy into account, tree-based models require specialized handling of splits and leaves:

- PPID3 (Privacy-Preserving ID3 [209]) shows how to construct decision trees on vertically-partitioned data with an arbitrary number of parties where only one party has the class attribute, albeit not in a DP context and without an experimental study on its effectiveness in terms of privacy, utility, or their trade-off.
- PPRDT (Privacy-Preserving Random Decision Trees [211]) leverages the fact that randomness in tree structure can provide strong privacy (better than simple perturbation of input/output) and requires significantly less time than alternative cryptographic approaches.
- SuLQ-based ID3 [84] is a privacy-preserving ID3 that uses noisy counts (i.e. adds Laplace noise to the observed frequencies) and is based on the SuLQ framework (Sub-Linear Queries [21]), a predecessor of differential privacy. However, as the count estimates required to evaluate the splitting criterion must be carried out for each attribute separately, the privacy budget must be split among all those separate queries. Consequently, the budget per query is small, resulting in a wasteful use of the privacy budget.
- DiffPID3 [84] provides a better approach: rather than evaluating each attribute separately, we can evaluate the attributes simultaneously in one query, the outcome of which is the attribute to use for splitting. A quality function q provided to the exponential mechanism scores each attribute according to the splitting criterion.
- In DiffPC4.5 [84], each time a partition reaches a pre-determined depth, or the number of samples in that partition is about the same scale as random noise, or the sample space corresponding to that partition is too small, the mechanism stops operating on that partition. It then assigns to that partition a noisy count of samples with each label. After the partitioning process has completed altogether, these noisy counts are used to decide whether to remove those nodes without having to consider privacy.
- LPDT [153] builds locally-private decision trees for non-parametric regression. On both synthetic and real-world data, LPDT exhibits superior performance when compared with other state-of-the-art LDP regression methods using private histograms [15] [102] and deconvolution kernels [78].

Combining multiple models, as in bagging, boosting, stacking, or meta-learning, is commonly used to improve the accuracy of single models. Applying DP requires managing privacy across the ensemble:

- *Ensembles of random decision trees* [76]: [112] and [113] use the Laplace mechanism to add noise to leaf frequencies and satisfy ϵ -DP (smaller values of ϵ imply that more noise is added). [81] proposes a query that outputs the most frequent label in some subset of the data with high probability (using the exponential mechanism) and uses this query in each leaf node of all the random trees in a forest.
- *Random forests* [26] are a popular method for building decision tree ensembles, based on the generation of bootstrap samples of training data and then, for each bootstrap sample, selecting a random subset of features to be used for each decision tree in the ensemble. Unfortunately, although ensemble approaches usually improve performance, they also weaken data privacy. A common strategy is to train each tree using a DP decision tree algorithm and divide the total privacy budget ϵ among the T trees using sequential composition (i.e. each tree

gets ϵ/T). If trees are trained on disjoint subsets of the data, parallel composition can be used, allowing each tree to potentially use the full budget ϵ , leading to better utility. The variance-preserving framework in [185] is designed so that, when an adversary attempts to estimate a specific attribute of any individual data point, the variance of the estimation error crucially depends on the number of trees in ensembles and, therefore, bounds can be derived on the maximum number of trees that are allowed in the ensemble while maintaining privacy. When the estimated variance may not be lower than the prior variance of the attributes, random forests may offer “privacy for free”.

- *Gradient-boosted decision trees*: Applying DP to GBDT is challenging due to the sequential nature of training, since each tree corrects the errors of the previous ones. This creates dependencies and makes sensitivity analysis and budget allocation complex. The privacy budget accumulates additively across trees in the sequence and, therefore, naive application of noise can lead to significant accuracy loss, since the gradients (or residuals) used to train subsequent trees depend on previous trees and the data, making sensitivity hard to bound tightly. DPBoost [140] employs a two-level boosting structure called “ensemble of ensembles” (EoE) for privacy budget allocation, combining parallel composition within each ensemble (i.e. training with disjoint training subsets) and sequential composition across ensembles. DPBoost adaptively controls the gradients of training data for each iteration and clips leaf nodes to tighten the sensitivity bounds. OpBoost [142][143] uses LDP for vertical federated learning, where users with non-overlapping attributes of the same data samples jointly train a model without directly sharing the raw data. FederBoost [202] observes that GBDT training primarily depends on the relative order of feature values for finding splits, not their exact values. Other alternatives offer different trade-offs among privacy, utility, and efficiency, such as SecureBoost [45], which relies on cryptography, or SimFL [139], based on locality-sensitive hashing.

Applying DP to decision trees presents unique challenges due to their recursive structure and reliance on data queries at multiple stages [80] [82]:

- *Node splitting*: Selecting the best attribute and threshold to split a node typically involves calculating impurity measures, like the Gini index or information gain, which are based on data frequencies within each node. Such computations can be privatized using the Laplace mechanism (applied to the counts used by the splitting criterion) or the exponential mechanism, which is often preferred (i.e. using the impurity measure (as the utility function) to probabilistically select a split that is close to optimal while preserving privacy). Handling continuous attributes requires care, as selecting an exact threshold from the data leaks privacy. Potential solutions include selecting a noisy threshold or using the exponential mechanism to select a range.
- *Leaf value determination*: Leaf nodes typically predict the majority class among the instances reaching the leaf. Again, both the Laplace and the exponential mechanism can be used: add Laplace noise to the counts of each class within the leaf (and output the class with the highest noisy count) or directly output a (potentially non-majority) class label, sacrificing count information but potentially improving accuracy.
- *Termination and pruning*: Standard criteria need adaptation. Termination can be based on reaching a maximum depth (since deeper trees consume more privacy budget), a minimum noisy count threshold, or when the noise magnitude overwhelms the signal. Pruning techniques can be adapted by using noisy counts or specifically removing nodes that were likely generated due to noise.

The privacy budget allocation is paramount in DP decision trees, since the total privacy budget must be distributed across all decisions made during tree construction, including deciding splits at internal nodes and determining the

final leaf values. Additionally, when used in decision tree ensembles, the budget must be properly allocated among the individual decision trees within the ensemble.

3.2 Differential privacy in Probabilistic AI

Estimators play a central role in probabilistic AI. For instance, they are used to infer the parameters of probability distributions or models that best explain observed data. Essentially, estimators are functions that provide estimates of unknown parameters based on observed data, i.e. approximations of quantities of interest based upon the evidence in a given dataset.

In probabilistic AI, estimators are used to learn model parameters (e.g., mean and variance in a Gaussian distribution), make predictions by estimating the most likely outcomes, quantify the uncertainty in predictions or model parameters, or support inference in Bayesian models or probabilistic graphical models.

- A point estimator provides a single “best guess” value for an unknown parameter. It does not quantify uncertainty but gives a specific value based on the data (e.g. the sample mean $\hat{\mu} = \frac{1}{n} \sum_{i=1}^n x_i$ is a point estimator of the population mean). The propose-test-release (PTR) [63] mechanism can be used to obtain (ϵ, δ) -DP robust statistical estimators (robust in the sense that they are resilient against outliers and small errors in data measurement). Convergence rates of differentially private approximations to statistical estimators [40] have also been studied within the smooth sensitivity framework (i.e. using local sensitivity [174]).
- M-estimators (short for maximum likelihood-type estimators) generalize point estimators by defining the estimate as the solution to an optimization problem: $\hat{\theta} = \arg \min_{\theta} \sum_{i=1}^n \rho(x_i, \theta)$, where $\rho(x_i, \theta)$ is a loss function that measures the discrepancy between the data and the model. Maximum likelihood estimators (MLEs) are a special case of M-estimators, where $\rho(x_i, \theta) = -\log p(x_i|\theta)$. M-estimators are useful for robust statistics, where you want to reduce the influence of outliers (e.g., using Huber loss instead of squared error). For a large class of parametric probability models, one can construct a ϵ -differentially private estimator whose distribution converges to that of the maximum likelihood estimator using the sample and aggregate strategy [197]: split D into k disjoint random subsets (a.k.a. blocks), estimate parameters on each subset $\hat{\theta}_i$, and use a differentially-private mean of $\hat{\theta}_i$ to approximate θ (i.e. compute the average estimate \bar{z} , draw a random observation R from a double-exponential distribution $Y \sim \text{Lap}(\frac{\Lambda}{2\epsilon})$, where Λ is the diameter of the parameter space Θ , and output $\theta^* = \bar{z} + R$). The estimator can also be made differentially private by replacing the empirical average with a differentially private aggregate using a widened Windsorized mean [198]. A third mechanism for obtaining ϵ -differentially private M-estimators is based on the use of perturbed histograms [137], i.e. adding double-exponential noise to each histogram count in order to obtain a noisy density function.

Estimators are used by many ML algorithms to learn from data, from both Naive Bayes to k-means and GMMs, yet they serve different purposes due to the nature of the algorithms, probabilistic in Naive Bayes and geometric in k-means and GMMs:

- *Naive Bayes* is a probabilistic classifier based on Bayes’ theorem with the assumption of feature independence. Estimators are used to compute the probabilities needed for classification (i.e. class prior probabilities and conditional feature likelihoods). These estimates are plugged into Bayes’ Theorem to compute posterior probabilities. An ϵ -differentially private Naive Bayes model (DP-NB) [210] is quite simple: just estimate the sensitivity for each attribute appropriately and add Laplacian noise of the model parameters.

A Naive Bayes classifier can also be learnt jointly by different data owners in a distributed environment, without the help of a trusted curator (i.e. without breaking the privacy of each data provider), with the help of additive homomorphic encryption (AHE-NB) [141], just by adding noise in the aggregated ciphertexts, which is functionally equivalent to performing the mechanism that implements differential privacy on the corresponding plaintexts. Also without a trusted data curator, individuals can send their perturbed class prior probabilities so that the data aggregator can estimate all the probabilities needed by the Naive Bayes classifier satisfying local differential privacy (LDP-NB) [234].

- *k-means* is an unsupervised clustering algorithm that partitions data into k clusters by minimizing within-cluster variance. A point estimator of the mean of the data points assigned to each cluster is used to describe the centroid of each cluster. A differentially-private k -means algorithm [174] can be implemented using the sample and aggregate strategy: randomly split the training set into many subsets, run the non-private k -means algorithm on each subset, choose the right aggregation function, and then use the smooth sensitivity framework (i.e. local sensitivity) to publish the final noisy output. This final step preserves privacy while the underlying k -means algorithm is essentially unchanged. Alternative privacy-preserving k -Means variations have also been proposed, beyond the DP framework, by not allowing to access the true data vectors at any stage of the clustering algorithm [19].

Like other ML techniques, the k -means algorithm has also been adapted to guarantee local differential privacy in a distributed environment [225]. In each iteration, the service provider (i.e. untrusted curator) aggregates the received perturbed feature vectors of users based on the clusters that they belong to in previous iteration, computes the k centroids, and then it publishes the centroids of the current iteration to users. Also in each iteration, after being informed of the computed centroids, each user computes the distances to the centroids based on its exact data and finds the nearest corresponding cluster; then, each user sends the cluster assignment to the service provider using binary strings with noise that preserve user's private information in terms of local differential privacy.

- *Gaussian Mixture Models (GMMs)* can be seen as a probabilistic generalization of k -means clustering. In k -means, each cluster is represented by a centroid (a point in space) and all points are assigned to the nearest centroid using Euclidean distance. In GMMs, each cluster is represented by a Gaussian distribution (defined by a mean vector and a covariance matrix). This allows clusters to have elliptical shapes, different sizes, and orientations. Gaussian mixture models have inspired several differentially-private algorithms with applications in ML:
 - PGME [123] is a differentially private algorithm for recovering the parameters of an unknown Gaussian mixture, provided that the components are sufficiently well separated, that uses private PCA [Principal Component Analysis] and private clustering steps.
 - DP-LMGMM [180] learns multiclass Gaussian mixture model-based classifiers that preserve differential privacy using a large margin loss function with a perturbed regularization term.
 - MR [232] adapts the expectation-maximization (EM) algorithm for locally differentially private (LDP) settings. Using a reduction framework inspired by Gaussian mixture models (GMM) and focusing on density estimation under LDP constraints, mixture reduction (MR) provides an approach that is suitable for decentralized or federated ML.
- PrivBayes [238] first spends half the privacy budget to learn a Bayesian network structure that captures the dependencies in the data, and then uses the remaining privacy budget to measure the statistics necessary to learn the Bayesian network parameters. It has been used as a differentially private mechanism that generates

synthetic data. Private-PGM [158] is a general-purpose post-processing tool to infer a data distribution given noisy measurements that uses a compact representation of the data distribution to avoid exponential complexity. Private-PGM can be used to improve the original PrivBayes [238] and has also been used to win the 2018 NIST differential privacy synthetic data competition [157].

In some situations, Bayesian learning can be inherently differentially private [221]. For instance, getting one sample from the posterior is a special case of the exponential mechanism and this sample as an estimator is near optimal for parametric learning (e.g. in logistic regression). Algorithmic procedures of stochastic gradient Langevin Dynamics (and variants) that attempt to sample from the posterior also guarantee differential privacy as a byproduct. There might be other cases where existing randomness could be exploited for privacy [221].

3.3 Differential privacy in Statistical AI

Statistical AI is a subfield of artificial intelligence that emphasizes the use of statistical methods to model, infer, and predict patterns in data. It forms a bridge between traditional statistics and modern machine learning. Key techniques within Statistical AI include traditional regression models and Support Vector Machines (SVMs).

3.3.1 Traditional regression models. Traditional regression models include linear regression for predicting a continuous outcome and logistic regression for predicting a binary outcome. In both cases, we can use objective perturbation (i.e. add noise to the target function) in order to make them differentially-private. For instance, adding a linear regularization term to the loss function can ensure privacy [42] [239].

- Linear regression models are of the form $\hat{y} = \beta_0 + \beta_1 x_1 + \dots + \beta_n x_n$. They are used to predict continuous variables in forecasting, trend analysis, or risk modeling. They are linear model, adjusted to minimize the Mean Squared Error (MSE), so they assume linearity, as well as independence, homoscedasticity, and the normality of errors. Objective perturbation [42] [239] and the sample and aggregate framework [63] can be used to design DP linear regression models. In linear regression, L2 regularization limits the magnitude of weights, reducing the L2-sensitivity of the output. Gaussian noise can then be added to achieve (ϵ, δ) -DP [42].
- Logistic regression models are of the form $P(y = 1|x) = \hat{y} = 1 / (1 + e^{-(\beta_0 + \beta_1 x_1 + \dots + \beta_n x_n)})$. They are used to predict a binary outcome in medical diagnosis, fraud detection, customer churn prediction, or spam detection. Unlike linear regression models, which predict a value directly, logistic regression models predict the probability of one class in binary classification problems. They use the sigmoid function to squash the output between 0 and 1 and minimize the log loss function (a.k.a. cross-entropy).

Logistic regression models can be made differentially-private by objective perturbation methods, including both the sensitivity and objective regularized (ϵ, δ) -DP logistic regression [41] [42], as well as the functional mechanism (FM) ϵ -DP regression [239]. Regularization (e.g., L2 regularization) can be used to bound the sensitivity of the model output (i.e. weights or predictions) and then noise can be added, proportional to this reduced sensitivity, to satisfy DP. Empirical results on logistic regression demonstrate improved utility [42].

3.3.2 Support Vector Machines (SVMs). SVMs aim to find a hyperplane that best separates data points belonging to different classes. Linear SVMs work when training examples are linearly separable, whereas kernel SVMs are used when data are not linearly separable:

- Linear SVMs find the hyperplane that maximizes the margin between the two classes in binary classification problems by solving a convex optimization problem with constraints. Both output perturbation mechanisms [42][191] and objective perturbation mechanisms [42] have been used for linear SVMs.
- Kernel SVMs apply a kernel function to map data into a higher-dimensional space where a linear separator can be found. This allows SVMs to create non-linear decision boundaries in the original space. Several ideas have been explored for private kernel SVMs:
 - For all translation-invariant kernels⁶, such as the RBF, Laplacian, and Cauchy kernels, you can approximate kernel functions in the original sample space with a linear kernel in another space, in order to avoid publishing training data [191].
 - For kernel functions that are not translation-invariant, such as the polynomial kernel or the sigmoid kernel, you can interpret a model as a function and use another function to approximate it, the basis of the Test Data-Independent Learner (TDIL) [115].

DP is typically applied to SVMs using objective perturbation [42][115], leveraging the convexity of the standard SVM objective function (hinge loss plus L2 regularization). Noise, often Gaussian or Laplace, depending on the specific formulation and desired privacy guarantees, is added directly to the SVM objective function. The optimization algorithm then finds the parameters (hyperplane weights and bias) that minimize this perturbed objective. The amount of noise depends on the sensitivity of the objective function, which can often be bounded. Perturbing the regularized objective function directly with random noise is simple to implement and leverages regularization smoothing. L2 regularization (strong convexity) reduces sensitivity in empirical risk minimization, enabling DP with less noise and improving utility [42].

While less common, output perturbation (adding noise to the final hyperplane parameters) [41][42][191] or gradient perturbation (when using SGD to solve the SVM optimization) [214] are also potential approaches. In both cases, care must be taken not to exacerbate bias and unfairness for different groups of individuals [205].

For high-dimensional data, applying dimensionality reduction techniques before SVM training can improve utility and reduce computational overhead. Differentially-private dimensionality reduction techniques allow the private release of a low dimensional approximation to a set of data records, e.g. by adding Gaussian noise to the covariance matrix in order to achieve a DP-compliant Principal Component Analysis algorithm [71]. When combining dimensionality reduction with SVMs, end-to-end privacy can be ensured either by adding noise in the PCA phase (i.e. to the covariance matrix), as in DPPCA-SVM [111], or by adding noise in the SVM phase (i.e. during SVM training), as in PCA-DPSVM [111]. Privacy can be adopted in both the PCA-based dimensionality reduction phase and the SVM training phase, as shown by Differential Privacy-compliant Federated Machine Learning with Dimensionality Reduction, FedDPDR-DPML [218], which enables multiple participants to collaboratively learn a global model based on weighted model averaging and knowledge aggregation, which is later distributed to each participant to improve local utility.

3.4 Differential privacy in Deep Learning

Deep learning models, with their vast number of parameters and non-convex optimization landscapes, pose significant challenges for differential privacy. The use of differential privacy in deep learning models typically relies on gradient perturbation, using a differentially-private version of the stochastic gradient descent algorithm [1].

⁶A kernel function of the form $k(x, y) = g(x - y)$, for some function g , is called translation-invariant.

3.4.1 Differentially-Private Stochastic Gradient Descent (DP-SGD). Differentially Private Stochastic Gradient Descent (DP-SGD) [1] adapts SGD to satisfy (ϵ, δ) -differential privacy by clipping per-example gradients to a fixed norm and adding Gaussian noise. This method builds on stochastic optimization methods [189] [128] [190] to ensure differential privacy in deep learning.

Gradient perturbation via DP-SGD has become the standard technique for DP deep learning. The standard SGD is modified so that DP is incorporated in each training iteration using clipped noisy gradients. First, as in the standard SGD, we sample a random mini-batch of data and compute per-example loss gradients for all training examples in the mini-batch. Then, the first modification to the standard algorithm is introduced: gradient clipping (i.e. clip the L2 norm of each per-example gradient). Clipped per-example gradients are averaged and a second modification completes the DP-SGD algorithm. Instead of updating the model parameters using the gradients as in SGD, we add Gaussian noise to the average gradients and update the model parameters using the noisy version of the gradients.

Both gradient clipping and gradient noise were incorporated into DP-SGD to achieve differential privacy:

- *Gradient noise:* Gaussian noise is added to the gradients to obscure individual data contributions. That Gaussian noise is scaled by the gradient clipping threshold. The noise level introduced into the gradients allows the privacy loss to be distributed over multiple training steps, decreasing the per-step privacy cost and enabling more epochs to be run within the same cumulative privacy budget.

The noise level has a significant impact on model performance, as larger noise levels reduce the risk of privacy leakage but can also degrade accuracy. DP-SGD, therefore, exhibits the privacy-utility trade-off, as higher privacy levels tend to decrease model accuracy due to the increased noise that is introduced during model training.

