Quantifying Classifier Utility under Local Differential Privacy

Ye Zheng
Rochester Institute of Technology
Rochester, NY, USA

Yidan Hu
Rochester Institute of Technology
Rochester, NY, USA

Abstract

Local differential privacy (LDP) offers rigorous, quantifiable privacy guarantees for personal data by introducing perturbations at the data source. Understanding how these perturbations affect classifier utility is crucial for both designers and users. However, a general theoretical framework for quantifying this impact is lacking and also challenging, especially for complex or black-box classifiers.

This paper presents a unified framework for theoretically quantifying classifier utility under LDP mechanisms. The key insight is that LDP perturbations are concentrated around the original data with a specific probability, allowing utility analysis to be reframed as robustness analysis within this concentrated region. Our framework thus connects the concentration properties of LDP mechanisms with the robustness of classifiers, treating LDP mechanisms as general distributional functions and classifiers as black boxes. This generality enables applicability to any LDP mechanism and classifier. A direct application of our utility quantification is guiding the selection of LDP mechanisms and privacy parameters for a given classifier. Notably, our analysis shows that a piecewise-based mechanism often yields better utility than alternatives in common scenarios.

Beyond the core framework, we introduce two novel refinement techniques that further improve utility quantification. We then present case studies illustrating utility quantification for various combinations of LDP mechanisms and classifiers. Results demonstrate that our theoretical quantification closely matches empirical observations, particularly when classifiers operate in lower-dimensional input spaces.

Keywords

local differential privacy, classifier utility, robustness analysis, utility quantification

1 Introduction

Classifiers are functions that map input data to class labels, playing a fundamental role in industrial applications ranging from predictive modeling and data clustering to image recognition [32, 37, 42]. When deployed as services, classifiers require users to provide data inputs, which often contain sensitive information such as medical records or financial data, raising significant privacy concerns. While users wish to benefit from classification services, they may be reluctant to reveal their actual data. A common approach to address this concern is data perturbation, e.g. adding noise to numerical data or applying blur filters to images before sending them to the classifier. Data perturbation remains a lightweight and intuitive solution for privacy-preserving classification [9, 69, 72].

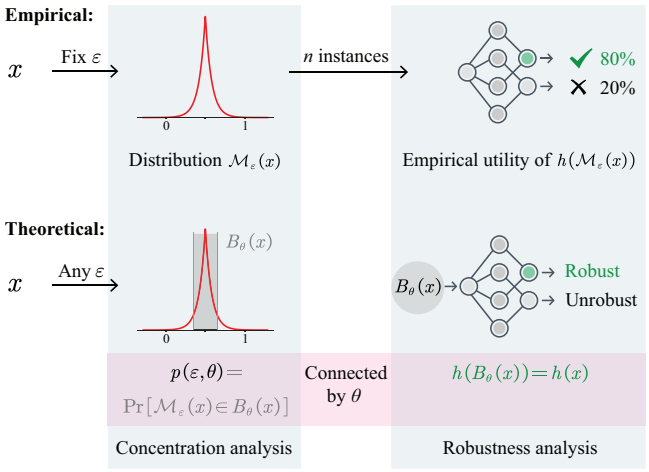


Figure 1: From empirical data utility to theoretical data utility. This paper provides a theoretical quantification of classifier utility under LDP-perturbed inputs by connecting the *concentration analysis* of LDP mechanisms with the *robustness analysis* of classifiers.

Local differential privacy (LDP) is a widely accepted mathematical framework for privacy protection through controlled data perturbation. It uses a parameter ε (the privacy parameter) to quantify the level of privacy protection. When data is processed through an LDP mechanism before being sent to a classifier, it receives provable privacy guarantees, meaning that any observer including the classifier cannot confidently infer the original sensitive data. Researchers have developed numerous LDP mechanisms for various data domains [18, 28, 33, 41, 60, 61], each offering different perturbation strategies while maintaining formal privacy guarantees.

Data utility is the most crucial metric when implementing LDP mechanisms. While privacy protection is controlled by the privacy parameter ε, data utility measures how well the perturbed data serves its intended application purpose. For simple applications like summation queries, utility can be directly quantified through the theoretical mean squared error (MSE) of LDP mechanisms [60, 61]. For classification tasks, utility corresponds to classification accuracy, which cannot be analytically expressed using MSE. A trivial approach to evaluate classifier utility under an LDP mechanism with a given privacy parameter ε is to repeatedly perturb the data and measure the proportion of correctly classified instances, as illustrated in Figure 1. While practical, this empirical method has significant limitations: it only applies to specific ε values and perturbed data instances; changing ε requires time-consuming re-evaluation, and different instances lead to different results. Meanwhile, it fails to

Proceedings on Privacy Enhancing Technologies 2026(3), 721–742

https://doi.org/10.56553/popets-2026-0103

establish relationships between utility and privacy parameters, making it inadequate for systematic comparison of LDP mechanisms. Analytical utility quantification, by contrast, provides valuable guidance for classifier designers and users; however, unlike the MSE for summation queries, we currently lack such a quantification framework for classifiers.

Quantifying the relationship between privacy and classifier utility presents significant analytical challenges. These challenges stem from two aspects: (i) LDP mechanisms inherently introduce perturbations across the entire data domain, potentially causing zero utility for classifiers. (ii) Classifiers are often complex or even black-box functions, making it difficult to analyze their behavior under perturbed inputs.

This paper develops a framework to theoretically quantify classifier utility under LDP-perturbed inputs. Our approach addresses the aforementioned challenges through two key insights: (i) LDP mechanisms generate perturbed data that concentrates within a bounded region around the original data with a specific high probability, making extreme perturbations rare. (ii) Within this concentration region, the utility analysis of a classifier can be reformulated as a robustness analysis problem. Robustness refers to how well a classifier maintains correct predictions when inputs are subjected to perturbations. We leverage established robustness analysis techniques to find the maximum permissible perturbation that preserves a classifier's output, thereby maintaining utility. By combining the concentration analysis of LDP mechanisms with the robustness analysis of classifiers, we establish theoretical data utility quantification in relation to the privacy parameter ε, expressed as:

Given classifier h, LDP mechanism $\mathcal{M}_\varepsilon$, and raw data x: with probability at least $p(\varepsilon, \theta)$, h preserves its correct classification under input $\mathcal{M}_\varepsilon(x)$.

This formulation provides a probabilistic guarantee for preserving the utility of classifier h when using mechanism $\mathcal{M}_\varepsilon$. The parameter θ represents a concentration threshold that determines the probability guarantee $p(\varepsilon, \theta)$, which directly depends on the privacy parameter ε. Figure 1 illustrates this intuition. By determining the maximum allowable perturbation $B_\theta(x)$ that a classifier h can tolerate, we can derive utility guarantees for h under $\mathcal{M}_\varepsilon$-perturbed inputs across different privacy parameters ε, without needing to repeatedly sample from the mechanism. Furthermore, if the distribution of raw data x is known, our framework also provides average-case and worst-case utility quantifications.

Additionally, we introduce two refinement techniques to refine the utility quantification $p(\varepsilon, \theta)$ for high-dimensional classifiers. First, we extend the widely used robustness notion of "robustness radius" to "robustness hyperrectangle", enabling a more precise analysis of classifier robustness. Second, we adapt a relaxed privacy notion, *Probably Approximately Correct* (PAC) privacy [68], as an alternative to (ε, δ)-LDP. Under this notion, we propose a novel privacy indicator and an extended Gaussian mechanism* applicable for any ε. The original Gaussian mechanism from Dwork [18] is only valid for $\varepsilon \in [0, 1]$. We improve the proof technique to extend it to $\varepsilon \in [0, \infty)$. These extensions allow the framework to provide more accurate utility quantification and incorporate more privacy-preserving mechanisms.

Applications. The proposed utility quantification framework has direct applications in privacy-preserving classification systems: (i) It enables systematic comparison of different LDP mechanisms for a given classifier by evaluating their probability guarantees $p(\varepsilon, \theta)$. For a fixed ε, a mechanism with a larger $p(\varepsilon, \theta)$ provides stronger utility preservation. (ii) It facilitates the selection of an appropriate privacy parameter ε to meet specific utility requirements. For example, given a utility threshold p^*, the framework identifies the ε that satisfies $p(\varepsilon, \theta) \geq p^*$, achieving a precise privacy-utility balance when using the classifier.

To summarize, our contributions are as follows:

- *Quantification framework.* To the best of our knowledge, this is the first framework that provides analytical utility quantification for classifiers under LDP-perturbed inputs. This framework bridges the concentration analysis of LDP mechanisms with the robustness analysis of classifiers, enabling systematic evaluation of classifier utility.
- *Refinement techniques.* We propose two refinement techniques to enhance utility quantification. The first extends the robustness notion from "robustness radius" to "robustness hyperrectangle," allowing for more precise robustness analysis. The second adapts the PAC privacy notion, introducing a novel privacy indicator and an extended Gaussian mechanism applicable to any ε.
- *Comparison and case studies.* We conduct case studies on typical classifiers, including logistic regression, random forests, and neural networks, under input perturbations by various LDP mechanisms. The results demonstrate that our theoretical utility quantification aligns closely with empirical observations, particularly in low-dimensional input spaces.

Structure. The main parts of this paper are organized as follows: Section 3 provides an illustrative example of utility quantification for a classifier under the Laplace mechanism. Section 4 introduces the formal framework for utility quantification. Section 5 presents refinement techniques, then Section 6 discusses related questions and Section 7 conducts case studies.

2 Preliminaries

This section formulates the problem and reviews the definition of LDP as well as the concept of classifier robustness.

2.1 Problem Formulation

Given a trained classifier h that provides public services, users must submit their data x as input to utilize the classifier. However, users' data x may contain sensitive information, and the third-party classifier h may not be fully trusted to protect data privacy. To address this concern, users perturb their data x using an LDP mechanism $\mathcal{M}_\varepsilon$ before sending it to the classifier h. The perturbation mechanism $\mathcal{M}_\varepsilon$ acts as an interface between users and the classifier, forming a trust boundary by presenting the perturbed data $\mathcal{M}_\varepsilon(x)$ to the classifier h.

While the perturbed data $\mathcal{M}_\varepsilon(x)$ protects user privacy, it may degrade the utility of the classifier h, potentially leading to incorrect outputs where $h(\mathcal{M}_\varepsilon(x)) \neq h(x)$. This degradation is undesirable

*It is well known that the Gaussian mechanism cannot satisfy pure (L)DP [18]; it must rely on a relaxed privacy notion such as (ε, δ)-LDP or PAC LDP.

for both the classifier provider and the users. Thus, a critical question arises: *How can classifier designers or users quantify the utility of the classifier h under the perturbation of $\mathcal{M}_\varepsilon$?* This paper aims to answer this question by introducing a theoretical framework to quantify the utility of h under the perturbation of $\mathcal{M}_\varepsilon$ w.r.t. the privacy parameter ε.

2.2 Local Differential Privacy

Definition 1 (ε-LDP [16]). *A perturbation mechanism $\mathcal{M} : \mathcal{X} \to \text{Range}(\mathcal{M})$ satisfies ε-LDP if, for any two arbitrary x_1 and x_2, the probability ratio of producing the same $\tilde{x}$ is bounded as follows:*

$$\forall x_1, x_2 \in \mathcal{X}, \forall \tilde{x} \in \text{Range}(\mathcal{M}) : \frac{\Pr[\mathcal{M}(x_1) = \tilde{x}]}{\Pr[\mathcal{M}(x_2) = \tilde{x}]} \leq e^\varepsilon.$$

If $\mathcal{M}(x)$ is continuous, the probability $\Pr[\cdot]$ is replaced by the probability density function (pdf). Definition 1 quantifies the difficulty of distinguishing between x_1 and x_2 based on the observed $\tilde{x}$. A smaller privacy parameter $\varepsilon \in [0, +\infty)$ indicates stronger privacy protection. For convenience, we use $\mathcal{M}_\varepsilon$ to denote an ε-LDP mechanism $\mathcal{M}$ when ε is necessary.

Some LDP mechanisms, such as the Laplace and Gaussian mechanisms [18], were originally designed for query functions on raw data. These mechanisms require the *sensitivity* of the query function to determine the scale of perturbation. The (global) sensitivity of a function $f : \mathcal{X} \to \text{Range}(f)$ is defined as the maximum change in f for any two data:

$$\Delta f = \max_{x_1, x_2 \in \mathcal{X}} |f(x_1) - f(x_2)|.$$

In the context of LDP, the privacy of the data x itself must be ensured, as the perturbation is applied directly to the data. Thus, here the query function f is the identity function $f(x) = x$, whose sensitivity is the diameter of input space $\mathcal{X}$. For classifiers, $\mathcal{X}$ is typically normalized to $\mathcal{X} = [0, 1]$, resulting in a sensitivity of 1.

2.3 Classifier and Robustness

Definition 2 (Classifier). *A classifier $h : \mathbb{R}^d \to \{1, 2, \ldots, K\}$ is a function that maps input data $x \in \mathbb{R}^d$ to a specific label $h(x) \in \{1, 2, \ldots, K\}$.*

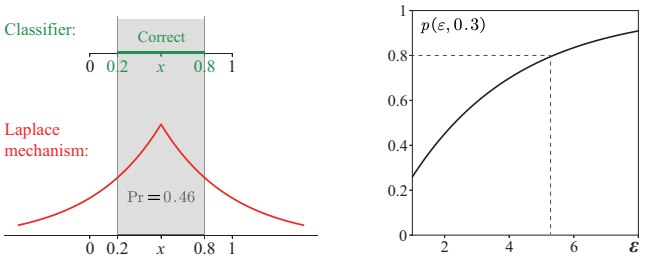
Based on representation complexity, classifiers can be categorized as follows: a *closed-form classifier* is represented by a closed-form function, such as a linear classifier; a *non-closed-form classifier* lacks a closed-form representation, such as a neural network. In both cases, if we know the classifier's structure and parameters, we call it a *white-box classifier*; otherwise, it is a *black-box classifier*.

Definition 3 (Local robustness). *Denote $B_\theta(x)$ as the ℓ_∞-norm ball centered at x with radius θ.[†] A classifier h is θ-locally-robust at x if:*

$$\forall \tilde{x} \in B_\theta(x),\ h(\tilde{x}) = h(x).$$

Intuitively, local robustness means that the classifier h is insensitive to small perturbations around the input x. The largest feasible θ is referred to as the *robustness radius* of the classifier h at x. Robustness radius serves as a key metric for evaluating the classifier's stability to unseen data.

[†]The ℓ_∞-norm is a common choice for measuring data perturbations and can be converted to other norms using norm inequalities. Here, $B_\theta(x)$ denotes the set of points $\tilde{x} \in \mathbb{R}^d$ where each dimension of $\tilde{x}$ is within a distance θ of the corresponding dimension of x, i.e. $B_\theta(x) \coloneqq \{\tilde{x} : |\tilde{x}^{(i)} - x^{(i)}| \leq \theta\}$.



(a) Example illustration. **(b) Classifier utility w.r.t. ε.**

Figure 2: Illustration of the example and utility quantification of the classifier under the Laplace mechanism for $x = 0.5$.

3 Illustrating the Framework: An Example

We illustrate our utility quantification framework with an intuitive example. Specifically, we consider a classifier under the Laplace mechanism and present a utility quantification statement.

Definition 4 (The Laplace mechanism [18]). *For any $x \in [0, 1]$, the Laplace mechanism $\mathcal{M}_\varepsilon$ is defined as: $\mathcal{M}_\varepsilon(x) = x + \xi$, where $\xi \sim Lap(1/\varepsilon)$ is a random variable drawn from Laplace distribution with mean 0 and scale $1/\varepsilon$.*

In the Laplace mechanism, $\mathcal{M}_\varepsilon(x)$ is unbounded due to the Laplace distribution. Since the classifier is typically defined on a bounded domain, such as $[0, 1]$, a common practice is to truncate the perturbed data to this range, i.e. $\mathcal{M}_\varepsilon(x)$ is truncated to $[0, 1]$.[‡]

3.1 Example

Consider a classifier $h : [0, 1] \to \{1, 2\}$ with one-dimensional input and two classes, defined as:

$$h(x) = \begin{cases} 1, & \text{if } x \in [0.2, 0.8], \\ 2, & \text{otherwise.} \end{cases}$$

Consider a specific sensitive data $x = 0.5$, which will be perturbed using the Laplace mechanism $\mathcal{M}_\varepsilon$ before submitting it to the classifier. As a result, the classifier's output, $h(\mathcal{M}_\varepsilon(0.5))$, becomes a random variable due to the randomness of $\mathcal{M}_\varepsilon$.

To quantify the utility of $h(\mathcal{M}_\varepsilon(0.5))$ with respect to ε, we analyze two key properties: the concentration of LDP mechanism $\mathcal{M}_\varepsilon$ and the robustness of classifier h.

Concentration analysis. Denote region $B_\theta(x)$ as the θ-neighborhood ball of x. For instance, $B_{0.3}(0.5) \coloneqq \{\tilde{x} : |\tilde{x} - 0.5| \leq 0.3\}$. Then the perturbed data $\mathcal{M}_\varepsilon(0.5)$ falls within $B_{0.3}(0.5)$ with a probability $p(\varepsilon, \theta = 0.3)$. Specifically, denote $F_{\text{Lap}}(\theta)$ as the cumulative distribution function (CDF) of the Laplace distribution used in $\mathcal{M}_\varepsilon$. We

[‡]This truncation can be treated as a post-processing step by the users that does not leak privacy [18, 26, 63]. We will also examine mechanisms with bounded perturbations in Section 4.1, where no truncation is needed. There are also truncated Laplace mechanisms [11, 24] that directly sample from the truncated Laplace distribution. It is also applicable to our framework; we use the original Laplace mechanism on $[0, 1]$ as an example for simplicity.

have:

$$p(\varepsilon, \theta = 0.3) = \Pr[\mathcal{M}_\varepsilon(x) \in B_{0.3}(x)] = F_{\text{Lap}}(0.3) - F_{\text{Lap}}(-0.3)$$
$$= 2F_{\text{Lap}}(0.3) - 1 = 1 - \exp(-0.3\varepsilon).$$

Therefore, we can state that: $\mathcal{M}_\varepsilon(0.5)$ outputs a value within $B_{0.3}(0.5)$ with probability $1-\exp(-0.3\varepsilon)$. For instance, if we set $\varepsilon = 2$, we have $p(2, 0.3) = 0.46$. Figure 2a shows the illustration of this instance.

