NOISETTE: Certifying Differential Privacy Mechanisms Efficiently

Qi Pang*[1], Radhika Garg[2], Ziling Liu[3], Hanshen Xiao[4], Virginia Smith[1], Wenting Zheng[1], and Xiao Wang[2]

[1]CMU, [2]Northwestern University, [3]National University of Singapore, [4]Purdue University

Abstract

Differential privacy (DP) has emerged as a rigorous framework for privacy-preserving data analysis, with widespread deployment in industry and government. Yet existing implementations typically assume that the party applying the mechanism can be trusted to sample noise correctly. This trust assumption is overly optimistic: a malicious party may deviate from the protocol to gain accuracy or avoid scrutiny, thereby undermining users' privacy guarantees.

In this paper, we introduce NOISETTE, a family of efficient protocols for certifying DP noise sampling across both discrete and continuous settings. We design a protocol that supports any discrete DP noise distribution through certifiable lookup table evaluation, and introduce a staircase-based optimization that greatly improves efficiency without compromising privacy or utility. We further extend this framework to continuous mechanisms, providing the first efficient protocol for certifiable continuous noise sampling with a rigorous end-to-end (ϵ, δ)-DP guarantee under standard floating-point arithmetic, closing the precision-induced privacy loss left open by prior work.

We demonstrate the practicality of our protocols through concrete DP applications, including mean estimation and federated learning. Our protocols outperform the prior state-of-the-art by orders of magnitude in runtime and communication, while preserving the same accuracy as uncertified DP mechanisms. These results establish NOISETTE as the first efficient, scalable, and general-purpose solution for certifiable DP noise sampling, making certified privacy guarantees practical in high-stakes applications.

1 Introduction

Differential privacy (DP) [DMNS06a] is a rigorous mathematical framework enabling privacy-preserving data analytics and machine learning, with strong and quantifiable privacy guarantees. For instance, Apple [DPT, DPT17] employs DP to collect usage statistics such as popular emojis without revealing individual user activity, Google [goo] uses DP for private telemetry in Chrome and federated learning, and the U.S. Census Bureau [MKA+08, Abo16, otC21] adopted DP to protect individual records in the 2020 Census.

Most deployments follow the centralized model: a trusted aggregator collects sensitive data and applies the DP mechanism, so the privacy guarantees hold only if the aggregator faithfully follows the protocol. This trust can be overly optimistic: since DP mechanisms trade utility for privacy, an aggregator who prefers higher accuracy has a strong incentive to add less noise, weakening users' privacy while still claiming DP protection. Even an honest aggregator can fail to deliver the promised guarantee: continuous DP mechanisms, which underpin most real-world deployments in machine

*Emails: qipang@cmu.edu, radhikaradhika2028@u.northwestern.edu

learning and federated learning [ACG+16a], run on floating-point (FP) arithmetic, and naive FP samplers leak information through low-order mantissa bits and intermediate rounding [Mir12]; a recent independent audit of Apple's widely deployed DP framework found that *every mechanism depending on floating-point noise fails to meet its advertised DP guarantee* [CBJX26]. Unlike malicious deviation, this failure lies in the mechanism itself rather than its execution: faithfully running such a sampler, or even proving that it was run correctly, yields no valid DP guarantee.

Such concerns about whether the promised guarantees actually hold have hindered broader adoption, particularly where outcomes drive high-stakes decisions such as resource allocation, policy evaluation, or fairness audits [LPDV23, YLBC25]. Major technology companies are increasingly required to demonstrate privacy compliance through independent audits: as part of a discrimination settlement, Meta must report platform statistics to third-party privacy auditors [met23], and under a Federal Trade Commission consent order, Google must undergo independent privacy assessments for 20 years [ftc11]. These audits, however, verify high-level procedural compliance and cannot certify each execution of a DP mechanism. This gap motivates fine-grained, privacy-preserving certification: proving that each execution samples noise from the promised distribution, while revealing neither the data nor the randomness, so that any utility degradation is attributable solely to the DP mechanism; for FP mechanisms, such certification is meaningful only if the certified DP guarantee holds under finite-precision arithmetic.

Existing work certifies DP noise sampling with zero-knowledge proofs (ZKPs) [BGKW24, BAT24, STC+24, BC23]. In addition, the MPC literature has developed efficient oblivious sampling circuits based on rejection sampling [CsU19, WYF+23]: although designed for joint noise generation among multiple parties, these circuits are data-oblivious and adapt directly to the ZKP setting. However, both approaches face the following fundamental limitations.

- *Poor efficiency or utility tradeoffs.* Some protocols [BGKW24, BC23, BAT24] implement crypto-friendly mechanisms (e.g., Binomial), which provide worse utility-privacy tradeoffs than widely used mechanisms such as Gaussian and Laplace. Others support Gaussian sampling but require expensive computation for crypto-unfriendly functions such as logarithms and square roots [STC+24], or rely on rejection sampling [WYF+23], which is highly inefficient in ZKP.

- *Lack of generalizability.* Existing approaches are tailored to specific DP mechanisms and cannot be easily generalized to others. Yet in practice, new mechanisms continue to be developed for specialized applications, such as the Cauchy mechanism for instance-based noise [NRS07] and the Skellam mechanism for federated learning [AKL21]. Manually crafting a new certifiable sampler for every distribution is tedious and unscalable.

- *No formal certification for floating-point noise.* Most critically, no existing ZKP protocol certifies the formal (ε, δ)-DP guarantee under FP arithmetic: existing protocols either operate only on discrete distributions [BGKW24, BAT24, BC23, WYF+23] or leave precision-induced privacy leakage unresolved even when targeting FP Gaussian noise [STC+24]. A certified mechanism can therefore still exhibit the very FP failures observed in deployment [CBJX26].

1.1 Contributions

To address these challenges, we propose Noisette, a family of efficient, scalable, and generalizable ZKP protocols for both discrete and floating-point DP noise distributions with formal (ε, δ)-DP guarantees. Table 1 compares Noisette with the most relevant prior protocols. We include Bell et al. [BGKW24] and Biswas et al. [BC23] as ZKP-based protocols for the Binomial mechanism,

Shamsabadi et al. [STC+24] for continuous Gaussian; Wei et al. [WYF+23] as the most efficient MPC sampler for discrete Gaussian and Laplace; and Bontekoe et al. [BAT24] as a representative shuffle-DP certification. None of them supports both discrete and continuous mechanisms, and none provides a formal (ε, δ)-DP guarantee under standard FP arithmetic. In summary, we make the following contributions:

Generalized Certifiable Discrete DP Noise Sampling. Our first observation is that it is possible to sample from *any discrete probability distribution* using inverse transform sampling, which represents the distribution using a finite-sized cumulative distribution function (CDF) table that is indistinguishable from the original distribution with negligible error. This technique is general and applies even to complex DP noise distributions, such as discrete Gaussian and Laplace. Crucially, we make this approach highly efficient in ZKP by leveraging well-studied set inclusion protocols [PFM+22, ZBK+22, CFG24], thereby avoiding the need to execute deep circuits for complex noise sampling. Based on these insights, we design Noisette-Base, a generalized ZKP protocol that can efficiently sample any additive discrete DP noise.

Extension to Continuous DP Noise Distributions. While inverse transform sampling provides a strong starting point, representing a continuous distribution would require an astronomically large CDF table (2^{32} for FP32), making lookup operations infeasible. To overcome this challenge, we introduce a staircase approximation that groups adjacent elements into buckets, and reduce the complexity by sampling the bucket index first and then sampling within the bucket. This enables Noisette-Cont, which makes efficiently sampling from continuous DP distributions feasible. As a key optimization, we also apply this staircase technique to discrete cases to construct Noisette-SC, which substantially reduces the CDF table size and improves the discrete sampling efficiency. We rigorously prove that these staircase approximations preserve DP guarantees.

Rigorous DP under FP Arithmetic. Noisette-Cont closes both FP leakage channels discussed above, including mantissa-bit leakage and intermediate rounding [Mir12], through a careful noise sampling design and an error-free FP expansion that combines the sampled noise with the query result and rounds exactly once. We formally bound the resulting deviation from the ideal continuous mechanism by a small total-variation distance and absorb it as a quantifiable additive increase in the δ of the final (ε, δ)-DP guarantee. To our knowledge, Noisette-Cont is the first certifiable continuous DP sampler with a rigorous end-to-end (ε, δ)-DP guarantee under standard FP arithmetic.

Evaluation on Concrete Applications. We benchmark Noisette-Base against state-of-the-art (SOTA) protocols for certifiable discrete Gaussian and Laplace sampling. Our results show that Noisette-Base achieves up to 23× faster runtime and 36× lower communication than the strongest prior protocol, and up to 52× and 83×, respectively, over oblivious-stack-based sampling. We also evaluate Noisette-SC and Noisette-Cont in two concrete DP applications: privacy-preserving mean estimation and differentially private federated learning, where we certify the noise for fine-tuning Transformer language models. Compared with SOTA protocols for continuous (floating-point) and discrete noise sampling, Noisette-SC and Noisette-Cont achieve up to 29× speedup in runtime and 34× reduction in communication, while providing the same DP and utility guarantees as uncertified mechanisms. Together, Noisette demonstrates efficiency and generalizability, supporting both discrete and FP certifiable DP noise sampling.