- *Gradient clipping* limits the gradient norm to a predefined threshold, bounding the sensitivity of each individual's contribution to the average gradient. Gradient clipping has two important effects: on the one hand, clipping reduces the variance of gradients and helps maintain privacy by bounding the maximum contribution of any individual training example; on the other hand, it can distort the unbiasedness of the gradient estimate.

The choice of an appropriate clipping bound is crucial, as a small bound can significantly affect the direction of the gradient (i.e. the gradients may deviate significantly from their true values), while a larger bound requires adding more noise to the gradients. A reasonable heuristic is taking the median of the norms of the unclipped gradients throughout the training process. This helps in balancing privacy and model accuracy, as excessive clipping may lead to a loss of useful information, while insufficient clipping could compromise privacy.

In DP-SGD, regularization is implicitly used to stabilize training, reducing sensitivity. In deep learning, regularization is common practice. DP-SGD also exploits subsampling privacy amplification. Whereas regularization reduces sensitivity, subsampling and noise in SGD (both natural and added) are used provide DP guarantees.

The original DP-SGD algorithm was sequential. Since SGD algorithms can be parallelized and executed asynchronously, distributed DP-SGD algorithms have been proposed, such as DSSGD, the Distributed Selective SGD [194]. In selective SGD, the learner chooses a fraction of parameters to be updated at each iteration. This selection can be completely random, but selecting the parameters with larger gradients is smarter (i.e. whose current values are farther away from their local optima). A sparse vector technique (SVT) [105] can be used to prevent indirect privacy leaks caused by participants sharing locally-updated parameters: randomly selecting a small subset of gradients whose values are above a threshold, and sharing perturbed values of the selected gradients. DSSGD assumes two or more participants training independently and concurrently. After each round of local training, participants asynchronously share with

each other the gradients they computed for some of the model parameters. Participants thus benefit from each other's training data, without actually seeing them, and produce more accurate models that they would have been able to learn in isolation, limited to their own training data.

In settings where only the labels are sensitive but the inputs are public, specialized techniques can be used, potentially achieving better utility than full DP-SGD [92].

Some challenges related to the use of DP-SGD include the following [184] [50]:

- *Hyperparameter sensitivity*: DP-SGD performance is highly sensitive to hyperparameters like the clipping threshold (for gradient clipping), the noise multiplier (for gradient noise), the learning rate, and the batch size, even the activation function. Bounded activation functions (e.g., tanh) are sometimes preferred over unbounded ones (e.g., ReLU) to help control gradient norms. Tuning these hyperparameters is critical but complex, and the tuning process itself should be accounted for in the privacy budget if using private data [50]. It should also be noted that architectural choices might affect privacy. For instance, batch normalization can break DP guarantees by creating dependencies across examples in a batch. Alternatives like group normalization or layer normalization, or specific DP-compatible variants, are often required [51].
- *Computational cost*: Computing per-example gradients is significantly more computationally and memory intensive than standard batch gradients. Techniques like efficient vectorization or "ghost clipping" aim to mitigate this [50].
- *Fairness*: DP-SGD can amplify existing biases. Gradient clipping disproportionately affects examples from underrepresented groups or those with more complex features (which tend to have larger gradients), reducing their influence on the model and leading to lower accuracy for these groups [50].
- *Utility degradation*: DP-SGD often leads to a noticeable drop in model accuracy compared to non-private training, especially for smaller privacy budgets (low ϵ). While DP-SGD is a robust method for ensuring privacy, it often leads to significant performance degradation, due to the added noise. Some variants, such as NoisySGD and NoisyAdam [29], can achieve higher accuracy under the same privacy budget.

An effective strategy for improving utility is to pre-train a large model on public data (non-privately) and then fine-tune it on the sensitive target task using DP-SGD. Larger pre-trained models often show better robustness to the noise introduced during private fine-tuning [236]. It should be noted, however, the assumption that large web-scraped datasets are truly "public" and free of privacy concerns is debatable [204].

The application of DP transforms the deep learning process. It requires careful algorithm design, introduces new hyperparameters, impacts computational requirements, and raises considerations about fairness and the very definition of privacy.

3.4.2 Privacy accounting. The privacy accountant tightly tracks cumulative privacy loss over all training iterations to ensure differential privacy in deep learning algorithms. Privacy accountants leverage composition theorems and privacy amplification by subsampling:

- The *Moments Accountant (MA)* was a key innovation introduced with DP-SGD [1]. MA bounds the moments of the privacy loss random variable, hence its name, and generalizes tracking (ϵ, δ) using the strong composition theorem. MA provides tight bounds on the cumulative privacy loss over multiple iterations (compositions) of the DP mechanism in DP-SGD, tighter than basic composition theorems. MA is efficient for Gaussian mechanisms and DP-SGD with subsampling, yet might not generalize well to other mechanisms. Widely adopted in practice

(e.g. TensorFlow Privacy), it requires the numerical computation of moments. Whereas composing directly from the tail bound (i.e. the privacy loss in the sense of differential privacy) can result in quite loose bounds, MA computes the log moments of the privacy loss random variable, which compose linearly. Then, MA converts the moments bound into the (ϵ, δ) -DP guarantee. Since moments compose linearly, the resulting privacy is proportional to the number of iterations.

It should be noted that, although the growth of privacy loss calculated with moments accountant is much less than the growth of privacy loss calculated by the strong-composition theorem, when the total privacy budget ϵ_{total} is very small, the bounds computed by the strong composition theorem are actually less than the bounds computed by the moments accountant.

- The *Analytical Moments Accountant (AMA)* [219] [220] generalizes the moments accounting technique, originally developed for the Gaussian mechanism [169], to any subsampled RDP mechanism (e.g. Gaussian, Laplace, exponential). AMA tracks the Rényi Differential Privacy (RDP) of a mechanism by computing the moments (expected values) of the privacy loss random variable and allows for the efficient conversion of RDP to (ϵ, δ) -DP. Using the relationship between RDP and the log moment generating function, log moments can be computed for any mechanism with known RDP guarantees. Once log moments are computed, numerical optimization can be used to find the tightest ϵ for a given δ .
- The *CLT-based privacy accountant* [29], the privacy accountant in the Central Limit Theorem (CLT) approach, as applied to NoisySGD and NoisyAdam, is a method for tracking and bounding the cumulative privacy loss during training of deep learning models under Gaussian differential privacy (GDP). It is called the CLT approach because the composition theorem of f-DP is tight and, more importantly, the privacy central limit theorem is asymptotic exact. It leverages the Central Limit Theorem to approximate the privacy loss distribution as a Gaussian (normal) distribution, which simplifies the analysis of privacy guarantees under composition. CLT composition is a probabilistic method that models the privacy loss random variable (PLRV) for each query and uses the Central Limit Theorem to approximate the sum of these losses. According to the CLT, the sum of many independent random variables tends toward a normal (Gaussian) distribution. This allows us to approximate the distribution of the total privacy loss. Using the Gaussian approximation, we can compute a tight bound on the total ϵ for a given δ (or vice versa).
- The *Amortized Privacy Accountant (APA)* [132] is another generalization of the Moments Accountant that amortizes the privacy cost over multiple steps or queries, especially under subsampling. APA provides tighter and more adaptive privacy guarantees by analyzing the expected privacy loss over multiple iterations rather than the worst-case loss at each step. By averaging the privacy cost over time, the amortized accountant can yield significantly tighter bounds on the total privacy loss, improving the utility of the trained model.

Amortized analysis is especially relevant when the actual noise added varies across steps, the sensitivity of the function being computed changes over time, or the data access pattern is not uniform or deterministic. It is particularly useful in iterative deep learning algorithms like NoisySGD and NoisyAdam, where privacy loss accumulates over many steps. Using Gaussian Differential Privacy (GDP), which naturally leads to amortized analysis, APA builds on the idea of privacy loss distributions and uses tools like supermartingales and hockey-stick divergences to track privacy loss more precisely. The amortized accountant uses privacy amplification theorems [13] [12] [125] [124] and strong composition theorems [70][10] to provide an amortized bound of privacy loss. APA uses approximations and precomputed bounds to efficiently estimate the total privacy loss. Instead of computing the privacy loss at each step, the amortized accountant uses a formula or lookup table to

estimate the total ϵ after all steps, given the noise multiplier and sampling rate. Amortized accountants may be less precise than exact methods like RDP or CLT composition, but they are much faster and still provide valid privacy guarantees.

- The *Edgeworth Accountant* [215] uses Edgeworth expansions to approximate the distribution of the privacy loss random variable under the f-DP framework, which generalizes traditional (ϵ, δ) -DP. The Edgeworth Accountant provides very tight non-asymptotic bounds on (ϵ, δ) -DP. and works with any noise mechanism (i.e. it can be applied to any noise-addition mechanism, not just Gaussian). It is computationally efficient even with many compositions (i.e. efficient for large-scale applications such as LLMs), albeit it is more mathematically complex than the Moments Accountant.

The Edgeworth accountant is an advanced method for composing differential privacy guarantees, offering even tighter approximations of cumulative privacy loss than CLT-based methods. It builds on the Edgeworth expansion, a refinement of the Central Limit Theorem (CLT), to more accurately approximate the distribution of the privacy loss random variable (PLRV). The Edgeworth expansion improves upon the CLT by incorporating higher-order moments (like skewness and kurtosis) of the distribution. While CLT approximates the sum of random variables as a normal distribution, Edgeworth adds correction terms to better match the true distribution. Since the Edgeworth accountant doesn't rely on closed-form expressions for specific mechanisms (like Gaussian or Laplace), it is mechanism-agnostic. As long as you can compute or estimate the moments of the PLRV for a mechanism, you can use the Edgeworth accountant. This includes non-standard or custom mechanisms, even if they don't have a known RDP or analytical form. This makes the Edgeworth Accountant very flexible, especially for research and advanced applications ⁷.

- The *RDP Accountant* [18] is a holistic Rényi Differential Privacy (RDP) accountant designed to provide tighter and more efficient privacy accounting for differentially private algorithms, especially DP-SGD with fixed-size subsampling (i.e. mini-batches). The RDP accountant supports both with replacement (FSwR) and without replacement (FSwoR) subsampling strategies.

How does it work? First, each mechanism is analyzed in terms of its Rényi divergence at various orders α . For instance, the Gaussian mechanism has a closed-form RDP expression. Then, RDP composes linearly: the total RDP at order α is the sum of the RDPs of individual mechanisms. Finally, after composition, the total RDP is converted back to a standard (ϵ, δ) -DP guarantee using a tight conversion formula.

As the Analytical Moments Accountant, the RDP accountant requires that, for each mechanism, you can compute its Rényi divergence. If a mechanism has no known or computable Rényi divergence, or is non-parametric or adaptive in a way that breaks the assumptions, then it cannot be directly used with the RDP accountant.

The RDP accountant is designed to be modular, composable, and efficient, making it suitable for complex training pipelines. It also supports many mechanisms, especially Gaussian and its subsampled variants.

Subsampling plays a crucial role in improving the privacy guarantees of differentially-private algorithms, especially in the context of privacy accountants. In the context of training deep learning models using stochastic gradient descent (e.g. DP-SGD, NoisySGD, or NoisyAdam), subsampling refers to selecting a random subset of the dataset at each training step, rather than using the full dataset (i.e. using a minibatch).

⁷DP in LLMs: Large language models (LLMs) present additional challenges due to their size and complexity. The EW-Tune framework [11] has been proposed to implement DP in LLMs with the help of the Edgeworth Accountant [215]. The Edgeworth series expansion models the distribution of noisy gradients by incorporating higher-order terms in its approximation. This approach enables a more precise calibration of noise injection for the differentially-private fine-tuning of large language models (LLMs) [237], effectively balancing model utility and privacy guarantees. Where DP-SGD [1] would introduce significant noise into the gradients, EW-Tune reduces variance while maintaining strict differential privacy guarantees.

Privacy Accountant	DP Framework	Composition Type	DP Mechanism
Moments [1]	(ϵ, δ) -DP	log moments	Gaussian
Analytical [219]	RDP	Subsampled RDP	Gaussian, Laplace, exponential
CLT [29]	GDP	Central Limit Theorem	Gaussian
Amortized [132]	GDP	Adaptive	Gaussian
Edgeworth [215]	f -DP	Edgeworth expansion	Any
RDP [18]	RDP	Subsampled RDP	Gaussian, Laplace, exponential

Table 3. Privacy accountants for DP in deep learning.

Subsampling reduces the effective ϵ by a factor proportional to the sampling rate q [126][127][68, Section 3.6]. This phenomenon is known as privacy amplification by subsampling: when a mechanism \mathcal{M} is (ϵ, δ) -DP on a dataset D , running the mechanism \mathcal{M} on a random subsample of D with sampling rate q improves privacy to approximately $(q\epsilon, q\delta)$ -DP. In other words, subsampling inherently boosts privacy, which is useful in deep learning and large-scale data analysis.

Subsampling introduces randomness in the data access pattern, which leads to privacy amplification. If a mechanism is differentially private when applied to the full dataset, then applying it to a random subset reduces the effective privacy loss. Intuitively, each individual has a lower chance of being included in any given minibatch, so their data is less exposed.

Subsampling is a privacy amplifier and a key enabler of practical DP training in deep learning. Privacy accountants leverage subsampling to tighten privacy bounds and enable efficient composition. Providing tighter and more realistic privacy guarantees and reducing per-step privacy cost enable longer training under fixed privacy budgets. This is especially important in deep learning, where thousands of gradient updates are common during model training.

Some accountants (e.g. the Amortized Accountant) are specifically designed to exploit subsampled mechanisms and provide accurate tracking over adaptive compositions. In fact, there are techniques that resort to privacy amplification by iteration. They efficiently handles multiple adaptive queries with minimal privacy budget. For instance, the Sparse Vector Technique (SVT) [66][68, Section 4], which answers whether a sequence of queries exceeds a noisy threshold, satisfies ϵ -DP while adaptively halting after a fixed number of "above-threshold" answers.

There are different types of subsampling. In Poisson subsampling, used by the Moments Accountant, each data point is included independently with some probability, which makes it easier to analyze. In uniform subsampling, a fixed-size minibatch is selected uniformly at random, which is more realistic in practice, yet harder to analyze. When sampling a dataset, you can also choose whether elements can be selected more than once (i.e. whether sampling is with or without replacement). That decision can affect the tightness of privacy bounds (e.g. see RDP Accountant [18]).

In some contexts, such as LLM training [39], additional sampling approaches might be useful to protect user-level privacy, in order to ensure that the contribution of individual users to the model is protected. Whereas example-level sampling (ELS) involves clipping gradients at the example level, user-level sampling (ULS) operates at the user level, allowing for gradient aggregation over all the examples provided by a single user. The distinction between ELS and ULS is also important in federated learning. In deep learning models, ELS is common practice in DP-SGD, since it is easier to analyze using tools like the moments and RDP accountants. ULS, on the other hand, requires more sophisticated privacy accounting (e.g., f -DP, Edgeworth, or user-level RDP accountants). In LLM fine-tuning [39], a novel user-level

DP accountant provides tight user-level privacy guarantees even when using ELS, helping to bridge the gap between the two approaches (by analyzing the privacy amplification effect of ELS at the user level). Under fixed computational budgets, ULS tends to provide stronger privacy guarantees and better model performance, particularly when stringent privacy protection is required or when larger computational resources are available.

Large Language Models (LLMs) require special care to achieve differential privacy. LLMs are trained over millions of gradient updates, each potentially leaking a small amount of information, so privacy accounting is paramount. Subsampling amplifies privacy, reduces per-step privacy cost, and enables tighter accounting in LLM training. The combined effect of privacy accounting and subsampling allows training LLMs with strong privacy guarantees efficiently.

In general, when training deep learning models, the privacy accountant keeps track of the accumulated privacy loss spending after each training step that updates model parameter values. If the accumulated privacy loss spent exceeds the allowed limits (i.e. the privacy budget), then training is stopped and no further updates are performed on the model. This is just a privacy-driven form of early stopping, a common regularization technique used in deep learning.

3.4.3 Privacy for free. Privacy for free, in the context of machine learning, refers to the idea that certain machine learning techniques can inadvertently provide some degree of privacy protection without requiring explicit privacy-preserving mechanisms like differential privacy. For instance, the regularization effect of dropout can also help to obscure individual data points in a dataset, making it harder to infer sensitive information about them.

Privacy for free” suggests that certain model architectures, training methods, or data-handling strategies can offer privacy benefits inherently, without the need for heavy cryptographic tools or added noise (as in differential privacy). You can design and train models in a way that inherently protects privacy without significantly sacrificing performance or requiring complex privacy-preserving techniques.

Can we get privacy benefits as a byproduct of good model design or training practices? Let us examine some examples and strategies that have been explored in research to answer that question:

- *Regularization:* Over-parameterized models (like large neural networks) can memorize training data, but with proper regularization, they generalize better and leak less private information. This can lead to reduced memorization, offering a form of implicit privacy. Regularization helps prevent overfitting, which in turn can reduce the risk of leaking sensitive data:
 - L1 and L2 regularization (a.k.a. weight decay) adds a penalty term to the loss function. L1 (Lasso) encourages sparsity, i.e. many weights become zero, whereas L2/ (ridge), encourages small, evenly distributed weights. Smaller weights reduce the model capacity to memorize specific data points.
 - Dropout randomly drops units (along with their connections) during training, which prevents the co-adaptation of neurons and forces the neural net to learn more robust features. Reducing reliance on specific neurons that might encode sensitive patterns provides an implicit form of differential privacy. In fact, dropout stochasticity mimics noise injection, potentially reducing sensitivity. Dropout is computationally cheap and widely used in deep learning.
 - Early stopping monitors validation loss and halts training when performance stops improving. From a privacy angle, preventing overfitting also limits the model exposure to training data, reducing memorization. Since training a model for too long can lead to memorization of specific data points, early stopping based on validation loss can reduce privacy leakage. Therefore, early stopping is another common (and simple) technique with privacy benefits.

- *Data minimization*: Using only the data necessary for the task (e.g. with the help feature selection algorithms and dimensionality reduction techniques) can reduce the risk of exposing sensitive information. This aligns with the principle of privacy by design.
- *Data condensation* [58], originally designed for improving training efficiency, can be used for private data generation, thus providing privacy for free. Instead of training models on the raw data, that can be attacked by MIA and suffer potential data leakages, we can generate a synthetic dataset from the raw data and then use the synthetic data for model training.
- *Data augmentation* applies transformations (e.g., rotation, cropping, noise) to training data with the goal of increasing data diversity without collecting more data. From the perspective of privacy, data augmentation makes it harder for the model to memorize specific examples, especially in image and text domains.
- *Adversarial training* trains the model on adversarial examples, i.e. inputs with small, intentional perturbations. As data augmentation, it is intended to improve robustness and generalization (this time, against adversarial attacks). From the privacy angle, both techniques encourage the model to learn more general patterns rather than memorizing specific details.
- *Data blending* blurs the identity of individual training samples, making memorization harder. You can combine two inputs and their labels linearly (“mixup”) or cut and paste patches between training images, while mixing labels according (“cutmix”).
- *Label smoothing* replaces hard labels (e.g., [0, 1]) with soft labels (e.g., [0.1, 0.9]) to prevent the model from becoming overconfident. As a form of input perturbation, it reduces the model tendency to memorize exact label associations.
- *Model compression and pruning*: Since smaller models tend to memorize less, techniques like pruning or knowledge distillation can reduce the model capacity to retain specific training examples, offering privacy as a side effect.
- *Unintended privacy from generalization*: Models that generalize well tend to memorize less and research has shown that generalization and privacy can be correlated, meaning that improving one can help the other, sometimes for free.

A caveat is in order: Even though all the aforementioned techniques might help improve privacy, they do not guarantee it. These methods reduce the privacy risk but don’t eliminate it. Unlike differential privacy, these strategies do not provide formal privacy guarantees in the form of quantifiable privacy bounds. Moreover, they are often context-dependent, what works in one domain (e.g., image classification) may not apply in another (e.g., NLP with LLMs).