Robustness analysis. Analyzing the utility of $h(\mathcal{M}_\varepsilon(0.5))$ requires analyzing the robustness of h under the input range $B_\theta(0.5)$. We know that h is robust under input range $B_{0.3}(0.5) = [0.2, 0.8]$ according to its definition. Then, we can state that: h preserves the correct classification under input range $B_{0.3}(0.5)$.

Utility quantification. Combining the above, we can quantify the utility of $h(\mathcal{M}_2(0.5))$: *with probability at least* $p(2, 0.3) = 0.46$, h *preserves the correct classification under input* $\mathcal{M}_2(0.5)$. Importantly, this utility quantification can be extended to any ε using the closed-form expression of $p(\varepsilon, \theta)$.

Application. For this classifier, the theoretical utility quantification $p(\varepsilon, 0.3)$ can guide the choice of the privacy parameter ε in the Laplace mechanism to achieve a desired utility level. Figure 2b shows $p(\varepsilon, 0.3)$ as a function of ε. For instance, to ensure the classifier preserves the correct classification with probability ≥ 0.8, privacy parameter ε should be set to at least 5.2.

3.2 Further Steps

The above example has demonstrated an analytical approach to quantify the utility of a classifier under the Laplace mechanism. However, this example is limited in scope as it focuses on a specific noise-adding mechanism and assumes a white-box classifier with a known robustness radius. In practice, several challenges arise: (i) Diverse LDP mechanisms. Not all LDP mechanisms behave as noise-adding mechanisms, which means the concentration analysis may differ significantly. (ii) Variety of classifiers. Classifiers come in various forms, and determining their robustness radius often requires different techniques. (iii) Conservative utility quantification. The utility provided in this example is a lower bound under strong pure LDP constraints and a conservative robustness radius. It can be refined, especially for higher-dimensional classifiers.

The subsequent sections will address these challenges by introducing a general framework for utility quantification and refinement techniques. Specifically, we will cover the following aspects:

- *General LDP mechanisms.* We model an LDP mechanism as a general distribution function and derive its concentration analysis $p(\varepsilon, \theta)$. As examples, we explore commonly used continuous and discrete mechanisms.
- *Unified method for finding robustness radius.* We treat classifiers as black-box functions and adapt a hypothesis testing framework to determine their robustness radius. This approach provides a unified method applicable to classifiers with different structures and access types (white-box or black-box).
- *Refinement techniques.* We refine the utility quantification by incorporating the concepts of robustness hyperrectangles and PAC privacy.

4 Formal Framework

This section presents the formal framework for quantifying classifier utility under input perturbations from various types of LDP mechanisms.

4.1 Concentration Analysis of LDP

In Definition 1 of LDP, the mechanism $\mathcal{M}(x)$ represents a family of probability distributions determined by the input x. Thus, the perturbed output $\tilde{x} = \mathcal{M}(x)$ is a random variable. The concentration property characterizes how the probability mass of the perturbed output $\tilde{x}$ concentrates around the original input x, and can be stated as follows.

Proposition 1 (Concentration property of LDP). *Let $\mathcal{M}(x)$ be an ε-LDP mechanism applied to data $x \in \mathbb{R}$, and let $F_\mathcal{M}$ denote the CDF of the perturbed data $\mathcal{M}(x)$. The concentration property of $\mathcal{M}(x)$ is given by:*

$$\forall a < b : \Pr[a \leq \mathcal{M}(x) \leq b] = F_\mathcal{M}(b) - F_\mathcal{M}(a).$$

This proposition follows directly from the definition of the CDF and provides a probabilistic characterization of the "boundedness" of the LDP mechanism $\mathcal{M}(x)$. As a special case, if $a = x - \theta$ and $b = x + \theta$, it becomes $\Pr[\mathcal{M}(x) \in B_\theta(x)]$. For practical mechanisms designed for better utility, $\mathcal{M}(x)$ is typically concentrated around x. This implies that $F_\mathcal{M}(b) - F_\mathcal{M}(a)$ captures most of the probability mass when $x \in [a, b]$. We analyze the concentration properties of typical continuous and discrete LDP mechanisms in the following subsections.

4.1.1 Continuous Mechanisms. When the data domain is continuous, the LDP mechanism corresponds to a continuous distribution. Based on whether the perturbation depends on the input x, continuous mechanisms are categorized into: (i) Noise-adding mechanisms, which add noise independent of the input x. (ii) Piecewise-based mechanisms, which sample outputs from input-dependent piecewise distributions.

Noise-adding Mechanisms. These mechanisms achieve LDP by adding carefully designed noise to the input data.

Definition 5. *A noise-adding LDP mechanism $\mathcal{M} : \mathcal{X} \to \text{Range}(\mathcal{M})$ is defined as: $\mathcal{M}(x) = x + \xi$, where ξ is a random variable (noise) drawn from a symmetric distribution with mean* 0.

Noise-adding mechanisms are input-independent, using the same noise distribution for all inputs. Examples include the Laplace and Gaussian mechanisms [18].

Concentration Analysis. For a noise-adding mechanism $\mathcal{M}(x)$, the probability of outputting a perturbed value $\mathcal{M}(x) \in B_\theta(x)$ is:

$$F_\mathcal{M}(x + \theta) - F_\mathcal{M}(x - \theta) = 2 \cdot F_\mathcal{M}(\theta) - 1,$$

where $F_\mathcal{M}$ is the CDF of the noise distribution. The last equality holds because of the symmetry of the noise distribution, making the result independent of x. For instance, the Laplace mechanism's concentration analysis is given by:

$$2F_{\text{Lap}}(\theta) - 1 = 1 - \exp(-\theta\varepsilon).$$

In the previous section, we have seen that for $\varepsilon = 2$ and $\theta = 0.3$, the concentration probability is 0.46. This indicates that while the

Laplace mechanism is defined on the whole real line, it is concentrated in a small region around x.

Piecewise-based Mechanisms. These mechanisms achieve LDP by sampling outputs from carefully designed piecewise distributions that vary based on the input x.

Definition 6. *A piecewise-based LDP mechanism* $\mathcal{M}(x) : \mathcal{X} \to \text{Range}(\mathcal{M})$ *is defined as:*

$$pdf[\mathcal{M}(x) = \tilde{x}] = \begin{cases} p_\varepsilon & \text{if } \tilde{x} \in [l_{x,\varepsilon}, r_{x,\varepsilon}], \\ p_\varepsilon / \exp(\varepsilon) & \text{otherwise}, \end{cases}$$

where p_ε is the sampling probability, and $[l_{x,\varepsilon}, r_{x,\varepsilon}]$ define the sampling interval based on x and ε.

Piecewise-based mechanisms are input-dependent, meaning the sampling distribution changes for different inputs x. Examples include PM [60] and SW [41], which use different p_ε and intervals $[l_{x,\varepsilon}, r_{x,\varepsilon}]$. New mechanisms can be designed by modifying these parameters.

Concentration Analysis. For a piecewise-based mechanism $\mathcal{M}(x)$, the probability of getting a perturbed value $\mathcal{M}(x) \in B_\theta(x)$ depends on θ. The concentration analysis is given by:

$$\begin{cases} 2\theta \cdot p_\varepsilon & \text{if } B_\theta(x) \subseteq [l_{x,\varepsilon}, r_{x,\varepsilon}], \\ (r_{x,\varepsilon} - l_{x,\varepsilon}) p_\varepsilon + \left(2\theta - (r_{x,\varepsilon} - l_{x,\varepsilon})\right) \frac{p_\varepsilon}{\exp(\varepsilon)} & \text{otherwise.} \end{cases}$$

This analysis is x-dependent due to the terms $l_{x,\varepsilon}$ and $r_{x,\varepsilon}$. For example, in the PM mechanism with $p_\varepsilon = \exp(\varepsilon/2)$, if $B_\theta(x) \subseteq [l_{x,\varepsilon}, r_{x,\varepsilon}]$, the probability is $2\theta \cdot \exp(\varepsilon/2)$. Concretely, for $\varepsilon = 2$ and $\theta = 0.3$, the concentration probability $\Pr[\mathcal{M}(x) \in B_\theta(x)]$ equals 0.86, indicating that the PM mechanism is more concentrated than the Laplace mechanism under these parameters.

4.1.2 Discrete Mechanisms. When the data domain is discrete and finite, the LDP mechanism corresponds to a discrete distribution. Many real-world datasets are discrete, such as image pixel values (e.g. 256 levels) or integer-valued attributes like age.

Definition 7. *If $|\mathcal{X}| < \infty$, a discrete LDP mechanism $\mathcal{M}(x) : \mathcal{X} \to \mathcal{X}$ is defined as:*

$$\Pr[\mathcal{M}(x) = \tilde{x}_i] = p_\varepsilon(\tilde{x}_i, x), \quad \tilde{x}_i \in \mathcal{X},$$

where $p_\varepsilon(\tilde{x}_i, x)$ is the probability of perturbing x to $\tilde{x}_i$, determined by ε, x, and $\tilde{x}_i$.

Examples of such mechanisms include the k-RR mechanism [33], where $p_\varepsilon(\tilde{x}_i, x)$ is uniform except for $\tilde{x}_i = x$, and the Exponential mechanism [18], where $p_\varepsilon(\tilde{x}_i, x)$ depends on the distance between $\tilde{x}_i$ and x.

Concentration Analysis. For discrete mechanisms, the support is a finite set $\mathcal{X}$, and the CDF is defined as: $F_\mathcal{M}(\tilde{x}) = \sum_{\tilde{x}_i \le \tilde{x}} p_\varepsilon(\tilde{x}_i, x)$. The probability of $\mathcal{M}(x)$ producing a value in $B_\theta(x)$ is:

$$F_\mathcal{M}(x + \theta) - F_\mathcal{M}(x - \theta) = \sum_{\tilde{x}_i \in [x-\theta, x+\theta]} p_\varepsilon(\tilde{x}_i, x).$$

For instance, in the k-RR mechanism:

$$p_\varepsilon(\tilde{x}_i, x) = \frac{\exp(\varepsilon)}{|\mathcal{X}| - 1 + \exp(\varepsilon)} \text{ if } \tilde{x}_i = x \text{ otherwise } \frac{1}{|\mathcal{X}| - 1 + \exp(\varepsilon)}.$$

Then, if $\mathcal{X}$ discretizes $[0, 1]$ evenly, i.e. $\mathcal{X} = \{0, 1/|\mathcal{X}|, 2/|\mathcal{X}|, \ldots, 1\}$, the concentration analysis becomes:

$$\sum_{\tilde{x}_i \in [x-\theta, x+\theta]} p_\varepsilon(\tilde{x}_i, x) = \frac{\exp(\varepsilon)}{|\mathcal{X}| - 1 + \exp(\varepsilon)} + \frac{2\theta|\mathcal{X}| - 1}{|\mathcal{X}| - 1 + \exp(\varepsilon)}.$$

When $|\mathcal{X}| = 100$, $\varepsilon = 2$, and $\theta = 0.3$, the concentration probability of the k-RR mechanism is 0.63.

4.1.3 Comparison of Concentration Properties. Fixing $\varepsilon = 2$ and $\theta = 0.3$, the concentration probabilities of the mentioned three mechanisms are:

- Laplace mechanism: $p(\varepsilon = 2, \theta = 0.3) = 0.46$;
- PM mechanism: $p(\varepsilon = 2, \theta = 0.3) = 0.86$;
- k-RR mechanism: $p(\varepsilon = 2, \theta = 0.3) = 0.63$.

These results highlight that different LDP mechanisms are differently concentrated under the same privacy parameter ε and region $B_\theta(x)$. This variation is critical for analyzing classifier utility and selecting appropriate LDP mechanisms. For instance, if a classifier's robustness radius θ is 0.3, and we set privacy level $\varepsilon = 2$, the PM mechanism achieves the highest probability of $\mathcal{M}(x) \in B_{0.3}(x)$, making it the best choice in this scenario.

4.2 Robustness Analysis of Classifier

We analyze the robustness of classifiers under the concept of *probabilistic robustness*, a probabilistic approximation of deterministic robustness.[§] Probabilistic robustness suffices for classifier utility analysis under LDP-perturbed inputs, as the concentration properties of LDP mechanisms are also probabilistic. Compared to deterministic robustness, it is computationally efficient and applicable to any classifier, including black-box classifiers, thus suitable as a general framework.

4.2.1 Probabilistic Robustness. Hypothesis testing provides an intuitive perspective for probabilistic robustness. We can state two counter hypotheses:

- H_0: h is robust under $B_\theta(x)$.
- H_1: h is not robust under $B_\theta(x)$.

These hypotheses can be statistically tested by querying the classifier on random samples in $B_\theta(x)$.[¶] If robustness holds for almost all samples, H_0 is accepted with high confidence. Formally, probabilistic robustness is defined as follows:

Definition 8 (Probabilistic robustness [57, 70]). *Given a classifier h, an input x, a radius θ, and a tolerance $\tau > 0$, h is θ-locally-robust at x with probability at least $1 - \omega$ if*

$$\Pr_{\tilde{x} \sim B_\theta(x)} \left[1 - \Pr[h(\tilde{x}) = h(x)] \le \tau\right] \ge 1 - \omega,^{\|}$$

where $\tilde{x}$ is a random variable drawn from $B_\theta(x)$.

[§] If the classifier is white-box, deterministic robustness can be derived analytically or through verification methods. See Appendix C.1 for details. In the most ideal case, the deterministic robustness can be expressed as a closed-form function of arbitrary x. We omit such easier cases for generality.

[¶] Robustness analysis is a different scenario from *using* the classifier, and does not leak users' privacy. Section 6 provides further discussions.

[‖] The inner probability $\Pr[h(\tilde{x}) = h(x)]$ is taken over the uncertainty of robustness, refer to Appendix C.2 for explanation.

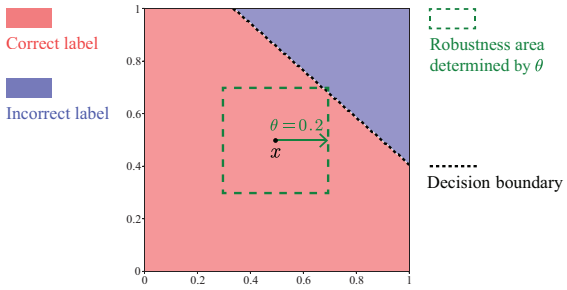


Figure 3: Robustness radius θ of a 2D classifier at x and the tested decision boundary by brute force.

This definition implies that the event $h(\tilde{x}) = h(x)$ is almost always true (within a tolerance τ) with probability at least $1-\omega$. Here, $1-\omega$ also corresponds to the confidence level (type I error) in the hypothesis testing. For minimal τ and ω, e.g. $\tau = 0.01$ and $\omega = 0.05$, h is nearly θ-locally robust at x. Deterministic robustness is a special case with $\tau = 0$ and $\omega = 0$.

Probabilistic robustness is typically computed using the Hoeffding bound [57, 70], which provides a probabilistic guarantee for the mean of a random variable. Let Z be the indicator function of $h(\tilde{x}) = h(x)$, i.e. $Z = 1$ if robust and $Z = 0$ otherwise. Then, the empirical mean $\hat{\mu}_Z$ (robust frequency) is calculated by sampling n points, and the true robustness probability μ_Z can be derived using the Hoeffding bound.

Theorem 1 (Hoeffding bound [31]). *For any $\tau \geq 0$ and $\omega > 0$, the probability that the empirical mean ($\hat{\mu}_Z$) is within τ of the true mean (μ_Z) is bounded by:*

$$\Pr_{Z_i \sim P_Z}\left[|\hat{\mu}_Z - \mu_Z| \leq \tau\right] \geq 1-\omega,$$

where $\omega = 2e^{-2n\tau^2}$ and n is the number of samples. Equivalently, at least $n(\omega,\tau) = (\ln\frac{2}{\omega})/(2\tau^2)$ samples are required to ensure the bound.

The number of samples n balances utility guarantees and computational cost. A smaller ω (more deterministic guarantee) requires more samples n. Given τ and ω, we check if the empirical mean $\hat{\mu}_Z$ on $n(\omega,\tau/2)$ samples satisfies $\hat{\mu}_Z \in [1-\tau/2, 1]$. If this is true, by the Hoeffding bound, $|\mu_Z - \hat{\mu}_Z| \leq \tau/2$ holds, which implies $\mu_Z \in [1-\tau, 1]$. Thus, Definition 8 holds, and we can claim that h is robust under $B_\theta(x)$ with probability at least $1-\tau$, and our confidence level is at least $1-\omega$.

Independent of the classifier. The above method treats the classifier as a black-box function, making it applicable to any classifier h without requiring knowledge of its internal structure or parameters.

4.2.2 Procedure for Finding θ. After setting tolerance τ and confidence level ω, we can find a probabilistic guaranteed robustness radius by the following procedure. The key idea is to increase θ from 0 until the misclassification rate exceeds $\tau/2$. Specifically,

(1) Compute the required number of samples $n(\omega,\tau/2)$ using the Hoeffding bound in Theorem 1.
(2) For the current θ, uniformly sample n points from $B_\theta(x)$, then query the classifier and calculate the misclassification rate.
(3) If the misclassification rate is below $\tau/2$, increase θ and repeat the second step.

As θ increases, the misclassification rate will eventually exceed $\tau/2$. The maximum θ before this point is the robustness radius. In practice, binary search is often used in the third step to reduce the complexity to a logarithmic scale.

Example 1. *Given a 2D neural network classifier $h : [0,1]^2 \to \{1,2\}$, treated as a black-box, and parameters $\tau = 0.02$ and $\omega = 0.05$, we determine the robustness radius θ at $x = (0.5, 0.5)$ using the described procedure. Specifically: (i) the required number of samples for testing is $n(\omega,\tau/2) \approx 1.8 \times 10^4$; (ii) by testing $\theta \in [0,1]$, we find the robustness radius to be $\theta = 0.20$.*

Figure 3 illustrates this example. The true decision boundary of h at x is shown by the black dashed line, computed via brute force. The green dashed box represents the robustness area $B_\theta(x)$ determined by the found $\theta = 0.20$. This result closely touches the true decision boundary.

In the above example, any perturbation within the robustness region $B_\theta(x)$ ensures that the classifier's output remains unchanged, thereby preserving utility. When the perturbation is performed by an LDP mechanism, the perturbed value $\tilde{x}$ will fall within $B_\theta(x)$ with a specific probability. The next subsection establishes the connection between $B_\theta(x)$ and this probability to quantify the classifier's utility under LDP-perturbed inputs.