2 Background

Notation. We use λ to denote the statistical security parameter. For a distribution $\mathcal{D}$, $x \leftarrow \mathcal{D}$ denotes x being sampled according to the distribution $\mathcal{D}$. We use upper-case letters like X (or bold lower case letters like $\boldsymbol{x}$) to denote a vector. For a vector X, X_i denotes the i-th component of X,

Method	Distributions	FP DP
Bell et al. [BGKW24]	Binomial	×
Biswas et al. [BC23]	Binomial	×
Bontekoe et al. [BAT24]	Bernoulli	×
Shamsabadi et al. [STC+24]	Gaussian	has precision leakage
Wei et al. [WYF+23]	dGauss, dLap	×
Noisette	**any additive**	**rigorous**

Table 1: Comparison of certifiable DP noise sampling protocols. "has precision leakage" means the formal (ε, δ)-DP guarantee fails under standard FP arithmetic; "rigorous" means it holds end-to-end.

Functionality $\mathcal{F}_{\mathsf{ZK}}$

Commit: Upon receiving (commit, w) from $\mathcal{P}$ and (commit) from $\mathcal{V}$. Pick a fresh id, send (id) to both parties and store (id, w).

Prove: Upon receiving $(\mathsf{prove}, \mathcal{C}, (\mathsf{id}_1, \mathsf{id}_2, \ldots, \mathsf{id}_n))$ from $\mathcal{P}$ and $\mathcal{V}$ where $\mathcal{C} : \{0,1\}^n \to \{0,1\}$, retrieve values $\boldsymbol{w}$ for all tags in $\mathbf{id} = (\mathsf{id}_1, \mathsf{id}_2, \ldots, \mathsf{id}_n)$. The functionality aborts if the circuits $(\mathcal{C})$ from both parties are different or any tag is invalid or if $\mathcal{C}(\boldsymbol{w}) \neq 1$, otherwise sends (true) to both parties.

Plookup: Upon receiving $(\mathsf{plookup}, \boldsymbol{a}, \mathbf{id})$ from both $\mathcal{P}$ and $\mathcal{V}$, retrieve values $\boldsymbol{w}$ for all tags in $\mathbf{id}$. The functionality aborts if $\boldsymbol{a}$ is inconsistent or any tag is invalid and sends (true) to both parties if $\boldsymbol{w} \subseteq \boldsymbol{a}$.

Figure 1: ZK functionality supporting commit and prove.

where X_0 is the first component. We use $[\![x]\!]$ to denote the commitment of x. We use the shorthand $[\![\boldsymbol{x}]\!]$ to denote the commitments $[\![\boldsymbol{x}_0]\!], [\![\boldsymbol{x}_1]\!], \ldots [\![\boldsymbol{x}_n]\!]$. $||$ is concatenation.

Zero-Knowledge Proof. Zero-knowledge proofs (ZKPs) for circuit satisfiability allow a prover $\mathcal{P}$ with witness w to prove to the verifier $\mathcal{V}$ that $\mathcal{C}(w) = 1$ for a public circuit $\mathcal{C}$ without revealing any information about w. We describe the functionality in Figure 1. This functionality can be instantiated efficiently using ZKP protocols, including zkSNARKs [Gro16, CHM+20, GWC19] and interactive protocols [YSWW21, WYY+22, BMRS21, BBMH+21, DILO22, BBMHS22]. We use the VOLE-based commitment scheme from QuickSilver [YSWW21], implemented in EMP-toolkit [WMK16].

Set Inclusion ZKP. Set inclusion protocols [PFM+22, ZBK+22, CFG24] allow the prover $\mathcal{P}$ to prove that $B \subset A$ where B is committed, and then the inclusion is proved based on the committed input $[\![B]\!]$ and auxiliary inputs. We describe the *plookup* functionality in Figure 1 and summarize the protocol from [GW20] below; the complete protocol description is provided in Appendix C. First, the prover commits to an auxiliary input $C = (A||B)$ that is sorted according to A, i.e., either $C[i] = C[i+1]$ or if $C[i] = A[j]$ then $C[i+1] = A[j+1]$. Then, we define polynomials f and g:

$$\begin{aligned} f(\beta, \gamma) &:= (1+\beta)^N \textstyle\prod_{i\in[N]}(\gamma + B[i]) \\ &\quad \cdot \textstyle\prod_{i\in[M-1]}(\gamma(1+\beta) + A[i] + \beta A[i+1]) \\ g(\beta, \gamma) &:= \textstyle\prod_{i\in[N+M-1]}(\gamma(1+\beta) + C[i] + \beta C[i+1]) \end{aligned} \tag{1}$$

Both polynomials are equal iff $B \subset A$ and C is defined correctly. Both parties $\mathcal{P}$ and $\mathcal{V}$ evaluate

the polynomials f and g on a random point $a, b \in \mathbb{F}_{ZK}$ and check if $f(a,b) \stackrel{?}{=} g(a,b)$. This requires $|A| + 2|B| - 2$ multiplications in $\mathbb{F}_{ZK}$.

Differential Privacy. Differential privacy (DP) [DMNS06a] mechanisms protect the privacy of sensitive data by randomizing the outputs of the query functions. The amount of privacy leakage is usually quantified by the parameters ε and δ, where ε is the privacy budget and δ the failure probability. When $\delta = 0$, mechanism $\mathcal{M}$ is said to satisfy pure ε-DP and it satisfies approximate (ε, δ)-DP for nonzero δ. (ε, δ)-DP is a relaxation where ε-DP holds with probability at least $1 - \delta$. Smaller ε and δ indicate less privacy budget and smaller failure probability, thus stronger privacy guarantees.

Definition 1 ((ε, δ)-DP)**.** *A randomized mechanism $\mathcal{M}$ satisfies (ε, δ)-DP, if for all neighboring datasets $\mathcal{X}, \mathcal{X}'$ that differ in a single record, and for all possible output S:*

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

Additive noise DP mechanisms are the most widely used DP mechanisms in practice, where additive noise is typically drawn from certain distributions like Laplace and Gaussian. For example, the Laplace mechanism $Lap(\Delta/\varepsilon)$ adds noise proportional to the function sensitivity Δ, and satisfies pure ε-DP. The function sensitivity Δ is defined as the maximum change in the output when any single record is added, removed, or replaced in the dataset.

Definition 2 (Sensitivity)**.** *Let $\mathcal{C}_D$ be a collection of all datasets, and $f : \mathcal{C}_D \to \mathbb{R}$. The sensitivity of function f is:*

$$\Delta = \max_{\mathcal{X}, \mathcal{X}' \in \mathcal{C}_D} \left| f(\mathcal{X}) - f(\mathcal{X}') \right|,$$

where the maximum is over all pairs of datasets $\mathcal{X}, \mathcal{X}' \in \mathcal{C}_D$ that differ in a single record. For higher dimensional outputs, the difference is usually measured by the L_p norm.

There are a lot of widely used DP mechanisms for both continuous and discrete applications, including Gaussian [DR+14], Laplace [DR+14], discrete Gaussian [CKS20], discrete Laplace [GRS08], Skellam [AKL21], etc. We present detailed introductions of these mechanisms in Appendix A. In this work, we propose a generic protocol for certifying arbitrary discrete and continuous (floating-point) DP noise sampling.

In this work, we use a commonly used technique Concentrated DP (CDP) (Definition 3 from [CKS20]) to track the privacy loss, which provides a tighter and more compositional analysis than the standard (ε, δ)-DP framework. We can also translate CDP into concrete (ε, δ)-DP guarantees. $(\varepsilon, 0)$-DP implies $\frac{1}{2}\varepsilon^2$-CDP and $\frac{1}{2}\varepsilon^2$-CDP implies $(\frac{1}{2}\varepsilon^2 + \varepsilon\sqrt{2\log(1/\delta)}, \delta)$-DP, $\forall \delta > 0$.

Definition 3 (CDP)**.** *A randomized algorithm $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$ satisfies $\frac{1}{2}\varepsilon^2$-CDP if, $\forall x, x' \in \mathcal{X}^n$ differing on a single element and $\forall \alpha > 1$, we have $D_\alpha(\mathcal{M}(x) || \mathcal{M}(x')) \leq \frac{1}{2}\varepsilon^2\alpha$, where $D_\alpha(P||Q) = \frac{1}{\alpha - 1}\log(\sum_{y \in \mathcal{Y}} P(y)^\alpha Q(y)^{1-\alpha})$ is the Rényi divergence of order α of distribution P from Q.*

DP is widely adopted in machine learning model training, and **DP-SGD** [ACG+16b] is a widely used optimization technique for differentially private model training. DP-SGD provides a DP guarantee for the trained model by clipping the model gradients during training and then applying a DP mechanism, such as Gaussian mechanism.