Can neural networks really provide privacy for free? Unintended memorization is a persistent, hard-to-avoid issue that can have serious consequences, e.g. extracting unique secrets such as credit card numbers [36]. Even well-generalized models can memorize and leak sensitive data. Average data points are rarely leaked, but outliers are frequently subject to memorization and, consequently, privacy leakage. However, removing the layer of outliers that are most vulnerable to a privacy attack exposes a new layer of previously-safe points to the same attack (the onion effect of memorization [35]). Defending against memorization without training with rigorous privacy guarantees are unlikely to be effective [35]. In short, privacy for free is possible in some cases, but not reliable without formal mechanisms.

Sometimes, a DP variant of an existing technique can be devised. For instance, a Bayesian interpretation of dropout, whose noise is added with regularization in mind, can lead to DP guarantees under the zCDP framework [73]. The

original dropout provides a form of randomization akin to DP, with implicit privacy benefits but no formal DP guarantee, even though the privacy-utility trade-off can be studied empirically for particular applications.

It has widely been hypothesized that regularization is a privacy enhancer. A theoretical discussion of sensitivity and stability [68], noting that regularization-like constraints (e.g., bounded weights) align with DP goals, provides some theoretical foundation for designing DP strategies (without an empirical study in this case).

As we already saw, objective perturbation in SVMs [42] directly uses regularization to achieve DP. Regularization reduces sensitivity in empirical risk minimization (ERM), enabling DP with less noise and improving utility.

Some studies [147] have compared DP techniques with traditional regularization methods, such as dropout and L2 regularization, a.k.a. weight decay. Classic regularization, commonly used to prevent overfitting, provides similar levels of protection against membership inference and model inversion attacks. Empirical results suggest that regularization methods may offer a more effective trade-off between privacy and utility (i.e. model performance). Unlike DP-SGD, which incurs in significant accuracy loss, regularization techniques can provide privacy protection with a minimal impact on performance.

3.4.4 Privacy-preserving Deep Learning algorithms. Several surveys cover recent advances in the application of differential privacy ideas within the field of deep learning [165] [24] [177] [51] [50].

In supervised learning, differential privacy has been used to train convolutional neural networks (CNNs) using the zero-concentrated (zCDP) [110] and the local (LDP) [155] differential privacy frameworks. Even without a formal underpinning, noise has been added to the inputs of a recurrent neural network to improve the robustness of LSTM predictions [144].

In distributed settings, multiparty learning mechanisms can make sharing input datasets unnecessary [194], whereas federated learning techniques can learn a shared model by aggregating locally-computed updates [160] [159].

DarKnight [106] is a framework for large DNN training that relies on cooperative execution between trusted execution environments (TEE) and accelerators, where the TEEs provide privacy and integrity verification, while accelerators perform the bulk of the linear algebraic computation to optimize the performance: customized data encodings based on matrix masking create input obfuscation within a TEE and, then, the obfuscated data is then offloaded to GPUs.

In unsupervised learning, DP techniques have been incorporated into autoencoders, generative adversarial networks, and other deep generative models.

An autoencoder is an artificial neural network used to learn efficient representations of data, typically for the purpose of dimensionality reduction, denoising, or feature learning. An autoencoder consists of two main parts: the encoder compresses the input into a smaller representation (called the latent space or bottleneck), while the decoder reconstructs the original input from the compressed representation. The goal is for the output to be as close as possible to the input, forcing the network to learn the most important features. There are different kinds of autoencoders: denoising autoencoders are trained to remove noise from input data, sparse Autoencoders encourage sparsity in the latent space, and variational autoencoders (VAE) learn a probabilistic latent space, useful in generative modeling. How can we ensure that autoencoders do not inadvertently leak information about any individual data point in the training set? Apart from DP-SGD, several specific techniques have been proposed:

- The deep private auto-encoder (dPA) [182] enforces ϵ -differential privacy by perturbing the objective function of the traditional deep auto-encoder, rather than its results. Objective perturbation, i.e. adding noise to the loss function, might be preferable to output perturbation, i.e. adding noise directly to the output of the encoder or decoder, which is simpler but may degrade reconstruction quality more significantly.

- A differentially-private autoencoder-based generative model (DP-AuGM) [43] is designed to defend against model inversion, membership inference, and GAN-based attacks. An autoencoder is trained with private data using a differentially private training algorithm and then the encoder is published (the decoder is dropped). New data will be generated (encoded) by feeding the users' own data (i.e., public data) into the encoder and the generated data can then be used to train any ML model. An analogous differentially-private variational autoencoder-based generative model (DP-VaeGM) [43] is also robust against membership inference attacks, result that leads to the conjecture that the key to defend against model inversion and GAN based attacks is not due to differential privacy but the perturbation of training data.
- The autoencoder-based differentially private text transformation (ADePT) [134] is a utility-preserving differentially private text transformation algorithm using auto-encoders that is designed to offer robustness against attacks and produce transformations with high semantic quality that perform well on downstream NLP tasks. In this case, the autoencoder first transforms a given text input into some latent representation, clipping and noising are applied on the latent sentence representations returned by the encoder, and text generation by the decoder on the noisy latent representation, offering another DP mechanism.
- PATE-AAE [230], with an adversarial autoencoder (AAE) that replaces the generative adversarial network (GAN) in a private aggregation of teacher ensembles (PATE), has been proposed as a solution for ensuring differential privacy in speech applications. Following the PATE scheme (i.e. an ensemble of noisy outputs label synthetic samples and guarantee ϵ -DP), the PATE-AAE framework consists of an AAE-based generator and a PATE-based classifier.

Generative adversarial networks (GANs) simultaneously train two competing neural network models. The first model, the generator, learns how to generate realistic synthetic data by mapping its input, which is a randomly sampled noise vector, to samples drawn from the distribution it learns about training data. The second model, the discriminator, learns how to distinguish between samples drawn from the training dataset and samples generated by the generator model, i.e. between real and generated data. The two models are trained until the generator produces data indistinguishable from real data. From the privacy point of view, the main problem with GANs is that the density of the learned generative distribution could concentrate on the training data points, meaning that they can easily remember specific training samples. Solutions often incorporate differential privacy into the training process of the discriminator, typically via DP-SGD with moments accountant:

- Differentially Private Discriminator (DP-GAN) [226]: DP-SGD is applied to the discriminator, which is the model that accesses the real data. The generator is trained using the feedback from the discriminator, so it never directly accesses the real data.
- DP-WGAN [226] [3] addresses the challenges of instability in GAN training by using the Wasserstein distance function as the GAN training objective.
- PATE-GAN [120] trains multiple discriminators (teachers) on disjoint subsets of the data. Then, it uses their noisy, aggregated outputs to train a student discriminator. The generator is trained using the student discriminator. Empirical studies [86] have shown that PATE, similarly to DP-SGD, has a disparate effect on the under/over-represented classes (DP-WGAN evens the classes while PATE-GAN increases the imbalance) but in a much milder magnitude making it more robust. For PATE, unlike DP-SGD, the privacy-utility trade-off is not monotonically decreasing but is much smoother and inverted U-shaped, meaning that adding a small degree of privacy actually helps generalization.

- G-PATE [148] is another framework based on PATE that can be used to train scalable data generator with DP guarantees while preserving high generated data utility. G-PATE combines GANs to generate data and private aggregation among different discriminators to ensure strong privacy guarantees. G-PATE trains a student data generator with an ensemble of teacher discriminators.
- DataLens [213] generates synthetic data in a differentially-private (DP) way given sensitive input data. Combining PATE with GANs, multiple discriminators are trained as teacher models and teachers vote using gradient compression techniques to train a student generator.
- GANobfuscator [227] uses GANs to obfuscate sensitive data, essentially generating synthetic data that mimic the statistical properties of the original data while preserving privacy. Sensitive data is fed into a discriminator with a privacy-preserving layer and, as in previous proposals, random perturbation is only applied to the discriminator.
- DTGAN [136] is designed specifically for generating synthetic tabular data with strong privacy guarantees. Based on a conditional Wasserstein GAN architecture, DTGAN confirms that applying DP-SGD on the discriminator achieves better data utility and training stability under strict privacy budgets and demonstrates strong resistance to membership inference attacks, keeping success rates close to random guessing ($\approx 50\%$).

Empirical studies [87] profile how DP generative models for tabular data distribute privacy budgets across rows and columns. Since deep generative models spend their budgets per iteration, their behavior is less predictable with varying dataset dimensions, but they are also more flexible. Probabilistic graphical models (e.g. MST or PrivBayes) outperform GANs at simple tasks like capturing statistics or marginal similarity, yet GANs perform better at more complex tasks like classification (and they are also more scalable than their probabilistic counterparts).

3.5 Model-agnostic differential privacy

Different approaches can be applied to make any machine learning technique differentially private, including federated learning and PATE (Private Aggregation of Teacher Ensembles).

3.5.1 Differentially-Private Federated Learning (DP-FL). Federated learning is a machine learning technique that enables multiple devices or servers (often called clients) to collaboratively train a shared model without sharing their raw data. Instead of sending data to a central server, each client trains the model locally on its own data and only shares model updates (like gradients or weights) with a central server. The server then aggregates these updates to improve the global model. Since raw data never leaves the local device, federated learning is privacy-preserving by design.

DP implementations can follow two different strategies: central DP [64], when users trust the data curator, and local DP [125] [75], which ensures privacy when clients do not trust the data curator. Both approaches can be used for DP federated learning. In local DP-FL, clients add noise to their model updates (or falsify their answers with some probability) before sending them to the server/curator. In central DP-FL, the server adds noise during the aggregation of updates.

Federated Learning (FL) enables collaborative model training across decentralized clients without centralizing sensitive datasets [159] by aggregating locally-computed updates to optimize a global objective (i.e. privately training ML models). Differentially-Private Federated Learning (DP-FL) [91] incorporates the principles of DP-SGD into the private training of deep learning models.

Not sharing raw data is not enough for strong privacy protection. A survey of DP-FL ideas [176] organizes existing proposals into three loose categories. Some proposals try to protect users' privacy during model training by using

mechanisms such as secure multiparty computing (SMC) [23] [170] or different cryptographic techniques (e.g. using homomorphic encryption [99] or functional [235] encryption). A second set of proposals act on the network architecture of the FL system (e.g. introducing proxies, a latent layer, or P2P communication) or constrain communication (e.g. sharing partial information [194]). A third category focus on controlling resource consumption (e.g. transmission rate allocation and communication overhead) in different contexts (over multiple access channels and on the Internet of things, IoT, respectively).

Since FL requires multiple rounds of synchronization, some privacy accounting is also needed. For instance, several techniques have been proposed for adjusting the variance of the Gaussian noise to guarantee that the privacy loss does not exceed the desired privacy budget ϵ after a given number of synchronization rounds using LDP [131] [223].

DP-FL [91] enables a set of clients, each with a private dataset, to collaboratively train a shared global model while ensuring client-level (ϵ, δ) -differential privacy using a distributed version of DP-SGD [1]. In each communication round, a random subset of clients is sampled to receive the current global model. Then, each client optimizes the model using its local data. Each local update of the model parameters is clipped. Finally, the global model is updated adding Gaussian noise to the averaged clipped updates.

The cost of privacy can be seen as the difference between the fitness of a privacy-preserving machine-learning model and the fitness of trained machine-learning model in the absence of privacy concerns. In DP-FL, this cost has an upper bound that is inversely proportional to the combined size of the training datasets squared and the sum of the privacy budgets squared [79].

Combining differential privacy with secure multiparty computation enables the reduction of noise injection as the number of parties increases without sacrificing privacy [206]. This combination protects against inference threats over both the messages exchanged during model training and the final trained model. It can be used to train a variety of ML models, including decision trees, CNNs, and SVMs.

Frameworks like FedDPDR-DPML [218] combine DP-PCA and objective-perturbed DP-SVM training across multiple participants holding distributed data. Models are trained locally with DP, and a central server aggregates them to produce a global model (e.g., using weighted averaging based on local data size).

As in central DP, regularization can act as a proxy for local differential privacy. In federated learning, regularization stabilizes local models before DP aggregation. We can apply regularization techniques to user-level models before privatizing outputs (via randomized responses or added noise). Regularization ensures models generalize across users, reducing overfitting to individual data and thus enhancing privacy. In fact, since the first proposals on training deep networks from decentralized data it was conjectured that, in addition to lowering communication costs, model averaging produces a regularization benefit similar to that achieved by dropout [159].

The distinction between example-level sampling (ELS) and user-level sampling (ULS) is especially important in federated learning.

- Example-level sampling (ELS) randomly selects individual examples, i.e. each individual data point is sampled independently with some probability, regardless of which user it came from. It is the common approach in centralized training, e.g. DP-SGD, where minibatches are formed by randomly selecting examples from the entire dataset, and per-example gradient clipping is applied to limit the influence of any single example. ELS, therefore, protects individual examples. It is simpler to implement and often yields better utility when users contribute few examples. However, ELS often leads to weaker privacy guarantees in federated settings, where

users may contribute many examples. Additionally, ELS is vulnerable to correlated data leakage from the same user (it does not protect user-level privacy if users contribute multiple examples).

- User-level sampling (ULS) randomly selects entire users and all their data. Entire users (and all their data) are sampled with some probability. When a specific user is selected, all their data is used in that round. ULS is common in federated learning, specially when data is naturally partitioned by user (e.g. mobile devices, health, or financial data). ULS protects entire users and all their data. It provides stronger privacy guarantees than ELS and it is more realistic for real-world privacy concerns. It is better suited for real-world applications where users contribute multiple, possibly correlated, examples (e.g., protecting a person’s entire browsing history). However, ULS requires more sophisticated privacy accounting (e.g., f-DP, Edgeworth Accountant, or user-level RDP).

Whereas ELS is simpler and fits centralized training, ULS is more suitable for real-world privacy expectations, especially in federated settings. The choice affects both the privacy guarantees and the accounting method used.

Empirical experiments on fine-tuning LLMs [39] found that ELS can outperform ULS in terms of utility when users contribute few examples or when computation budgets are tight (i.e. limited computational resources). On the other hand, ULS is superior when strong privacy guarantees are needed, users contribute many or diverse examples, and the computation budget allows for more sophisticated training.

3.5.2 Private Aggregation of Teacher Ensembles (PATE). The PATE framework is a method for training ML models with strong differential privacy guarantees. PATE trains multiple models (teacher models) on disjoint subsets of the training data. Teachers models are aggregated to form an ensemble whose predictions are guaranteed not to leak those of individual models. The noisy aggregation of the teacher model outputs is then used to train a student model. This student model is learned without direct access to the raw data nor the parameters of individual teacher models. The resulting student model can be shared and, due to the aggregator’s guarantee, is theoretically guaranteed not to leak information (from a DP point of view).

The federated learning methods of the previous section protect the privacy of data privacy for multiple users in distributed ML, but the encryption algorithm and security protocols needed increase their computational overhead. PATE also provides data protection with some overhead, since it needs to train an additional student model. Training is more complex but can offer stronger privacy guarantees.

- The original PATE [178] uses an ensemble of teacher models trained on disjoint subsets of sensitive data. These teachers vote on labels for unlabeled public data, and a student model is trained on this labeled public data. Laplacian noise is added to the teacher votes to ensure (ϵ, δ) -differential privacy.

The noisy aggregated labels are used to train a student model, effectively shielding individual data contributions while preserving utility. A key detail is the use of a confidence-based thresholding mechanism, so that predictions with sufficient agreement among teachers are released, reducing noise variance and enhancing the quality of the transferred knowledge. PATE offers good performance when public unlabeled data is available.

The canonical PATE computes output labels by aggregating the predictions of a (potentially distributed) collection of teacher models via a voting mechanism. The mechanism adds noise to attain a DP guarantee with respect to the teachers’ training data. However, the use of noise, which is needed for DP and makes PATE predictions stochastic, enables new forms of leakage of sensitive information [217]. For a given input, an adversary can exploit this stochasticity to extract high-fidelity histograms of the votes submitted by the underlying teachers. From these histograms, the adversary can learn sensitive data, even when DP guarantees are not violated. Counterintuitively, this attack becomes easier as we add more noise to provide stronger

differential privacy. When learning from sensitive data, care must be taken to ensure that training algorithms address privacy norms and expectations.

An empirical evaluation of PATE robustness against adversaries using Monte Carlo sampling [216] shows that teacher votes can be inferred. The results confirm that adversaries are more successful in recovering voting information when the vote-aggregation mechanism introduces noise with a larger variance, revealing a trade-off between achieving confidentiality and differential privacy in collaborative ML. Unfortunately, in practice, PATE provides virtually no privacy guarantees when an adversary is allowed unlimited access to the trained model.

- Individualized/personalized PATE (iPATE) [22] adapts the privacy mechanism to provide personalized privacy budgets. Standard PATE assumes uniform privacy needs across all data points. However, in real-world scenarios, some individuals may require stronger privacy than others (e.g. when dealing with healthcare data, some patients may consent to broader data use, while others require strict privacy).

In iPATE, each data point (or individual) is assigned a custom privacy budget. The noise added during aggregation is adjusted based on the individual's privacy requirement. This adjustment allows for more utility (better model performance) while still respecting individual privacy constraints.

Instead of using a uniform privacy budget across all participants, iPATE [22] provides individualized privacy guarantees introducing some variants into the PATE framework:

- *Upsampling PATE* replicates data points with higher privacy budgets across multiple teacher models. Unlike the original PATE, whose data partitions were disjoint, upsampling allows data with higher privacy budgets to be learned by several teachers. This upsampling mechanism increases the influence of less privacy-sensitive data while preserving privacy for more sensitive data. Upsampling either increases the number of teachers or allocates more data per teacher, with the former generally providing better utility by reducing vote count variance.
- *Vanishing PATE* reduces the influence of data points with stricter privacy requirements by limiting their participation in teacher training. This way, privacy cost is minimized for sensitive individuals.
- *Weighting PATE* assigns weights to teacher votes based on the privacy budget of the data they were trained on. Weighting balances utility and privacy by giving more influence to less sensitive data. Instead of changing the composition of teacher model training datasets, as in the previous variants, the weighting mechanism is applied during the aggregation of teacher votes. As a result, teachers handling data with higher privacy budgets are given more influence, improving the accuracy of the final student model.

Given a set of teacher models, the aggregation mechanism in iPATE assigns a personalized privacy budget ϵ_i to each user, rather than enforcing a global ϵ . The aggregation mechanism adds personalized Laplacian or Gaussian noise to each teacher vote, calibrating that noise to an individualized sensitivity function that is adaptively determined from the user's privacy budget ϵ_i . By adapting noise levels individually, iPATE balances privacy and utility on a per-user basis, making it particularly effective in applications with heterogeneous privacy preferences.

- Scalable PATE (sPATE) [179] extends PATE to large scale tasks and uncured, imbalanced training data with errors. Specifically, sPATE introduces new noisy aggregation mechanisms for the ensemble of teacher models that require less noise to be added to provide privacy guarantees.

The original PATE can be computationally expensive because many teacher models must be trained and their votes must be aggregated. Scalable PATE introduces some optimizations to make the framework more efficient and applicable to larger datasets.

sPATE resorts to techniques such as the parallel training of teachers, model compression, and efficient vote aggregation. Training teacher models in parallel speeds up the process. Model compression reduces the size or number of teacher models. More efficient vote aggregation can be devised to reduce computation requirements. The Confident Aggregator and the GNMax (Gaussian Noise Maximum) mechanism come into play to replace the LNMax (Laplace noise maximum) mechanism in the original PATE.

- The Confident Aggregator [179] only answers to student queries when teacher votes are confident. As a benefit, this aggregation mechanism reduces privacy cost by avoiding noisy answers when teachers disagree.

The Confident Aggregator improves the original PATE framework by introducing a confidence threshold before answering a query. When a query is made, we first check whether the consensus among teachers is sufficiently high. If the number of votes assigned to the most popular label among teachers (or the vote margin, i.e. difference between the top two vote counts) exceeds a predefined threshold, we accept the query. If not, we reject it. The threshold itself is randomized in order to provide privacy during this checking process.

Once a query has been accepted, the original noisy aggregation mechanism is applied. Noise is added to each of the vote counts corresponding to each label and the label with more votes is returned. When the votes are not confident and the query is rejected, no label is returned.