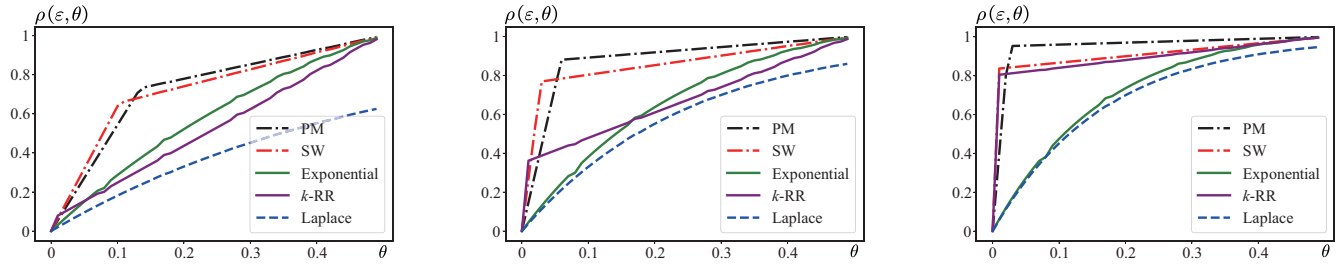
4.3 Utility Quantification

By combining the concentration analysis of LDP mechanisms and the robustness analysis of classifiers, we derive a utility quantification for a given classifier under inputs perturbed by an LDP mechanism w.r.t. any privacy parameter ε. This quantification framework provides a systematic evaluation of classifier performance under LDP-perturbed inputs. In contrast, the empirical approach requires re-evaluation for different ε and fails to establish analytical relationships between classifier utility and ε.

Formally, by Definition 8, the classifier is robust under $B_\theta(x)$ with tolerance τ and confidence level $1-\omega$. Therefore, under the same confidence level $(1-\omega)$, a probabilistic estimate of the event $h(\tilde{x}) = h(x)$, i.e. exact robustness without tolerance, is at least $1-\tau$. Meanwhile, the LDP mechanism $\mathcal{M}$ perturbs x into $B_\theta(x)$ with probability $F_{\mathcal{M}}(x+\theta) - F_{\mathcal{M}}(x-\theta)$. Thus, the utility guarantee of the classifier under the LDP mechanism is quantified by the product of these two probabilities. Specifically, given a d-dimensional classifier h and raw data x, denote $\mathcal{M}^d(x)$ as the perturbed d-dimensional input by d independent $\mathcal{M}$.** We claim: *Under confidence level at least $1-\omega$, with probability at least $\rho(\varepsilon,\theta)$, the classifier h preserves the correct classification result under the input perturbed by $\mathcal{M}^d$, where*

$$\rho(\varepsilon,\theta) = \Pr[\mathcal{M}^d(x) \in B_\theta(x)] \cdot \Pr_{\tilde{x} \sim B_\theta(x)}[h(\tilde{x}) = h(x)]$$
$$\geq (F_{\mathcal{M}}(x+\theta) - F_{\mathcal{M}}(x-\theta))^d \cdot (1-\tau),$$

**For simplicity, we assume $\mathcal{M}$ is applied to all dimensions of h. In practice, it may only be applied to a subset of sensitive features.



(a) $\varepsilon = 2$ **(b)** $\varepsilon = 4$ **(c)** $\varepsilon = 6$

Figure 4: Comparison of the robustness probability $\rho(\varepsilon, \theta)$ for different LDP mechanisms with $\varepsilon = 2, 4, 6$. Details of the mechanisms and discussions on the curves of $\rho(\varepsilon, \theta)$ are provided in Appendix C.3.

with $F_{\mathcal{M}}$ the CDF of the LDP mechanism $\mathcal{M}$.

For a given mechanism $\mathcal{M}$, the term $F_{\mathcal{M}}(x+\theta) - F_{\mathcal{M}}(x-\theta)$ is a closed form w.r.t. ε and θ, directly computable from the mechanism's parameters. ω and τ are predetermined based on Definition 8 of robustness. We fix $\omega = 0.05$ and $\tau = 0.01$ for all subsequent analyses. Thus, with the known robustness radius θ of the classifier h at x, $\rho(\varepsilon, \theta)$ provides an immediate utility quantification for any ε.

Furthermore, if the distribution of x as P_x is known, we can extend the utility quantification to provide average-case (expected) and worst-case guarantees:[††]

$$\rho_{\text{avg}}(\varepsilon) = \mathbb{E}_{x\sim P_x}[\rho(\varepsilon, \theta)], \text{and } \rho_{\text{wor}}(\varepsilon) = \min_{x\sim P_x}[\rho(\varepsilon, \theta)].$$

Such guarantees provide input-distribution-aware utility quantification for the classifier. For example, the average-case utility guarantee can be stated as follows: *On average, under confidence level at least $1-\omega$,[‡‡] with probability at least $\rho_{\text{avg}}(\varepsilon)$, the classifier h preserves the correct classification result for an input $x \sim P_x$ perturbed by $\mathcal{M}^d$.*

Application 1: Select the best LDP mechanism. A direct application of this utility quantification is comparing the classifier's utility under different LDP mechanisms to select the best one. Mechanisms with higher $\rho(\varepsilon, \theta)$ yield higher utility guarantees for classifiers. However, this varies with θ for different classifiers and ε for different mechanisms. In fact, there is not a single best mechanism for all classifiers under all ε.

Example 2. *Figure 4 shows the utility guarantee $\rho(\varepsilon, \theta)$ for different LDP mechanisms, with θ ranging from 0.1 to 0.5 and at $x = 0.5$. Detailed instantiations of these mechanisms are provided in Appendix C.3. A higher $\rho(\varepsilon, \theta)$ indicates a higher utility for classifiers with robustness radius θ. It is evident that no single mechanism is universally optimal for all ε and θ. For instance, when ε and θ are small (e.g. $\varepsilon = 2$ and $\theta \le 0.1$), the SW mechanism performs best. As ε and θ increase, the PM mechanism becomes the superior choice.*

Takeaway results. Although no mechanism is universally optimal for all classifiers (with different robustness radius θ) under all ε, heuristic guidelines can assist in mechanism selection. From the above example and Figure 4, we observe that if the privacy parameter ε is not too small (≥ 2), and the classifier's robustness radius θ is not too small (≥ 0.1), then the PM mechanism is generally the best choice. Experiments in Section 7 further support this conclusion for higher-dimensional classifiers.

Application 2: Select ε for a utility requirement. With a chosen mechanism, the utility quantification can guide the selection of the privacy parameter ε to meet a desired utility guarantee. For example, if the classifier's robustness radius θ is 0.3, and we require the classifier to preserve the correct classification result with probability $\rho(\varepsilon, \theta) \ge 0.8$ under the PM mechanism, we can set $\varepsilon = 1.8$ to meet this utility requirement while ensuring a strong privacy guarantee.

Impact of dimensionality. For high-dimensional classifiers, the primary challenge lies in the (sensitive) input dimension d. The utility guarantee $\rho(\varepsilon, \theta)$ decreases exponentially with d, and capturing the robustness of high-dimensional classifiers becomes more difficult, making the utility quantification less meaningful.

The second part of our contribution addresses this dimensionality issue. By introducing more precise robustness analysis methods and relaxing the privacy constraint, we refine the utility guarantee $\rho(\varepsilon, \theta)$, making the utility quantification more practical for high-dimensional classifiers.

5 Refinement Techniques

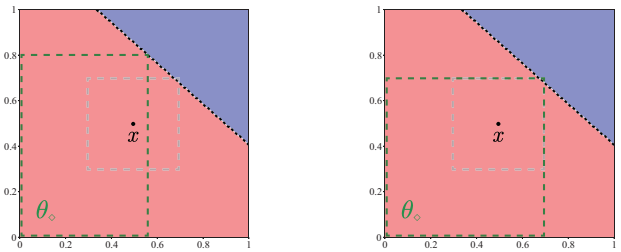
This section introduces two techniques for refining the utility quantification presented in Section 4.3. The utility guarantee $\rho(\varepsilon, \theta)$ is an at-least guarantee based on the robustness radius under pure ε-LDP. To refine it, we propose the following techniques.

- *Robustness refinement.* We generalize the robustness radius to a robustness hyperrectangle, which assigns distinct radii to different dimensions, enabling a more precise computation of $\rho(\varepsilon, \theta)$.
- *Privacy relaxation.* We relax the pure ε-LDP constraint to (ε, δ)-PAC LDP, which tolerates a small probability of failure. Under this relaxation, we introduce a δ-probabilistic indicator for applying the ε-LDP mechanism and propose an extended Gaussian mechanism tailored for PAC LDP.[§§]

[††]Evaluating average-case and worst-case utility requires knowledge of the input distribution P_x. In most papers, analyses of worst-case utility for their proposed mechanisms (defined on a specific domain) implicitly assume P_x is defined on that domain with non-zero probability. In practice, P_x can be estimated from historical data or domain knowledge.

[‡‡]For clarity, we omit the confidence level in subsequent statements, as it is a predefined constant with $1 - \omega \approx 1$.

[§§]The primary motivation for adopting PAC LDP is to accommodate the Gaussian mechanism, which is widely used in privacy-preserving machine learning but does not satisfy pure LDP.



(a) $\theta_\diamond = [0, 0.55] \times [0, 0.8]$. **(b)** $\theta_\diamond = [0, 0.7] \times [0, 0.7]$.

Figure 5: Two robustness hyperrectangles (green dashed boxes) for the classifier in Figure 3. The original robustness radius $\theta = 0.2$ corresponds to the gray dashed box.

5.1 Robustness Hyperrectangle

To refine the utility quantification, we introduce the notion of a robustness hyperrectangle, which provides a more precise description of the robustness area.

The commonly used robustness radius for classifiers [19] is a conservative measure of robustness. It defines the robustness area as an ℓ_∞-ball,[¶¶] where all dimensions share the same radius. However, in practice, different dimensions often have varied robustness radii, making the true robustness area larger than that defined by an ℓ_∞-ball. In our context, accurately describing the robustness area is crucial for refining utility quantification. To this end, we define the robustness hyperrectangle, which accounts for the robustness of each dimension individually. The formal definition is as follows.

Definition 9 (Robustness hyperrectangle). *Given a d-dimensional classifier h and a data point x, the robustness hyperrectangle $\theta_\diamond := [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_d, b_d]$ is a d-dimensional hyperrectangle that includes x and satisfies*

$$\forall \tilde{x} \in \theta_\diamond,\ h(\tilde{x}) = h(x).$$

The robustness radius is a special case of the robustness hyperrectangle, where each dimension has the same size θ around x. More generally, we say the classifier h is $\theta_\diamond$-locally robust at x.

Finding robustness hyperrectangle. The robustness hyperrectangle $\theta_\diamond$ is not unique, unlike the robustness radius. Different methods for finding $\theta_\diamond$ may yield different results. Heuristically, we aim to include the robustness radius (i.e. $\theta \subseteq \theta_\diamond$) to ensure an improved utility quantification.

After determining the robustness radius θ using the procedure in Section 4.2.2, we initialize the robustness area of each dimension as $[a_i, b_i]$ with θ. We then try to expand $[a_i, b_i]$ along the a_i or b_i directions. This process outputs a maximal robustness hyperrectangle $\theta_\diamond$ that satisfies the definition in Definition 9.

Example 3. *Figure 5 illustrates two examples of robustness hyperrectangles for the classifier in Example 1. The original robustness radius is $\theta = 0.2$, corresponding to the hyperrectangle $[0.4, 0.6] \times [0.4, 0.6]$ (gray dashed boxes). Two different hyperrectangles (green dashed boxes) are shown, both larger than the original robustness radius. Figure 5a partially includes the robustness radius, while Figure 5b fully encompasses it. Our proposed procedure will obtain the hyperrectangle depicted in Figure 5b.*

Each dimension of $\theta_\diamond$, represented as $[a_i, b_i]$, influences the concentration analysis of LDP mechanisms. Larger $[a_i, b_i]$ allows for greater perturbation, leading to a stronger utility guarantee. Specifically, the concentration level θ in the utility guarantee $\rho(\varepsilon, \theta)$ is replaced with the refined $\theta_\diamond$, resulting in a strictly improved utility guarantee $\rho(\varepsilon, \theta_\diamond)$.

Refined utility quantification. Given a d-dimensional classifier h, LDP mechanism $\mathcal{M}_\varepsilon$ for each dimension, and sensitive data x. Denote the utility quantification $\rho(\varepsilon, \theta)$ as defined in Section 4.3. It can be refined to: *With probability at least $\rho(\varepsilon, \theta_\diamond)$, the classifier h preserves the correct classification under the input perturbed by d independent $\mathcal{M}_\varepsilon$ (i.e. pure $d\varepsilon$-LDP), where*[***]

$$\rho(\varepsilon, \theta_\diamond) = \prod_{i=1}^{d} F_{\mathcal{M}}(b_i) - F_{\mathcal{M}}(a_i),$$

where $[a_i, b_i]$ represents the i-th dimension of the hyperrectangle $\theta_\diamond$.

5.2 PAC Privacy

Probably approximately correct (PAC) privacy [68] provides a formal framework for relaxing the worst-case privacy guarantee. While pure LDP requires the privacy constraint to hold in all cases, PAC privacy allows a small probability of failure, i.e. probabilistic LDP. This relaxation allows the use of a wider range of mechanisms, notably the Gaussian mechanism, which is widely used in privacy-preserving machine learning.

To simplify notation, we denote the privacy loss [18] (of a mechanism $\mathcal{M}$) between x_1 and x_2 at an observation $\tilde{x}$ as

$$\mathcal{L}_{\mathcal{M},x_1,x_2}(\tilde{x}) := \ln\left(\frac{\Pr[\mathcal{M}(x_1) = \tilde{x}]}{\Pr[\mathcal{M}(x_2) = \tilde{x}]}\right).$$

Pure ε-LDP can then be expressed as: $\mathcal{L}_{\mathcal{M},x_1,x_2}(\tilde{x}) \le \varepsilon$ for all $x_1, x_2, \tilde{x}$. In contrast, (ε, δ)-PAC LDP is defined by a relaxed δ-probabilistic condition.[†††]

Definition 10 (PAC LDP, adopted from [68]). *A randomized mechanism $\mathcal{M} : \mathcal{X} \to \text{Range}(\mathcal{M})$ satisfies (ε, δ)-PAC LDP if*

$$\forall x_1, x_2, \tilde{x} : \Pr[\mathcal{L}_{\mathcal{M},x_1,x_2}(\tilde{x}) \le \varepsilon] \ge 1 - \delta,$$

i.e. $\mathcal{M}$ satisfies ε-LDP with probability at least $1 - \delta$.

Pure LDP is a special case of (ε, δ)-PAC LDP with $\delta = 0$. Although PAC LDP allows a δ-probabilistic failure, the adversary can only infer the original data with probability at most δ when ε-LDP fails.

Combination of PAC LDP mechanisms. When applying d independent PAC LDP mechanisms to d dimensions, if we follow the combination theorem from Dwork [18], it gives $(d\varepsilon, d\delta)$-PAC LDP. However, this result is not tight, as it allows the failure probability $d\delta$ to exceed 1, which is impossible. We provide a tighter result for combining d independent PAC LDP mechanisms.

[¶¶] Other ℓ_p-balls, such as ℓ_1-ball and ℓ_2-ball, are also used, but they are also symmetric across all dimensions.

[***] We omit the terms $(1 - \omega)$ and $(1 - \tau)$ related to $\rho(\varepsilon, \theta_\diamond)$ for simplicity in presentation. They are almost 1 in practice (e.g. $\omega = 0.05, \tau = 0.01$) and thus also negligible.

[†††] Readers may wonder about the relationship between (ε, δ)-PAC LDP and (ε, δ)-LDP. In fact, the traditional (ε, δ)-LDP from Dwork [18] is ambiguous and has many interpretations. Concentrated privacy [16] from Dwork, Rényi privacy [50], and PAC privacy [68] can address this ambiguity. See Appendix C.4 for a detailed discussion.

Theorem 2 (Combination of (ε, δ)-PAC LDP). *Given d independent mechanisms $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_d$ that satisfy (ε, δ)-PAC LDP, their combination satisfies $(d\varepsilon, 1-(1-\delta)^d)$-PAC LDP.*

Proof. (Sketch) The result of δ comes from computing the total failure probability of d independent mechanisms. Appendix B.3 provides the details. □

With the notion of PAC LDP, we present two techniques to improve the utility quantification. The first is a privacy indicator, which extends any pure LDP mechanism to PAC LDP. The second is an extended Gaussian mechanism, which cannot satisfy pure LDP but satisfy PAC LDP.

5.2.1 Privacy Indicator. The failure probability in (ε, δ)-PAC LDP allows us to design a δ-probabilistic indicator for the application of the pure LDP mechanism. Specifically, we define

$$\mathcal{I}_\delta(\mathcal{M}(x)) = \begin{cases} \mathcal{M}(x) & \text{w.p. } 1-\delta, \\ x & \text{w.p. } \delta, \end{cases}$$

i.e. with probability $1-\delta$, the mechanism $\mathcal{M}$ is applied, and with probability δ, the original data x is returned. The mechanism $\mathcal{I}_\delta(\mathcal{M}(x))$ satisfies (ε, δ)-PAC LDP.

Theorem 3. *Assume $\mathcal{M}$ is a randomized mechanism that satisfies ε-LDP. Then the mechanism $\mathcal{I}_\delta(\mathcal{M}(x))$ defined above satisfies (ε, δ)-PAC LDP.*

Proof. (Sketch) The proof follows directly from the definition of $\mathcal{I}_\delta(\mathcal{M}(x))$. Appendix B.1 provides the details. □

Privacy indicator for multiple mechanisms. For a d-dimensional data x, instead of applying the privacy indicator to each dimension independently, we can treat d mechanisms applied to each dimension as a single mechanism designed for d-dimensional data, denoted as $\mathcal{M}^d(x)$. The privacy indicator $\mathcal{I}_\delta(\mathcal{M}^d(x))$ then controls the application of all mechanisms together. By substituting the pure LDP mechanism $\mathcal{M}^d(x)$ with $\mathcal{I}_\delta(\mathcal{M}^d(x))$, we can refine the utility quantification while relaxing the privacy guarantee to (ε, δ)-PAC LDP.