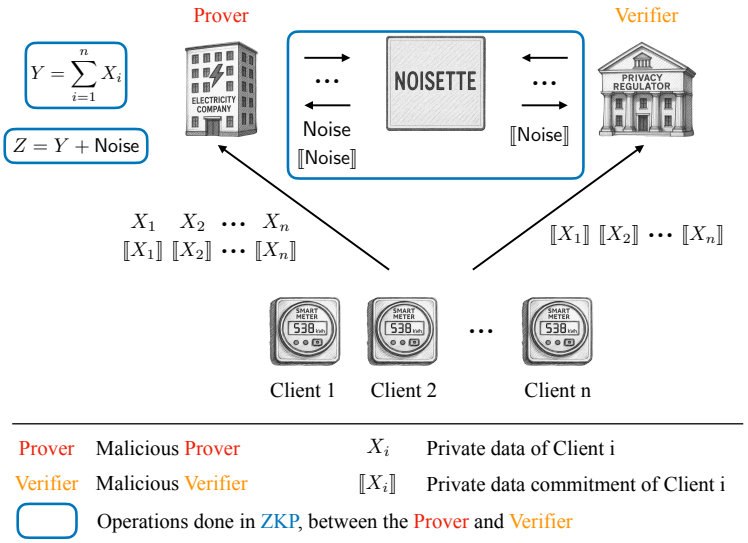


Figure 2: Setup Overview.

3 Overview

Before introducing our techniques, we outline the setup of DP certification with a concrete example. Consider an electricity company that collects data from household smart meters. These measurements are valuable for demand forecasting and grid optimization, but they also reveal sensitive living patterns, e.g., when residents are home or asleep. To protect privacy, the company can apply DP by adding calibrated noise before releasing aggregate statistics. The guarantee, however, depends entirely on the aggregator applying the DP mechanism correctly. The aggregator may have incentives to deviate: adding too little noise to gain accuracy, or too much to obscure underlying patterns. Yet it cannot publish raw data under public and regulatory pressure. In our setting, each smart meter (client) commits to its collected data and forwards the commitments to both the aggregator and an independent regulator. The aggregator acts as the prover, generating DP-protected statistics, and the regulator is the verifier that confirms these statistics were produced with the promised DP mechanism, as illustrated in Figure 2.

Threat Model. We consider a set of clients holding sensitive data and a central server that aggregates them. The server, responsible for adding DP noise, acts as the prover. The prover can act maliciously during noise sampling and may deviate from the DP mechanism to gain higher accuracy or avoid scrutiny, but it will not release raw data due to public and regulatory pressure. An independent regulator acts as the verifier, checking that the released outputs were produced with the promised DP mechanism. The verifier is also untrusted: it may deviate from the protocol and attempt to learn the clients' sensitive data.

We assume clients honestly follow the data-commitment and transfer protocol. Data manipulation attacks (e.g., backdoor attacks in federated learning) are orthogonal and out of scope for our focus on DP certification. We assume no collusion between the prover and the verifier. We rigorously account for the finite-precision effects of FP32 arithmetic in our DP analysis (§ 5 and Appendix H).

The essential task in this setting is DP noise sampling certification, which NOISETTE is designed to support efficiently, and we provide certifiable DP noise sampling for both discrete and continuous mechanisms. For discrete mechanisms, we propose NOISETTE-BASE and its staircase-optimized variant NOISETTE-SC (§ 4, § 5). For continuous mechanisms, we design NOISETTE-CONT with

Functionality $\mathcal{F}_{\mathsf{sample}}$

Input: Number of samples N, finite and discrete distribution to be sampled from $\mathcal{D}$.
Functionality: Upon receiving $(\mathsf{sample}, \mathcal{D}, N)$ from $\mathcal{P}$ and $\mathcal{V}$, sample $\boldsymbol{x}_i \leftarrow \mathcal{D}$ for $i \in [N]$.
Output: Send $\boldsymbol{x}$, $[\![\boldsymbol{x}]\!]$ to $\mathcal{P}$ and $[\![\boldsymbol{x}]\!]$ to $\mathcal{V}$.

Figure 3: Functionality for sampling $\boldsymbol{x} \leftarrow \mathcal{D}$.

end-to-end DP guarantees under FP arithmetic (§ 5; FP precision analysis in § 5.4). § 6 demonstrates the efficiency of all three on concrete DP applications.

4 Certifying Discrete Noise Sampling

In this section, we introduce Noisette-Base, a new zero-knowledge protocol for certifying sampling correctness for *any* discrete DP noise distribution. Figure 3 describes the corresponding ideal functionality $\mathcal{F}_{\mathsf{sample}}$. $\mathcal{F}_{\mathsf{sample}}$ takes as input two parameters: $\mathcal{D}$, a finite and discrete noise distribution; and N, the number of samples. It then samples $\boldsymbol{x}_i \leftarrow \mathcal{D}$ for $i \in [N]$. $\mathcal{F}_{\mathsf{sample}}$ outputs these samples $\boldsymbol{x}$ to the prover $\mathcal{P}$, and commitments $[\![\boldsymbol{x}]\!]$ to both $\mathcal{P}$ and the verifier $\mathcal{V}$.

To design an efficient protocol, our first observation is that, given any concentrated DP noise distribution, it is possible to truncate to a finite domain such that the truncated distribution is statistically indistinguishable from the original distribution. The truncated distribution can be instantiated using a lookup table that encodes its CDF. We then use a technique called *inverse transform sampling* to sample from the CDF table. For the rest of this section, we first formally describe properties of truncation, followed by inverse transform sampling, uniform randomness generation, and finally the Noisette-Base protocol.

4.1 Distribution Truncation

Any concentrated discrete DP distribution $\mathcal{D}$ (one with a tail bound $\Pr_{X\sim\mathcal{D}}(|X| > M) \leq f(M)$) can be truncated to distribution $\mathcal{D}'$ by restricting $\mathcal{D}$ to $[-M, M]$ and renormalizing. Lemma 1 formally defines statistical indistinguishability between $\mathcal{D}$ and $\mathcal{D}'$ in terms of the security parameter λ.

Lemma 1. *Let $\mathcal{D}$ be a discrete distribution over $\mathbb{Z}$ and let f be a tail bound satisfying $\Pr_{X\sim\mathcal{D}}(|X| > M) \leq f(M)$. Let $\mathcal{D}'$ be the distribution obtained by truncating $\mathcal{D}$ to $[-M, M]$, i.e., for all $x \in [-M, M]$,*

$$\Pr_{X'\sim\mathcal{D}'}(X' = x) = \frac{\Pr_{X\sim\mathcal{D}}(X = x)}{\Pr_{X\sim\mathcal{D}}(|X| \leq M)}.$$

Then for any adversary $\mathcal{A}$ that sees N i.i.d. samples, $\mathcal{D}^{\otimes N}$ and $\mathcal{D}'^{\otimes N}$ are statistically indistinguishable whenever $Nf(M) \leq 2^{-\lambda}$, for statistical security parameter λ.

Truncation Parameter M. Most additive DP noise distributions are concentrated, including Gaussian [CKS20, DR$^+$14], Laplace [CKS20], Skellam [AKL21], etc. We use a Chernoff bound [Che52] to determine the truncation parameter as shown in Table 2. We include both discrete and continuous distributions here and will discuss continuous (FP) sampling in § 5.3.

Distribution	Truncation Bound M
Discrete Gaussian $\mathcal{N}_{\mathbb{Z}}(0, \sigma^2)$	$\sigma\sqrt{(2\ln 2)(\lambda+1)+2\ln N}+1$
Continuous Gaussian $\mathcal{N}(0, \sigma^2)$	$\sigma\sqrt{(2\ln 2)(\lambda+1)+2\ln N}$
Discrete Laplace $Lap_{\mathbb{Z}}(0, t)$	$t\ln\frac{2N\cdot 2^{\lambda}}{e^{1/t}+1}$
Continuous Laplace $Lap(0, t)$	$t\ln(N\cdot 2^{\lambda})$

Table 2: Truncation parameters M for commonly used DP noise distributions. λ denotes the statistical security parameter, and N is the total number of noise samples.