The final result is a reduction of the number of queries that consume the privacy budget unnecessarily. Since the queries that are answered have a high consensus among teachers, the budget needed to answer them is very small. Ambiguous or low-consensus queries are avoided to prevent wasting the privacy budget. In other words, the Confident Aggregator filters the queries that would actually deplete the privacy budget in the original PATE framework.

For comparable levels of student performance, the privacy budget afforded by the Confident Aggregator is smaller than in the original PATE noisy aggregation mechanism .

- The GNMax Aggregator uses Gaussian noise instead of Laplace noise in the vote aggregation phase. GNMax stands for Gaussian Noisy Maximum: After teachers vote, the vote tally is adjusted with Gaussian noise, and the label with the highest noisy vote count is chosen

The Gaussian distribution is more concentrated than the Laplace distribution used to introduce noise in the original PATE framework. Compared to Laplace noise, Gaussian noise often results in less distortion, especially when the number of classes is large. In addition, the chance of teacher consensus is increased by using more concentrated noise.

Under Rényi Differential Privacy (RDP), which is more suitable for composition over many queries, tighter privacy bounds can be provided for Gaussian noise. RDP allows for more accurate tracking of cumulative privacy loss over multiple queries.

GNMax can be combined with smooth sensitivity analysis to adaptively adjust noise based on the sensitivity of each query. When using in conjunction with a confident aggregator, the Confident GNMax only releases labels when the noisy vote margin exceeds a threshold. There is also an Interactive GNMax that dynamically adjusts the noise or decision threshold based on query history. Yet another variant, called PV-PATE [109], uses an Improved Confident-GNMax Aggregator to further refine query filtering.

The PATE framework transfers to a student model the knowledge of an ensemble of teacher models, with intuitive privacy provided by training teachers on different subsets of data and strong privacy guaranteed by noisy aggregation of teachers' answers.

The PATE framework has been combined with multiple techniques for different applications, including adversarial autoencoders for speech classification [230] and generative adversarial networks, in PATE-GAN [120], G-PATE [148], and DataLens [213].

Existing PATE variants, however, assume that teachers, who cast their votes, are honest. When they are corrupted by an adversary, attacks can become drastically more potent as the attacker is able to control more participants [216]. Defending against this kind of attacks might not be trivial.

Even though PATE offers DP guarantees, its robustness is in question. Several vulnerabilities have been found (e.g. vote inference and corrupt teachers) and stronger noise mechanisms and vote aggregation strategies are needed.

4 Differential privacy in practice: Empirical evaluation of differential privacy methods

Measuring the effectiveness of algorithmic techniques proposed to achieve differential privacy involves evaluating both their privacy guarantees and their utility (i.e., how well they preserve the usefulness of the data). Differential privacy is inherently a trade-off between these two aspects, and precise measurement requires a combination of theoretical analysis, empirical evaluation, and statistical tools.

Table 4 and its continuation, Table 5, summarize many of the ML algorithms discussed in the previous Section, indicating the DL framework they are based on and the mechanism they employ to add differential privacy to the ML models they train (i.e. input, mechanism design, objective, and output perturbation, as well as the sample and aggregate strategy common in parallel and distributed ML). Those tables also report how privacy effectiveness, performance/utility, and the privacy-utility trade-off are evaluated through experiments in the publications where the different privacy-preserving algorithms were originally proposed.

4.1 Measuring privacy effectiveness

Most publication on DP-ML techniques tend to ignore the empirical evaluation of privacy, since it is theoretically guaranteed by the corresponding DP mechanism. They just report the calculated (ϵ, δ) or other DP parameters (e.g., RDP, zCDP) achieved by the DP model, based on the DP mechanism and privacy accountant used.

However, implementing DP algorithms correctly requires careful attention to details, including numerical stability (e.g., avoiding floating-point vulnerabilities [166]), proper sensitivity analysis, and correct privacy accounting [89]. Subtle algorithmic errors or analytical mistakes, such as incorrect noise calibration or the misuse of composition rules (e.g. most variants of SVT are actually not private [151]), can silently undermine the claimed privacy guarantees [150].

Even when the DP analysis and implementation are correct, a timing side-channel attack [104] may lead to the catastrophic compromise of a DP system.⁸

While DP provides theoretical upper bounds on privacy loss (ϵ, δ) , these bounds are worst-case estimates. Effectiveness in terms of privacy should be empirically measured by ensuring the algorithm adheres to the (ϵ, δ) -DP definition

⁸DP guarantees are defined in the language of probability distributions. DP implementations use floating-point arithmetic precision. A single sample from a naive implementation of the Laplace mechanism allows distinguishing between two adjacent datasets with probability more than 35% [166]. Unlike operations on integers that are typically constant-time on modern CPUs, floating-point arithmetic exhibit input-dependent timing variability. Covert channels are notoriously difficult to avoid and they devastating for DP: "if a side channel allows the adversary to learn even a single bit of private information, the differential privacy guarantees are already broken!" [104]

ML Algorithm	DP		Empirical evaluation		
	Framework	Perturbation	Privacy	Utility	Trade-off
<i>Symbolic AI</i>					
PPID3 [209] & PPRDT [211]	-	-	-	-	-
DiffPID3 [84]	ϵ	mechanism	-	$acc(D)$	-
DiffPC4.5 [84]	ϵ	mechanism	-	$acc(D)$	$acc(\epsilon)$
Private-RDT [112] [113]	ϵ	output	-	-	$acc(\epsilon)$
DPRF [185]	ϵ	mechanism	-	AUC	$AUC(\epsilon)$
DPDF [81]	ϵ	mechanism	S	$acc(\Theta), F_1, AUC$	$acc(\epsilon)$
LPDT [153]	LDP	aggregate	-	$MSE(\Theta)$	$MSE(\epsilon)$
DPBoost [140]	ϵ	mechanism	-	$acc(E)$	$acc(\epsilon)$
OpBoost [143]	LDP	mechanism	-	-	$acc(\epsilon), MSE(\epsilon)$
FederBoost [202]	LDP	mechanism	-	$AUC(\Theta)$	$AUC(\epsilon)$
<i>Probabilistic AI</i>					
DP-NB [210]	ϵ	output	-	-	$acc(\epsilon)$
AHE-NB [141]	ϵ	aggregate	-	-	-
LDP-NB [234]	LDP	aggregate	-	-	$acc(\epsilon)$
PrivBayes [238]	ϵ	aggregate	-	$acc(\Theta)$	$SMI(\epsilon), acc(\epsilon)$
MST [157]	RDP	mechanism	-	-	-
DPSyn [138]	(ϵ, δ)	mechanism	-	-	-
PrivatePGM [158]	ϵ	mechanism	-	acc	$acc(\epsilon)$
DPKMeans [174]	ϵ	aggregate	-	-	-
PPKMeans [19]	-	mechanism	-	$F_1(\Theta), NMI(\Theta)$	-
LDPKMeans [225]	LDP	aggregate	-	$NMI(\Theta), RE(\Theta)$	$NMI(\epsilon), RE(\epsilon)$
PGME [123]	(ϵ, δ)	mechanism	-	-	-
DP-LMGMM [180]	ϵ	mechanism	-	-	$acc(\epsilon)$
MR [232]	LDP	mechanism	-	-	$MAE(\epsilon)$
<i>Statistical AI</i>					
PTR linear regression [63]	(ϵ, δ)	aggregate	-	-	-
Objective LR [41] [42]	(ϵ, δ)	objective	-	$acc(D)$	$acc(\epsilon)$
FM linear regression [239]	ϵ	objective	-	$MSE(D), MSE(d)$	$MSE(\epsilon)$
FM logistic regression [239]	ϵ	objective	-	$acc(D), acc(d)$	$acc(\epsilon)$
PrivateSVM [191]	(ϵ, δ)	output	-	-	-
Objective SVM [42]	(ϵ, δ)	objective	-	$acc(D)$	$acc(\epsilon)$
TDDL [115]	(ϵ, δ)	objective	-	-	$acc(\epsilon), MSE(\epsilon)$
DPPCA-SVM [111]	$(\epsilon, 0)$	output	-	$acc(D)$	$acc(\epsilon)$
PCA-DPSVM [111]	$(\epsilon, 0)$	output	-	-	$acc(\epsilon)$
DPDR-DPML [218]	(ϵ, δ)	objective	-	$acc(D), acc(d)$	$acc(\epsilon)$

Table 4. DP in ML algorithms: Differential privacy in symbolic, probabilistic, and statistical AI. Legend: ϵ (privacy budget), S (sensitivity), acc (accuracy, for classification problems), MSE (mean-squared error, for regression problems), MAE (mean absolute error), F_1 (F1 score), AUC (area under the ROC curve), NMI (normalized mutual information), NMI (sum of mutual information), RE (relative error), d (number of dimensions), $|D|$ (training examples), $|E|$ (ensemble size), θ (model parameters), Θ (model hyperparameters).

ML Algorithm	DP		Empirical evaluation		
	Framework	Perturbation	Privacy	Utility	Trade-off
<i>Deep Learning</i>					
DP-SGD [1]	ϵ	mechanism	-	$acc(\Theta)$	$acc(\epsilon)$
NoisySGD [29]	GDP	mechanism	$\delta(\epsilon), \epsilon(\delta)$	$acc(\Theta)$	$acc(\epsilon)$
DPAGD-CNN [110]	zCDP	mechanism	-	-	$acc(\epsilon), acc(\delta)$
LATENT [155]	LDP	mechanism	-	-	$acc(\epsilon)$
DP-LSTM [144]	-	input	-	MSE, acc	-
DSSGD [194]	-	aggregate	-	$acc(\Theta)$	-
FedAvg [159]	-	aggregate	-	$acc(\Theta)$	-
dPA [182]	ϵ	objective	-	$acc(D)$	$acc(\epsilon)$
DP-AuGM [43]	(ϵ, δ)	mechanism	-	$acc(\Theta)$	$acc(\epsilon), acc(\delta)$
DP-VaeGM [43]	(ϵ, δ)	mechanism	-	$acc(\Theta)$	$acc(\epsilon), acc(\delta)$
ADePT [134]	(ϵ, δ)	mechanism	-	-	$MIA(acc)$
PATE-AAE [230]	(ϵ, δ)	aggregate	-	-	$acc(\epsilon)$
DP-[W]GAN [226]	(ϵ, δ)	mechanism	-	$W(\Theta)$	$acc(\epsilon), W(\epsilon), DW(\epsilon)$
PATE-GAN [120]	(ϵ, δ)	aggregate	-	$AUC(\Theta)$	$AUC(\epsilon), SRA(\epsilon)$
G-PATE [148]	RDP	aggregate	-	$AUC, acc(\Theta)$	$acc(\epsilon), IS(\epsilon)$
DataLens [213]	RDP	aggregate	$\epsilon(\Theta)$	$acc(\Theta)$	$acc(\epsilon), IS(\epsilon)$
DTGAN [136]	RDP	mechanism	MIA, AIA	$acc, F_1, AUC, APR, JS, W$	
GANobfuscator [227]	(ϵ, δ)	mechanism	-	$acc(D)$	$IS(\epsilon), JS(\epsilon), MIA(\epsilon)$

Table 5. DP in ML algorithms (continued): Differential privacy in deep learning. Legend: ϵ (privacy budget), S (sensitivity), acc (accuracy, for classification problems), MSE (mean-squared error, for regression problems), MAE (mean absolute error), F_1 (F1 score), AUC (area under the ROC curve), APR (average precision score), NMI (normalized mutual information), IS (Inception score [100] to measure the quality of generated data), JS (Jensen-Shannon divergence), W (Wasserstein distance), DW (dimension-wise probability & prediction), SRA (synthetic ranking agreement), RE (relative error), MIA (membership inference attack success rate, as a quantifiable measure of privacy preservation), AIA (attribute inference attack), d (number of dimensions), $|D|$ (training examples), $|E|$ (ensemble size), θ (model parameters), Θ (model hyperparameters).

and by analyzing how tightly this guarantee is achieved. Auditing DP implementations can help detect bugs and also potential misuse [150].

The empirical validation of DPML techniques can be performed by simulation studies for testing DPML algorithms and adversarial testing for evaluating a particular ML model.

In simulation studies, you generate synthetic datasets D and D' differing in one record. Then, by running the ML training algorithm, you estimate the empirical ratio $P[M(D) \in S]/P[M(D') \in S]$. If it exceeds $e^\epsilon + \delta$, the mechanism is less effective than claimed. You can also vary dataset size, dimensionality, and noise levels to test robustness. Statistical tests (e.g., t-test, Kolmogorov-Smirnov) can be used to compare output distributions $M(D)$ and $M(D')$. Effective DPML algorithms should make these distributions statistically indistinguishable up to the (ϵ, δ) bound. Such simulation studies are essential because, even when formal results might imply that learning while preserving differential privacy is possible, theoretical results often depend on technical assumptions, which do not always hold in practice [129].

For testing particular ML models, adversarial testing is a reasonable empirical validation strategy. You can simulate privacy attacks on the ML model to estimate the actual privacy loss and compare it to the theoretical ϵ it should provide.

The experimental evaluation using simulated privacy attacks can provide a lower bound on the actual privacy leakage of a specific DP implementation [183], by measuring the actual success rate of adversarial attacks.

- Membership inference attacks (MIA) [196], sometimes known as tracing attacks [103], pose a binary classification problem in which the attacker tries to predict whether a particular example was used to train the model from the model outputs (i.e. a black-box attack). They are the most common type of attack for the empirical privacy evaluation of DPML implementations. Attacks typically observe the model behavior on the target: when the model is highly confident or achieves very low loss on the example, the example is more likely to have been in the training set (due to potential overfitting or memorization) [183]. DP effectiveness is high if the attack success rate approaches the random guess baseline (e.g., 50% for binary membership).
- Attribute inference attacks (AIA) [224], also called model inversion attacks [83], pose a regression problem where the attacker attempts to predict the values of a sensitive target attribute, provided he has black-box access to the trained ML model. The attacker might even infer sensitive attributes that were not used directly as features during training [163].
- Data reconstruction/extraction attacks [146] try to recover specific training data instances (or parts thereof) from the trained model. These are generally considered stronger attacks than MIA. Privacy is effective if the reconstruction error remains high despite the model output.

Other attacks try to infer specific information from the target model [116]: memorization attacks try to exploit the ability of high capacity models to memorize sensitive patterns [36], model stealing attacks try to recover the model parameters [203], hyperparameter stealing attacks try to recover the hyperparameters used during model training, and property inference attacks try to infer whether the training data set has a specific property.

Common attack strategies include thresholding (loss < threshold or confidence > threshold) [233], shadow models (auxiliary models trained on similar data learn a classifier that distinguishes members from non-members based on the target model output [196]), perturbation-based attacks (analyzing the change in loss when the input is slightly perturbed, i.e., Merlin attack [117]), and combined attacks (combining loss and perturbation information, i.e., Morgan attack [117]).

Attack success can be measured using standard classification metric (e.g. accuracy, precision, recall, AUC) or tailored metrics, such as membership advantage ($MA = TPR - FPR$). The precision, or positive predictive value ($PPV = TP/P$), is suitable for evaluating risk when the base rate of membership might be low (i.e. for imbalanced classes). Instead of using averages across a whole dataset, it is recommended that attack success should focus on the most vulnerable examples, since privacy is often a per-instance property, e.g. TPR at low FPR for the most vulnerable sample [2].

Security evaluation curves [16] [17] can also be used to evaluate the security of DPML systems. The security evaluation curves characterize the system performance under different attack strengths and attackers with different level of knowledge, thus providing a more comprehensive evaluation of the system performance under attack.

Attempts to define purely deterministic privacy metrics (e.g., based on k-anonymity [49] [192] or non-stochastic information theory [77]) generally fail to provide the robustness of DP, particularly against adversaries with auxiliary information. Some researchers have explored “noiseless” privacy by leveraging the inherent randomness in data [34]. However, DP strength lies precisely in its probabilistic nature and its ability to handle arbitrary side knowledge.

It should also be noted that the DP definition is not always the same. Neighboring datasets are allowed to change in different ways: add-or-remove one record, zero-out one record, or replace one record. The first two have comparable

semantics for a fixed ϵ , whereas the guarantee for replace-one is approximately twice as strong. Care should therefore be taken when comparing ϵ based on different criteria [183].

Strong formal privacy guarantees ($\epsilon \leq 1$) frequently result in large utility drop for large ML models, and may be infeasible in practice [183]. Reasonable privacy guarantees ($\epsilon \leq 10$) provide a reasonable level of anonymization for many applications [183]. Any finite ϵ is an improvement over a model with no privacy protections, even with weak to no formal privacy guarantees: a finite ϵ moves the model into a regime where further privacy improvements can be quantified and even large ϵ s can indicate a substantial decrease in a model's ability to memorize user data [183]. However, for a large ϵ (e.g., > 10), the DP guarantee is not sufficient evidence of data anonymization and additional measures (e.g., empirical privacy auditing, demonstrated robustness to attacks, or pre-processing to remove privacy-sensitive data from the training set) may be necessary before releasing or deploying a ML model [183].

It is important to note that empirical attacks provide only a lower bound on privacy leakage [183]. They demonstrate vulnerability against known attack strategies but do not guarantee protection against future, more sophisticated attacks. Their effectiveness also depends heavily on the adversary's assumptions and capabilities [149].

Even under DP, classifiers can be used to infer private attributes accurately in realistic data, given that coarse properties of the population taken together can be combined to build a model that can be applied to individuals with high accuracy. Existing anonymization methods tend to ignore this issue. Why? Because attacks are inherently probabilistic [47]. Although we can empirically determine that some attacks are correct in associating an individual with their true value a large fraction of the time, they do not indicate with certainty for which individuals this is a true inference and for which it is a false positive. The classifier might often be right, but we are never sure when.

In any case, it is not sufficient to naively apply differential privacy to data, and assume that this is sufficient to address all privacy concerns. You should always evaluate the privacy leakage for a given privacy budget ϵ before publishing a dataset or a ML model [176].

4.2 Measuring utility effectiveness

In differential privacy, utility refers to how well the algorithm preserves the performance or usefulness of the data analysis. The metrics used to estimate utility depend on the specific task to be done, from statistical queries and machine learning to synthetic data generation.

The standard experimental methodology for evaluating DPML algorithms typically involves training a non-private version of the ML model on the target dataset to establish a baseline utility performance (baseline model) and training one or more versions of the DPML algorithm on the same dataset (DP models), varying the privacy parameters (e.g., ϵ , δ , noise multiplier, clipping threshold) and other learning algorithm hyperparameters to explore different points on the privacy-utility spectrum. Then, the utility of both the baseline and the DP models are evaluated on a held-out test dataset using standard ML metrics relevant to the task at hand.

Assessing the utility of a DPML model depends heavily on its intended use case. For ML models, standard performance metrics are often used. In classification problems, common metrics include accuracy, precision, recall, F1-score (often macro-averaged for multi-class problems), the whole confusion matrix, the ROC curve, and Area Under the Curve (AUC) for both ROC and precision-recall curves. In regression problems, common metrics include Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), or Mean Absolute Percentage Error (MAPE) for regression tasks. For special-purpose applications (e.g. synthetic data generation using generative AI techniques), customized metrics are also used, such as normalized mutual information (NMI), the Inception score (IS) [100], Jensen-Shannon divergence (JS),

total variation distance (TVD), Kullback–Leibler divergence (KL), Wasserstein distance (W), dimension-wise probability (DW), synthetic ranking agreement (SRA), average precision score (APR), or relative error (RE).

Apart from using metrics corresponding to the intended goal of the ML, the difference in the objective function (e.g., cross-entropy loss) between private and non-private models could also be used. In some situations, such as synthetic data generation, you can also evaluate downstream task performance (i.e. train models on synthetic data and compare their performance against models trained on real data). Effective techniques yield similar performance in downstream tasks.

Sometimes, the sensitivity of those metrics is evaluated with respect to different ML algorithm hyperparameters, such as the number of dimensions in input data, the number of training examples, the ensemble size, or even the number of training epochs. For instance, you can quantify how many additional samples are needed to achieve the same accuracy as a non-private model (sample complexity). Effective algorithms minimize this overhead [1].