Refined utility quantification. Given a d-dimensional classifier h, combined LDP mechanism $\mathcal{M}_\varepsilon^d$, privacy indicator $\mathcal{I}_\delta(\cdot)$, and raw data x, the utility quantification $\rho(\varepsilon, \theta)$ can be refined to: *With probability at least $\delta + (1-\delta)\rho(\varepsilon, \theta_\diamond)$, the classifier h preserves the correct classification under the input perturbed by $\mathcal{I}_\delta(\mathcal{M}^d(x))$, which guarantees $(d\varepsilon, \delta)$-PAC LDP.*

The refined utility guarantee $\delta + (1-\delta)\rho(\varepsilon, \theta_\diamond)$ is always larger than $\rho(\varepsilon, \theta_\diamond)$, as it equals $\rho(\varepsilon, \theta_\diamond) + \delta(1-\rho(\varepsilon, \theta_\diamond))$, and the latter term is non-negative. Therefore, the new utility guarantee is always a refinement over the original one.

5.2.2 Extended Gaussian Mechanism. It is already known that the Gaussian mechanism [18] cannot satisfy pure ε-LDP, making the privacy indicator inapplicable. Nonetheless, Dwork has shown that it can satisfy (ε, δ)-LDP for $\varepsilon \leq 1$. Under the notion of PAC LDP, we provide an extended Gaussian mechanism to work for any ε.[‡‡‡] The following theorem is the form for data domain $x \in [0, 1]$.

[‡‡‡]Analytical Gaussian mechanism [5] works for any ε, but it is based on an alternative privacy definition and lacks analytical form of the noise scale σ. Appendix B.2.3 provides the detailed comparison.

Theorem 4 (Extended Gaussian mechanism). *The Gaussian mechanism $\mathcal{M}_\varepsilon(x) = x + \mathcal{N}(0, \sigma^2)$ satisfies (ε, δ)-PAC LDP for $x \in [0, 1]$ if σ is defined as*

$$\sigma \geq \frac{\sqrt{2}}{2}\left(\frac{\sqrt{\ln(2/\delta) + \varepsilon} + \sqrt{\ln(2/\delta)}}{\varepsilon}\right).$$

Proof. (Sketch) The proof generally follows the structure of Dwork et al [18], but uses a rewritten privacy loss and a different tail bound for the Gaussian distribution. Appendix B.2 provides the details and compares this bound with others in the literature. □

The above theorem applies to a single dimension. For a d-dimensional data, we can apply the Gaussian mechanism to each dimension with the same σ. Then the combination result from Theorem 2 provides a PAC privacy guarantee.

Refined utility quantification. Given a classifier h, combined Gaussian mechanism $\mathcal{M}^d$ with δ, and d-dimensional sensitive data x, the utility quantification $\rho(\varepsilon, \theta_\diamond)$ can be refined to: *With probability at least $\rho(\varepsilon, \theta_\diamond)$, the classifier h preserves the correct classification under the input perturbed by $\mathcal{M}^d$, which guarantees $(d\varepsilon, 1-(1-\delta)^d)$-PAC LDP.*

The extended Gaussian mechanism and the privacy indicator provide two approaches to achieve (ε, δ)-PAC LDP. While both operate under the same privacy notion, the Gaussian mechanism actually provides a stronger effective privacy: when it fails to satisfy the pure ε-LDP (with probability δ), the privacy loss exceeds ε but remains bounded by a larger (unknown) value. In contrast, the privacy indicator exposes the original data when it fails.

6 Discussions

This section discusses detailed privacy questions in the utility quantification framework and summarizes other discussions that are deferred to the appendix.

Detailed Privacy Discussions. In the LDP paradigm, it is assumed that the classifier is curious about users' sensitive data but honest in reporting results, i.e. it outputs $h(\mathcal{M}(x))$ rather than fabricating results. This is a weak assumption, as any utility evaluation becomes meaningless if the classifier fabricates results.

Utility quantification, as a different scenario from *using* the classifier, focuses on a specific input x but does not involve interaction with users' data. It can be conducted either by the classifier designer or by the users, and in both cases, there is no privacy leakage for the users. (i) If conducted by the classifier designer, e.g. for improving design, there is no interaction with users, and thus no privacy concerns arise. (ii) If conducted by the users, e.g. to achieve a better privacy-utility trade-off for their data, they can obtain the robustness radii without revealing their original data. For white-box classifiers, users know the classifier parameters and can compute the robustness radii. For black-box classifiers, users can uniformly query the classifier over the entire input domain (i.e. prediction for nearly all input $x \in [0, 1]^d$), without revealing their specific data. This one-time procedure yields robustness radii for all inputs and privacy parameters. Thus, the robustness radius at any input can be considered public knowledge, which is a reasonable assumption.

Other Discussions. For brevity, additional discussions are deferred to Appendix C.5, including the following topics:

(1) Complexity of finding the robustness radius.
(2) Per-instance data utility vs aggregated data utility.
(3) Fixed classifier with noisy input vs retrained classifier with noisy data.
(4) Robustness of LDP mechanisms.
(5) Independent vs correlated LDP mechanisms.

They provide additional insights into potential questions that may arise related to the utility quantification framework.

7 Case Studies

This section presents case studies of our utility quantification framework. We use representative LDP mechanisms and classifiers as examples, though the proposed framework is generalizable to any LDP mechanism and classifier type.

7.1 Setup

We use the following datasets to train three types of classifiers.

- Stroke Prediction [**?**]: Predicts stroke likelihood based on features like age, hypertension, and BMI. This dataset contains 6 numerical input dimensions and a binary output. The sensitive features are "Age" and "BMI".
- Bank Customer Attrition [**?**]: Predicts whether a bank customer will leave based on features like age, salary, and tenure. This dataset has 9 numerical input dimensions and a binary output. The sensitive features are "Age" and "Estimated Salary".
- MNIST-7×7 (variant of [12]): Predicts handwritten digits (0 to 9) based on pixel values. This dataset has 49 input dimensions and a 10-class output. All dimensions are treated as sensitive features.

The first two datasets involve direct sensitive user inputs, while the third is a standard image classification dataset. For the first two datasets, we use the `sklearn` library with cross-validation to train Logistic Regression and Random Forest classifiers. These traditional classifiers are widely used in medical and financial domains due to their interpretability and lower data requirements [2]. For the third dataset, which consists of high-dimensional images, we use the `torch` library to train a Convolutional Neural Network (CNN) classifier. All datasets are normalized to $[0, 1]$ for training, as normalization aids classifier convergence and is a common practice.

Given a trained classifier, users apply LDP mechanisms to sensitive features (e.g. age, salary, or specific parts of an image) before sending their data to the classifier for prediction. We consider typical LDP mechanisms in literature, which are summarized in Table 1, along with the classifiers used.

We present theoretical utility quantification for the classifiers under the LDP mechanisms discussed above. This quantification is based on the classifier's robustness hyperrectangle $\theta_\diamond$ (at a record), determined using the search method described in Section 5.1. We also compare the theoretical results with empirical utility observations. Specifically, for a given classifier and mechanism, we provide and compare:

Table 1: Summary of LDP mechanisms, types of classifiers, and datasets used in the case studies.

Mechanism	Laplace [18], Gaussian (Theorem 4), PM [60], SW [41], Exponential [52], k-RR [33]
Classifier	Low-dim: Logistic Regression, Random Forest; High-dim: Neural Network
Dataset	Low-dim: Stroke Prediction [**?**], Bank Customer Attrition [**?**]; High-dim: MNIST-7×7 (variant of [12])

- *Theoretical* $\rho(\varepsilon, \theta_\diamond)$: Derived using our utility quantification framework.
- *Empirical* $\hat{\rho}(\varepsilon)$: Estimated by sampling $n = 2000$ instances $\{\tilde{x}\}_n$ from the LDP mechanism and testing the classifier for each ε.

The theoretical $\rho(\varepsilon, \theta_\diamond)$ is a closed-form expression that can be directly computed from the mechanism's parameters for any ε. It serves as a lower bound of the actual robustness probability and is generally smaller than $\hat{\rho}(\varepsilon)$. A smaller gap indicates more accurate theoretical utility quantification.

Without loss of generality, given the input data distribution P_x determined by each dataset, we can also provide average-case and worst-case utility quantification: $\rho_{\text{avg}}(\varepsilon)$, $\rho_{\text{wor}}(\varepsilon)$ and their empirical counterparts $\hat{\rho}_{\text{avg}}(\varepsilon)$, $\hat{\rho}_{\text{wor}}(\varepsilon)$.

7.2 Utility Quantification Results

We present the utility quantification results for the classifiers under input perturbation by different LDP mechanisms.

7.2.1 Stroke Prediction. We begin by considering pure LDP for this medical dataset. For a randomly selected record "Age: 79, BMI: 24, Hypertension: 1, …", we analyze theoretical and empirical utility for the classifiers under the Laplace, PM, and Exponential mechanisms. Additional records and mechanisms can be found in Appendix C.6.1.

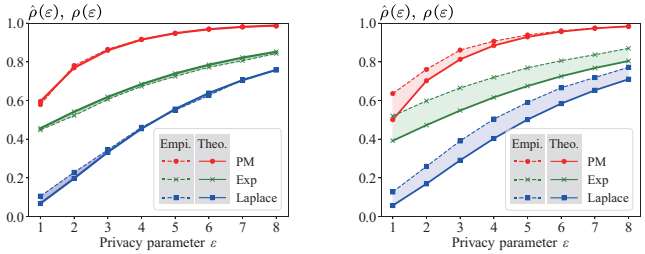
Logistic Regression. The robustness hyperrectangle at the given record is $\theta_\diamond = [0.63, 1] \times [0, 1]$ for this classifier. From this information, the utility guarantee under the PM mechanism can be directly derived as follows:

For this trained Logistic Regression classifier on the Stroke Prediction dataset, at the record "Age: 79, Hypertension: 1, …, BMI: 24", with probability at least $\rho(\varepsilon, \theta_\diamond)$, the classifier preserves the correct prediction under the PM mechanism applied to "Age" and "BMI" (i.e. 2ε-LDP), where

$$\begin{aligned}\rho(\varepsilon, \theta_\diamond) =& [F_{\text{PM}_\varepsilon,\text{Age}}(1) - F_{\text{PM}_\varepsilon,\text{Age}}(0.63)] \cdot [F_{\text{PM}_\varepsilon,\text{BMI}}(1) - F_{\text{PM}_\varepsilon,\text{BMI}}(0)] \\ =& 1 - F_{\text{PM}_\varepsilon,\text{Age}}(0.63).\end{aligned}$$

The last equality holds because the CDF of the PM mechanism is 0 at 0, and 1 at 1. Term $F_{\text{PM}_\varepsilon,\text{Age}}(0.63)$ is the CDF of the PM mechanism (applied to the feature "Age") at 0.63, which can be computed from the PM mechanism's parameters, as detailed in Appendix C.6.2.

Random Forest. For this classifier, the robustness hyperrectangle at the same record is $\theta_\diamond = [0.50, 1] \times [0, 0.62]$. Based on this information, the utility guarantee under the PM mechanism can be derived as follows:



(a) Logistic Regression. (b) Random Forest.

Figure 6: Empirical and theoretical utility for two classifiers trained on the Stroke Prediction dataset.

For this trained Random Forest classifier on the Stroke Prediction dataset, at the record "Age: 79, Hypertension: 1, …, BMI: 24", with probability at least $\rho(\varepsilon, \theta_\diamond)$, the classifier preserves the correct prediction under the PM mechanism applied to "Age" and "BMI" (i.e. 2ε-LDP), where

$$\rho(\varepsilon, \theta_\diamond) = (1 - F_{PM_{\varepsilon,Age}}(0.5)) \cdot F_{PM_{\varepsilon,BMI}}(0.62).$$

Figure 6 compares the theoretical utility $\rho(\varepsilon, \theta_\diamond)$ with the empirical utility $\hat{\rho}(\varepsilon)$ for ε values ranging from 1 to 8, applied to each sensitive feature. Both utilities increase with ε, with the PM mechanism consistently providing the highest utility. This observation aligns with our takeaway that PM is the best choice for moderate ε and θ values. Similarly, the Exponential mechanism outperforms the Laplace mechanism. For both classifiers, the theoretical utility $\rho(\varepsilon, \theta_\diamond)$ closely matches the empirical utility $\hat{\rho}(\varepsilon)$, particularly for the Logistic Regression classifier, showing the accuracy of our theoretical utility.

The Random Forest classifier exhibits a larger gap between $\rho(\varepsilon, \theta_\diamond)$ and $\hat{\rho}(\varepsilon)$ compared to the Logistic Regression classifier. This discrepancy arises from the robustness hyperrectangle $\theta_\diamond$, which is a conservative estimate of the overall robustness region (difficult to determine precisely). For Logistic Regression, the decision boundary is a hyperplane [54], making it more predictable and easier to approximate a hyperrectangle, resulting in a more accurate $\rho(\varepsilon, \theta_\diamond)$. Appendix C.6.3 provides additional details on decision boundaries.

Time cost. In addition to a sampling-independent theoretical guarantee, our utility quantification also has an extremely low time cost. It does not require any sampling from the LDP mechanism, the utility statement can be directly computed for any ε from the mechanism's parameters and the robustness hyperrectangle $\theta_\diamond$. Therefore, for mechanisms having high sampling complexity, e.g. the Exponential mechanism ($O(m)$ with m as the domain size), the theoretical utility quantification is much more efficient than the empirical one.

Table 2 shows the average time of computing one $\rho(\varepsilon, \theta_\diamond)$ and $\hat{\rho}(\varepsilon)$ for different mechanisms in this case study. For each empirical result $\hat{\rho}(\varepsilon)$, we sample 2000 times from the mechanism and feed the samples to the classifier for inference. The theoretical approach is significantly faster, especially for the Exponential mechanism. For this mechanism, computing one empirical $\hat{\rho}(\varepsilon)$ requires 859.83 milliseconds for 2000 samples, whereas computing one theoretical $\rho(\varepsilon, \theta_\diamond)$ takes only 0.94 milliseconds, which is less than 0.1%. The

Table 2: Time cost comparison (in milliseconds).

	PM	Exponential	Laplace
Empirical[a]	6.56 + 1.38	859.83 + 1.53	11.29 + 1.53
Theoretical[b]	0.24	0.94	0.30

[a] Time for generating 2000 samples + running inference.
[b] Time to compute $\rho(\varepsilon, \theta_\diamond)$ only; computing $\theta_\diamond$ takes 5.50 ms but is a one-time cost amortized across all ε values.

efficiency stems from directly substituting the privacy parameter ε and the robustness hyperrectangle $\theta_\diamond$ into the closed-form expression of $\rho(\varepsilon, \theta_\diamond)$. Moreover, although computing $\rho(\varepsilon, \theta_\diamond)$ requires the robustness hyperrectangle, $\theta_\diamond$ needs to be computed only once and can then be reused for all ε values. In this case, computing $\theta_\diamond$ takes only 5.50 milliseconds, which is still more efficient than estimating the empirical utility $\hat{\rho}(\varepsilon)$ separately for each ε.

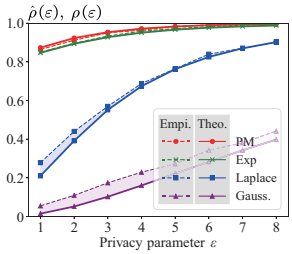
Average-case and worst-case analysis. All data points in the dataset collectively define a data distribution P_x over the sensitive features. Therefore, we evaluate the utility at each $\{\text{Age, BMI}\} \sim P_x$, then compute the mean for the average-case utility and the minimum for the worst-case utility. This approach yields average-case and worst-case utility guarantees that account for the input data distribution. Comprehensive results are provided in Appendix C.6.4.

7.2.2 Bank Customer Attrition. We now evaluate PAC LDP for this financial dataset. For a randomly selected record "Age: 22, Estimated Salary: 101,348, Credit Score: 619, …", we analyze both theoretical and empirical utility for the two classifiers under PAC LDP mechanisms. Specifically, we apply the privacy indicator $\mathcal{I}_\delta$ to the Laplace, PM, and Exponential mechanisms, while also including the Gaussian mechanism. The failure probability is fixed at $\delta = 0.1$ for all mechanisms. Detailed utility quantification statements for the privacy indicator and Gaussian mechanism under PAC LDP are provided in Appendix C.7.1.

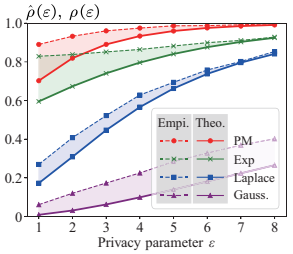
Figure 7 compares $\rho(\varepsilon, \theta_\diamond)$ and $\hat{\rho}(\varepsilon)$ for ε values from 1 to 8 assigned to each sensitive feature. The results show a similar trend to the Stroke Prediction dataset, with the Exponential mechanism performing much better in this case, while the Gaussian mechanism lags behind the others.

Two key observations emerge in this case study: (i) The Exponential mechanism performs comparably to the PM mechanism for Logistic Regression. This is because its robustness hyperrectangle $\theta_\diamond = [0, 0.72] \times [0, 1]$ is large enough to cover most of the sensitive feature ranges, enabling excellent performance for all mechanisms defined on $[0, 1]$. Figure 4 illustrates this phenomenon: larger θ values lead to $\rho(\varepsilon, \theta)$ approaching 1 for all such mechanisms. However, Laplace and Gaussian mechanisms are less effective here due to their probability mass extending beyond $[0, 1]$. (ii) The Gaussian mechanism is not as good as the other mechanisms. The main reason is from the noise scale σ in Theorem 4. Although we can extend the Gaussian mechanism to $\varepsilon \geq 1$, this leads to a larger σ, i.e. less concentrated.

Average-case and worst-case analysis. We can similarly provide these utility guarantees as done for the Stroke Prediction dataset. Detailed results are presented in Appendix C.7.2.

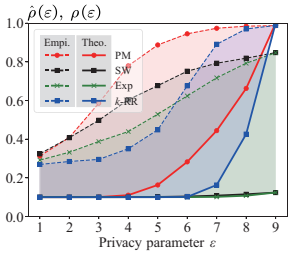


(a) Logistic Regression.

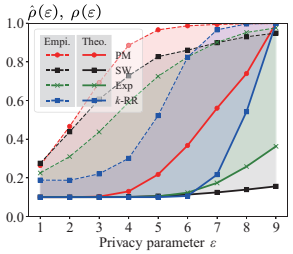


(b) Random Forest.

Figure 7: Empirical and theoretical utility for two classifiers trained on the Bank Customer Attrition dataset.