4.2 Inverse Transform Sampling

After truncating the discrete distribution $\mathcal{D}$ to $\mathcal{D}'$ with a finite domain, we use the inverse transform method to sample from $\mathcal{D}'$. The CDF is defined as $\mathsf{CDF}_{\mathcal{D}'}(x) = \Pr_{X\sim\mathcal{D}'}(X \le x)$. We use a sorted vector $\boldsymbol{x}$ to denote the support of $\mathcal{D}'$ such that $\mathsf{CDF}_{\mathcal{D}'}(\boldsymbol{x}_{i-1}) < \mathsf{CDF}_{\mathcal{D}'}(\boldsymbol{x}_i)$ and $\mathsf{CDF}_{\mathcal{D}'}(\boldsymbol{x}_{-1}) := 0$. Then the probability $\Pr_{X\sim\mathcal{D}'}(X = \boldsymbol{x}_i) = \mathsf{CDF}_{\mathcal{D}'}(\boldsymbol{x}_i) - \mathsf{CDF}_{\mathcal{D}'}(\boldsymbol{x}_{i-1})$.

We represent the distribution $\mathcal{D}'$ as a table T with its i-th entry defined as $T[i] = (\mathsf{CDF}_{\mathcal{D}'}(\boldsymbol{x}_{i-1}), \mathsf{CDF}_{\mathcal{D}'}(\boldsymbol{x}_i), \boldsymbol{x}_i)$. To sample from the distribution $\mathcal{D}'$, we first sample a uniform value $u \in (0, 1]$. Then, we go through the CDF table and find an index i such that $\mathsf{CDF}_{\mathcal{D}'}(\boldsymbol{x}_{i-1}) < u \le \mathsf{CDF}_{\mathcal{D}'}(\boldsymbol{x}_i)$. The sampled value is $\boldsymbol{x}_i$.

Optimization for Symmetric Distributions. For symmetric distributions, we can halve the table size to sample the distribution's right half and additionally sample a sign bit to extend the sampled value to the whole domain. Thus, we construct the CDF table containing $M+1$ elements and sample $\boldsymbol{x}_i \in [0, M]$ using the above procedure. Then we sample a random sign $s \in \{-1, 1\}$ and output $s \cdot \boldsymbol{x}_i$.

Distribution Shift under Fixed-Point Representation. Inverse transform sampling assumes real-valued CDF probabilities and uniform $u \in (0, 1]$. Finite-precision computation cannot represent arbitrary reals, so we encode both the CDF probabilities and u using n-bit fixed-point: an n-bit string U encodes the value $u = (U+1)/2^n$, so u is uniform over the grid $\{2^{-n}, 2\cdot 2^{-n}, \ldots, 1\}$ of $(0, 1]$ and falls in exactly one CDF interval. We show the resulting deviation from the ideal sampler is at most $2^{-\lambda}$ for statistical security parameter λ.

Lemma 2. *Let $\mathcal{D}'$ be the truncated discrete distribution represented by a CDF table with $M+1$ entries, and let $\mathcal{D}''$ be the distribution produced by inverse transform sampling when the CDF probabilities and the uniform value $u \in (0, 1]$ are represented in n-bit fixed-point. Then, for statistical security parameter λ, any adversary $\mathcal{A}$ that sees N i.i.d. samples cannot distinguish $\mathcal{D}'$ from $\mathcal{D}''$ whenever*

$$n \ge \log_2 N + \log_2(M+1) + \lambda.$$

We describe the table construction in Appendix B. Now the essential part is to sample an n-bit uniform value u; we address how to ensure fair randomness in the next section.

4.3 Fair Randomness

To sample a uniform n-bit fixed-point u, $\mathcal{F}_{\mathsf{sample}}$ makes use of an ideal functionality $\mathcal{F}_{\mathsf{rand}}$ that takes the number of bits n as input and outputs n random bit commitments to both parties (see Figure 4). We instantiate the $\mathcal{F}_{\mathsf{rand}}$ functionality using Π_{rand} as shown in Figure 5. The prover $\mathcal{P}$ samples n random bits and sends their commitments to the verifier $\mathcal{V}$. Upon receiving these commitments

Functionality $\mathcal{F}_{\mathsf{rand}}$

This functionality has all the same features as $\mathcal{F}_{\mathsf{ZK}}$ and the following command:

Rand. Upon receiving(rand, n) from $\mathcal{P}$ and $\mathcal{V}$, sample $\boldsymbol{x} \leftarrow \{0,1\}^n$ and send $(\mathbf{id})$ to both parties and store $(\mathbf{id}, \boldsymbol{x})$.

Figure 4: Functionality to get committed random bits.

Protocol Π_{rand}

$\mathcal{R}$ is a random oracle.

1. $\mathcal{P}$ samples $\boldsymbol{x} \leftarrow \{0,1\}^n$ and sends $[\![\boldsymbol{x}]\!]$ to the $\mathcal{V}$.
2. On receiving $[\![\boldsymbol{x}]\!]$, $\mathcal{V}$ sends $s \leftarrow \{0,1\}^\kappa$ to $\mathcal{P}$.
3. $\mathcal{P}$ and $\mathcal{V}$ obtain $\boldsymbol{y} \leftarrow \mathcal{R}(s)$
4. $\mathcal{P}$ receives $\boldsymbol{r}, [\![\boldsymbol{r}]\!]$ and $\mathcal{V}$ receives $[\![\boldsymbol{r}]\!]$, where $\boldsymbol{r} = \boldsymbol{x} \oplus \boldsymbol{y}$.

Figure 5: Protocol for sampling n uniformly random bits.

from the $\mathcal{P}$, $\mathcal{V}$ sends a random seed s to $\mathcal{P}$. $\mathcal{P}$ then uses s to generate n random bits and outputs the final bits as the XOR of both $\mathcal{P}$'s and $\mathcal{V}$'s randomness. The interaction cannot be avoided via the Fiat-Shamir transform: $\mathcal{P}$ must not know $\mathcal{V}$'s seed before committing to its own random bits.

4.4 Noisette-Base Protocol Design

We present our full sampling protocol Noisette-Base in Figure 6. To sample from $\mathcal{D}''$, $\mathcal{P}$ and $\mathcal{V}$ first use Π_{rand} to obtain the commitment of a uniformly random value $[\![u]\!]$. $\mathcal{P}$ outputs $\boldsymbol{x}_i$ such that $\mathsf{CDF}(\boldsymbol{x}_{i-1}) < u \leq \mathsf{CDF}(\boldsymbol{x}_i)$. $\mathcal{P}$ commits to $\boldsymbol{x}_i$, $\mathsf{CDF}(\boldsymbol{x}_{i-1})$ and $\mathsf{CDF}(\boldsymbol{x}_i)$. To prove that the sampling was done honestly, the parties verify that the committed values satisfy the condition $\mathsf{CDF}(\boldsymbol{x}_{i-1}) < u \leq \mathsf{CDF}(\boldsymbol{x}_i)$. Finally, we need to prove that the committed tuple exists in the table T that defines the public distribution $\mathcal{D}''$. Both parties compute the set defined as follows:

$$A = \{\alpha^2 \boldsymbol{x}_i + \alpha \mathsf{CDF}(\boldsymbol{x}_i) + \mathsf{CDF}(\boldsymbol{x}_{i-1}) | i \in [M]\}$$

where $\alpha \leftarrow \mathbb{F}_{ZK}$ is a random value chosen by the verifier $\mathcal{V}$, and $\mathbb{F}_{ZK}$ denotes the finite field over which $\mathcal{F}_{\mathsf{ZK}}$ defines its arithmetic constraints. $\mathcal{P}$ and $\mathcal{V}$ compute the same linear combination for the chosen value and execute a set inclusion check. When sampling N values, the set inclusion check can be amortized across all samples. We use the Plookup protocol [GW20] Π_{plookup} described in § 2 for the set inclusion check and $\mathcal{F}_{\mathsf{ZK}}$ to verify the range check. For clarity, Figure 6 omits the optional sign-bit sampling used for symmetric noise; it can be added as in § 4.2 without affecting the analysis.

Complexity Analysis. We analyze the cost of obtaining N samples from $\mathcal{D}''$ where M is its CDF table size. The prover and verifier perform $M + 2N - 2$ multiplications in $\mathbb{F}_{ZK}$ for Π_{plookup} at step 5b. For step 5a, we require 2 comparisons per sample, i.e., $O(N)$ comparisons for N samples. We use the Fiat-Shamir technique to make step 3 non-interactive. The cost per noise sample is thus

Protocol Π_{sample}

Input:

- Truncated distribution to be sampled from $\mathcal{D}'$.
- Number of samples N.

Publicly Compute:

- $\mathcal{P}$ and $\mathcal{V}$ construct the public table $T \leftarrow \Pi_{\mathsf{TConstruct}}(\mathcal{D}', N)$, where $T_i = (p_i^{min}, p_i^{max}, \boldsymbol{x}_i)$, $i \in [M]$, $p_i^{min}, p_i^{max} \in \{0,1\}^n$ and $\boldsymbol{x}_i \in \mathcal{D}'$.