With respect to hyperparameter optimization, the recommended approach, when possible, is to do all model architecture search and hyperparameter tuning on a proxy public dataset (with a distribution similar to the private data), and only use the private training dataset to train the final DP model.

When a single data set is used in multiple computations, the composition rule for privacy implies that we should choose a total ϵ for all computations on the data and budget privacy for each computation. When hyperparameter tuning must be performed on private data, any privacy guarantees reported should clearly state that all ‘data touches’ are accounted for. This should include, at least, a guarantee that applies only to the use of private data in training the final model, but ideally also a (weaker) guarantee that accounts for the use of private data in hyperparameter tuning [183].

It should be noted that no single utility metric is universally applicable [25]. The choice of metrics should be guided by the specific goals of the task the ML model is intended for. A suite of metrics is often necessary to provide a comprehensive assessment.

Comparing these metrics between the DP model and a non-private baseline quantifies the utility loss incurred for privacy. The cost of DP is a reduction in model utility. In other words, effectiveness is quantified by comparing the privatized output (the DP model) to the true non-private result (the non-private baseline model). Effectiveness is high if the utility drop is small (lower deviation indicates better utility). Effectiveness is excellent if DP models outperform baselines in utility for the same privacy budget ϵ .

Finally, it has been observed that the cost of DP is not borne equally [6]: accuracy of DP models drops much more for the underrepresented classes and subgroups. Critically, this gap is bigger in the DP model than in the non-DP model, i.e., if the original model is unfair, the unfairness becomes worse once DP is applied.

4.3 The privacy-utility trade-off

Effectiveness is ultimately about optimizing the privacy-utility trade-off. In practice, this is quantified by measuring utility for different privacy budgets ϵ :

- *Privacy-utility curves*: $\text{utility}(\epsilon)$

Plot utility (e.g., accuracy, MSE) against ϵ . Since lower ϵ means stronger privacy, the curve typically shows that as privacy increases (ϵ decreases), utility decreases. Effective algorithms shift the curve upward (higher utility for the same ϵ) or leftward (same utility for smaller ϵ). The curve is usually monotonically decreasing: better privacy leads to worse utility. Small changes in ϵ can have large or small effects on utility depending on the particular DP mechanism and dataset, making it hard to predict outcomes without empirical testing. The

resulting curve is often non-linear, with steep drops in utility at low ϵ values and diminishing returns at higher ϵ . For instance, for the Laplace mechanism, utility decreases linearly with $1/\epsilon$, while the Gaussian mechanism with zCDP may offer sublinear degradation. Since the privacy budget ϵ is not an intuitive concept, empirical curves help understand what a given ϵ means in practice.

- *Rate-distortion theory analogy: $\epsilon(\text{utility})$*

Treat privacy loss ϵ as “rate” and utility loss as “distortion”. In rate-distortion theory, the goal is to compress data while minimizing the distortion (loss of fidelity) under a constraint on the bit rate (amount of information transmitted). The privacy-utility trade-off in DP is analogous to the rate-distortion trade-off in lossy compression: stronger privacy (lower ϵ) requires more noise and, therefore, produces higher distortion (lower utility); weaker privacy (higher ϵ) requires less noise and leads to lower distortion (higher utility). Analogous to the rate-distortion function that gives the minimum rate needed to achieve a given distortion level, a privacy-utility function $\epsilon(\text{utility})$ gives the minimum privacy loss needed to achieve a certain utility level. Effective algorithms achieve a lower distortion for a given rate, analogous to optimal compression.

In rate-distortion theory, mutual information plays a central role in quantifying the trade-off between the amount of information retained and the distortion introduced during compression. Mutual information measures how much information the compressed version retains about the original data by quantifying the reduction in uncertainty about the original data given the compressed data. The mutual information represents the minimum number of bits needed to encode the data so that the distortion does not exceed the desired level. Lower distortion (better fidelity) requires higher mutual information (more bits), and vice versa.

The information bottleneck method, a generalization of rate-distortion theory, can also be applied to DP to formalize how much information about the input can be retained while satisfying privacy constraints. The information bottleneck (IB) framework formalizes how to extract the most relevant information from a variable while compressing it. Its goal is finding a compressed representation that retains as much information about the model output as possible and, at the same time, discards as much irrelevant information as possible about the input. The IB method aligns naturally with the DP goal of releasing information about a dataset while limiting what can be inferred about individuals. In deep learning, IB-inspired objectives can be used to learn private embeddings that retain task-relevant information while discarding sensitive details.

- *Pareto frontier: $(\epsilon, \text{utility})$*

Identify the set of $(\epsilon, \text{utility})$ pairs where no improvement in utility is possible without increasing ϵ . Each point represents a DP mechanism with a specific privacy level (ϵ) and corresponding utility. The Pareto-optimal points are the mechanisms for which no other mechanism is both more private and more useful. Points not on the frontier are dominated, i.e. there exists another mechanism that is better in both privacy and utility.

The Pareto frontier helps identify the optimal trade-offs between privacy and utility, visualizing the cost of stronger privacy in terms of utility loss. For mechanism selection, you just choose the particular DP mechanism on the Pareto frontier that best fits your privacy and utility requirements. For benchmarking, you compare new DP algorithm implementations against the frontier to assess their efficiency. Effective algorithms lie closer to the ideal frontier.

It is essential that experiments are performed to estimate the privacy-utility trade-off for each particular ML model and training dataset. The actual privacy loss might be overestimated. Conservative theoretical bounds may overestimate ϵ , reducing perceived effectiveness. Formal privacy guarantees assume a worst-case adversary; real-world effectiveness

may differ if adversaries are weaker. But privacy loss could also be underestimated. Apart from implementation subtleties, which might negatively affect differential privacy guarantees, effectiveness also varies with dataset properties such as size, dimensionality, or variance. This is particularly clear in small datasets, which might yield poor utility when strong privacy is required.

For a given privacy level, we need a larger sample size n to achieve the same level of utility or approximation error. When data is scarce, we have to settle for the lowest ϵ for which the sacrifice in utility is acceptable. There is little consensus on how to choose that budget [193]. A KDD'2008 paper [88] suggested appropriate δ should be less than $1/n^2$.

Unlike metrics like accuracy or error rates, ϵ has no upper bound and no universally-accepted good or bad values. What counts as acceptable privacy varies by context. Selecting optimal values for ϵ and δ requires expertise, as overly restrictive parameters exacerbate utility loss, while lenient ones weaken privacy. Methods like individualized PATE require intricate tuning of upsampling and weighting parameters, with misconfigurations leading to up to 10% accuracy degradation. In distributed settings, DP-FL faces challenges from data heterogeneity, non-i.i.d. distributions amplify noise impact, reducing convergence rates by up to 20% compared to centralized DP-SGD. Furthermore, privacy amplification in federated settings scales with the number of clients, requiring larger noise to achieve the same ϵ , further affecting utility. With strict privacy constraints (i.e., low privacy budgets), dataset sampling and/or early stopping are likely to prove beneficial with respect to utility. For a specific ML algorithm, effectiveness might depend on its noise calibration, sensitivity handling, and composition properties, all of which must be tailored to the task at hand.

The aforementioned practical difficulties, rooted in the mathematical constraints of differential privacy and the practical demands of machine learning, highlight the need for experimental evaluation. The effectiveness of differential privacy techniques is the result of the interaction among theoretical privacy bounds, utility metrics, learning algorithm hyperparameters, and problem-specific details. Empirical validation and benchmarking is essential for the real-world application of DP in ML models.

5 Conclusion

Differential Privacy (DP) is the leading theoretical framework for data privacy, offering mathematically rigorous and provable guarantees against a wide range of inference attacks. Its evolution from the foundational ϵ -DP definition to more nuanced variants like (ϵ, δ) -DP, zCDP, RDP, GDP, and LDP reflects a continuous effort to enhance analytical tractability, particularly for composition, and capture the privacy properties of specific mechanisms. Core theoretical properties like sensitivity calibration, composition theorems, and post-processing immunity provide the tools for building complex privacy-preserving systems.

Differential Privacy provides an invaluable standard for reasoning about and quantifying privacy loss in an era of large-scale data analysis and machine learning. In machine learning, DP has been successfully adapted to a wide range of algorithms. While objective perturbation is common for convex models like SVMs, and specialized techniques exist for decision trees and ensembles, gradient perturbation via DP-SGD has become the de facto standard for deep learning. This involves careful gradient clipping, noise addition, and sophisticated privacy accounting over iterative training.

Evaluating DPML systems needs a multi-pronged approach. Theoretical (ϵ, δ) bounds provide worst-case guarantees, standard ML metrics assess utility loss, and empirical privacy attacks (like MIA) offer practical lower bounds on information leakage.

The significance of DP extends beyond technical achievements, as it addresses ethical and regulatory demands in data-sensitive domains. For instance, DP aligns with privacy regulations such as the European Union’s General Data Protection Regulation (GDPR) [74], which mandates strict controls on personal data processing [32].

ML security-related issues go beyond differential privacy. They can be broadly classified into five categories [229]: training set poisoning, backdoors in the training set, adversarial example attacks, model theft, and recovery of sensitive training data. Differential Privacy (DP) offers a robust mechanisms for privacy-preserving machine learning, yet its deployment entails significant trade-offs.

Deploying DP introduces additional challenges and trade-offs, including increased computational costs, potential utility degradation due to noise, and the complexity of tuning privacy parameters like ϵ and δ . DP, however, is well-aligned with the general goals of machine learning. Memorizing a particular training example is a violation of privacy, but it is also a form of overfitting that harms ML models generalization capability. Hence, DP implies some form of stability in ML.

Once the models are trained, two additional concerns must be considered when models are deployed to make predictions [37]: prediction requests might contain sensitive information (user privacy) and the knowledge embedded in the trained models should be protected (both training data and model hyperparameters). DP might help in protecting training data, but additional measures should be taken to address the remaining privacy considerations.

A critical evaluation of DP’s benefits and liabilities is needed, particularly in the context of machine learning, where high-dimensional data and iterative model training amplify these issues. By evaluating DP’s strengths and weaknesses, we conclude our comprehensive survey on the current state of DP in machine learning and its practical implications for data-sensitive applications.

On the positive side, DP use in ML offers some significant benefits in practice:

- *Sensitive data analysis*: DP allows valuable statistical analysis and ML model training on sensitive datasets that might otherwise be inaccessible due to privacy regulations or concerns.
- *Provable privacy guarantees*: In sharp contrast with the heuristic nature of older anonymization techniques, DP provides a rigorous, mathematical definition of privacy, allowing for quantifiable upper bounds on information leakage concerning individuals’ data. Formal guarantees can potentially streamline data access protocols within organizations.
- *Privacy accounting*: DP mechanisms are composable, allowing the construction of complex privacy-preserving workflows and algorithms from simpler building blocks. Composition theorems provide bounds on the cumulative privacy loss. Advanced privacy accounting enhances the tightness of (ϵ, δ) -guarantees.
- *Post-processing immunity*: Any computation performed on the output of a DP mechanism, without accessing the original private data, inherits the same DP guarantee. This greatly simplifies the design and analysis of data pipelines.
- *Robustness to auxiliary information*: DP formal guarantees hold irrespective of any side information an adversary might possess, now or in the future. This offers resilience that is not provided by other techniques.
- *Defense against inference attacks*: DP provides inherent protection against a range of common ML privacy attacks, including membership inference, attribute inference, and data reconstruction/extraction.
- *Scalability*: DP methods extend privacy to large-scale and distributed systems. DP-SGD [1] can be used to train large neural networks, whereas EW-Tune [11], using the Edgeworth Accountant [215] that leverages Edgeworth

series expansions to model privacy loss distributions, can be used for LLM fine-tuning. DP-FL [159] scales DP to federated settings, applying local DP-SGD and secure aggregation to train models across multiple clients.

Despite its many strengths, DP faces significant limitations and challenges in the context of ML:

- *Privacy-utility trade-off*: The most fundamental DP challenge is the inherent trade-off between the strength of the privacy guarantee (ϵ, δ) and the utility of the output (e.g., accuracy in classification, fidelity in data generation). Stronger privacy (smaller ϵ) generally requires adding more noise, which degrades utility. Empirical studies indicate that the trade-off between model utility and privacy loss is complex. The fundamental trade-off between privacy and utility often leads to difficult choices, particularly for complex ML models where achieving strong privacy guarantees can substantially degrade performance. Achieving high utility with strong privacy guarantees remains difficult for many complex tasks.
- *Disparate impact and fairness*: A major concern is that DP can disproportionately affect ML model utility for underrepresented classes and subgroups, exacerbating existing biases. The potential for DP mechanisms to exacerbate unfairness and biases requires careful consideration and mitigation strategies.
- *Interpretability of privacy guarantees*: The interpretation of the meaning of specific (ϵ, δ) guarantees is far from intuitive. It can create communication barriers among project stakeholders, specially those without a theoretical background on DP.
- *Sensitivity analysis*: Accurately bounding the global sensitivity of complex functions or ML training procedures can be difficult, potentially leading to overly conservative noise addition. Many practical deployments use relatively large ϵ values, where theoretical worst-case guarantees are weak, and rely on empirical evidence of resistance to known attacks.
- *Implementation complexity and correctness*: The many subtleties involved in the proper implementation of DP might lead to unnoticed privacy problems and additional privacy and security risks.
- *Hyperparameter tuning*: DPML models are often highly sensitive to many hyperparameters (ϵ, δ , clipping norm, noise scale, learning rate, batch size...). Finding optimal settings is complex, computationally expensive, and the tuning process itself can leak information if not done carefully or accounted for in the privacy budget.
- *Computational overhead*: DP mechanisms, e.g, DP-SGD for deep learning which requires per-example gradient computations, can impose significant computational and memory overhead compared to non-private methods. DP-SGD and EW-Tune can increase training time by 2x-3x, while EW-Tune's Edgeworth Accountant requires an additional 20% computational cost for iterative statistical computations. These computational costs limit DP applicability in resource-constrained situations.

The tension between DP theoretical foundations and practical issues with respect to utility, complexity, and fairness is unlikely to disappear. While DP provides robust protection against certain privacy threats in theory, its application requires careful consideration of its many trade-offs and limitations. Integrating DP with other PETs (privacy-enhancing technologies, like cryptography) and considering it within broader ethical frameworks for responsible AI will be increasingly important. Bridging the gap between the theoretical promises of DP and its practical, equitable, and widespread deployment requires continued innovation and interdisciplinary collaboration between computer scientists, statisticians, domain experts, ethicists, and policymakers. While not a panacea, DP is a critical component in the ongoing effort to harness the power of data responsibly.

The challenges outlined above suggest some avenues for further research, such as improving the privacy-utility trade-off by designing more sophisticated DP algorithms, addressing fairness and bias by understanding the mechanisms

causing unfairness and designing interventions, reducing the computational and memory footprint of DP algorithms, and developing tailored DP algorithms for newer ML techniques like Large Language Models (LLMs) or Graph Neural Networks (GNNs). From a theoretical point of view, improved privacy accounting and tighter composition theorems would enable a more efficient use of the privacy budget. Formal verification tools and techniques are needed to increase trust in the implementation of complex DP algorithms. Open questions remain regarding fundamental limits (lower bounds), the relationships between different DP variants, the optimal choice of privacy parameters, and the interplay between privacy and other constraints like computation and communication. Future progress of DP in ML will likely involve context-specific solutions tailored to particular algorithms, data types, and application domains, moving beyond generic mechanisms.

Some final considerations regarding DP should be mentioned, as they are related to the opportunities and risks of DP adoption in diverse fields and applications. The trade-off between individual privacy and model utility is inherent to DP and it should be always be made explicit: higher privacy sacrifices model utility, higher utility sacrifices individual privacy. Appropriate best practices and guidelines on privacy-preserving technologies like DP should be promoted. Even though DP quantifies privacy loss, as a theoretical upper bound, it might also encourage more data (and model) sharing. Due to the many subtleties associated to the use of DP, its use can create a false sense of security and that can actually increase actual privacy risks. For auditing and evaluating DP robustness, the use of privacy attacks (i.e. MIAs, AIAs, and reconstruction attacks) is recommended to assess the actual privacy offered by DP systems in practice and identify potential implementation flaws.

References

- [1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (Vienna, Austria) (CCS '16)*. Association for Computing Machinery, New York, NY, USA, 308–318. <https://doi.org/10.1145/2976749.2978318>
- [2] Michael Aerni, Jie Zhang, and Florian Tramèr. 2024. Evaluations of Machine Learning Privacy Defenses are Misleading. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security (Salt Lake City, UT, USA) (CCS '24)*. Association for Computing Machinery, New York, NY, USA, 1271–1284. <https://doi.org/10.1145/3658644.3690194>
- [3] Moustafa Alzantot and Mani Srivastava. 2019. Differential Privacy Synthetic Data Generation using WGANs. https://github.com/nsl/nist_differential_privacy_synthetic_data_challenge/
- [4] Shahab Asodeh, Fady Alajaji, and Tamás Linder. 2014. Notes on information-theoretic privacy. In *2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. 1272–1278. <https://doi.org/10.1109/ALLERTON.2014.7028602>
- [5] Shahab Asodeh, Jiachun Liao, Flavio P. Calmon, Oliver Kosut, and Lalitha Sankar. 2021. Three Variants of Differential Privacy: Lossless Conversion and Applications. *IEEE Journal on Selected Areas in Information Theory* 2, 1 (2021), 208–222. <https://doi.org/10.1109/JSAIT.2021.3054692>
- [6] Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. 2019. Differential Privacy Has Disparate Impact on Model Accuracy. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2019/file/fc0de4e0396fff257ea362983c2dda5a-Paper.pdf
- [7] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. 2001. On the (Im)possibility of Obfuscating Programs. In *Advances in Cryptology — CRYPTO 2001*, Joe Kilian (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 1–18.
- [8] Rina Foygel Barber and John C. Duchi. 2014. Privacy and Statistical Risk: Formalisms and Minimax Bounds. [arXiv:1412.4451 \[math.ST\]](https://arxiv.org/abs/1412.4451) <https://arxiv.org/abs/1412.4451>
- [9] Raef Bassily, Kobbi Nissim, Uri Stemmer, and Abhradeep Thakurta. 2017. Practical Locally Private Heavy Hitters. [arXiv:1707.04982 \[cs.DS\]](https://arxiv.org/abs/1707.04982) <https://arxiv.org/abs/1707.04982>
- [10] Raef Bassily, Adam Smith, and Abhradeep Thakurta. 2014. Differentially Private Empirical Risk Minimization: Efficient Algorithms and Tight Error Bounds. [arXiv:1405.7085 \[cs.LG\]](https://arxiv.org/abs/1405.7085) <https://arxiv.org/abs/1405.7085>
- [11] Rouzbeh Behnia, Mohammadreza Reza Ebrahimi, Jason Pacheco, and Balaji Padmanabhan. 2022. EW-Tune: A Framework for Privately Fine-Tuning Large Language Models with Differential Privacy. In *2022 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, Orlando, FL, USA, 560–566. <https://doi.org/10.1109/icdmw58026.2022.00078>
- [12] Amos Beimel, Hai Brenner, Shiva Prasad Kasiviswanathan, and Kobbi Nissim. 2014. Bounds on the sample complexity for private learning and private data release. *Machine learning* 94 (2014), 401–437.