(a) Neural Network on the first image in the dataset.



(b) Neural Network on the second image in the dataset.

Figure 8: Empirical and theoretical utility for a Neural Network classifier trained on the MNIST-7×7 dataset.

7.2.3 MNIST-7×7. We evaluate both theoretical and empirical utility under the PM, SW, Exponential, and k-RR mechanisms for the first two images in this dataset, labeled as "digit 5" and "digit 0." Consistent with the setup in previous case studies, we focus on PAC LDP by applying the privacy indicator $\mathcal{I}_\delta$ to these mechanisms, with the failure probability fixed at $\delta = 0.1$. The Laplace and Gaussian mechanisms are excluded, as they are prone to perturbing pixel values outside the valid range $[0, 1]$ in this high-dimensional dataset, making fair comparisons with other mechanisms difficult.

Figure 8 compares $\rho(\varepsilon, \theta_\diamond)$ and $\hat{\rho}(\varepsilon)$ for ε values from 1 to 8 assigned to each pixel. Larger gaps between $\rho(\varepsilon, \theta_\diamond)$ and $\hat{\rho}(\varepsilon)$ are observed in this high-dimensional Neural Network classifier, especially for smaller ε values. Even for the best mechanism, PM, the theoretical utility only starts to increase significantly when $\varepsilon > 4$, with an exponential growth rate. Similarly, the theoretical utility of the k-RR mechanism shows noticeable improvement when $\varepsilon > 6$. Despite the larger gap, the trends between theoretical and empirical results remain consistent: mechanisms with higher theoretical utility also exhibit higher empirical utility.

Reasons for the results. The larger gap between $\rho(\varepsilon, \theta_\diamond)$ and $\hat{\rho}(\varepsilon)$ in the Neural Network classifier compared to traditional classifiers arises from two factors: (i) The complex decision boundary of the Neural Network, which is difficult to approximate using a single hyperrectangle. (ii) Error accumulation in high-dimensional data, which is more pronounced than in low-dimensional data. Specifically, the Neural Network classifier partitions the high-dimensional space into intricate regions (one for each digit class), making it challenging to approximate with a single hyperrectangle around a digit. Additionally, theoretical utility is derived from the product of utility analyses for each pixel, making it more sensitive to error accumulation in high-dimensional data. In contrast, empirical utility is unaffected by the robustness hyperrectangle $\theta_\diamond$, avoiding such error accumulation.

Improvements for high-dimensional classifiers. (i) In practice, not all dimensions are sensitive and require privacy protection. While we treat all dimensions as sensitive in the MNIST-7×7 dataset, users can opt to apply LDP mechanisms to only a subset of dimensions, i.e. specific parts of the image. In such cases, the theoretical utility quantification can be more accurate. (ii) At present, we identify a connected (local) robustness rectangle that contains x; however, the classifier may admit additional disconnected and non-rectangular robust regions. One can use Monte Carlo sampling to discover such regions and then combine this with Monte Carlo integration to obtain a more accurate theoretical utility quantification for high-dimensional classifiers. Appendix C.8.1 provides details.

Average-case and worst-case analysis. We use the first 50 images from the MNIST-7×7 dataset as representative samples for analyzing the average-case and worst-case utility under LDP-perturbed inputs. Appendix C.8.2 presents the results of this analysis.

7.3 Summary of Case Studies

We presented case studies of theoretical utility statements for different classifiers under inputs perturbed by typical LDP mechanisms and compared them with empirical utility. The key findings from our case studies are as follows:

- Theoretical utility quantification is computationally efficient, requiring negligible time due to its closed-form expression for any ε.
- For low-dimensional classifiers, theoretical utility often closely matches empirical utility. For high-dimensional classifiers, theoretical utility has a larger gap with empirical utility, due to complex decision boundaries and error accumulation. However, mechanisms with higher theoretical utility also exhibit higher empirical utility.
- The PM mechanism generally outperforms other mechanisms, providing the best utility across our case studies.

8 Conclusions

This paper introduces a framework for quantifying classifier utility under LDP-perturbed inputs. The framework connects the concentration analysis of LDP mechanisms with the robustness analysis of classifiers. It provides guidelines for selecting the best mechanism and privacy parameter for a given classifier. Beyond the core framework, we introduce two novel refinement techniques that further improve utility quantification. Case studies on typical classifiers under LDP-perturbed inputs demonstrate the framework's effectiveness and applicability.

Future work: (i) The robustness is not limited to classifiers; it can be extended to robustness of programs and control systems. Connecting differential privacy with them is a promising direction. (ii) This paper analyzes data dimensions independently. Extending the utility quantification to multidimensional LDP mechanisms presents another direction for future work.

Acknowledgments

We thank the anonymous reviewers and the revision editor for their valuable feedback and guidance, which significantly improved this paper. We also acknowledge the use of GPT-5 for language refinement in this paper. This research is supported in part by the U.S. National Science Foundation under grants CNS-2245689 (CRII), as well as by the 2022 Meta Research Award for Privacy-Enhancing Technologies.

References

[1] Martín Abadi, Andy Chu, Ian J. 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, October 24-28, 2016*, Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi (Eds.). ACM, 308–318. https://doi.org/10.1145/2976749.2978318

[2] Nasrin Amini, Mahdi Mahdavi, Hadi Choubdar, Atefeh Abedini, Ahmad Shalbaf, and Reza Lashgari. 2023. Automated prediction of COVID-19 mortality outcome using clinical and laboratory data based on hierarchical feature selection and random forest classifier. *Computer Methods in Biomechanics and Biomedical Engineering* 26, 2 (2023), 160–173. https://doi.org/10.1080/10255842.2022.2050906 arXiv:https://doi.org/10.1080/10255842.2022.2050906 PMID: 35297747.

[3] Héber Hwang Arcolezi, Karima Makhlouf, and Catuscia Palamidessi. 2023. (Local) Differential Privacy has NO Disparate Impact on Fairness. In *Data and Applications Security and Privacy XXXVII - 37th Annual IFIP WG 11.3 Conference, DBSec 2023, Sophia-Antipolis, France, July 19-21, 2023, Proceedings (Lecture Notes in Computer Science, Vol. 13942)*, Vijayalakshmi Atluri and Anna Lisa Ferrara (Eds.). Springer, 3–21. https://doi.org/10.1007/978-3-031-37586-6_1

[67]]lda-qda Sergio Bacallado. [n. d.]. Lecture 9: Classification, LDA, QDA. https://web.stanford.edu/~lmackey/stats202/content/lec9.pdf

[5] Borja Balle and Yu-Xiang Wang. 2018. Improving the Gaussian Mechanism for Differential Privacy: Analytical Calibration and Optimal Denoising. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018 (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer G. Dy and Andreas Krause (Eds.). PMLR, 403–412. http://proceedings.mlr.press/v80/balle18a.html

[6] Ridha Ilyas Bendjillali, Mohammed Beladgham, Khaled Merit, and Abdelmalik Taleb-Ahmed. 2020. Illumination-robust face recognition based on deep convolutional neural networks architectures. *Indonesian Journal of Electrical Engineering and Computer Science* 18, 2 (2020), 1015–1027. https://doi.org/10.11591/ijeecs.v18.i2.pp1015-1027

[7] Clément L. Canonne, Gautam Kamath, and Thomas Steinke. 2020. The Discrete Gaussian for Differential Privacy. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/b53b3a3d6ab90ce0268229151c9bde11-Abstract.html

[8] Hongge Chen, Huan Zhang, Si Si, Yang Li, Duane S. Boning, and Cho-Jui Hsieh. 2019. Robustness Verification of Tree-based Models. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 12317–12328. https://proceedings.neurips.cc/paper/2019/hash/cd9508fdaa5c1390e9cc329001cf1459-Abstract.html

[9] Keke Chen and Ling Liu. 2005. Privacy Preserving Data Classification with Rotation Perturbation. In *Proceedings of the 5th IEEE International Conference on Data Mining ICDM 2005, 27-30 November 2005, Houston, Texas, USA*. IEEE Computer Society, 589–592. https://doi.org/10.1109/ICDM.2005.121

[10] Albert Cheu, Adam D. Smith, Jonathan R. Ullman, David Zeber, and Maxim Zhilyaev. 2019. Distributed Differential Privacy via Shuffling. In *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 11476)*, Yuval Ishai and Vincent Rijmen (Eds.). Springer, 375–403. https://doi.org/10.1007/978-3-030-17653-2_13

[11] William L. Croft, Jörg-Rüdiger Sack, and Wei Shi. 2022. Differential Privacy via a Truncated and Normalized Laplace Mechanism. *J. Comput. Sci. Technol.* 37, 2 (2022), 369–388. https://doi.org/10.1007/S11390-020-0193-Z

[12] Li Deng. 2012. The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]. *IEEE Signal Process. Mag.* 29, 6 (2012), 141–142. https://doi.org/10.1109/MSP.2012.2211477

[13] Samuel Fuller Dodge and Lina J. Karam. 2016. Understanding how image quality affects deep neural networks. In *Eighth International Conference on Quality of Multimedia Experience, QoMEX 2016, Lisbon, Portugal, June 6-8, 2016*. IEEE, 1–6. https://doi.org/10.1109/QOMEX.2016.7498955

[14] Minxin Du, Xiang Yue, Sherman S. M. Chow, and Huan Sun. 2023. Sanitizing Sentence Embeddings (and Labels) for Local Differential Privacy. In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, Ying Ding, Jie Tang, Juan F. Sequeda, Lora Aroyo, Carlos Castillo, and Geert-Jan Houben (Eds.). ACM, 2349–2359. https://doi.org/10.1145/3543507.3583512

[15] D.P. Dubhashi and A. Panconesi. 2009. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press. https://books.google.com/books?id=UUohAwAAQBAJ

[16] John C. Duchi, Martin J. Wainwright, and Michael I. Jordan. 2016. Minimax Optimal Procedures for Locally Private Estimation. *CoRR* abs/1604.02390 (2016). arXiv:1604.02390 http://arxiv.org/abs/1604.02390

[17] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel. 2012. Fairness through awareness. In *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*, Shafi Goldwasser (Ed.). ACM, 214–226. https://doi.org/10.1145/2090236.2090255

[18] Cynthia Dwork and Aaron Roth. 2014. The Algorithmic Foundations of Differential Privacy. *Found. Trends Theor. Comput. Sci.* 9, 3-4 (2014), 211–407. https://doi.org/10.1561/0400000042

[19] Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. 2016. Robustness of classifiers: from adversarial to random noise. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (Eds.). 1624–1632. https://proceedings.neurips.cc/paper/2016/hash/7ce3284b743aefde80ffd9aec500e085-Abstract.html

[67]]stroke-dataset Fedesoriano. [n. d.]. Stroke Prediction Dataset. https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset

[21] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon M. Lin, David Page, and Thomas Ristenpart. 2014. Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing. In *Proceedings of the 23rd USENIX Security Symposium, San Diego, CA, USA, August 20-22, 2014*, Kevin Fu and Jaeyeon Jung (Eds.). USENIX Association, 17–32. https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/fredrikson_matthew

[22] Jie Fu, Yuan Hong, Xinpeng Ling, Leixia Wang, Xun Ran, Zhiyu Sun, Wendy Hui Wang, Zhili Chen, and Yang Cao. 2024. Differentially Private Federated Learning: A Systematic Review. *CoRR* abs/2405.08299 (2024). https://doi.org/10.48550/ARXIV.2405.08299 arXiv:2405.08299

[23] Chang Ge, Xi He, Ihab F. Ilyas, and Ashwin Machanavajjhala. 2019. APEx: Accuracy-Aware Differentially Private Data Exploration. In *Proceedings of the 2019 International Conference on Management of Data, SIGMOD Conference 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019*, Peter A. Boncz, Stefan Manegold, Anastasia Ailamaki, Amol Deshpande, and Tim Kraska (Eds.). ACM, 177–194. https://doi.org/10.1145/3299869.3300092

[24] Quan Geng, Wei Ding, Ruiqi Guo, and Sanjiv Kumar. 2020. Tight Analysis of Privacy and Utility Tradeoff in Approximate Differential Privacy. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy] (Proceedings of Machine Learning Research, Vol. 108)*, Silvia Chiappa and Roberto Calandra (Eds.). PMLR, 89–99. http://proceedings.mlr.press/v108/geng20a.html

[25] Benyamin Ghojogh and Mark Crowley. 2019. Linear and Quadratic Discriminant Analysis: Tutorial. *CoRR* abs/1906.02590 (2019). arXiv:1906.02590 http://arxiv.org/abs/1906.02590

[26] Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. 2009. Universally utility-maximizing privacy mechanisms. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, Michael Mitzenmacher (Ed.). ACM, 351–360. https://doi.org/10.1145/1536414.1536464

[67]]googledifferential-privacy-2026 Google. [n. d.]. *google/differential-privacy*. https://github.com/google/differential-privacy original-date: 2019-09-04T13:04:15Z.

[28] Xiaolan Gu, Ming Li, Yueqiang Cheng, Li Xiong, and Yang Cao. 2020. PCKV: Locally Differentially Private Correlated Key-Value Data Collection with Optimized Utility. In *29th USENIX Security Symposium, USENIX Security 2020, August 12-14, 2020*, Srdjan Capkun and Franziska Roesner (Eds.). USENIX Association, 967–984. https://www.usenix.org/conference/usenixsecurity20/presentation/gu

[67]]GMM Mark Hasegawa-Johnson. [n. d.]. ECE 417 Multimedia Signal Processing. https://courses.grainger.illinois.edu/ece417/fa2021/lectures/lec11.pdf

[30] Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition*. Springer. https://doi.org/10.1007/978-0-387-84858-7

[31] Wassily Hoeffding. 1963. Probability Inequalities for Sums of Bounded Random Variables. *J. Amer. Statist. Assoc.* 58, 301 (1963), 13–30. http://www.jstor.org/stable/2282952

[32] Anil K. Jain, M. Narasimha Murty, and Patrick J. Flynn. 1999. Data Clustering: A Review. *ACM Comput. Surv.* 31, 3 (1999), 264–323. https://doi.org/10.1145/331499.331504

[33] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. 2014. Extremal Mechanisms for Local Differential Privacy. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger (Eds.). 2879–2887. https://proceedings.neurips.cc/paper/2014/hash/86df7dcfd896fcaf2674f757a2463eba-Abstract.html

[34] Can Kanbak, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. 2018. Geometric Robustness of Deep Networks: Analysis and Improvement. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. Computer Vision Foundation / IEEE Computer Society, 4441–4449. https://doi.org/10.1109/CVPR.2018.00467

[35] Guy Katz, Clark W. Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. 2017. Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks. In *Computer Aided Verification - 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 10426)*, Rupak Majumdar and Viktor Kuncak (Eds.). Springer, 97–117. https://doi.org/10.1007/978-3-319-63387-9_5

[36] Sosuke Kobayashi. 2018. Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, Marilyn A. Walker, Heng Ji, and Amanda Stent (Eds.). Association for Computational Linguistics, 452–457. https://doi.org/10.18653/V1/N18-2072

[37] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura E. Barnes, and Donald E. Brown. 2019. Text Classification Algorithms: A Survey. *Inf.* 10, 4 (2019), 150. https://doi.org/10.3390/INFO10040150

[38] Jinyu Li, Li Deng, Yifan Gong, and Reinhold Haeb-Umbach. 2014. An Overview of Noise-Robust Automatic Speech Recognition. *IEEE ACM Trans. Audio Speech Lang. Process.* 22, 4 (2014), 745–777. https://doi.org/10.1109/TASLP.2014.2304637

[39] Xiaoguang Li, Ninghui Li, Wenhai Sun, Neil Zhenqiang Gong, and Hui Li. 2023. Fine-grained Poisoning Attack to Local Differential Privacy Protocols for Mean and Variance Estimation. In *32nd USENIX Security Symposium, USENIX Security 2023, Anaheim, CA, USA, August 9-11, 2023*, Joseph A. Calandrino and Carmela Troncoso (Eds.). USENIX Association, 1739–1756. https://www.usenix.org/conference/usenixsecurity23/presentation/li-xiaoguang

[40] Xiaoguang Li, Zitao Li, Ninghui Li, and Wenhai Sun. 2025. On the Robustness of LDP Protocols for Numerical Attributes under Data Poisoning Attacks. 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/ndss-paper/on-the-robustness-of-ldp-protocols-for-numerical-attributes-under-data-poisoning-attacks/

[41] Zitao Li, Tianhao Wang, Milan Lopuhaä-Zwakenberg, Ninghui Li, and Boris Skoric. 2020. Estimating Numerical Distributions under Local Differential Privacy. In *Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020*, David Maier, Rachel Pottinger, AnHai Doan, Wang-Chiew Tan, Abdussalam Alawini, and Hung Q. Ngo (Eds.). ACM, 621–635. https://doi.org/10.1145/3318464.3389700

[42] D. Lu and Q. Weng. 2007. A survey of image classification methods and techniques for improving classification performance. *International Journal of Remote Sensing* 28, 5 (2007), 823–870. https://doi.org/10.1080/01431160600746456

[43] Gabriel Resende Machado, Eugênio Silva, and Ronaldo Ribeiro Goldschmidt. 2023. Adversarial Machine Learning in Image Classification: A Survey Toward the Defender's Perspective. *ACM Comput. Surv.* 55, 2 (2023), 8:1–8:38. https://doi.org/10.1145/3485133

[44] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. 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=rJzIBfZAb

[45] Karima Makhlouf, Tamara Stefanovic, Héber Hwang Arcolezi, and Catuscia Palamidessi. 2024. A Systematic and Formal Study of the Impact of Local Differential Privacy on Fairness: Preliminary Results. In *37th IEEE Computer Security Foundations Symposium, CSF 2024, Enschede, Netherlands, July 8-12, 2024*. IEEE, 1–16. https://doi.org/10.1109/CSF61375.2024.00039

[67]]bank-attrition Sagar Maru. [n. d.]. Bank Customer Attrition Insights. https://www.kaggle.com/datasets/marusagar/bank-customer-attrition-insights

[47] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera 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

[48] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. 2018. Learning Differentially Private Recurrent Language Models. In *International Conference on Learning Representations*. https://openreview.net/forum?id=BJ0hF1Z0b

[49] Mark Huasong Meng, Guangdong Bai, Sin Gee Teo, Zhe Hou, Yan Xiao, Yun Lin, and Jin Song Dong. 2022. Adversarial Robustness of Deep Neural Networks: A Survey from a Formal Verification Perspective. *CoRR* abs/2206.12227 (2022). https://doi.org/10.48550/ARXIV.2206.12227 arXiv:2206.12227