Protocol:

1. $\mathcal{P}$ and $\mathcal{V}$ send (rand, nN) to $\mathcal{F}_{\mathsf{rand}}$. $\mathcal{P}$ obtains U and $[\![U]\!]$ and $\mathcal{V}$ obtains $[\![U]\!]$, where $[\![U]\!] = \{[\![U_0]\!], [\![U_1]\!], \ldots, [\![U_{N-1}]\!]\}$, $U_i \in \{0,1\}^n$, encoding the fixed-point values $u_i = (U_i + 1)/2^n \in (0, 1]$.
2. $\mathcal{P}$ defines the vector S, where $S_i := T_j$, s.t. $p_j^{min} < u_i \leq p_j^{max}$, $i \in [N]$. $\mathcal{P}$ and $\mathcal{V}$ use $\mathcal{F}_{\mathsf{ZK}}$ to obtain the commitments to $S_{i,0}$, $S_{i,1}$ and $S_{i,2}$ for all $i \in [N]$.
3. $\mathcal{V}$ samples $\alpha \leftarrow \mathbb{F}_{\mathsf{ZK}}$ and sends it to $\mathcal{P}$.
4. $\mathcal{P}$ and $\mathcal{V}$ compute A and $[\![B]\!]$ such that $A_j = T_{j,0} + \alpha T_{j,1} + \alpha^2 T_{j,2}, j \in [M]$ and $[\![B_i]\!] = [\![S_{i,0}]\!] + \alpha[\![S_{i,1}]\!] + \alpha^2[\![S_{i,2}]\!], i \in [N]$ respectively.
5. $\mathcal{P}$ and $\mathcal{V}$ perform the following checks:
   (a) For all $i \in [N]$, $\mathcal{P}$ and $\mathcal{V}$ use $\mathcal{F}_{\mathsf{ZK}}$ to prove that $S_{i,0} < u_i \leq S_{i,1}$.
   (b) $\mathcal{P}$ and $\mathcal{V}$ send $(\mathsf{plookup}, A, [\![B]\!])$ to $\mathcal{F}_{\mathsf{ZK}}$ to check $B \subseteq A$.

   If both checks pass, $\mathcal{V}$ outputs true; otherwise, $\mathcal{V}$ outputs false.

Output: $\mathcal{P}$ receives $S_{i,2}, [\![S_{i,2}]\!]$; $\mathcal{V}$ receives $[\![S_{i,2}]\!]$, $i \in [N]$.

Figure 6: NOISETTE-BASE protocol for verified sampling from a finite and discrete distribution $\mathcal{D}'$.

$O(M/N + c')$ where c' is a small constant. Thus, NOISETTE-BASE can scale to a large number of noise samples, and the cost is amortized over the number of samples N.

Security Analysis. Let $\mathcal{D}''$ be the truncation of a discrete DP noise distribution $\mathcal{D}$ as in Lemma 1 with n-bit fixed-point CDF table representations as in Lemma 2; adding noise from $\mathcal{D}$ is (ε, δ)-DP. In the $(\mathcal{F}_{\mathsf{ZK}}, \mathcal{F}_{\mathsf{rand}})$-hybrid model, Theorem 1 shows that Π_{sample} realizes $\mathcal{F}_{\mathsf{sample}}(\mathcal{D}'', N)$: honest execution outputs i.i.d. samples from $\mathcal{D}''$ with consistent commitments, any deviation by a malicious prover is detected except with probability at most $2MN/|\mathbb{F}_{\mathsf{ZK}}|$ (plus negligible instantiation error), and a corrupted verifier's view is simulatable by zero-knowledge. Lemma 1 bounds the truncation error from $\mathcal{D}$ to $\mathcal{D}'$ and Lemma 2 bounds the additional sampling deviation between $\mathcal{D}''$ and $\mathcal{D}'$; thus these errors can be absorbed as an additive increase to the final δ in the (ε, δ)-DP guarantee, as shown in Corollary 1. We present the proof of Theorem 1 in Appendix D.

Theorem 1. *The protocol Π_{sample} securely instantiates the functionality $\mathcal{F}_{\mathsf{sample}}$ in the $(\mathcal{F}_{\mathsf{ZK}}, \mathcal{F}_{\mathsf{rand}})$-*

hybrid model for any finite and discrete distribution $\mathcal{D}''$.

Corollary 1 (DP Guarantee of NOISETTE-BASE)**.** *Suppose an ideal mechanism using N i.i.d. samples from $\mathcal{D}$ is (ε, δ)-DP, then the mechanism drawing its N samples via Π_{sample} differs from the ideal by a TV of at most $\eta_{\mathsf{trunc}} + \eta_{\mathsf{cdf}}$, where $\eta_{\mathsf{trunc}} = 2^{-\lambda}$ is the truncation error as in Lemma 1 and $\eta_{\mathsf{cdf}} = 2^{-\lambda}$ is the CDF-table representation error as in Lemma 2. By the TV-closeness bound (see Appendix H.3), and adding the negligible dataset-independent ZK protocol soundness error of Step 5 (Theorem 1), $\delta_{\mathsf{cert}} = \mathsf{negl}(\kappa) + 2MN/|\mathbb{F}_{\mathsf{ZK}}|$, it is*

$$\Big(\varepsilon,\ \delta + \delta_{\mathsf{cert}} + (1 + e^{\varepsilon})\big(\eta_{\mathsf{trunc}} + \eta_{\mathsf{cdf}}\big)\Big)\text{-}DP.$$

Public verifiability. Our protocol naturally extends to the public verifiability setting described in [BGKW24]. For this adaptation, we assume the existence of a public registrar. During the setup phase, the prover $\mathcal{P}$ and the registrar jointly generate commitments to unbiased random bits using $\mathcal{F}_{\mathsf{rand}}$. When the verifier $\mathcal{V}$ issues a query q to both the prover and the registrar, the registrar responds by sending the commitments to the randomness. Given these commitments, $\mathcal{P}$ and $\mathcal{V}$ can execute steps 3–5 to verify the sampling process.

5 Extension to Floating-Point Noise Sampling

With NOISETTE-BASE, we can efficiently sample from discrete DP noise distributions. However, NOISETTE-BASE cannot be directly generalized to sampling continuous (floating-point) DP noise. Naively representing floating-point DP noise using a CDF table leads to a very large table – the support size of the floating-point distribution can be as large as 2^{32} if FP32 is used, even after truncation.

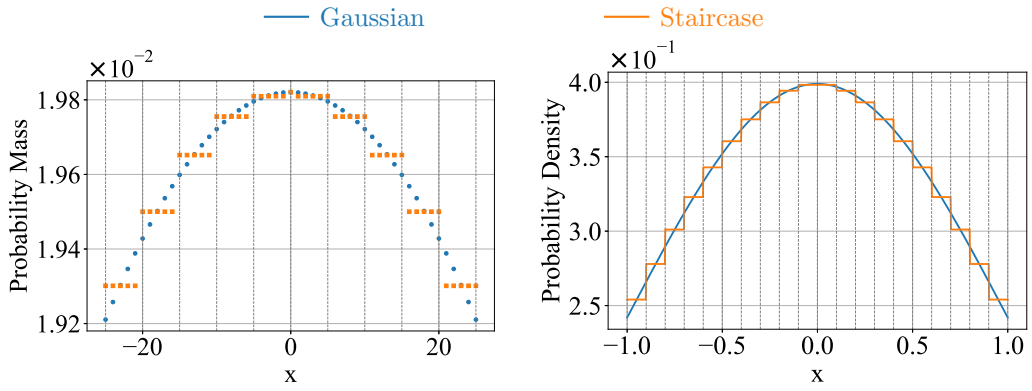
To address this challenge, we propose a novel technique called *staircase approximation*. Our insight is that, by slightly modifying the original DP noise distribution to a staircase approximation, we can significantly reduce the CDF table size. Reducing the table size first improves NOISETTE-BASE's efficiency for discrete noise sampling and then enables efficient FP noise sampling. We prove that staircase optimizations maintain privacy guarantees.

In the rest of this section, we first use staircase approximation to reduce a discrete noise distribution's CDF table size, and derive a more efficient discrete noise sampling protocol NOISETTE-SC. Then, we generalize our optimization to build a certifiable FP noise sampling protocol NOISETTE-CONT with end-to-end (ε, δ)-DP guarantees.

5.1 Staircase DP Noise

We first introduce the discrete staircase DP noise and analyze its privacy guarantees. Consider a symmetric, truncated discrete distribution with CDF table size $M + 1$. To construct its staircase version, our algorithm takes a bucket width parameter m: the value 0 keeps its exact probability, and each group of m neighboring values (together with its mirror image) forms a bucket whose values share the bucket's average probability. This reduces the CDF table to $M/m+1$ entries, which significantly improves the sampling efficiency. We visualize the PMFs of discrete Gaussian and its staircase version in Figure 7a. Formally, we present the definition of staircase discrete DP noise:

Definition 4 (Staircase Discrete DP Noise Distribution)**.** *Consider a symmetric discrete distribution $\mathcal{D}$ with PMF $p(x), x \in \mathbb{Z}$, and a bucket width $m \in \mathbb{Z}_{>0}$. The staircase noise distribution $\mathcal{D}_s$'s PMF*



(a) Discrete. (b) Continuous.