- [13] Amos Beimel, Shiva Prasad Kasiviswanathan, and Kobbi Nissim. 2010. Bounds on the Sample Complexity for Private Learning and Private Data Release. In *Theory of Cryptography*, Daniele Micciancio (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 437–454.
- [14] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. 1998. Relations among notions of security for public-key encryption schemes. In *Advances in Cryptology — CRYPTO '98*, Hugo Krawczyk (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 26–45.
- [15] Thomas B. Berrett, László Györfi, and Harro Walk. 2021. Strongly universally consistent nonparametric regression and classification with privatised data. *Electronic Journal of Statistics* 15, 1 (2021), 2430 – 2453. <https://doi.org/10.1214/21-EJS1845>
- [16] Battista Biggio and Fabio Roli. 2018. Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (Toronto, Canada) (CCS '18). Association for Computing Machinery, New York, NY, USA, 2154–2156. <https://doi.org/10.1145/3243734.3264418>
- [17] Battista Biggio and Fabio Roli. 2018. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition* 84 (2018), 317–331. <https://doi.org/10.1016/j.patcog.2018.07.023>
- [18] Jeremiah Birrell, Reza Ebrahimi, Rouzbeh Behnia, and Jason Pacheco. 2024. Differentially Private Stochastic Gradient Descent with Fixed-Size Minibatches: Tighter RDP Guarantees with or without Replacement. In *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Eds.), Vol. 37. Curran Associates, Inc., 11087–11131. https://proceedings.neurips.cc/paper_files/paper/2024/file/14fef58f09f2ebe69306e0a322e3be2b-Paper-Conference.pdf
- [19] Chandan Biswas, Debasis Ganguly, Dwaipayan Roy, and Ujjwal Bhattacharya. 2019. Privacy Preserving Approximate K-means Clustering. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (Beijing, China) (CIKM '19). Association for Computing Machinery, New York, NY, USA, 1321–1330. <https://doi.org/10.1145/3357384.3357969>
- [20] Matthieu Bloch, Onur Günlü, Aylin Yener, Frédérique Oggier, H. Vincent Poor, Lalitha Sankar, and Rafael F. Schaefer. 2021. An Overview of Information-Theoretic Security and Privacy: Metrics, Limits and Applications. *IEEE Journal on Selected Areas in Information Theory* 2, 1 (2021), 5–22. <https://doi.org/10.1109/JSAIT.2021.3062755>
- [21] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. 2005. Practical privacy: the SuLQ framework. In *Proceedings of the Twenty-Fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems* (Baltimore, Maryland) (PODS '05). Association for Computing Machinery, New York, NY, USA, 128–138. <https://doi.org/10.1145/1065167.1065184>
- [22] Franziska Boenisch, Christopher Mühl, Roy Rinberg, Jannis Ihrig, and Adam Dzedzic. 2022. Individualized PATE: Differentially Private Machine Learning with Individual Privacy Guarantees. arXiv:2202.10517 [cs.LG] <https://arxiv.org/abs/2202.10517>
- [23] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2016. Practical Secure Aggregation for Federated Learning on User-Held Data. arXiv:1611.04482 [cs.CR] <https://arxiv.org/abs/1611.04482>
- [24] Amine Boulemtafes, Abdelouahid Derhab, and Yacine Challal. 2020. A review of privacy-preserving techniques for deep learning. *Neurocomputing* 384 (2020), 21–45. <https://doi.org/10.1016/j.neucom.2019.11.041>
- [25] Claire McKay Bowen. 2021. Utility Metrics for Differential Privacy: No One-Size-Fits-All. <https://www.nist.gov/blogs/cybersecurity-insights/utility-metrics-differential-privacy-no-one-size-fits-all>
- [26] Leo Breiman. 2001. Random forests. *Machine Learning* 45 (2001), 5–32.
- [27] Hai Brenner and Kobbi Nissim. 2010. Impossibility of Differentially Private Universally Optimal Mechanisms. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23–26, 2010, Las Vegas, Nevada, USA*. IEEE Computer Society, 71–80. <https://doi.org/10.1109/FOCS.2010.13>
- [28] Hai Brenner and Kobbi Nissim. 2014. Impossibility of Differentially Private Universally Optimal Mechanisms. *SIAM J. Comput.* 43, 5 (2014), 1513–1540. <https://doi.org/10.1137/110846671> arXiv:https://doi.org/10.1137/110846671
- [29] Zhiqi Bu, Jinshuo Dong, Qi Long, and Weijie J Su. 2020. Deep learning with gaussian differential privacy. *Harvard data science review* 2020, 23 (2020), 10–1162.
- [30] Valerii Vladimirovich Buldygin and IU V Kozachenko. 2000. *Metric characterization of random variables and random processes*. Vol. 188. American Mathematical Soc., Providence, USA.
- [31] Mark Bun, Cynthia Dwork, Guy N. Rothblum, and Thomas Steinke. 2018. Composable and versatile privacy via truncated CDP. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing* (Los Angeles, CA, USA) (STOC 2018). Association for Computing Machinery, New York, NY, USA, 74–86. <https://doi.org/10.1145/3188745.3188946>
- [32] Mark Bun and Thomas Steinke. 2016. Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds. In *Theory of Cryptography*, Martin Hirt and Adam Smith (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 635–658.
- [33] Mark Bun and Thomas Steinke. 2016. Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds. arXiv:1605.02065 [cs.CR] <https://arxiv.org/abs/1605.02065>
- [34] Paul Burchard, Anthony Daoud, and Dominic Dötterrer. 2023. Empirical Differential Privacy. arXiv:1910.12820 [cs.LG] <https://arxiv.org/abs/1910.12820>
- [35] Nicholas Carlini, Matthew Jagielski, Chiyuan Zhang, Nicolas Papernot, Andreas Terzis, and Florian Tramèr. 2022. The Privacy Onion Effect: Memorization is Relative. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 13263–13276. https://proceedings.neurips.cc/paper_files/paper/2022/file/564b5f8289ba846ebc498417e834c253-Paper-Conference.pdf

- [36] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*. 267–284.
- [37] Shan Chang and Chao Li. 2018. Privacy in Neural Network Learning: Threats and Countermeasures. *IEEE Network* 32, 4 (2018), 61–67. <https://doi.org/10.1109/MNET.2018.1700447>
- [38] Anne-Sophie Charest and Yiwei Hou. 2017. On the Meaning and Limits of Empirical Differential Privacy. *Journal of Privacy and Confidentiality* 7, 3 (May 2017), 53–66. <https://doi.org/10.29012/jpc.v7i3.406>
- [39] Zachary Charles, Arun Ganesh, Ryan McKenna, H. Brendan McMahan, Nicole Mitchell, Krishna Pillutla, and Keith Rush. 2024. Fine-Tuning Large Language Models with User-Level Differential Privacy. arXiv:2407.07737 [cs.LG] <https://arxiv.org/abs/2407.07737>
- [40] Kamalika Chaudhuri and Daniel J. Hsu. 2012. Convergence Rates for Differentially Private Statistical Estimation. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress. <http://icml.cc/2012/papers/663.pdf>
- [41] Kamalika Chaudhuri and Claire Monteleoni. 2008. Privacy-preserving logistic regression. In *Advances in Neural Information Processing Systems*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou (Eds.), Vol. 21. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2008/file/8065d07da4a77621450aa84fee5656d9-Paper.pdf
- [42] Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. 2011. Differentially Private Empirical Risk Minimization. *Journal of Machine Learning Research* 12, 29 (2011), 1069–1109. <http://jmlr.org/papers/v12/chaudhuri11a.html>
- [43] Qingrong Chen, Chong Xiang, Minhui Xue, Bo Li, Nikita Borisov, Dali Kaarfar, and Haojin Zhu. 2018. Differentially Private Data Generative Models. arXiv:1812.02274 [cs.CR] <https://arxiv.org/abs/1812.02274>
- [44] Rui Chen, Benjamin C. Fung, Philip S. Yu, and Bipin C. Desai. 2014. Correlated network data publication via differential privacy. *The VLDB Journal* 23, 4 (Aug. 2014), 653–676. <https://doi.org/10.1007/s00778-013-0344-8>
- [45] Kewei Cheng, Tao Fan, Yilun Jin, Yang Liu, Tianjian Chen, Dimitrios Papadopoulos, and Qiang Yang. 2021. SecureBoost: A Lossless Federated Learning Framework. *IEEE Intelligent Systems* 36, 6 (2021), 87–98. <https://doi.org/10.1109/MIS.2021.3082561>
- [46] Léo Colisson. 2016. L3 internship report: Quantum analog of differential privacy in term of Rényi divergence.
- [47] Graham Cormode. 2011. Personal privacy vs population privacy: learning to attack anonymization. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (San Diego, California, USA) (KDD '11)*. Association for Computing Machinery, New York, NY, USA, 1253–1261. <https://doi.org/10.1145/2020408.2020598>
- [48] Paul Cuff and Lanqing Yu. 2016. Differential Privacy as a Mutual Information Constraint. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (Vienna, Austria) (CCS '16)*. Association for Computing Machinery, New York, NY, USA, 43–54. <https://doi.org/10.1145/2976749.2978308>
- [49] Tore Dalenius. 1986. Finding a needle in a haystack or identifying anonymous census records. *Journal of Official Statistics* 2, 3 (1986), 329.
- [50] Saswat Das and Subhankar Mishra. 2024. *Advances in Differential Privacy and Differentially Private Machine Learning*. Springer Nature Singapore, Singapore, 147–188. https://doi.org/10.1007/978-981-97-0407-1_7
- [51] Lea Demelius, Roman Kern, and Andreas Trügler. 2025. Recent Advances of Differential Privacy in Centralized Deep Learning: A Systematic Survey. *ACM Comput. Surv.* 57, 6, Article 158 (Feb. 2025), 28 pages. <https://doi.org/10.1145/3712000>
- [52] Dorothy E. Denning. 1980. Secure statistical databases with random sample queries. *ACM Transactions on Database Systems* 5, 3 (Sept. 1980), 291–315. <https://doi.org/10.1145/320613.320616>
- [53] Dorothy E. Denning, Peter J. Denning, and Mayer D. Schwartz. 1979. The tracker: A threat to statistical database security. *ACM Transactions on Database Systems* 4, 1 (March 1979), 76–96. <https://doi.org/10.1145/320064.320069>
- [54] Christos Dimitrakakis, Blaine Nelson, Aikaterini Mitrokotsa, and Benjamin I. P. Rubinstein. 2014. Robust and Private Bayesian Inference. In *Algorithmic Learning Theory*, Peter Auer, Alexander Clark, Thomas Zeugmann, and Sandra Zilles (Eds.). Springer International Publishing, Cham, 291–305.
- [55] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. 2017. Collecting Telemetry Data Privately. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/253614bbac999b38b5b60cae531c4969-Paper.pdf
- [56] Irit Dinur and Kobbi Nissim. 2003. Revealing information while preserving privacy. In *Proceedings of the Twenty-Second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (San Diego, California) (PODS '03)*. Association for Computing Machinery, New York, NY, USA, 202–210. <https://doi.org/10.1145/773153.773173>
- [57] Jinshuo Dong, Aaron Roth, and Weijie J. Su. 2022. Gaussian Differential Privacy. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 84, 1 (02 2022), 3–37. <https://doi.org/10.1111/rssb.12454> arXiv:https://academic.oup.com/jrssb/article-pdf/84/1/3/49324238/jrssb_84_1_3.pdf
- [58] Tian Dong, Bo Zhao, and Lingjuan Lyu. 2022. Privacy for Free: How does Dataset Condensation Help Privacy?. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.). PMLR, 5378–5396. <https://proceedings.mlr.press/v162/dong22c.html>
- [59] John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. 2018. Minimax Optimal Procedures for Locally Private Estimation. *J. Amer. Statist. Assoc.* 113, 521 (2018), 182–201. <https://doi.org/10.1080/01621459.2017.1389735> arXiv:<https://doi.org/10.1080/01621459.2017.1389735>
- [60] Cynthia Dwork. 2008. Differential Privacy: A Survey of Results. In *Theory and Applications of Models of Computation (TMAC) (Lecture Notes in Computer Science, Vol. 4978)*, Manindra Agrawal, Dingzhu Du, Zhenhua Duan, and Angsheng Li (Eds.). Springer Berlin Heidelberg, Berlin,

- Heidelberg, 1–19.
- [61] Cynthia Dwork. 2011. A firm foundation for private data analysis. *Commun. ACM* 54, 1 (Jan. 2011), 86–95. <https://doi.org/10.1145/1866739.1866758>
 - [62] Cynthia Dwork, Krishnamurthy Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. 2006. Our Data, Ourselves: Privacy Via Distributed Noise Generation. In *Advances in Cryptology - EUROCRYPT 2006*, Serge Vaudenay (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 486–503.
 - [63] Cynthia Dwork and Jing Lei. 2009. Differential privacy and robust statistics. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing* (Bethesda, MD, USA) (STOC '09). Association for Computing Machinery, New York, NY, USA, 371–380. <https://doi.org/10.1145/1536414.1536466>
 - [64] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography: Proceedings of the Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006 (Lecture Notes in Computer Science, Vol. 3876)*, Shai Halevi and Tal Rabin (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 265–284.
 - [65] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2017. Calibrating Noise to Sensitivity in Private Data Analysis. *Journal of Privacy and Confidentiality* 7, 3 (May 2017), 17–51. <https://doi.org/10.29012/jpc.v7i3.405>
 - [66] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. 2010. Differential privacy under continual observation. In *Proceedings of the Forty-Second ACM Symposium on Theory of Computing* (Cambridge, Massachusetts, USA) (STOC '10). Association for Computing Machinery, New York, NY, USA, 715–724. <https://doi.org/10.1145/1806689.1806787>
 - [67] Cynthia Dwork, Moni Naor, Toniann Pitassi, Guy N. Rothblum, and Sergey Yekhanin. 2010. Pan-Private Streaming Algorithms. In *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, Andrew Chi-Chih Yao (Ed.). Tsinghua University Press, 66–80. <http://conference.iis.tsinghua.edu.cn/ICS2010/content/papers/6.html>
 - [68] Cynthia Dwork and Aaron Roth. 2014. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3–4 (2014), 211–407. <https://doi.org/10.1561/04000000042>
 - [69] Cynthia Dwork and Guy N. Rothblum. 2016. Concentrated Differential Privacy. arXiv:1603.01887 [cs.DS] <https://arxiv.org/abs/1603.01887>
 - [70] Cynthia Dwork, Guy N. Rothblum, and Salil Vadhan. 2010. Boosting and Differential Privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*. IEEE, Las Vegas, NV, USA, 51–60. <https://doi.org/10.1109/FOCS.2010.12>
 - [71] Cynthia Dwork, Kunal Talwar, Abhradeep Thakurta, and Li Zhang. 2014. Analyze gauss: optimal bounds for privacy-preserving principal component analysis. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing* (New York, New York) (STOC '14). Association for Computing Machinery, New York, NY, USA, 11–20. <https://doi.org/10.1145/2591796.2591883>
 - [72] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (Scottsdale, Arizona, USA) (CCS '14). Association for Computing Machinery, New York, NY, USA, 1054–1067. <https://doi.org/10.1145/2660267.2660348>
 - [73] Beyza Ermis and Ali Taylan Cemgil. 2017. Differentially Private Dropout. arXiv:1712.01665 [stat.ML] <https://arxiv.org/abs/1712.01665>
 - [74] European Parliament and Council of the European Union. 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). , 88 pages. Available at <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R0679>.
 - [75] Alexandre Evfimievski, Johannes Gehrke, and Ramakrishnan Srikant. 2003. Limiting privacy breaches in privacy preserving data mining. In *Proceedings of the Twenty-Second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems* (San Diego, California) (PODS '03). Association for Computing Machinery, New York, NY, USA, 211–222. <https://doi.org/10.1145/773153.773174>
 - [76] W. Fan, H. Wang, P.S. Yu, and S. Ma. 2003. Is random model better? On its accuracy and efficiency. In *Third IEEE International Conference on Data Mining*. 51–58. <https://doi.org/10.1109/ICDM.2003.1250902>
 - [77] Farhad Farokhi. 2019. Development and Analysis of Deterministic Privacy-Preserving Policies Using Non- Stochastic Information Theory. *IEEE Transactions on Information Forensics and Security* 14, 10 (2019), 2567–2576. <https://doi.org/10.1109/TIFS.2019.2903660>
 - [78] Farhad Farokhi. 2020. Deconvoluting kernel density estimation and regression for locally differentially private data. *Scientific Reports* 10, 1 (2020), 21361.
 - [79] Farhad Farokhi, Nan Wu, David Smith, and Mohamed Ali Kaafar. 2021. The Cost of Privacy in Asynchronous Differentially-Private Machine Learning. *IEEE Transactions on Information Forensics and Security* 16 (2021), 2118–2129. <https://doi.org/10.1109/TIFS.2021.3050603>
 - [80] Sam Fletcher and Md Zahidul Islam. 2016. Decision Tree Classification with Differential Privacy: A Survey. arXiv:1611.01919 [cs.DB] <https://arxiv.org/abs/1611.01919>
 - [81] Sam Fletcher and Md Zahidul Islam. 2017. Differentially private random decision forests using smooth sensitivity. *Expert Systems with Applications* 78 (2017), 16–31. <https://doi.org/10.1016/j.eswa.2017.01.034>
 - [82] Sam Fletcher and Md. Zahidul Islam. 2019. Decision Tree Classification with Differential Privacy: A Survey. *ACM Comput. Surv.* 52, 4, Article 83 (Aug. 2019), 33 pages. <https://doi.org/10.1145/3337064>
 - [83] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (Denver, Colorado, USA) (CCS '15). Association for Computing Machinery, New York, NY, USA, 1322–1333. <https://doi.org/10.1145/2810103.2813677>
 - [84] Arik Friedman and Assaf Schuster. 2010. Data mining with differential privacy. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Washington, DC, USA) (KDD '10). Association for Computing Machinery, New York, NY, USA, 493–502.