[50] Ilya Mironov. 2017. Rényi Differential Privacy. In *30th IEEE Computer Security Foundations Symposium, CSF 2017, Santa Barbara, CA, USA, August 21-25, 2017*. IEEE Computer Society, 263–275. https://doi.org/10.1109/CSF.2017.11

[51] Jeet Mohapatra, Tsui-Wei Weng, Pin-Yu Chen, Sijia Liu, and Luca Daniel. 2020. Towards Verifying Robustness of Neural Networks Against A Family of Semantic Perturbations. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, 241–249. https://doi.org/10.1109/CVPR42600.2020.00032

[52] Kobbi Nissim, Rann Smorodinsky, and Moshe Tennenholtz. 2012. Approximately optimal mechanism design via differential privacy. In *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*, Shafi Goldwasser (Ed.). ACM, 203–213. https://doi.org/10.1145/2090236.2090254

[53] Luis Perez and Jason Wang. 2017. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *CoRR* abs/1712.04621 (2017). arXiv:1712.04621 http://arxiv.org/abs/1712.04621

[54] Suchismita Sahu. 2024. Decision Boundary for Classifiers: An Introduction. https://medium.com/analytics-vidhya/decision-boundary-for-classifiers-an-introduction-cc67c6d3da0e

[55] Shafizur Rahman Seeam, Ye Zheng, and Yidan Hu. 2025. Frequency Estimation of Correlated Multi-attribute Data under Local Differential Privacy. *CoRR* abs/2507.17516 (2025). https://doi.org/10.48550/ARXIV.2507.17516 arXiv:2507.17516

[56] 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 2017, San Jose, CA, USA, May 22-26, 2017*. IEEE Computer Society, 3–18. https://doi.org/10.1109/SP.2017.41

[57] R. Tempo, E.W. Bai, and F. Dabbene. 1996. Probabilistic robustness analysis: explicit bounds for the minimum number of samples. In *Proceedings of 35th IEEE Conference on Decision and Control*, Vol. 3. 3424–3428 vol.3. https://doi.org/10.1109/CDC.1996.573690

[58] Elisabet Lobo Vesga, Alejandro Russo, and Marco Gaboardi. 2020. A Programming Framework for Differential Privacy with Accuracy Concentration Bounds. In *2020 IEEE Symposium on Security and Privacy, SP 2020, San Francisco, CA, USA, May 18-21, 2020*. IEEE, 411–428. https://doi.org/10.1109/SP40000.2020.00086

[59] Sajani Vithana, Viveck R. Cadambe, Flávio P. Calmon, and Haewon Jeong. 2025. Correlated Privacy Mechanisms for Differentially Private Distributed Mean Estimation. In *IEEE Conference on Secure and Trustworthy Machine Learning, SaTML 2025, Copenhagen, Denmark, April 9-11, 2025*. IEEE, 590–614. https://doi.org/10.1109/SATML64287.2025.00039

[60] Ning Wang, Xiaokui Xiao, Yin Yang, Jun Zhao, Siu Cheung Hui, Hyejin Shin, Junbum Shin, and Ge Yu. 2019. Collecting and Analyzing Multidimensional Data with Local Differential Privacy. In *35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, April 8-11, 2019*. IEEE, 638–649. https://doi.org/10.1109/ICDE.2019.00063

[61] Tianhao Wang, Jeremiah Blocki, Ninghui Li, and Somesh Jha. 2017. Locally Differentially Private Protocols for Frequency Estimation. In *26th USENIX Security Symposium, USENIX Security 2017, Vancouver, BC, Canada, August 16-18, 2017*, Engin Kirda and Thomas Ristenpart (Eds.). USENIX Association, 729–745. https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/wang-tianhao

[62] Tianhao Wang, Ninghui Li, and Somesh Jha. 2018. Locally Differentially Private Frequent Itemset Mining. In *2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA*. IEEE Computer Society, 127–143. https://doi.org/10.1109/SP.2018.00035

[63] Tianhao Wang, Milan Lopuhaä-Zwakenberg, Zitao Li, Boris Skoric, and Ninghui Li. 2020. Locally Differentially Private Frequency Estimation with Consistency. In *27th Annual Network and Distributed System Security Symposium, NDSS 2020, San Diego, California, USA, February 23-26, 2020*. The Internet Society. https://www.ndss-symposium.org/ndss-paper/locally-differentially-private-frequency-estimation-with-consistency/

[64] Jason W. Wei and Kai Zou. 2019. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, 6381–6387. https://doi.org/10.18653/V1/D19-1670

[67]]naive-bayes Kilian Weinberger. [n. d.]. Machine Learning for Intelligent Systems. https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote05.html

[66] Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. 2018. Evaluating the Robustness of Neural Networks: An Extreme Value Theory Approach. 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=BkUHlMZ0b
[67]]wiki-k-means Wikipedia. [n. d.]. *k*-means clustering. https://en.wikipedia.org/w/index.php?title=K-means_clustering&oldid=1258447782
[68] Hanshen Xiao and Srinivas Devadas. 2023. PAC Privacy: Automatic Privacy Measurement and Control of Data Processing. In *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 14082)*, Helena Handschuh and Anna Lysyanskaya (Eds.). Springer, 611–644. https://doi.org/10.1007/978-3-031-38545-2_20
[69] Nan Zhang, Shengquan Wang, and Wei Zhao. 2005. A new scheme on privacy-preserving data classification. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, Illinois, USA, August 21-24, 2005*, Robert Grossman, Roberto J. Bayardo, and Kristin P. Bennett (Eds.). ACM, 374–383. https://doi.org/10.1145/1081870.1081913
[70] Tianle Zhang, Wenjie Ruan, and Jonathan E. Fieldsend. 2022. PRoA: A Probabilistic Robustness Assessment Against Functional Perturbations. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2022, Grenoble, France, September 19-23, 2022, Proceedings, Part III (Lecture Notes in Computer Science, Vol. 13715)*, Massih-Reza Amini, Stéphane Canu, Asja Fischer, Tias Guns, Petra Kralj Novak, and Grigorios Tsoumakas (Eds.). Springer, 154–170. https://doi.org/10.1007/978-3-031-26409-2_10
[71] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. 2023. Domain Generalization: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 45, 4 (2023), 4396–4415. https://doi.org/10.1109/TPAMI.2022.3195549
[72] Ezgi Zorarpaci and Selma Ayse Özel. 2021. Privacy preserving classification over differentially private data. *WIREs Data Mining Knowl. Discov.* 11, 3 (2021). https://doi.org/10.1002/WIDM.1399

A Related Work

This paper studies the data utility of classifiers under perturbed inputs. We first review closely related concepts—semantic perturbations, concentration properties of random variables, and classifier robustness—and clarify how they differ from our setting. We then discuss broader related work.

A.1 Closely Related Concepts

Semantic perturbations. A number of the existing literature focuses on empirical evaluations of data utility under semantic perturbations [51], i.e. perturbations that involve semantic meanings. For image classifiers, such perturbations include brightness adjustments [6], blur [13], and data augmentation such as rotation and scaling [34, 53]. For text classification, examples include synonym replacement and random swap [36, 64], etc. For speech recognition, there are acoustic distortions [38]. These perturbations do not provide formal privacy guarantees, and some of them even preserve the semantic meanings of the original data.

This paper focuses on a specific category of mathematically tractable perturbations, LDP mechanisms. These perturbations, such as the Laplace or Gaussian noise, have weaker semantic meanings but provide provable privacy guarantees against adversaries. Additionally, there are variants of LDP mechanisms tailored for semantic perturbations, such as sentence embedding perturbation [14]. The LDP mechanisms examined in this paper are typical building blocks for numerous LDP protocols in practical applications.

Concentration property of random variables. The term concentration property mostly relates to the *concentration inequalities* in probability theory, e.g. Chernoff bound and Hoeffding inequality (see book [15]). These inequalities provide upper bounds on the probability that the sum of independent random variables deviates from its expected value. From these inequalities, the notion (α, β)-accuracy [10, 15] is defined as: the error is bounded by α with probability at least β. In DP, this notion has been used to analyze the error of the summation query in the shuffle model [10]. It also appears in accuracy analysis of data exploring languages under DP guarantee, such as Haskell and SQL [23, 58].

In comparison to (α, β)-accuracy, the theoretical utility quantification $\rho(\varepsilon, \theta)$ includes the concentration property of LDP mechanisms and the robustness property of classifiers. Among these two properties, the concentration property shares similarities with (α, β)-accuracy by providing a probabilistic error bound for perturbed data, but it specifically focuses on the distribution of an LDP mechanism. Meanwhile, this property is used to connect with the robustness of classifiers in this paper.

Robustness of classifiers. Robustness refers to a classifier's ability to maintain performance (utility) under input perturbations [19]. This property is crucial as classifiers often encounter data that deviates from their training distribution. Robustness also serves as a metric for evaluating a classifier's generalization ability [71]. Beyond semantic perturbations, robustness has been extensively studied under ℓ_p-norm perturbations [19, 43, 44], where input data is perturbed within an ℓ_p-norm ball. Within this perturbation model, various verification techniques have been developed to analyze classifier robustness. In addition to approximating the robustness of a classifier as described in Section 4.2, we can also approximate the classifier using a "smoothed" classifier via randomized smoothing [47] and then analyze the robustness radius of the smoothed classifier.

Unlike these studies, this paper focuses on classifier robustness under LDP perturbations. These perturbations can be viewed as occurring within *probabilistic* balls defined by the concentration property, controlled by the privacy parameter ε.

A.2 Broader Related Work

Impact of (L)DP mechanisms on fairness of classifiers. Fairness dictates that two individuals who are similar w.r.t. a particular task should be classified similarly. Under some metrics of similarity, it relates to the concept of DP [17]. DP mechanisms can impact the fairness of classifiers by changing biases in the input data distribution. For instance, if certain groups are over- or under-represented in the data provided to a classifier, applying DP mechanisms can shift the input distribution and thereby alter the fairness (or bias) of the resulting predictions [3, 45].

DP mechanisms for privacy-preserving classifier training. This paper addresses data privacy when (L)DP mechanisms are applied by users before sharing data with classifiers, i.e. ensuring privacy during classifier usage. An orthogonal line of research examines privacy risks posed by classifiers themselves, focusing on data leakage during training. Such works aim to counter membership inference attacks [21, 56], which infer training data from model parameters. DP-SGD [1] is a well-known defense, applying Gaussian noise to gradient updates during training.

Federated learning [47] is a stricter privacy setting, where sensitive training data is distributed across multiple users and cannot be shared with a central server. In this context, DP-Fed-SGD [48] perturbs local gradients before aggregation to the central server. For a comprehensive overview, see a recent survey [22].

While these approaches focus on protecting training data privacy, this paper examines the utility of trained classifiers under input perturbations caused by LDP mechanisms. In the former case, the classifier is trusted and used for inference. In the latter, the classifier is untrusted, and users perturb their input data before sharing it.

Utility of LDP mechanisms. Data utility is the most crucial metric for LDP mechanisms. For simple queries such as mean, summation, and histogram [60–62], data utility is essentially determined by the theoretical mean squared error (MSE) of the mechanism. While empirical evaluations can be conducted by sampling from the mechanism, the empirical utility will converge to the theoretical MSE when the sample size is large enough. Meanwhile, MSE is also the most common metric in mechanism design and analysis.

However, a mechanism with lower MSE does not always guarantee better utility for all applications, particularly classifiers. This paper has shown that classifier utility depends on both the concentration property of the mechanism and the classifier's robustness. We provide a framework to quantify classifier utility under LDP-perturbed data, offering an alternative to empirical evaluations.

B Proofs

B.1 Proof of Theorem 3

Proof. From the definition of $\mathcal{I}_\delta(\mathcal{M}(x))$, we have

$$\Pr[\mathcal{I}_\delta(\mathcal{M}(x)) = \mathcal{M}(x)] = 1 - \delta.$$

Since $\mathcal{M}(x)$ satisfies ε-LDP, it means the privacy loss $\mathcal{L}_{\mathcal{M},x_1,x_2}(\tilde{x}) \leq \varepsilon$ always holds. Thus, with probability at least $1-\delta$, the $\mathcal{L}_{\mathcal{M},x_1,x_2}(\tilde{x})$ is bounded by ε, i.e.

$$\Pr[\mathcal{L}_{\mathcal{M},x_1,x_2}(\tilde{x}) \leq \varepsilon] \geq 1 - \delta,$$

which proves that $\mathcal{I}_\delta(\mathcal{M}(x))$ satisfies (ε, δ)-PAC LDP. □

B.2 Proof of Theorem 4 and Discussion

B.2.1 Our Gaussian Mechanism. The proof generally follows the structure of Dwork's proof of the Gaussian mechanism on page 261 in [18]. Their proof assumes $\varepsilon \leq 1$ for the soundness of the $\ln(\varepsilon, \Delta f)$ function. We eliminate this limitation by refining the proof technique and providing a new noise bound.

Proof. For the Gaussian distribution and an original data $x_1 \in [0,1]$, the probability of seeing $x_1 + \tilde{x}$ with $\tilde{x} \sim \mathcal{N}(0, \sigma^2)$ is given by the probability density function at $\tilde{x}$, i.e. $pdf[\tilde{x}]$. The probabilities of observing *the same perturbed value* when the original input is $x_2 \in [0,1]$ form the set $\{\, pdf[\tilde{x} + \Delta] : \Delta \in [-1,1] \,\}$. Thus, the privacy loss is

$$\mathcal{L}_{\mathcal{M},x_1,x_2}(\tilde{x}) = \ln\left(\frac{pdf[\tilde{x}]}{pdf[\tilde{x}+\Delta]}\right) = \ln\frac{\exp(-\frac{1}{2\sigma^2}\cdot\tilde{x}^2)}{\exp(-\frac{1}{2\sigma^2}\cdot(\tilde{x}+\Delta)^2)}$$
$$= \frac{1}{2\sigma^2}\cdot\left(2\tilde{x}\Delta + \Delta^2\right) = \frac{\Delta}{\sigma^2}\cdot\tilde{x} + \frac{\Delta^2}{2\sigma^2}.$$

Since $\tilde{x} \sim \mathcal{N}(0, \sigma^2)$, the privacy loss is distributed as a Gaussian random variable with mean $\Delta^2/(2\sigma^2)$ and variance $(\Delta/\sigma)^2 \cdot \sigma^2 = \Delta^2/\sigma^2$. We can see that the Gaussian mechanism can not satisfy pure ε-LDP for any ε because $\tilde{x} \in (-\infty, \infty)$, meaning the privacy loss is unbounded.

Nonetheless, the privacy loss random variable can be bounded by ε with a certain probability $1-\delta$. Specifically, letting $Z \sim \mathcal{N}(0,1)$, the privacy loss can be rewritten as a random variable:

$$\mathcal{L}_{\mathcal{M},x_1,x_2}(\tilde{x}) \sim \frac{\Delta}{\sigma}\cdot Z + \frac{\Delta^2}{2\sigma^2}.$$

In the remainder of the proof, we will find a value of σ that ensures $\mathcal{L}_{\mathcal{M},x_1,x_2}(\tilde{x}) \leq \varepsilon$ with probability at least $1-\delta$.

Step 1: Gaussian tail bound. When $\mathcal{L}_{\mathcal{M},x_1,x_2}(\tilde{x}) \geq \varepsilon$, we have

$$\frac{\Delta}{\sigma}\cdot Z + \frac{\Delta^2}{2\sigma^2} \geq \varepsilon, \text{which means } |Z| \geq \frac{\varepsilon\sigma - \frac{\Delta^2}{2\sigma}}{\Delta}.$$

Since $\Delta \in [-1,1]$, we can set $\Delta = 1$ to minimize the right-hand side of the above inequality, yielding the worst-case $|Z|$, which gives $|Z| \geq \varepsilon\sigma - 1/(2\sigma)$. We need to bound the probability of this event by δ:

$$\Pr\left[|Z| \geq \varepsilon\sigma - \frac{1}{2\sigma}\right] \leq \delta, \text{ implying } \Pr\left[Z \geq \varepsilon\sigma - \frac{1}{2\sigma}\right] \leq \frac{\delta}{2}.$$

By the Gaussian tail bound (Chernoff bound):

$$\Pr[Z \geq t] \leq \exp\left(\frac{-t^2}{2}\right),$$

we have

$$\exp\left(\frac{-\left(\varepsilon\sigma - \frac{1}{2\sigma}\right)^2}{2}\right) \leq \frac{\delta}{2}.$$

Step 2: Solving σ. Define $C = \sqrt{-2\ln(\delta/2)}$. We can rewrite the above inequality as

$$\left(\varepsilon\sigma - \frac{1}{2\sigma}\right)^2 \geq C^2.$$

Taking the square root of both sides and rearranging gives us two cases:

$$\varepsilon\sigma - \frac{1}{2\sigma} \geq C \quad \text{or} \quad \varepsilon\sigma - \frac{1}{2\sigma} \leq -C.$$

Therefore, the best choice of σ is the one that satisfies:

$$\varepsilon\sigma - \frac{1}{2\sigma} = \pm C.$$

Solving this quadratic equation gives us two solutions for σ:

$$\sigma = \frac{\pm C \pm \sqrt{C^2 + 2\varepsilon}}{2\varepsilon}.$$

Since σ must be positive, we take the positive root, which gives us the final solution:

$$\sigma = \frac{\sqrt{-2\ln(\delta/2)} + \sqrt{-2\ln(\delta/2) + 2\varepsilon}}{2\varepsilon},$$

which is equivalent to the result in Theorem 4. □

B.2.2 Comparison with Dwork's Proof. There are two main concerns with Dwork's proof of the Gaussian mechanism:

- The proof is hard to interpret within the (ε, δ)-DP notion;
- The proof is based on a problematic Gaussian tail bound.

Specifically, (i) the proof is based on the assumption that the privacy loss can be probabilistically bounded by $1-\delta$, which is consistent with the PAC privacy notion but hard to interpret in the original (ε, δ) privacy notion. The (ε, δ) privacy notion intuitively says that the distance between two distributions is *always* bounded by the given (ε, δ) pair, i.e. strictly cannot be unbounded. From this notion,

it is hard to interpret the probabilistic bound of $1-\delta$ in Dwork's proof. (ii) The proof is based on a tail bound (p262 in [18]):

$$\Pr[Z \geq t] \leq \frac{\sigma}{\sqrt{2\pi}} \cdot \exp\left(\frac{-t^2}{2\sigma^2}\right),$$

where $Z \sim \mathcal{N}(0, \sigma^2)$. This bound is problematic. As a counterexample, if $t = 0$ and $\sigma = 1$, the bound gives $\Pr[Z \geq 0] \leq 1/\sqrt{2\pi} \approx 0.399$, which is incorrect since $\Pr[Z \geq 0] = 0.5$ for arbitrary Gaussian distributions with mean 0.