Figure 7: Gaussian noise and its staircase versions.

p_s is defined as $p_s(0) = p(0)$ and, for $x \neq 0$,

$$p_s(x) = \frac{w_k}{m}, \quad k = \left\lceil \frac{|x|}{m} \right\rceil, \quad w_k = \sum_{r=(k-1)m+1}^{km} p(r),$$

where w_k is the probability mass of the k-th bucket $C_k = \{(k-1)m+1, \ldots, km\}$.

Since the zero atom and the buckets $\pm C_k$ partition $\mathbb{Z}$, $\mathcal{D}_s$ is normalized and symmetric; when p is non-increasing in $|x|$ (as for all DP noise we consider), p_s is also non-increasing in $|x|$, because $p(0) \geq w_1/m$ and $w_k \geq w_{k+1}$. The staircase bucket width m controls the utility-efficiency trade-off. A larger m further reduces the CDF table size and improves the sampling efficiency, but the distribution deviates more from the original analytical DP noise, complicating the worst-case analysis. We discuss selecting the staircase hyper-parameters in concrete applications in § 6.2.1.

We now analyze discrete staircase DP noise's privacy guarantees using Concentrated DP (CDP; see § 2). Theorem 2 formalizes the privacy guarantees of the staircase discrete noise. We show that by choosing the bucket size m such that the sensitivity Δ is a multiple of m, the privacy loss of the staircase noise is bounded by the original discrete noise's privacy loss. Intuitively, since staircase optimization "coarsens" the original discrete distribution by grouping probability mass into buckets, such coarsening reduces information leakage and provides stronger privacy guarantees [GV15]. We present the proof in Appendix E and discuss the generalization to multivariate scenarios in Appendix F.

Theorem 2 (DP Guarantee of Staircase Discrete Noise)**.** *If the original discrete DP noise $\mathcal{D}$ satisfies $\frac{1}{2}\varepsilon^2$-CDP, then the discrete staircase noise $\mathcal{D}_s$ also satisfies $\frac{1}{2}\varepsilon^2$-CDP, if the bucket width m satisfies $\Delta = tm$, $t \in \mathbb{Z}$, Δ is the sensitivity, and the original DP discrete noise distribution $\mathcal{D}$'s PMF is symmetric and monotone, i.e., $p(x) = p(-x)$ and $p(x) \geq p(x+1)$, $\forall x \in \mathbb{Z}_{\geq 0}$.*

In fact, our proof shows a stronger statement: at shifts that are multiples of m, the staircase noise's entire privacy profile, including every (ε, δ) trade-off and every Rényi order, is dominated by the original noise's (Lemma 4).

5.2 Noisette-SC: Staircase Discrete Sampling

The staircase discrete DP noise can be efficiently sampled using the CDF-table lookup technique in Noisette-Base. After transforming the original discrete distribution into a staircase version, the

Protocol $\Pi_{\mathsf{sampleStaircase}}$

Input:

- Truncated discrete distribution $\mathcal{D}''$, number of samples N.
- Invoke system type, NOISETTE-SC for discrete and NOISETTE-CONT for FP sampling.

Publicly Compute:

- $\mathcal{P}$ and $\mathcal{V}$ convert $\mathcal{D}''$ to its staircase version, and obtain bucket width m and staircase discrete distribution $\mathcal{D}''_s$ over the table rows: the zero atom and bucket bases $\{1, m{+}1, 2m{+}1, \cdots\}$ for NOISETTE-SC ($m \in \mathbb{Z}_{>0}$), or bucket bases $\{0, m, 2m, \cdots\}$ for NOISETTE-CONT ($m \in \mathbb{R}_{>0}$). Each row stores its base $\boldsymbol{x}'$ and an indicator b, which is 0 for the zero atom and 1 otherwise.
- Determine the bit length B to represent random values from $[0, m)$.

Protocol:

1. $\mathcal{P}$ and $\mathcal{V}$ send $(\mathsf{sample}, \mathcal{D}''_s, N)$ to $\mathcal{F}_{\mathsf{sample}}$. $\mathcal{P}$ obtains $(\boldsymbol{x}'_i, \boldsymbol{b}_i)$ with commitments, and $\mathcal{V}$ obtains $[\![\boldsymbol{x}'_i]\!], [\![\boldsymbol{b}_i]\!]$, $i \in [N]$.
2. $\mathcal{P}$ and $\mathcal{V}$ send (rand, BN) to $\mathcal{F}_{\mathsf{rand}}$. $\mathcal{P}$ obtains $\boldsymbol{u}$ and $[\![\boldsymbol{u}]\!]$ and $\mathcal{V}$ obtains $[\![\boldsymbol{u}]\!]$, where $[\![\boldsymbol{u}]\!] = \{[\![\boldsymbol{u}_0]\!], [\![\boldsymbol{u}_1]\!], \ldots, [\![\boldsymbol{u}_{N-1}]\!]\}$, $\boldsymbol{u}_i \in \{0,1\}^B$ are interpreted as uniformly random values from $[0, m)$.
3. $\mathcal{P}$ and $\mathcal{V}$ use $\mathcal{F}_{\mathsf{ZK}}$ to prove $\boldsymbol{x}_i = \boldsymbol{x}'_i + \boldsymbol{b}_i \boldsymbol{u}_i$, $i \in [N]$.

Output: $\mathcal{P}$ receives $\boldsymbol{x}_i, [\![\boldsymbol{x}_i]\!]$; $\mathcal{V}$ receives $[\![\boldsymbol{x}_i]\!]$, $i \in [N]$.

Figure 8: NOISETTE-SC and NOISETTE-CONT protocols for sampling discrete and continuous staircase DP noise.

original truncation parameter M stays the same, while the lookup table only needs one entry for the zero atom and one per bucket. The staircase optimization thus reduces the table size by $m\times$, from $M+1$ to $M/m+1$. To sample from the staircase distribution, we first sample a row from the CDF table: either the zero atom, which outputs 0, or a bucket base $x = (k-1)m + 1$, where k is the sampled bucket index. In the latter case, we uniformly sample an integer $u \in [0, m-1]$ (m is a power of two, so u is $\log_2 m$ uniform bits) and output $z = x + u$, which follows the staircase discrete noise distribution; each table row stores an indicator b that is 0 for the zero atom and 1 otherwise, so both cases are covered by the single relation $z = x + b \cdot u$. We present the NOISETTE-SC protocol in Figure 8; as in Figure 6, the sign-bit sampling for symmetric noise is omitted.

5.3 Noisette-Cont: Continuous (FP) Sampling

So far, we have shown staircase optimizations significantly reduce the CDF table size for discrete DP noise sampling. Now, we generalize our technique to floating-point noise sampling and introduce NOISETTE-CONT. In this section, we temporarily assume all operations are over the reals so that

the analysis excludes FP precision leakage. We will bound the leakage and present an end-to-end DP guarantee under FP arithmetic in the next section.

Definition 5 (Staircase Continuous DP Noise Distribution)**.** *Consider a symmetric continuous noise distribution $\mathcal{D}$ with PDF $p(x), x \in \mathbb{R}$, the staircase noise distribution $\mathcal{D}_s$'s PDF $p_s(x)$ is defined as follows:*

$$p_s(x) = \frac{w_k}{m},\ k = \lfloor \frac{|x|}{m} \rfloor \cdot \text{SIGN}(x),\ w_k = \int_{|k|m}^{(|k|+1)m} p(x)dx,$$

where $m \in \mathbb{R}_{>0}$ is the bucket width, w_k is k-th bucket's probability integral.

We visualize staircase continuous Gaussian in Figure 7b. Similar to the discrete Gaussian, we assign the values within each bucket the same probability, which is that bucket's probability integral. The staircase optimization effectively discretizes the continuous distributions and preserves their tail behaviors. For the one-dimensional case or multidimensional cases with independent sensitivities, similar to Theorem 2, we set the bucket size m such that the sensitivity $\Delta = km$, where $k \in \mathbb{Z}$. The privacy loss of the staircase noise is bounded by that of the original noise.

For multivariate noise of dimension d, the dimensions only need to satisfy a joint L_p sensitivity constraint, and the shift assigned to each dimension can be arbitrarily small for continuous cases, so we can no longer choose m to divide every per-dimension shift as Theorem 2 requires. Computing the actual privacy cost numerically is also challenging due to the large space of continuous value representations such as FP32. To address this challenge, we bound the privacy loss in CDP via a rounding argument (we treat the multivariate discrete case separately in Appendix F). The high-level intuition is that the divergence of the staircase noise is monotone in each dimension's shift (Appendix E). Thus, we round each dimension's shift up to the next multiple of m, so that Theorem 2 applies coordinate-wise, and the privacy loss of the staircase noise is bounded by that of the original mechanism under a slightly inflated sensitivity. We can then maintain the original privacy target by calibrating the noise scale to the inflated sensitivity.