- <https://doi.org/10.1145/1835804.1835868>
- [85] Andrea Gadotti, Luc Rocher, Florimond Houssiau, Ana-Maria Crețu, and Yves-Alexandre de Montjoye. 2024. Anonymization: The imperfect science of using data while preserving privacy. *Science Advances* 10, 29 (2024), eadn7053. <https://doi.org/10.1126/sciadv.adn7053> arXiv:<https://www.science.org/doi/pdf/10.1126/sciadv.adn7053>
 - [86] Georgi Ganev. 2021. DP-SGD vs PATE: Which Has Less Disparate Impact on GANs? arXiv:2111.13617 [cs.LG] <https://arxiv.org/abs/2111.13617>
 - [87] Georgi Ganev, Kai Xu, and Emiliano De Cristofaro. 2024. Graphical vs. Deep Generative Models: Measuring the Impact of Differentially Private Mechanisms and Budgets on Utility. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security* (Salt Lake City, UT, USA) (CCS '24). Association for Computing Machinery, New York, NY, USA, 1596–1610. <https://doi.org/10.1145/3658644.3690215>
 - [88] Srivatsava Ranjit Ganta, Shiva Prasad Kasiviswanathan, and Adam Smith. 2008. Composition attacks and auxiliary information in data privacy. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Las Vegas, Nevada, USA) (KDD '08). Association for Computing Machinery, New York, NY, USA, 265–273. <https://doi.org/10.1145/1401890.1401926>
 - [89] Gonzalo Munilla Garrido, Xiaoyuan Liu, Florian Matthes, and Dawn Song. 2022. Lessons Learned: Surveying the Practicality of Differential Privacy in the Industry. arXiv:2211.03898 [cs.CR] <https://arxiv.org/abs/2211.03898>
 - [90] Craig Gentry. 2009. Fully homomorphic encryption using ideal lattices. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing* (Bethesda, MD, USA) (STOC '09). Association for Computing Machinery, New York, NY, USA, 169–178. <https://doi.org/10.1145/1536414.1536440>
 - [91] Robin C. Geyer, Tassilo Klein, and Moin Nabi. 2018. Differentially Private Federated Learning: A Client Level Perspective. arXiv:1712.07557 [cs.CR] <https://arxiv.org/abs/1712.07557>
 - [92] Badih Ghazi, Noah Golowich, Ravi Kumar, Pasin Manurangsi, and Chiyuan Zhang. 2021. Deep Learning with Label Differential Privacy. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 27131–27145. https://proceedings.neurips.cc/paper_files/paper/2021/file/e3a54649aeec04cf1c13907b6c65c8aa-Paper.pdf
 - [93] Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. 2009. Universally Utility-Maximizing Privacy Mechanisms. arXiv:0811.2841 [cs.DB] <https://arxiv.org/abs/0811.2841>
 - [94] Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. 2009. Universally utility-maximizing privacy mechanisms. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing* (Bethesda, MD, USA) (STOC '09). Association for Computing Machinery, New York, NY, USA, 351–360. <https://doi.org/10.1145/1536414.1536464>
 - [95] Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. 2012. Universally Utility-maximizing Privacy Mechanisms. *SIAM J. Comput.* 41, 6 (2012), 1673–1693. <https://doi.org/10.1137/09076828X> arXiv:<https://doi.org/10.1137/09076828X>
 - [96] O. Goldreich, S. Micali, and A. Wigderson. 1987. How to play ANY mental game. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing* (New York, New York, USA) (STOC '87). Association for Computing Machinery, New York, NY, USA, 218–229. <https://doi.org/10.1145/28395.28420>
 - [97] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. 1985. The knowledge complexity of interactive proof-systems. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing* (Providence, Rhode Island, USA) (STOC '85). Association for Computing Machinery, New York, NY, USA, 291–304. <https://doi.org/10.1145/22145.22178>
 - [98] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. 1989. The Knowledge Complexity of Interactive Proof Systems. *SIAM J. Comput.* 18, 1 (1989), 186–208. <https://doi.org/10.1137/0218012> arXiv:<https://doi.org/10.1137/0218012>
 - [99] Maoguo Gong, Jialun Feng, and Yu Xie. 2020. Privacy-enhanced multi-party deep learning. *Neural Networks* 121 (2020), 484–496. <https://doi.org/10.1016/j.neunet.2019.10.001>
 - [100] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved Training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/892c3b1c6dcd52936e27cbd0ff683d6-Paper.pdf
 - [101] Mangesh Gupte and Mukund Sundararajan. 2010. Universally optimal privacy mechanisms for minimax agents. In *Proceedings of the Twenty-Ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems* (Indianapolis, Indiana, USA) (PODS '10). Association for Computing Machinery, New York, NY, USA, 135–146. <https://doi.org/10.1145/1807085.1807105>
 - [102] László Györfi and Martin Kroll. 2022. On rate optimal private regression under local differential privacy. arXiv:2206.00114 [math.ST] <https://arxiv.org/abs/2206.00114>
 - [103] Trung Ha, Tran Khanh Dang, Tran Tri Dang, Tuan Anh Truong, and Manh Tuan Nguyen. 2019. Differential Privacy in Deep Learning: An Overview. In *2019 International Conference on Advanced Computing and Applications (ACOMP)*. 97–102. <https://doi.org/10.1109/ACOMP.2019.00022>
 - [104] Andreas Haeberlen, Benjamin C. Pierce, and Arjun Narayan. 2011. Differential Privacy Under Fire. In *20th USENIX Security Symposium (USENIX Security 11) (SEC'11)*. USENIX Association, San Francisco, CA, 33. <https://www.usenix.org/conference/usenix-security-11/differential-privacy-under-fire>
 - [105] Moritz Hardt and Guy N. Rothblum. 2010. A Multiplicative Weights Mechanism for Privacy-Preserving Data Analysis. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*. 61–70. <https://doi.org/10.1109/FOCS.2010.85>
 - [106] Hanieh Hashemi, Yongqin Wang, and Murali Annavaram. 2021. DarKnight: An Accelerated Framework for Privacy and Integrity Preserving Deep Learning Using Trusted Hardware. In *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture* (Virtual Event, Greece) (MICRO '21). Association for Computing Machinery, New York, NY, USA, 212–224. <https://doi.org/10.1145/3466752.3480112>

- [107] Xi He, Ashwin Machanavajjhala, and Bolin Ding. 2014. Blowfish privacy: tuning privacy-utility trade-offs using policies. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data* (Snowbird, Utah, USA) (SIGMOD '14). Association for Computing Machinery, New York, NY, USA, 1447–1458. <https://doi.org/10.1145/2588555.2588581>
- [108] Shen-Shyang Ho and Shuhua Ruan. 2011. Differential privacy for location pattern mining. In *Proceedings of the 4th ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS* (Chicago, Illinois) (SPRINGL '11). Association for Computing Machinery, New York, NY, USA, 17–24. <https://doi.org/10.1145/2071880.2071884>
- [109] Hongyu Hu, Qilong Han, Zhiqiang Ma, Yukun Yan, Zuobin Xiong, Linyu Jiang, and Yuemin Zhang. 2024. PV-PATE: An Improved PATE for Deep Learning with Differential Privacy in Trusted Industrial Data Matrix. In *Web and Big Data*, Xiangyu Song, Ruyi Feng, Yunliang Chen, Jianxin Li, and Geyong Min (Eds.). Springer Nature Singapore, Singapore, 477–491.
- [110] Xixi Huang, Jian Guan, Bin Zhang, Shuhan Qi, Xuan Wang, and Qing Liao. 2019. Differentially Private Convolutional Neural Networks with Adaptive Gradient Descent. In *2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC)*. 642–648. <https://doi.org/10.1109/DSC.2019.00105>
- [111] Yuxian Huang, Geng Yang, Yahong Xu, and Hao Zhou. 2021. Differential Privacy Principal Component Analysis for Support Vector Machines. *Security and Communication Networks* 2021, 1 (2021), 5542283. <https://doi.org/10.1155/2021/5542283> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1155/2021/5542283>
- [112] Geetha Jagannathan, Krishnan Pillaipakkamnatt, and Rebecca N. Wright. 2009. A Practical Differentially Private Random Decision Tree Classifier. In *2009 IEEE International Conference on Data Mining Workshops*. 114–121. <https://doi.org/10.1109/ICDMW.2009.93>
- [113] Geetha Jagannathan, Krishnan Pillaipakkamnatt, and Rebecca N. Wright. 2012. A Practical Differentially Private Random Decision Tree Classifier. *Transactions on Data Privacy* 5, 1 (2012), 273–295. <https://www.tdp.cat/issues11/tdp.a082a11.pdf>
- [114] Aayush Jain, Huijia Lin, and Amit Sahai. 2021. Indistinguishability obfuscation from well-founded assumptions. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing* (Virtual, Italy) (STOC 2021). Association for Computing Machinery, New York, NY, USA, 60–73. <https://doi.org/10.1145/3406325.3451093>
- [115] Prateek Jain and Abhradeep Thakurta. 2013. Differentially Private Learning with Kernels. In *Proceedings of the 30th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 28)*, Sanjoy Dasgupta and David McAllester (Eds.). PMLR, Atlanta, Georgia, USA, 118–126. <https://proceedings.mlr.press/v28/jain13.html>
- [116] Bargav Jayaraman and David Evans. 2019. Evaluating Differentially Private Machine Learning in Practice. In *28th USENIX Security Symposium (USENIX Security 19)*. USENIX Association, Santa Clara, CA, 1895–1912. <https://www.usenix.org/conference/usenixsecurity19/presentation/jayaraman>
- [117] Bargav Jayaraman, Lingxiao Wang, Katherine Knipmeyer, Quanquan Gu, and David Evans. 2020. Revisiting Membership Inference Under Realistic Assumptions. arXiv:2005.10881 [cs.CR] <https://arxiv.org/abs/2005.10881>
- [118] Zhanglong Ji, Zachary C. Lipton, and Charles Elkan. 2014. Differential Privacy and Machine Learning: a Survey and Review. arXiv:1412.7584 [cs.LG] <https://arxiv.org/abs/1412.7584>
- [119] Noah Johnson, Joseph P. Near, and Dawn Song. 2018. Towards practical differential privacy for SQL queries. *Proc. VLDB Endow.* 11, 5 (Jan. 2018), 526–539. <https://doi.org/10.1145/3187009.3177733>
- [120] James Jordon, Jinsung Yoon, and Mihaela van der Schaar. 2019. PATE-GAN: Generating Synthetic Data with Differential Privacy Guarantees. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6–9, 2019*. OpenReview.net. <https://openreview.net/forum?id=S1zk9iRqF7>
- [121] J Kahane. 1960. Propriétés locales des fonctions à séries de Fourier aléatoires. *Studia Mathematica* 19, 1 (1960), 1–25.
- [122] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. 2015. The Composition Theorem for Differential Privacy. In *Proceedings of the 32nd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 37)*, Francis Bach and David Blei (Eds.). PMLR, Lille, France, 1376–1385. <https://proceedings.mlr.press/v37/kairouz15.html>
- [123] Gautam Kamath, Or Sheffet, Vikrant Singhal, and Jonathan Ullman. 2019. Differentially Private Algorithms for Learning Mixtures of Separated Gaussians. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2019/file/68d30a9594728bc39aa24be94b319d21-Paper.pdf
- [124] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2011. What Can We Learn Privately? *SIAM J. Comput.* 40, 3 (2011), 793–826. <https://doi.org/10.1137/090756090> arXiv:<https://doi.org/10.1137/090756090>
- [125] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. 2008. What Can We Learn Privately?. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25–28, 2008, Philadelphia, PA, USA*. IEEE Computer Society, 531–540. <https://doi.org/10.1109/FOCS.2008.27>
- [126] Shiva Prasad Kasiviswanathan and Adam Smith. 2008. A note on differential privacy: Defining resistance to arbitrary side information. arXiv:0803.39461 [cs.CL] <https://eprint.iacr.org/2008/144>
- [127] Shiva P. Kasiviswanathan and Adam Smith. 2014. On the 'Semantics' of Differential Privacy: A Bayesian Formulation. *Journal of Privacy and Confidentiality* 6, 1 (Jun. 2014). <https://doi.org/10.29012/jpc.v6i1.634>
- [128] J. Kiefer and J. Wolfowitz. 1952. Stochastic Estimation of the Maximum of a Regression Function. *The Annals of Mathematical Statistics* 23, 3 (1952), 462 – 466. <https://doi.org/10.1214/aoms/1177729392>
- [129] Daniel Kifer and Ashwin Machanavajjhala. 2011. No free lunch in data privacy. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data* (Athens, Greece) (SIGMOD '11). Association for Computing Machinery, New York, NY, USA, 193–204. <https://doi.org/10.1145/1979369.1979411>

- 1145/1989323.1989345
- [130] Daniel Kifer and Ashwin Machanavajjhala. 2012. A rigorous and customizable framework for privacy. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems* (Scottsdale, Arizona, USA) (PODS '12). Association for Computing Machinery, New York, NY, USA, 77–88. <https://doi.org/10.1145/2213556.2213571>
 - [131] Muah Kim, Onur Günlü, and Rafael F. Schaefer. 2021. Federated Learning with Local Differential Privacy: Trade-Offs Between Privacy, Utility, and Communication. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2650–2654. <https://doi.org/10.1109/ICASSP39728.2021.9413764>
 - [132] Antti Koskela, Marlon Tobaben, and Antti Honkela. 2023. Individual Privacy Accounting with Gaussian Differential Privacy. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net. https://openreview.net/forum?id=JmC_Tld3v-f
 - [133] Ios Kotsogiannis, Stelios Doudalis, Sam Haney, Ashwin Machanavajjhala, and Sharad Mehrotra. 2020. One-sided Differential Privacy. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. 493–504. <https://doi.org/10.1109/ICDE48307.2020.00049>
 - [134] Satyapriya Krishna, Rahul Gupta, and Christophe Dupuy. 2021. ADEPT: Auto-encoder based Differentially Private Text Transformation. arXiv:2102.01502 [cs.CR] <https://arxiv.org/abs/2102.01502>
 - [135] S. Kullback and R. A. Leibler. 1951. On Information and Sufficiency. *The Annals of Mathematical Statistics* 22, 1 (1951), 79–86. <http://www.jstor.org/stable/2236703>
 - [136] Aditya Kumar, Robert Birke, Zilong Zhao, and Lydia Chen. 2021. DTGAN: Differential Private Training for Tabular GANs. arXiv:2107.02521 [cs.LG] <https://arxiv.org/abs/2107.02521>
 - [137] Jing Lei. 2011. Differentially Private M-Estimators. In *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger (Eds.), Vol. 24. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2011/file/f718499c1c8cef6730f9fd03c8125cab-Paper.pdf
 - [138] Ninghui Li, Zhikun Zhang, and Tianhao Wang. 2021. DPSyn: Experiences in the NIST Differential Privacy Data Synthesis Challenges. *Journal of Privacy and Confidentiality* 11 (2021), 2.
 - [139] Qinbin Li, Zeyi Wen, and Bingsheng He. 2020. Practical Federated Gradient Boosting Decision Trees. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 04 (Apr. 2020), 4642–4649. <https://doi.org/10.1609/aaai.v34i04.5895>
 - [140] Qinbin Li, Zhaomin Wu, Zeyi Wen, and Bingsheng He. 2020. Privacy-Preserving Gradient Boosting Decision Trees. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 01 (Apr. 2020), 784–791. <https://doi.org/10.1609/aaai.v34i01.5422>
 - [141] Tong Li, Jin Li, Zheli Liu, Ping Li, and Chunfu Jia. 2018. Differentially private Naive Bayes learning over multiple data sources. *Information Sciences* 444 (2018), 89–104. <https://doi.org/10.1016/j.ins.2018.02.056>
 - [142] Xiaochen Li, Yuke Hu, Weiran Liu, Hanwen Feng, Li Peng, Yuan Hong, Kui Ren, and Zhan Qin. 2022. OpBoost: A Vertical Federated Tree Boosting Framework Based on Order-Preserving Desensitization. arXiv:2210.01318 [cs.LG] <https://arxiv.org/abs/2210.01318>
 - [143] Xiaochen Li, Yuke Hu, Weiran Liu, Hanwen Feng, Li Peng, Yuan Hong, Kui Ren, and Zhan Qin. 2022. OpBoost: a vertical federated tree boosting framework based on order-preserving desensitization. *Proc. VLDB Endow.* 16, 2 (Oct. 2022), 202–215. <https://doi.org/10.14778/3565816.3565823>
 - [144] Xinyi Li, Yinchuan Li, Hongyang Yang, Liuqing Yang, and Xiao-Yang Liu. 2019. DP-LSTM: Differential Privacy-inspired LSTM for Stock Prediction Using Financial News. arXiv:1912.10806 [q-fin.ST] <https://arxiv.org/abs/1912.10806>
 - [145] Changchang Liu, Supriyo Chakraborty, and Prateek Mittal. 2016. Dependence Makes You Vulnerable: Differential Privacy Under Dependent Tuples. In *23rd Annual Network and Distributed System Security Symposium, NDSS 2016, San Diego, California, USA, February 21-24, 2016*. The Internet Society. <http://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2017/09/dependence-makes-you-vulnerable-differential-privacy-under-dependent-tuples.pdf>
 - [146] Sheng Liu, Zihan Wang, Yuxiao Chen, and Qi Lei. 2025. Data Reconstruction Attacks and Defenses: A Systematic Evaluation. arXiv:2402.09478 [cs.CR] <https://arxiv.org/abs/2402.09478>
 - [147] Eugenio Lomurno and Matteo Matteucci. 2023. On the Utility and Protection of Optimization with Differential Privacy and Classic Regularization Techniques. In *Machine Learning, Optimization, and Data Science (LOD 2022 Revised Selected Papers) (Lecture Notes in Computer Science, Vol. 13810)*, Giuseppe Nicosia, Varun Ojha, Emanuele La Malfa, Gabriele La Malfa, Panos Pardalos, Giuseppe Di Fatta, Giovanni Giuffrida, and Renato Umeton (Eds.). Springer Nature Switzerland, Cham, 223–238.
 - [148] Yunhui Long, Boxin Wang, Zhuolin Yang, Bhavya Kailkhura, Aston Zhang, Carl Gunter, and Bo Li. 2021. G-PATE: Scalable Differentially Private Data Generator via Private Aggregation of Teacher Discriminators. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 2965–2977. https://proceedings.neurips.cc/paper_files/paper/2021/file/171ae1bbb81475eb96287dd78565b38b-Paper.pdf
 - [149] Andrew Lowy, Zhuohang Li, Jing Liu, Toshiaki Koike-Akino, Kieran Parsons, and Ye Wang. 2024. Why Does Differential Privacy with Large Epsilon Defend Against Practical Membership Inference Attacks? arXiv:2402.09540 [cs.CR] <https://arxiv.org/abs/2402.09540>
 - [150] Fred Lu, Joseph Munoz, Maya Fuchs, Tyler LeBlond, Elliott Zaresky-Williams, Edward Raff, Francis Ferraro, and Brian Testa. 2022. A General Framework for Auditing Differentially Private Machine Learning. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 4165–4176. https://proceedings.neurips.cc/paper_files/paper/2022/file/1add3bbdbc20c403a383482a665eb5a4-Paper-Conference.pdf

- [151] Min Lyu, Dong Su, and Ninghui Li. 2016. Understanding the Sparse Vector Technique for Differential Privacy. arXiv:1603.01699 [cs.CR] <https://arxiv.org/abs/1603.01699>
- [152] Xin Lyu. 2022. Composition Theorems for Interactive Differential Privacy. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 9700–9712. https://proceedings.neurips.cc/paper_files/paper/2022/file/3f52b555967a95ee850fcecbbd29ee52d-Paper-Conference.pdf
- [153] Yuheng Ma, Han Zhang, Yuchao Cai, and Hanfang Yang. 2023. Decision Tree for Locally Private Estimation with Public Data. In *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36. Curran Associates, Inc., 43676–43705. https://proceedings.neurips.cc/paper_files/paper/2023/file/88237ac4e9941b1be5c6d3c1ad408184-Paper-Conference.pdf
- [154] Ashwin Machanavajjhala, Daniel Kifer, John Abowd, Johannes Gehrke, and Lars Vilhuber. 2008. Privacy: Theory meets Practice on the Map. In *2008 IEEE 24th International Conference on Data Engineering*, 277–286. <https://doi.org/10.1109/ICDE.2008.4497436>
- [155] Pathum Chamikara Mahawaga Arachchige, Peter Bertok, Ibrahim Khalil, Dongxi Liu, Seyit Camtepe, and Mohammed Atiquzzaman. 2020. Local Differential Privacy for Deep Learning. *IEEE Internet of Things Journal* 7, 7 (2020), 5827–5842. <https://doi.org/10.1109/JIOT.2019.2952146>
- [156] Mourad El Maouchi, Oğuzhan Ersoy, and Zekeriya Erkin. 2019. DECOUPLES: a decentralized, unlinkable and privacy-preserving traceability system for the supply chain. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing* (Limassol, Cyprus) (SAC '19). Association for Computing Machinery, New York, NY, USA, 364–373. <https://doi.org/10.1145/3297280.3297318>
- [157] Ryan McKenna, Gerome Miklau, and Daniel Sheldon. 2021. Winning the NIST Contest: A Scalable and General Approach to Differentially Private Synthetic Data. *Journal of Privacy and Confidentiality* 11 (2021), 3.
- [158] Ryan McKenna, Daniel Sheldon, and Gerome Miklau. 2019. Graphical-model based estimation and inference for differential privacy. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 4435–4444. <https://proceedings.mlr.press/v97/mckenna19a.html>
- [159] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 54)*, Aarti Singh and Jerry Zhu (Eds.). PMLR, 1273–1282. <https://proceedings.mlr.press/v54/mcmahan17a.html>
- [160] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2016. Federated Learning of Deep Networks using Model Averaging. arXiv:1602.05629 [cs.LG] <https://arxiv.org/abs/1602.05629>
- [161] Frank McSherry and Kunal Talwar. 2007. Mechanism Design via Differential Privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*. IEEE, Providence, RI, USA, 94–103. <https://doi.org/10.1109/FOCS.2007.66>
- [162] Frank D. McSherry. 2009. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data* (Providence, Rhode Island, USA) (SIGMOD '09). Association for Computing Machinery, New York, NY, USA, 19–30. <https://doi.org/10.1145/1559845.1559850>
- [163] Luise Mehner, Saskia Nuñez von Voigt, and Florian Tschorsch. 2021. Towards Explaining Epsilon: A Worst-Case Study of Differential Privacy Risks. In *2021 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 328–331. <https://doi.org/10.1109/EuroSPW54576.2021.00041>
- [164] Sebastian Meiser. 2018. Approximate and Probabilistic Differential Privacy Definitions. Cryptology ePrint Archive, Paper 2018/277. <https://eprint.iacr.org/2018/277>
- [165] Fatemehsadat Mirehghallah, Mohammadkazem Taram, Praneeth Vepakomma, Abhishek Singh, Ramesh Raskar, and Hadi Esmaeilzadeh. 2020. Privacy in Deep Learning: A Survey. arXiv:2004.12254 [cs.LG] <https://arxiv.org/abs/2004.12254>
- [166] Ilya Mironov. 2012. On significance of the least significant bits for differential privacy. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security* (Raleigh, North Carolina, USA) (CCS '12). Association for Computing Machinery, New York, NY, USA, 650–661. <https://doi.org/10.1145/2382196.2382264>
- [167] Ilya Mironov. 2017. Rényi Differential Privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. Santa Barbara, CA, USA, 263–275. <https://doi.org/10.1109/CSF.2017.11>
- [168] Ilya Mironov, Omkant Pandey, Omer Reingold, and Salil Vadhan. 2009. Computational Differential Privacy. In *Advances in Cryptology - CRYPTO 2009*, Shai Halevi (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 126–142.
- [169] Ilya Mironov, Kunal Talwar, and Li Zhang. 2019. Rényi Differential Privacy of the Sampled Gaussian Mechanism. arXiv:1908.10530 [cs.LG] <https://arxiv.org/abs/1908.10530>
- [170] Payman Mohassel and Yupeng Zhang. 2017. SecureML: A System for Scalable Privacy-Preserving Machine Learning. In *2017 IEEE Symposium on Security and Privacy (SP)*, 19–38. <https://doi.org/10.1109/SP.2017.12>
- [171] Jack Murtagh and Salil Vadhan. 2016. The Complexity of Computing the Optimal Composition of Differential Privacy. In *Theory of Cryptography*, Eyal Kushilevitz and Tal Malkin (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 157–175.
- [172] Arvind Narayanan and Vitaly Shmatikov. 2007. How To Break Anonymity of the Netflix Prize Dataset. arXiv:cs/0610105 [cs.CR] <https://arxiv.org/abs/cs/0610105>
- [173] Arvind Narayanan and Vitaly Shmatikov. 2008. Robust De-anonymization of Large Sparse Datasets. In *2008 IEEE Symposium on Security and Privacy (SP 2008)*. IEEE, Oakland, CA, USA, 111–125. <https://doi.org/10.1109/SP.2008.33>
- [174] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2007. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing* (San Diego, California, USA) (STOC '07). Association for Computing Machinery, New York, NY, USA, 75–84. <https://doi.org/10.1145/1250790.1250803>