B.2.3 Comparison with the Analytic Gaussian. Another Gaussian mechanism is the analytic Gaussian mechanism [5]. The name originates from the analytic form of the Gaussian distribution in its result. We don't adopt this mechanism for two reasons: (i) It lacks analytical form of noise scale σ due to the complexity of the given formula to satisfy their privacy definition. i.e. Theorem 8 in [5]. As stated in their paper: "we propose to find σ using a numerical algorithm... (p4, line 7)", which complicates analytical utility quantification. (ii) It is based on an alternative privacy definition:

$$\Pr\left[\mathcal{L}_{\mathcal{M},x_1,x_2} \geq \varepsilon\right] - \exp(\varepsilon) \cdot \Pr\left[\mathcal{L}_{\mathcal{M},x_1,x_2} \leq -\varepsilon\right] \leq \delta.$$

While they proved this definition satisfies (ϵ, δ)-DP, it is hard to interpret, because it bounds the *difference* between the probability of the privacy loss being too large ($\mathcal{L} \geq \epsilon$) and too small ($\mathcal{L} \leq \epsilon$), different as Dwork's bounding $\Pr[\mathcal{L} \geq \epsilon] \leq \delta$, which is more direct and interpretable.

B.3 Proof of Theorem 2

Proof. We calculate the total failure probability of d mechanisms to show that it is $1-(1-\delta)^d$.

The combination of d independent (ε, δ)-PAC LDP mechanisms satisfies pure ε-LDP only if all d mechanisms satisfy pure ε-LDP. Therefore, the probability of no failure is at least $(1-\delta)^d$ by independence. Thus, the total failure probability of d independent (ε, δ)-PAC LDP mechanisms is at most $1-(1-\delta)^d$.

The combination result for ε is straightforward, and it is the same as the combination of pure ε-LDP mechanisms. □

Remark. Compare with the original combination theorem for d mechanisms, i.e. $(d\varepsilon, d\delta)$-LDP [18], our result is guaranteed to be more precise, as $1-(1-\delta)^d \leq d\delta$ for any $\delta \in (0,1)$ and $d \geq 1$.

C Complementary Materials

Notations used in this paper are summarized in Table 3.

C.1 Deterministic Robustness of White-box Classifiers (Section 4.2)

When the classifier is a (public) white-box model, we can directly analyze its exact robustness radius from the model parameters. According to their representativeness, classifiers can be categorized into two types: closed-form classifiers and non-closed-form classifiers.

C.1.1 Closed-form Classifiers. A closed-form classifier has a mathematical expression that can be directly analyzed. Examples include the naive Bayes classifier [**?**], QDA [25**?**], certain variants of SVM [30], and clustering models like k-means [**?**] and Gaussian mixture models [**?**]. Closed-form expression means excellent interpretability, including robustness analysis. For these classifiers, decision boundaries can be analytically derived from the model parameters, allowing for analytical expression of robustness radius θ for any input variable x_0.

Table 3: Notations used in this paper.

Notation	Meaning
$\mathcal{M}(x)$	LDP mechanism applied to input x
$\mathcal{X}$	Input domain of an LDP mechanism
$\tilde{x} = \mathcal{M}(x)$	Perturbed version of x (sent to h)
h	Classifier function
$B_\theta(x)$	θ-ball centered at x
$\rho(\varepsilon, \theta)$	Utility quantification of a classifier
$F_{\mathcal{M}}(\cdot)$	CDF of LDP mechanism $\mathcal{M}$
ω	$1-\omega$ indicates the confidence level
τ	Robustness tolerance

Formally, if the decision boundary of h is a closed-form curve (or surface) $C(x) = 0$, then θ at an input variable x_0 is the distance from x_0 to $C(x)$. Its analytical expression w.r.t variable x_0 can be calculated by minimizing the distance between x_0 and $C(x)$.

Example 4. *Consider a 2D QDA classifier $h_{1,2}(x) : (x_u, x_v) \to \{1, 2\}$, where the discriminant function for the first class is $h_1(x) = -0.5(x_u^2+x_v^2)+\ln(0.5)$, and for the second class is $h_2(x) = -0.5(0.5(x_u-1)^2 + 0.5(x_v-1)^2) - 0.5\ln(4) + \ln(0.5)$. The decision boundary $h_1(x) - h_2(x) = 0$ is a circle with radius $2\sqrt{\ln 2 + 1}$ centered at $(-1,-1)$. Thus, the ℓ_2 robustness radius at $x_0 = (x_u, x_v)$ is $\theta = |2\sqrt{\ln 2 + 1} - \sqrt{(x_u+1)^2 + (x_v+1)^2}|$.*[*]

C.1.2 Non-closed-form Classifiers. Non-closed-form classifiers often lack explicit mathematical expressions for their decision boundaries. Examples include neural network [66] and decision tree [8]. Their decision boundaries are analytically intractable, making it challenging to derive the robustness radius θ directly. Finding θ for these classifiers is known as the robustness verification problem [35, 49], i.e. verify whether $h(B_\theta(x)) = h(x)$ for a given θ at x. The main insight is to encode classifiers as a set of constraints, transforming the problem of verifying $h(B_\theta(x)) = h(x)$ into a satisfiability problem that can be solved by constraints solvers.

Formally, given a concrete sample x_0 and θ, we have $B_\theta(x_0) = \{x : ||x - x_0||_\infty \leq \theta\}$ as a linear constraint on x. Denote $h_i(x)$ as the score function of the i-th class of $h(x)$, and let c be the correct class of x_0. By encoding $B_\theta(x_0)$ and the classifier $h(x)$ into a satisfiability problem

$$x \in B_\theta(x_0) \ \wedge \ h_c(x) < h_i(x) \text{ for } i \neq c,$$

we can verify the problem using a constraint solver. If it is unsatisfiable, then $h(x)$ outputs the correct class within $B_\theta(x_0)$. The maximum θ that makes the problem unsatisfiable is the robustness radius θ.

[*] A trivial lower bound of the ℓ_∞ robustness radius can be obtained by using $\theta/\sqrt{2}$, based on the norm inequality.

Example 5. *(To replicate the QDA classifier from the previous example, which is also the same classifier depicted in Figure 3.) Consider a 2D neural network classifier* $h_{1,2}(x) : [0,1]^2 \to \{1,2\}$ *with structure* $a_2(\text{ReLU}(a_1(x)))$*, where* $\text{ReLU} = \max(0,x)$ *and* a_1, a_2 *are affine functions of the hidden layer and output layer that*

$$a_1(x) = \begin{bmatrix} -1.86 & -2.09 \\ 0.12 & -0.46 \end{bmatrix} x + \begin{bmatrix} 3.71 \\ -0.08 \end{bmatrix},$$
$$a_2(x) = \begin{bmatrix} -3.05 & 0.40 \\ 4.02 & -0.22 \end{bmatrix} x + \begin{bmatrix} 0.94 \\ -0.58 \end{bmatrix}.$$

Denote $h_1(x)$ *and* $h_2(x)$ *as the 1st and 2nd dimension of* $h_{1,2}(x)$*. Given* $x_0 = (0.5, 0.5)$ *and* $\theta = 0.1$*, we have* $B_{0.1}(x_0) = ||x - x_0||_\infty \leq 0.1$ *as a linear constraint on* x*. If the correct class is* $h_{1,2}(x_0) = 1$*, the robustness verification problem is*

$$x \in B_{0.1}(x_0) \ \wedge \ h_1(x) < h_2(x).$$

If this problem is verified as unsatisfiable, it means $h_{1,2}(x)$ *always outputs class* 1 *within* $B_{0.1}(x_0)$*. Finally, finding* $\arg\max_\theta h(B_\theta(x)) = h(x)$ *is a sequence of decision problems on* θ*.*

C.2 Explanation of Definition 8

There are two randomness sources in Definition 8: (i) Denote the deterministic robustness $h(\tilde{x}) = h(x)$ an indicator $\phi(\tilde{x}) \in \{0,1\}$. We can say a system is robust under random $\tilde{x}$ with $\Pr[\phi(\tilde{x}) = 1] > 1 - \omega$. In this setting, the robustness boundary is assumed to be known, and the randomness comes from $\tilde{x}$. (ii) When the robustness boundary is unknown, $\phi(\tilde{x})$ must be estimated and thus takes values in $[0,1]$ instead of $\{0,1\}$. In this case, the condition $\phi(\tilde{x}) > 1 - \tau$ becomes a random event, where the randomness comes from the estimation of ϕ. This leads to two-level probabilistic guarantee $\Pr[1 - \Pr[\phi(\tilde{x}) \leq \tau]] \geq 1 - \omega$. (Such interpretation also is lacked in existing definitions [57, 70].)

C.3 LDP Mechanisms in Figure 4

We provide concrete instantiations of the LDP mechanisms in Figure 4 and discuss their $\rho(\varepsilon, \theta)$ curves.

C.3.1 Piecewise-based Mechanism. The original PM mechanism [60] is defined on $[-1,1] \to [-C_\varepsilon, C_\varepsilon]$, where C_ε is a variable dependent on ε. Such design ensures unbiasedness, as it was originally designed for mean estimation. The original SW mechanism [41] is defined on $[0,1] \to [-b_\varepsilon, 1 + b_\varepsilon]$. Unlike PM, SW is biased and designed for distribution estimation.

Although defined on different domains, both PM and SW mechanisms can be "normalized" to the same domain $[0,1] \to [0,1]$ with the same privacy parameter ε. We give their formulation as the following two definitions.

Definition 11 (The PM mechanism). *The PM mechanism takes an input* $x \in [0,1]$ *and outputs a random variable* $\tilde{x} \in [0,1]$ *as follows:*

$$pdf[\mathcal{M}(x) = \tilde{x}] = \begin{cases} p_\varepsilon & \text{if } \tilde{x} \in [l_{x,\varepsilon}, r_{x,\varepsilon}], \\ p_\varepsilon / \exp(\varepsilon) & \tilde{x} \in [0,1] \setminus [l_{x,\varepsilon}, r_{x,\varepsilon}], \end{cases}$$

where $p_\varepsilon = \exp(\varepsilon/2)$*,*

$$[l_{x,\varepsilon}, r_{x,\varepsilon}] = \begin{cases} [0, 2C) & \text{if } x \in [0, C), \\ x + [-C, C] & \text{if } x \in [C, 1-C], \\ (1-2C, 1] & \text{otherwise}, \end{cases}$$

with $C = (\exp(\varepsilon/2) - 1)/(2\exp(\varepsilon) - 2)$*.*

The SW mechanism is similar but uses different instantiations for p_ε and $[l_{x,\varepsilon}, r_{x,\varepsilon})$.

Definition 12 (The SW mechanism). *The SW mechanism takes an input* $x \in [0,1]$ *and outputs a random variable* $\tilde{x} \in [0,1]$ *as the same formulation as the PM mechanism, but with*

$$p_\varepsilon = \frac{\exp(\varepsilon) - 1}{\varepsilon}, \quad C = \frac{\exp(\varepsilon)(\varepsilon - 1) + 1}{2(\exp(\varepsilon) - 1)^2}.$$

Both mechanisms guarantee ε-LDP through their piecewise design, as the probability of seeing the same value $\tilde{x}$ when the original data is x_1 and x_2 is bounded by $\exp(\varepsilon)$.

$\rho(\varepsilon, \theta)$ curves. The concentration analysis $F(x+\theta) - F(x-\theta)$ of the PM and SW mechanisms results in a linear curve with two segments. When θ is small, the concentration is determined by the $[l_{x,\varepsilon}, r_{x,\varepsilon})$ segment, which gives a slope proportional to p_ε. When θ is large, the concentration is also determined by the $[0,1) \setminus [l_{x,\varepsilon}, r_{x,\varepsilon})$ segment, which gives a slope proportional to $p_\varepsilon / \exp(\varepsilon)$. Thus, the $\rho(\varepsilon, \theta)$ curves for the PM and SW mechanisms exhibit a linear pattern with two distinct slopes.

C.3.2 Discrete Mechanism. By discretizing the $[0,1]$ domain into bins, LDP mechanisms defined on discrete domains can also be applied. In Figure 4, we fixed the number of bins to 100, i.e. the domain is $\mathcal{X} := \{0, 0.01, 0.02, \ldots, 1\}$. The according Exponential mechanism and k-RR mechanism is defined as follows.

Definition 13 (The Exponential mechanism). *Given score function* $d(x, \tilde{x})$*, the Exponential mechanism defined on* $\mathcal{X}$ *takes an input* $x \in \mathcal{X}$ *and outputs a random variable* $\tilde{x} \in \mathcal{X}$ *as follows:*

$$\Pr[\mathcal{M}_{\exp}(x) = \tilde{x}] = \frac{\exp\left(\frac{\varepsilon d(x,\tilde{x})}{2\Delta d}\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(\frac{\varepsilon d(x,\tilde{x}')}{2\Delta d}\right)},$$

where $\Delta d = \max_{x,\tilde{x},\tilde{x}' \in \mathcal{X}} |d(x,\tilde{x}) - d(x,\tilde{x}')|$ *is the sensitivity of the score function* d*.*

We choose the score function as the negative absolute distance, i.e. $d(x,\tilde{x}) := -|x - \tilde{x}|$, so that outputs closer to x are assigned higher probabilities. Accordingly, the sensitivity is $\Delta d = 1$.

Definition 14 (The k-RR mechanism). *k-RR is a sampling mechanism* $\mathcal{M} : \mathcal{X} \to \mathcal{X}$ *that, given input* $x \in \mathcal{X}$*, outputs* $\tilde{x} \in \mathcal{X}$ *according to*

$$\Pr[\mathcal{M}(x) = \tilde{x}] = \begin{cases} \dfrac{\exp(\varepsilon)}{|\mathcal{X}| - 1 + \exp(\varepsilon)} & \text{if } \tilde{x} = x, \\ \dfrac{1}{|\mathcal{X}| - 1 + \exp(\varepsilon)} & \text{otherwise.} \end{cases}$$

$\rho(\varepsilon, \theta)$ curves. The concentration analysis $F(x+\theta) - F(x-\theta)$ of the Exponential mechanism results in a smooth curve, as the probability of seeing value $\tilde{x}$ further away from x decreases smoothly w.r.t θ. In contrast, the concentration analysis of the k-RR mechanism exhibits a pattern similar to the PM and SW mechanisms.

When θ is small, the probability is determined by the bin containing x, results in a slope $\propto \exp(\varepsilon)/(|\mathcal{X} - 1 + \exp(\varepsilon)|)$. When θ becomes larger, the probability is also determined by the other bins, which gives a slope $\propto 1/(|\mathcal{X} - 1 + \exp(\varepsilon)|)$.

C.4 PAC LDP vs (ε, δ)-LDP (Section 5.2)

This section discusses the difference between PAC LDP, (ε, δ)-LDP, and other related privacy notions.

Definition 15 ((ε, δ)-LDP [18]). *A randomized mechanism $\mathcal{M}$ satisfies (ε, δ)-LDP if for all $x_1, x_2 \in \mathcal{X}$ and all $S \subseteq \mathcal{Y}$,*

$$\Pr[\mathcal{M}(x_1) \in S] \leq \exp(\varepsilon) \cdot \Pr[\mathcal{M}(x_2) \in S] + \delta.$$

This definition says that the distance between the distributions of $\mathcal{M}(x_1)$ and $\mathcal{M}(x_2)$ is bounded by a (ε, δ) pair. More specifically, it can be rewritten as:

$$\frac{\Pr[\mathcal{M}(x_1) \in S]}{\Pr[\mathcal{M}(x_2) \in S] + \delta/\exp(\varepsilon)} \leq \exp(\varepsilon).$$

Given concrete values of pair (ε, δ),[†] the term $\delta/\exp(\varepsilon)$ is a constant. Thus, it means that the distance between the distributions of $\mathcal{M}(x_1)$ and $\mathcal{M}(x_2)$ must be strictly bounded, i.e. similar to pure ε-LDP. However, this definition is often interpreted as: it is ε-LDP with a failure probability of δ. Meanwhile, the Gaussian mechanism relies on this interpretation.

Other relaxed privacy notions, including the Concentrated privacy from Dwork [16], Rényi privacy [50], and PAC privacy [68], provide patches to (ε, δ)-LDP. The Gaussian mechanism has natural results under the Concentrated privacy and Rényi privacy notions, while we provide a PAC privacy result in Appendix B.2. Among these privacy notions, (ε, δ)-PAC LDP provides the same meaning as the common interpretation of (ε, δ)-LDP: it is ε-LDP with a failure probability of δ.

C.5 Other Discussions (Section 6)

C.5.1 Complexity of Finding the Robustness Radius. The procedure for finding a probabilistic robustness radius is independent of the classifier's structure and parameters. Its complexity arises from the iterations required to determine the robustness radius and the testing of n perturbed data points in each iteration.

Using binary search to find the radius results in a complexity of $\Theta(\log(1/\kappa))$, where κ is the precision of the radius. For instance, if $\kappa = 0.01$ (two decimal places), the complexity is $\Theta(\log 100) \approx \Theta(7)$. Meanwhile, classifiers are often implemented with parallel inference on GPUs, which allows for $\Theta(1) \times T$ complexity for n data points, where T is the inference time for a single data point, provided n is not excessively large. Consequently, the total complexity for finding a two-decimal precision robustness radius is $\Theta(7T)$. In practice, T is typically in the order of milliseconds, making the complexity of finding the robustness radius sufficiently low. Moreover, the robustness radius is a one-time computation for a given classifier, which can be used for all future utility quantification.

[†]In fact, δ can be defined in a more intricate manner, such as incorporating the noise scale σ as in [7]. However, this approach introduces a recursive dependency thus uncontrollable in practical applications, as σ itself should be defined in terms of δ.