Theorem 3 (DP Guarantee of Multivariate Staircase Continuous Noise)**.** *If the original continuous noise $\mathcal{D}$ satisfies $\frac{1}{2}\varepsilon'^2$-CDP under the inflated L_p sensitivity $(1+c)\Delta_p$, then the staircase noise $\mathcal{D}_s$ satisfies $\frac{1}{2}\varepsilon'^2$-CDP under the original sensitivity Δ_p, if the bucket width m satisfies $m = c\Delta_p/d^{1/p}$ with $c > 0$, and $\mathcal{D}$'s PDF is symmetric and monotone, i.e., $p(x) = p(-x)$ and $p(x) \geq p(y)$, $\forall y \geq x \geq 0$.*

With Theorem 3 (see proof in Appendix G), the staircase noise inherits the original mechanism's guarantee at the slightly inflated sensitivity $(1+c)\Delta_p$. The scaling of the CDP parameter with the sensitivity depends on the noise family: for the Gaussian and Laplace mechanisms it is linear, so $\varepsilon' = (1+c)\varepsilon$ and the total privacy loss is $(1+c)\varepsilon$. The CDF table size is linear in $1/c$. Thus, c can be set to balance privacy guarantees and sampling efficiency. To achieve a target ε, we calibrate the base noise to satisfy $\frac{1}{2}\varepsilon^2$-CDP at the inflated sensitivity $(1+c)\Delta_p$ and then construct the staircase distribution and the CDF table accordingly. For Gaussian, this amounts to choosing the noise scale for $\frac{\varepsilon}{1+c}$ at sensitivity Δ_p.

The continuous noise sampling protocol NOISETTE-CONT is similar to the discrete noise sampling protocol NOISETTE-SC. We first sample $x = km, k \in \mathbb{Z}_{\geq 0}$ from the staircase CDF table, where k represents the sampled bucket index. Then we sample a value from $[0, 1)$. In a concrete implementation, real values are not representable by finite computers, and we sample a B-bit integer U using Π_{rand}. Then we compute the product of the sampled uniform and the public m to obtain $u = m \cdot U/2^B \in [0, m)$. For symmetric noise optimization, we sample a sign indicator $s \in \{-1, 1\}$, and compute the continuous staircase noise as $z = s \cdot (x + u)$. We omit this step for clarity in

Figure 8. We discuss the FP sampling implementation and DP analysis that gives an end-to-end DP guarantee under FP arithmetic in the following section.

5.4 Rigorous DP under FP32 Arithmetic

Naive floating-point implementations of continuous DP mechanisms leak information through low-order mantissa bits and intermediate rounding [Mir12], breaking the formal (ε, δ)-DP guarantee. Noisette-Cont closes both channels directly. The uniform offset is drawn as $U/2^B$ from a B-bit integer grid, and the addition $f + km + Um/2^B$ of the query result f and the noise $km + Um/2^B$ is evaluated in an error-free FP32 expansion that rounds exactly once at the end. We choose the bucket width such that $m = 2^{-a}, a \in \mathbb{Z}$, and the resulting deviation from the ideal continuous mechanism can be captured by the following total-variation bound, which we absorb into the δ parameter of the final (ε, δ)-DP guarantee. We present the detailed proofs and FP32 expansion implementation and analysis in Appendix H.

Theorem 4 (TV bound for the B-bit FP32 implementation)**.** *Let $|a| \le A$ and $B \ge 23$. Under IEEE-754 round-to-nearest, ties-to-even, for any FP32 query result $f \in \mathbb{F}_{32}$ within the finite FP32 range, the total-variation distance between the ideal continuous-staircase mechanism's output and its FP32 implementation with B-bit uniform discretization is at most*

$$\eta_B \;:=\; 2^{24-B} + H_{A,B}\, 2^{23-B} + \left(2H_{A,B} + \tfrac{1}{2}\right) 2^{-B},$$

where $H_{A,B} := \min\{128 + A,\ B - 21\}$.

Drawing N samples thus inherits the ideal (ε, δ)-DP guarantee up to an extra $(1 + e^{\varepsilon})\, N\eta_B$ on δ (Corollary 3). Selecting a large enough B can make this error term sufficiently small. To our knowledge, Noisette-Cont is the first certifiable FP DP sampler with a rigorous end-to-end (ε, δ)-DP guarantee under standard FP32 arithmetic, addressing the precision-induced privacy loss identified by prior work [CBJX26] directly rather than relying on external mitigation [Mir12] or leaving it unsolved [STC+24].

5.5 Security and Privacy Analysis

The protocols Noisette-SC and Noisette-Cont are constructed by invoking Noisette-Base on a reduced public CDF table T to certify sampling of the row $(\boldsymbol{x}'_i, \boldsymbol{b}_i)$, and then applying a (proved-in-ZK) post-processing step that draws an independent uniform offset $\boldsymbol{u}_i \in [0, m)$ (and a sign bit s_i) via $\mathcal{F}_{\mathsf{rand}}$ and outputs $\boldsymbol{x}_i = \boldsymbol{x}'_i + \boldsymbol{b}_i \boldsymbol{u}_i$ (and $s_i \cdot \boldsymbol{x}_i$). Hence, in the $(\mathcal{F}_{\mathsf{ZK}}, \mathcal{F}_{\mathsf{rand}})$-hybrid model, completeness, soundness, and zero-knowledge follow exactly as in § 4. For DP guarantees, the only substantive change relative to Noisette-Base is replacing $\mathcal{D}$ by its staircase approximation, whose guarantees are captured by Theorem 2 and Theorem 3: Noisette-SC preserves the original $\frac{1}{2}\varepsilon^2$-CDP, while Noisette-Cont inherits the original mechanism's guarantee at the inflated sensitivity $(1 + c)\Delta_p$, which we compensate by calibrating the base noise to the inflated sensitivity (for Gaussian, scaling the noise target from ε to $\varepsilon/(1 + c)$). Combining the staircase CDP adjustment with the truncation and certification errors of Corollary 1 and the FP32 TV bound of § 5.4, all residual errors are absorbed as an additive increase to the final δ, as in Corollary 3. Appendix I gives the concrete budget split and the parameters used in our evaluation.

6 Evaluation

In § 6.1, we benchmark the performance of Noisette-Base and evaluate its scalability w.r.t. the number of samples and the CDF table size. We then compare Noisette-Base with state-

of-the-art (SOTA) baselines on discrete Gaussian and Laplace sampling. In § 6.2, we show that the staircase optimizations in Noisette-SC and Noisette-Cont further improve efficiency and enable continuous DP noise sampling with the same privacy guarantees and minor utility influence. We consider two widely deployed real-world DP applications: mean estimation [DMNS06b] and federated learning (FL) [AKL21].

Evaluation setup. We implement Noisette using EMP-toolkit [WMK16] for the interactive zero-knowledge interface. We benchmark Noisette-Base on an AWS `c5.xlarge` instance, and evaluate Noisette-SC and Noisette-Cont in the DP applications on `c6i.32xlarge`. We consider two network settings:

- LAN: Bandwidth up to 1 Gbps and latency 0.1 ms.
- WAN: Bandwidth up to 100 Mbps and latency 100 ms.

Baselines. As mentioned in § 1, most prior certifiable DP work targets a specific noise distribution. To certify complex distributions such as discrete Gaussian and Laplace, the most efficient existing approach besides our lookup-based technique is rejection sampling, for which the MPC literature provides the most optimized oblivious circuits. Because these circuits are data-oblivious, they adapt directly to the ZKP setting; we instantiate them with the same ZKP primitive as Noisette, yielding the strongest available baselines. For each distribution, we consider two such baselines: (1) the SOTA sampling protocols for discrete Gaussian and Laplace proposed in [WYF+23], denoted as Prior-Disc; and (2) the Bernoulli sampling protocol from [CsU19], instantiated with an oblivious stack (Ostack) and combined with Prior-Disc following [FW24], denoted as Prior-Disc-Ostack.

For continuous cases, we compare Noisette-Cont against the SOTA approach for certifiable floating-point Gaussian sampling [STC+24], denoted as Prior-Cont.

6.1 Discrete DP Noise Sampling Benchmark

We first benchmark Noisette-Base, then compare it with Prior-Disc and Prior-Disc-Ostack on discrete Gaussian and Laplace sampling. Since the staircase optimizations influence utility, we evaluate Noisette-SC and Noisette-Cont in concrete applications in § 6.2.

6.1.1 Scalability of Noisette-Base

We evaluate the runtime and communication cost of Noisette-Base w.r.t. the number of samples and the table size.