- [175] Se-Ra Oh, Young-Duk Seo, Euijong Lee, and Young-Gab Kim. 2021. A Comprehensive Survey on Security and Privacy for Electronic Health Data. *International Journal of Environmental Research and Public Health* 18, 18 (2021). <https://doi.org/10.3390/ijerph18189668>
- [176] Ahmed El Ouadrhiri and Ahmed Abdelhadi. 2022. Differential Privacy for Deep and Federated Learning: A Survey. *IEEE Access* 10 (2022), 22359–22380. <https://doi.org/10.1109/ACCESS.2022.3151670>
- [177] Ke Pan, Yew-Soon Ong, Maoguo Gong, Hui Li, A.K. Qin, and Yuan Gao. 2024. Differential privacy in deep learning: A literature survey. *Neurocomputing* 589 (2024), 127663. <https://doi.org/10.1016/j.neucom.2024.127663>
- [178] Nicolas Papernot, Martin Abadi, Úlfar Erlingsson, Ian J. Goodfellow, and Kunal Talwar. 2017. Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data. In *5th International Conference on Learning Representations, ICLR 2017, April 24–26, 2017, Conference Track Proceedings*. Toulon, France, 16 pages. <https://openreview.net/forum?id=HkwoSDPgg>
- [179] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. 2018. Scalable Private Learning with PATE. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 – May 3, 2018, Conference Track Proceedings*. OpenReview.net. <https://openreview.net/forum?id=rkZB1XbRZ>
- [180] Manas A. Pathak and Bhiksha Raj. 2012. Large Margin Gaussian Mixture Models with Differential Privacy. *IEEE Transactions on Dependable and Secure Computing* 9, 4 (2012), 463–469. <https://doi.org/10.1109/TDSC.2012.27>
- [181] Balázs Pejó and Damien Desfontaines. 2022. *Guide to differential privacy modifications: a taxonomy of variants and extensions*. Springer Nature.
- [182] NhatHai Phan, Yue Wang, Xintao Wu, and Dejing Dou. 2016. Differential Privacy Preservation for Deep Auto-Encoders: an Application of Human Behavior Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence* 30, 1 (Feb. 2016). <https://doi.org/10.1609/aaai.v30i1.10165>
- [183] Natalia Ponomareva, Hussein Hazimeh, Alex Kurakin, Zheng Xu, Carson Denison, H Brendan McMahan, Sergei Vassilvitskii, Steve Chien, and Abhradeep Guha Thakurta. 2023. How to DP-fy ML: A Practical Guide to Machine Learning with Differential Privacy. *Journal of Artificial Intelligence Research* 77 (2023), 1113–1201.
- [184] Natalia Ponomareva, Sergei Vassilvitskii, Zheng Xu, Brendan McMahan, Alexey Kurakin, and Chiyaun Zhang. 2023. How to DP-fy ML: A Practical Tutorial to Machine Learning with Differential Privacy. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (Long Beach, CA, USA) (KDD '23)*. Association for Computing Machinery, New York, NY, USA, 5823–5824. <https://doi.org/10.1145/3580305.3599561>
- [185] Santu Rana, Sunil Kumar Gupta, and Ssvetha Venkatesh. 2015. Differentially Private Random Forest with High Utility. In *2015 IEEE International Conference on Data Mining*. 955–960. <https://doi.org/10.1109/ICDM.2015.76>
- [186] Alfréd Rényi. 1961. On measures of entropy and information. In *Proceedings of the fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, Vol. 4. University of California Press, 547–562.
- [187] Omar Rivasplata. 2012. Subgaussian random variables: An expository note.
- [188] Ronald L Rivest, Len Adleman, and Michael L Dertouzos. 1979. On Data Banks and Privacy Homomorphisms. *Foundations of Secure Computations* (1979), 171.
- [189] Herbert Robbins and Sutton Monro. 1951. A Stochastic Approximation Method. *The Annals of Mathematical Statistics* 22, 3 (1951), 400 – 407. <https://doi.org/10.1214/aoms/1177729586>
- [190] Frank Rosenblatt. 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review* 65, 6 (1958), 386.
- [191] Benjamin I. P. Rubinstein, Peter L. Bartlett, Ling Huang, and Nina Taft. 2009. Learning in a Large Function Space: Privacy-Preserving Mechanisms for SVM Learning. arXiv:0911.5708 [cs.LG] <https://arxiv.org/abs/0911.5708>
- [192] Pierangela Samarati and Latanya Sweeney. 1998. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression.
- [193] Anand D. Sarwate and Kamalika Chaudhuri. 2013. Signal Processing and Machine Learning with Differential Privacy: Algorithms and Challenges for Continuous Data. *IEEE Signal Processing Magazine* 30, 5 (2013), 86–94. <https://doi.org/10.1109/MSP.2013.2259911>
- [194] Reza Shokri and Vitaly Shmatikov. 2015. Privacy-Preserving Deep Learning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (Denver, Colorado, USA) (CCS '15)*. Association for Computing Machinery, New York, NY, USA, 1310–1321. <https://doi.org/10.1145/2810103.2813687>
- [195] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership Inference Attacks against Machine Learning Models. arXiv:1610.05820 [cs.CR] <https://arxiv.org/abs/1610.05820>
- [196] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership Inference Attacks Against Machine Learning Models. In *2017 IEEE Symposium on Security and Privacy (SP)*. 3–18. <https://doi.org/10.1109/SP.2017.41>
- [197] Adam Smith. 2008. Efficient, Differentially Private Point Estimators. arXiv:0809.4794 [cs.CR] <https://arxiv.org/abs/0809.4794>
- [198] Adam Smith. 2011. Privacy-preserving statistical estimation with optimal convergence rates. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing (San Jose, California, USA) (STOC '11)*. Association for Computing Machinery, New York, NY, USA, 813–822. <https://doi.org/10.1145/1993636.1993743>
- [199] Shuang Song, Yizhen Wang, and Kamalika Chaudhuri. 2017. Pufferfish Privacy Mechanisms for Correlated Data. In *Proceedings of the 2017 ACM International Conference on Management of Data (Chicago, Illinois, USA) (SIGMOD '17)*. Association for Computing Machinery, New York, NY, USA, 1291–1306. <https://doi.org/10.1145/3035918.3064025>
- [200] Jordi Soria-Comas, Josep Domingo-Ferrer, David Sánchez, and David Megías. 2017. Individual Differential Privacy: A Utility-Preserving Formulation of Differential Privacy Guarantees. *IEEE Transactions on Information Forensics and Security* 12, 6 (2017), 1418–1429. <https://doi.org/10.1109/TIFS>

- 2017.2663337
- [201] Apple Differential Privacy Team. 2017. Learning with Privacy at Scale. Retrieved from <https://machinelearning.apple.com/research/learning-with-privacy-at-scale>.
 - [202] Zhihua Tian, Rui Zhang, Xiaoyang Hou, Lingjuan Lyu, Tianyi Zhang, Jian Liu, and Kui Ren. 2024. FederBoost: Private Federated Learning for GBDT. *IEEE Transactions on Dependable and Secure Computing* 21, 3 (2024), 1274–1285. <https://doi.org/10.1109/TDSC.2023.3276365>
 - [203] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. 2016. Stealing machine learning models via prediction {APIs}. In *25th USENIX security symposium (USENIX Security 16)*. 601–618.
 - [204] Florian Tramèr, Gautam Kamath, and Nicholas Carlini. 2024. Position: Considerations for Differentially Private Learning with Large-Scale Public Pretraining. arXiv:2212.06470 [cs.LG] <https://arxiv.org/abs/2212.06470>
 - [205] Cuong Tran, My Dinh, and Ferdinando Fioretto. 2021. Differentially Private Empirical Risk Minimization under the Fairness Lens. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 27555–27565. https://proceedings.neurips.cc/paper_files/paper/2021/file/e7e8f8e5982b3298c8addedf6811d500-Paper.pdf
 - [206] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. 2019. A Hybrid Approach to Privacy-Preserving Federated Learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security (London, United Kingdom) (AISeC'19)*. Association for Computing Machinery, New York, NY, USA, 1–11. <https://doi.org/10.1145/3338501.3357370>
 - [207] Salil Vadhan and Tianhao Wang. 2021. Concurrent Composition of Differential Privacy. In *Theory of Cryptography*, Kobbi Nissim and Brent Waters (Eds.), Springer International Publishing, Cham, 582–604.
 - [208] Salil Vadhan and Wanrong Zhang. 2023. Concurrent Composition Theorems for Differential Privacy. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (Orlando, FL, USA) (STOC 2023)*. Association for Computing Machinery, New York, NY, USA, 507–519. <https://doi.org/10.1145/3564246.3585241>
 - [209] Jaideep Vaidya, Chris Clifton, Murat Kantarcioglu, and A. Scott Patterson. 2008. Privacy-preserving decision trees over vertically partitioned data. *ACM Transactions on Knowledge Discovery from Data* 2, 3, Article 14 (Oct. 2008), 27 pages. <https://doi.org/10.1145/1409620.1409624>
 - [210] Jaideep Vaidya, Basit Shafiq, Anirban Basu, and Yuan Hong. 2013. Differentially Private Naive Bayes Classification. In *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, Vol. 1. 571–576. <https://doi.org/10.1109/WI-IAT.2013.80>
 - [211] Jaideep Vaidya, Basit Shafiq, Wei Fan, Danish Mehmood, and David Lorenzi. 2014. A Random Decision Tree Framework for Privacy-Preserving Data Mining. *IEEE Transactions on Dependable and Secure Computing* 11, 5 (2014), 399–411. <https://doi.org/10.1109/TDSC.2013.43>
 - [212] Tim van Erven and Peter Harremoos. 2014. Rényi Divergence and Kullback-Leibler Divergence. *IEEE Transactions on Information Theory* 60, 7 (2014), 3797–3820. <https://doi.org/10.1109/TIT.2014.2320500>
 - [213] Boxin Wang, Fan Wu, Yunhui Long, Luka Rimanic, Ce Zhang, and Bo Li. 2021. DataLens: Scalable Privacy Preserving Training via Gradient Compression and Aggregation. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (Virtual Event, Republic of Korea) (CCS '21)*. Association for Computing Machinery, New York, NY, USA, 2146–2168. <https://doi.org/10.1145/3460120.3484579>
 - [214] Di Wang, Minwei Ye, and Jinhui Xu. 2017. Differentially Private Empirical Risk Minimization Revisited: Faster and More General. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/f337d999d9ad116a7b4f3d409fcc6480-Paper.pdf
 - [215] Hua Wang, Sheng Gao, Huan Yu Zhang, Milan Shen, and Weijie J. Su. 2022. Analytical Composition of Differential Privacy via the Edgeworth Accountant. arXiv:2206.04236 [cs.CR] <https://arxiv.org/abs/2206.04236>
 - [216] Jiaqi Wang. 2024. In *Differential Privacy, There Is Truth: Evaluating Pate With Monte Carlo Adversaries*. Master's thesis. University of Toronto. <https://www.proquest.com/dissertations-theses/differential-privacy-there-is-truth-evaluating/docview/3128067282/se-2>
 - [217] Jiaqi Wang, Roei Schuster, I Shumailov, David Lie, and Nicolas Papernot. 2022. In Differential Privacy, There is Truth: on Vote-Histogram Leakage in Ensemble Private Learning. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 29026–29037. https://proceedings.neurips.cc/paper_files/paper/2022/file/ba8d1b46292c5e82cbfb3b3dc3b968af-Paper-Conference.pdf
 - [218] Teng Wang, Yao Zhang, Jiangguo Liang, Shuai Wang, and Shuanggen Liu. 2024. Differentially Private Support Vector Machines with Knowledge Aggregation. *Computers, Materials & Continua* 78, 3 (2024), 3891–3907. <https://doi.org/10.32604/cmc.2024.048115>
 - [219] Yu-Xiang Wang, Borja Balle, and Shiva Kasiviswanathan. 2018. Subsampled Rényi Differential Privacy and Analytical Moments Accountant. arXiv:1808.00087 [cs.LG] <https://arxiv.org/abs/1808.00087>
 - [220] Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. 2019. Subsampled Renyi Differential Privacy and Analytical Moments Accountant. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 89)*, Kamalika Chaudhuri and Masashi Sugiyama (Eds.). PMLR, 1226–1235. <https://proceedings.mlr.press/v89/wang19b.html>
 - [221] Yu-Xiang Wang, Stephen Fienberg, and Alex Smola. 2015. Privacy for Free: Posterior Sampling and Stochastic Gradient Monte Carlo. In *Proceedings of the 32nd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 37)*, Francis Bach and David Blei (Eds.). PMLR, Lille, France, 2493–2502. <https://proceedings.mlr.press/v37/wangg15.html>
 - [222] Stanley L. Warner. 1965. Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias. *J. Amer. Statist. Assoc.* 60, 309 (1965), 63–69. <https://doi.org/10.1080/01621459.1965.10480775> arXiv:<https://www.tandfonline.com/doi/pdf/10.1080/01621459.1965.10480775> PMID: 12261830.

- [223] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Hang Su, Bo Zhang, and H. Vincent Poor. 2022. User-Level Privacy-Preserving Federated Learning: Analysis and Performance Optimization. *IEEE Transactions on Mobile Computing* 21, 9 (2022), 3388–3401. <https://doi.org/10.1109/TMC.2021.3056991>
- [224] Xi Wu, Matthew Fredrikson, Wentao Wu, Somesh Jha, and Jeffrey F. Naughton. 2015. Revisiting Differentially Private Regression: Lessons From Learning Theory and their Consequences. arXiv:1512.06388 [cs.CR] <https://arxiv.org/abs/1512.06388>
- [225] Chang Xia, Jingyu Hua, Wei Tong, and Sheng Zhong. 2020. Distributed K-Means clustering guaranteeing local differential privacy. *Computers & Security* 90 (2020), 101699. <https://doi.org/10.1016/j.cose.2019.101699>
- [226] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. 2018. Differentially Private Generative Adversarial Network. arXiv:1802.06739 [cs.LG] <https://arxiv.org/abs/1802.06739>
- [227] Chugui Xu, Ju Ren, Deyu Zhang, Yaoxue Zhang, Zhan Qin, and Kui Ren. 2019. GANobfuscator: Mitigating Information Leakage Under GAN via Differential Privacy. *IEEE Transactions on Information Forensics and Security* 14, 9 (2019), 2358–2371. <https://doi.org/10.1109/TIFS.2019.2897874>
- [228] Zheng Xu, Yanxiang Zhang, Galen Andrew, Christopher A. Choquette-Choo, Peter Kairouz, H. Brendan McMahan, Jesse Rosenstock, and Yuanbo Zhang. 2023. Federated Learning of Gboard Language Models with Differential Privacy. arXiv:2305.18465 [cs.LG] <https://arxiv.org/abs/2305.18465>
- [229] Mingfu Xue, Chengxiang Yuan, Heyi Wu, Yushu Zhang, and Weiqiang Liu. 2020. Machine Learning Security: Threats, Countermeasures, and Evaluations. *IEEE Access* 8 (2020), 74720–74742. <https://doi.org/10.1109/ACCESS.2020.2987435>
- [230] Chao-Han Huck Yang, Sabato Marco Siniscalchi, and Chin-Hui Lee. 2021. PATE-AAE: Incorporating Adversarial Autoencoder into Private Aggregation of Teacher Ensembles for Spoken Command Classification. In *Interspeech 2021*. 881–885. <https://doi.org/10.21437/Interspeech.2021-640>
- [231] Andrew C. Yao. 1982. Protocols for secure computations. In *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*. 160–164. <https://doi.org/10.1109/SFCS.1982.38>
- [232] Yutong Ye, Tianhao Wang, Min Zhang, and Dengguo Feng. 2025. Revisiting EM-based Estimation for Locally Differentially Private Protocols. In *32nd Annual Network and Distributed System Security Symposium, NDSS 2025, San Diego, California, USA, February 24-28, 2025*. The Internet Society. <https://www.ndss-symposium.org/wp-content/uploads/2025-521-paper.pdf>
- [233] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*. 268–282. <https://doi.org/10.1109/CSF.2018.00027>
- [234] Emre Yilmaz, Mohammad Al-Rubaie, and J. Morris Chang. 2019. Locally Differentially Private Naive Bayes Classification. arXiv:1905.01039 [cs.LG] <https://arxiv.org/abs/1905.01039>
- [235] Lihua Yin, Jiyuan Feng, Hao Xun, Zhe Sun, and Xiaochun Cheng. 2021. A Privacy-Preserving Federated Learning for Multiparty Data Sharing in Social IoTs. *IEEE Transactions on Network Science and Engineering* 8, 3 (2021), 2706–2718. <https://doi.org/10.1109/TNSE.2021.3074185>
- [236] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A. Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, Sergey Yekhanin, and Huishuai Zhang. 2022. Differentially Private Fine-tuning of Language Models. arXiv:2110.06500 [cs.LG] <https://arxiv.org/abs/2110.06500>
- [237] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A. Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, Sergey Yekhanin, and Huishuai Zhang. 2022. Differentially Private Fine-tuning of Language Models. arXiv:2110.06500 [cs.LG] <https://arxiv.org/abs/2110.06500>
- [238] Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. 2017. PrivBayes: Private Data Release via Bayesian Networks. *ACM Transactions on Database Systems* 42, 4, Article 25 (Oct. 2017), 41 pages. <https://doi.org/10.1145/3134428>
- [239] Jun Zhang, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, and Marianne Winslett. 2012. Functional mechanism: regression analysis under differential privacy. *Proceedings of the International Conference on Very Large Data Bases* 5, 11 (July 2012), 1364–1375. <https://doi.org/10.14778/2350229.2350253>
- [240] Keyu Zhu, Ferdinando Fioretto, and Pascal Van Hentenryck. 2022. Post-processing of Differentially Private Data: A Fairness Perspective. arXiv:2201.09425 [cs.CR] <https://arxiv.org/abs/2201.09425>

Received June 2025