C.5.2 Per-instance Data Utility vs Aggregated Data Utility. Our utility quantification framework focuses on classifiers and per-instance data utility, providing utility guarantees at the level of individual data points. This stands in contrast to aggregated data utility analyses such as mean estimation, which evaluate the accuracy of aggregated statistics computed from the perturbed data of many users. Per-instance data utility is crucial for several reasons: (i) Classifiers and other personalized services like recommendation systems operate on individual data points. Their performance depends directly on per-instance utility and has no direct connection to aggregated utility. (ii) Per-instance data utility focuses on mechanism-level utility analysis, characterizing the privacy-utility tradeoff curves without relying on sample size or aggregation effects. These two reasons relate to our focus on per-instance data utility in this paper.

C.5.3 Fixed Classifier with Noisy Input vs Retrained Classifier with Noisy Data. These are two different scenarios. (i) In the former scenario, users locally apply an LDP mechanism to their data for privacy protection, agnostic to the service provider (i.e. the classifier), and are affected by the resulting utility of the existing classifier. (ii) In the latter scenario, the service provider applies LDP to users' data and retrains a new classifier tailored to the perturbed data, which relates to "robust training".

We focus on the former user-privacy-centered scenario; in the latter scenario, users expose their raw data to the service provider. Moreover, from users' perspective, the classifier is fixed at the time of use (i.e. when they query the service). If a retrained classifier is deployed, then the classifier utility is for the retrained classifier (from users' perspective).

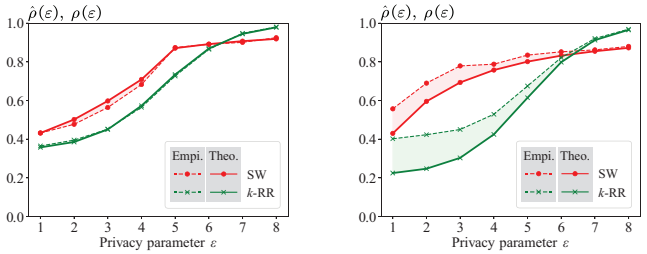
C.5.4 Robustness of LDP Mechanisms. Intuitively, LDP mechanisms with higher concentration around raw data are more prone to reconstruction (with sufficient reports) and poisoning attacks, as malicious data are less averaged. In our concentration analysis, PM, SW, and k-RR (especially with small k) are more concentrated, thus expected to be more vulnerable. This aligns with prior findings: PM and k-RR show more vulnerability to poisoning attacks [39, 40].

C.5.5 Independent LDP Mechanisms vs Correlated LDP Mechanisms. We adopt independent perturbations for cleanness of presentation. In practice, DP libraries (e.g. Google DP [**?**]) typically provide independent (L)DP primitives and do not explicitly support correlated perturbations. The essential reason is that the underlying data correlations are often unknown.

When the correlation coefficients are known or estimable, independent LDP can be extended via (i) post-processing of perturbed data to enforce a given correlation structure, or (ii) adopting correlated-perturbation mechanisms (e.g. [55, 59]; although they are primarily tailored to mean estimation). Our utility quantification framework also applies to such correlated mechanisms by replacing the CDF of independent mechanisms with the conditional CDF of correlated mechanisms.

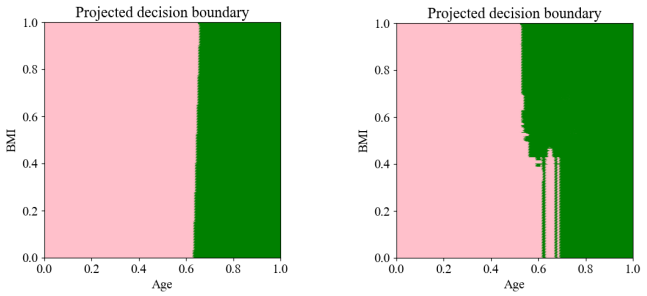
C.6 More Case Studies for the Stroke Prediction Dataset (Section 7.2.1)

This section provides a detailed utility analysis of the two classifiers trained on the Stroke Prediction dataset.



(a) Logistic Regression. (b) Random Forest.

Figure 9: Empirical and theoretical utility for two classifiers trained on the Stroke Prediction dataset.



(a) Logistic Regression (LR). (b) Random Forest (RF).

Figure 10: Projected decision boundary of the two classifiers on the stroke dataset.

C.6.1 Other Records and Mechanisms. We also focus on the first record in the dataset, i.e. the record "Age: 67, BMI: 36.6, Hypertension: 0, …", to evaluate the performance of other two LDP mechanisms: the SW mechanism and the k-RR mechanism.

Figure 9 shows the theoretical and empirical utility of the two classifiers under the SW and k-RR mechanisms. The theoretical utility of both mechanisms is almost the same as the empirical utility for the Logistic Regression classifier. For the Random Forest classifier, the theoretical utility is a lower bound for the empirical utility, especially when ε is small. Meanwhile, we can see that the SW mechanism does not always outperform the k-RR mechanism. When ε exceeds 6, k-RR outperforms SW in both classifiers.

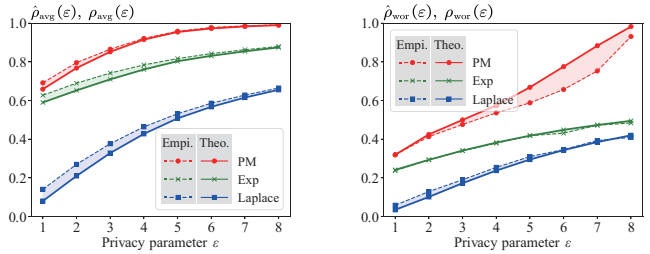
C.6.2 Closed Form CDF of the PM Mechanism. This section provides the closed form expression for $\rho(\varepsilon, \theta)$ when using the PM mechanism.

The input "Age: 79" corresponds to the normalized value $x = 0.79$ in domain $[0, 1]$. Then according to Definition 11 of the PM mechanism, we have

$$F_{\mathrm{PM}_{\varepsilon,\mathrm{Age}}}(0.63) = \int_0^{0.63} pdf[\mathcal{M}(0.79) = \tilde{x}]\mathrm{d}\tilde{x}.$$

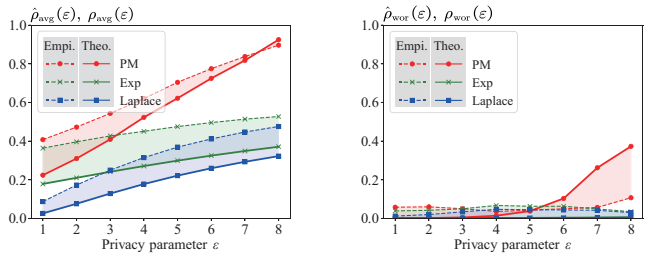
For piecewise-based mechanisms, we need to consider the range of $[l_{0.79,\varepsilon}, r_{0.79,\varepsilon}]$ relative to 0.63. For ε that makes $0.63 \notin [l_{0.79,\varepsilon}, r_{0.79,\varepsilon}]$, the above integral is

$$F_{\mathrm{PM}_{\varepsilon,\mathrm{Age}}}(0.63) = 0.63 \cdot \exp(-\frac{\varepsilon}{2}).$$



(a) LR: Average-case utility. (b) LR: Worst-case utility.

Figure 11: Average-case and worst-case utility for the Logistic Regression classifier trained on the Stroke Prediction dataset.



(a) RF: Average-case utility. (b) RF: Worst-case utility.

Figure 12: Average-case and worst-case utility for the Random Forest classifier trained on the Stroke Prediction dataset.

For ε that makes $0.63 \in [l_{0.79,\varepsilon}, r_{0.79,\varepsilon}]$, the above integral is

$$F_{\mathrm{PM}_{\varepsilon,\mathrm{Age}}}(0.63) = (0.79 - C) \cdot \exp(-\frac{\varepsilon}{2}) + (C - 0.16) \cdot \exp(\frac{\varepsilon}{2}),$$

where $C = (\exp(\varepsilon/2) - 1)/(2\exp(\varepsilon) - 2)$, as define in Definition 11.

C.6.3 Projected Decision Boundary. The gap between the theoretical and empirical utility for the Random Forest classifier comes from the complexity of its decision boundary.

Figure 10 illustrates the projected decision boundaries of the two classifiers on the "Age" and "BMI" features. We can see that the decision boundary of the Random Forest classifier is significantly more complex than that of the Logistic Regression classifier, i.e. a combination of many hyperrectangles, making it hard to be approximated by one hyperrectangle. Consequently, the theoretical utility for the Random Forest classifier is conservative compared to its actual utility. In contrast, the simpler decision boundary of the Logistic Regression classifier can be well approximated by hyperrectangles, leading to a precise theoretical utility.

C.6.4 Average-case and Worst-case Utility. Figure 11 shows the average-case and worst-case utility of the Logistic Regression classifier under three LDP mechanisms. For the average-case utility, the trends closely mirror those observed in the point-wise utility quantification presented in the main text. For the worst-case utility, the curves are less regular but still exhibit the same overall ordering among mechanisms: the PM mechanism consistently achieves the highest utility, followed by the Exponential mechanism, and then the Laplace mechanism. In the worst-case scenario, we observe that the theoretical utility is not always lower than the empirical utility (though they are close), especially for the PM mechanism when ε

is large. This occurs because the worst-case robustness hyperrectangle is typically two-dimensional and small, which amplifies the impact of sampling errors in empirical evaluation. In such cases, the actual utility is often higher than the empirical utility.

Figure 12 shows the average-case and worst-case utility of the Random Forest classifier. The worst-case utility for the Random Forest classifier is significantly smaller than that of the Logistic Regression classifier. This is due to the complex decision boundary of the Random Forest classifier, as shown in Figure 10b. Around Age, BMI $\approx \{0.65, 0.3\}$, the robustness hyperrectangle is significantly smaller than that of the Logistic Regression classifier, leading to both small theoretical and empirical utilities.

Conclusion on Robustness. From the results of average-case and worst-case utility analysis of the Logistic Regression and Random Forest classifiers, we can conclude that the Logistic Regression classifier is more robust under LDP-perturbed inputs than the Random Forest classifier.

C.7 More Case Studies for the Bank Customer Attrition Dataset (Section 7.2.2)

C.7.1 Detailed Quantification. We take the Logistic Regression classifier as an example to show the detailed utility quantification under the PAC LDP.

The robustness hyperrectangle at this record is $\theta_\diamond = [0, 0.72] \times [0, 1]$. Using this information, we can provide the quantification of the utility at the record under the PM mechanism.

For this trained Logistic Regression classifier on the Bank Customer Attrition dataset, at the record "Age: 22, Estimated Salary: 101 348, Credit Score: 619, …", with probability at least $\rho(\varepsilon, \theta_\diamond)$, the classifier preserves the correct prediction under the privacy indicator $\mathcal{I}_{0.1}$ and the PM mechanism applied to "Age" and "Estimated Salary" (i.e. $(2\varepsilon, 0.1)$-PAC LDP), where

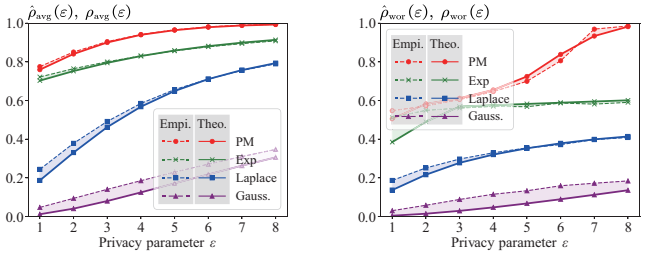
$$\begin{aligned}\rho(\varepsilon, \theta) =& [F_{PM_{\varepsilon,Age}}(0.72) - F_{PM_{\varepsilon,Age}}(0)][F_{PM_{\varepsilon,Sal}}(1) - F_{PM_{\varepsilon,Sal}}(0)] \\ =& F_{PM_{\varepsilon,Age}}(0.72).\end{aligned}$$

For the Gaussian mechanism, the utility quantification is similar but uses the Gaussian CDF.

For this trained Logistic Regression classifier on the Bank Customer Attrition dataset, at the record "Age: 22, Estimated Salary: 101 348, Credit Score: 619, …", with probability at least $\rho(\varepsilon, \theta_\diamond)$, the classifier preserves the correct prediction under the Gaussian mechanism (Theorem 4) applied to "Age" and "Estimated Salary" (i.e. $(2\varepsilon, 0.1)$- PAC LDP), where

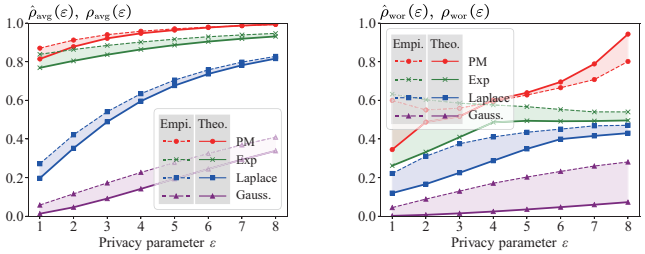
$$\begin{aligned}\rho(\varepsilon, \theta) =& [F_{Gau_{\varepsilon,Age}}(0.72) - F_{Gau_{\varepsilon,Age}}(0)][F_{Gau_{\varepsilon,Sal}}(1) - F_{Gau_{\varepsilon,Sal}}(0)] \\ =& F_{Gau_{\varepsilon,Age}}(0.72) - F_{Gau_{\varepsilon,Age}}(0).\end{aligned}$$

C.7.2 Average-case and Worst-case Utility. Figure 13 and Figure 14 summarize the average-case and worst-case utility for classifiers trained on the Bank Customer Attrition dataset. The observed trends are consistent with those from the Stroke Prediction dataset: (i) The theoretical utility closely matches the empirical utility, especially in the average-case scenario. (ii) The PM mechanism consistently achieves the highest utility, followed by the Exponential mechanism, then the Laplace mechanism, and finally the Gaussian mechanism.



(a) LR: Average-case utility. (b) LR: Worst-case utility.

Figure 13: Average-case and worst-case utility for the Logistic Regression classifier trained on the Bank Customer Attrition dataset.



(a) RF: Average-case utility. (b) RF: Worst-case utility.

Figure 14: Average-case and worst-case utility for the Random Forest classifier trained on the Bank Customer Attrition dataset.

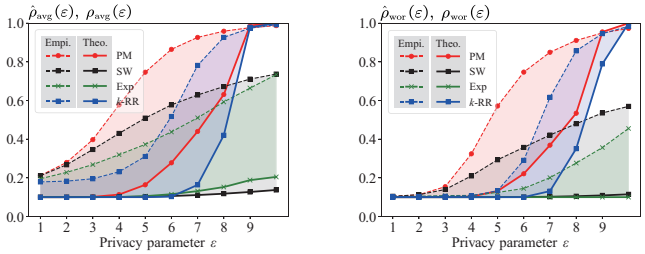
C.8 More Details for the MNIST Classifier (Section 7.2.3)

C.8.1 Monte Carlo Estimation of the Utility. We emphasize the conceptual connection between LDP and robustness while adopting robustness notions that largely align with the standard *local robustness* used in prior work. Although these notions are clean and well established, they can yield conservative utility estimates in high-dimensional settings.

To obtain tighter utility quantification in high dimensions, one can consider more flexible robustness notions, e.g. robustness regions given by irregular sets rather than axis-aligned hyperrectangles (as induced by complex decision boundaries). Such regions can be estimated via Monte Carlo sampling in high-dimensional spaces. This estimation is performed once; afterward, for any given ε, the utility can be theoretically quantified by approximately integrating the d-dimensional LDP mechanism's probability distribution over the estimated robustness region.

Formally, let $h : [0, 1]^d \to \{1, \ldots, K\}$ be a d-dimensional classifier and let $x \in [0, 1]^d$ be an input with predicted (and correct) label $h(x)$. Draw $\tilde{x}_1, \ldots, \tilde{x}_N$ i.i.d. uniformly from $[0, 1]^d$, and for each sample check whether $h(\tilde{x}_i) = h(x)$ holds. Assume that m of these samples satisfy the check, and let $\mathcal{S} \subseteq [0, 1]^d$ denote an estimated robustness region that contains the accepted samples. Under a d-dimensional (continuous) LDP mechanism,[‡] the utility

[‡]For discrete LDP mechanisms, the utility can be computed directly by summing the probabilities that the mechanism outputs samples in $\mathcal{S}$, eliminating the need for volume estimation. We omit this discussion for simplicity.



(a) NN: Average-case utility. **(b) NN: Worst-case utility.**

Figure 15: Average-case and worst-case utility for the Neural Network classifier trained on the MNIST-7×7 dataset.

can then be quantified as

$$\rho(\varepsilon, \mathcal{S}) = \int_{\mathcal{S}} pdf[\mathcal{M}_\varepsilon(x) = \tilde{x}]\mathrm{d}\tilde{x}$$
$$\approx \mathrm{Vol}(\mathcal{S}) \cdot \frac{1}{m} \sum_{i=1}^{m} pdf[\mathcal{M}_\varepsilon(x) = \tilde{x}_i],$$

where $\mathrm{Vol}(\mathcal{S})$ is the volume of the region $\mathcal{S}$, which can be estimated by $\mathrm{Vol}([0,1]^d) \cdot m/N$. It is still a theoretical measure, since samples are fixed and $pdf[\mathcal{M}_\varepsilon(x) = \tilde{x}_i]$ is determined analytically by the LDP mechanism and can be evaluated for any ε.

This approach follows the connection between concentration analysis and robustness analysis established in this paper, with the concentration analysis now performed over the estimated robustness region $\mathcal{S}$ rather than the hyperrectangle $\theta_\diamond$. The error is now governed by the Monte Carlo estimation error (i.e. $O(1/\sqrt{N})$) and volume estimation error, instead of Hoeffding bound in Theorem 1.

We experimented with this approach on the MNIST-7×7 dataset. While it is theoretically sound, in such a high-dimensional space the samples that pass the uniform-sampling check are sparse. As a result, for most accepted samples $\tilde{x}_i$, the density $pdf[\mathcal{M}_\varepsilon(x) = \tilde{x}_i]$ is close to zero, which drives the estimated utility to an unrealistically small value. Moreover, we cannot directly apply importance sampling using $pdf[\mathcal{M}_\varepsilon(x)]$ as the proposal distribution, since this proposal distribution depends on ε and would essentially reduce the procedure to empirical utility estimation.

C.8.2 Average-case and Worst-case Utility. We use the first 50 images from the MNIST-7×7 dataset as representative samples for analyzing the average-case and worst-case utility under LDP-perturbed inputs. Figure 15 presents the results of this analysis. Both the average-case and worst-case utility show a similar trend to the point-wise utility quantification, with the PM mechanism consistently achieving the highest utility.