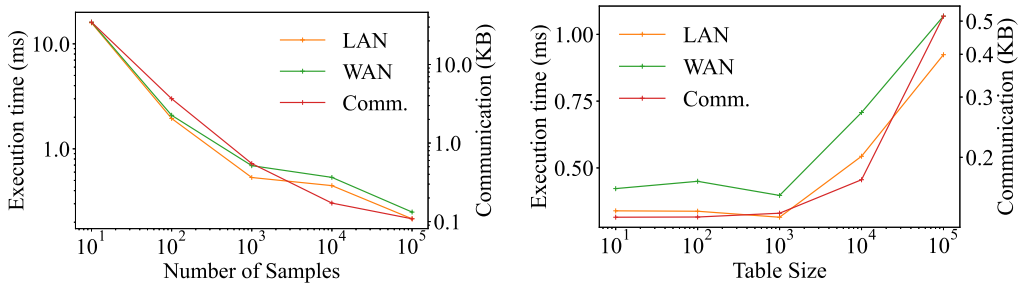
Scalability w.r.t. number of samples. We benchmark Noisette-Base on a discrete distribution with CDF table size $M = 10^4$ and 64-bit probability entries. As shown in Figure 9a, the per-sample runtime and communication costs decrease as the sample number N increases, reflecting the amortization analyzed in § 4.

Scalability w.r.t. table size. Figure 9b illustrates the cost of sampling $N = 10^4$ values from discrete distributions with varying CDF table sizes M. The cost scales linearly with M, driven by the linear increase in $\mathbb{F}_{ZK}$ multiplications.

6.1.2 Comparison With Prior Work

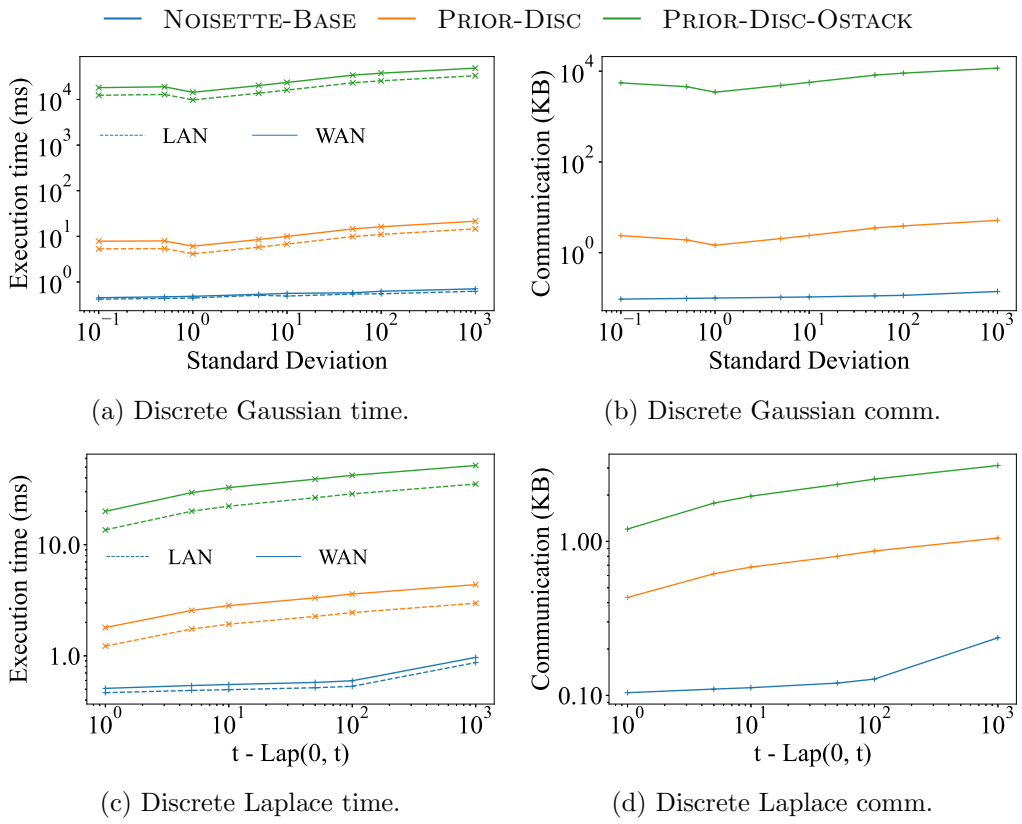
We compare Noisette-Base with Prior-Disc and Prior-Disc-Ostack, obtaining the AND gate counts of their oblivious circuits from the prior MPC implementation[1] to compute their costs. Figure 10 presents the comparison in both network settings with $N = 2^{14}$.

[1] https://github.com/yuchengxj/Secure-sampling-benchmark



(a) Varying sample numbers. (b) Varying table sizes.

Figure 9: Benchmarks of NOISETTE-BASE: per-sample runtime (ms) and communication (KB) when **(a)** sampling from a distribution with CDF table size $M = 10^4$, **(b)** drawing $N = 10^4$ samples with varying table sizes.



(a) Discrete Gaussian time. (b) Discrete Gaussian comm.

(c) Discrete Laplace time. (d) Discrete Laplace comm.

Figure 10: Comparison with SOTA baselines. Table size is determined by noise scales. Number of samples $N = 2^{14}$.

Discrete Gaussian. NOISETTE-BASE outperforms both rejection sampling protocols in all settings (Figure 10a and 10b), demonstrating a 10–30× runtime improvement and 10–36× communication reduction compared to PRIOR-DISC. Specifically, for $\sigma = 1000$ in the LAN setting, our protocol requires 23× less time and 36× less communication than PRIOR-DISC. Furthermore, we achieve a 52× (resp. 83×) improvement in runtime (resp. communication) compared to discrete Gaussian sampling with Ostack.

Discrete Laplace. Noisette-Base also significantly outperforms Prior-Disc for discrete Laplace distributions. For $t = 100$, Noisette-Base yields a 6× improvement in both time and communication. Compared to Prior-Disc-Ostack, we observe a 17× improvement in runtime and 19× in communication.

As expected, for both discrete Gaussian and Laplace, Noisette-Base greatly outperforms prior work by avoiding large circuits for complex noise sampling, while set inclusion executes efficiently and further benefits from a large number of samples.

6.2 Evaluation on Concrete Applications

We evaluate the utility and efficiency of Noisette on two widely deployed DP applications: privacy-preserving mean estimation and differentially private federated learning (DPFL). We measure runtime and communication costs alongside utility metrics to assess the utility-efficiency tradeoff. Detailed settings for each use case are provided below.

6.2.1 Privacy Preserving Mean Estimation

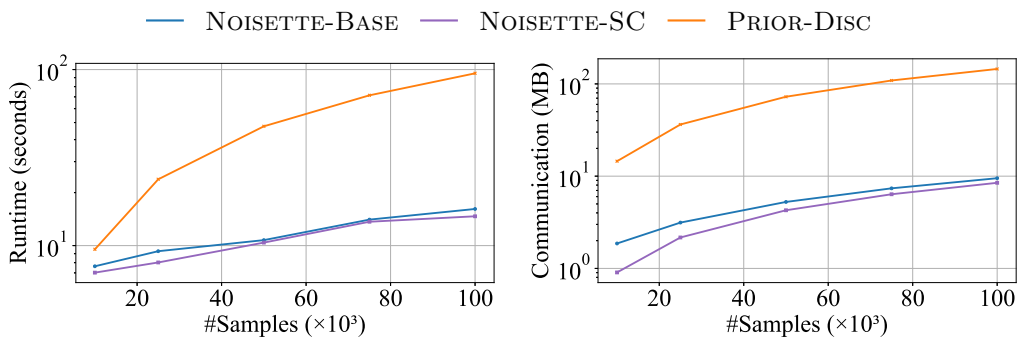
Privacy-preserving mean estimation is a fundamental application of DP, widely used in real-world scenarios and as a benchmark for DP protocols [FW24, AKL21]. We evaluate the utility and efficiency of Noisette-Base, Noisette-SC, and Noisette-Cont for mean estimation with discrete and continuous Gaussian mechanisms. We measure scalability w.r.t. the number of noise samples and the privacy budget (table size), and analyze the impact of the staircase bucket width m to demonstrate the utility-efficiency tradeoff of our staircase optimizations.

Application setup. To simulate a real-world scenario, we consider s datasets $X_i \in \mathbb{Z}^{h \times d}, i \in [s]$ representing smart meter data collected from s neighborhoods. Each neighborhood contains h devices, where each data point is a d-dimensional vector. Smart meters commit these data samples and send them to an electricity company for aggregation. The aggregator computes the average of the collected data and applies DP mechanisms before releasing the aggregated electricity usage statistics. Crucially, the aggregator proves to a regulator that the DP mechanism is executed correctly, ensuring a certifiable privacy guarantee of the released statistics. In total, the aggregator generates $s \cdot d$ DP noise samples.

We synthesize each dataset by randomly sampling $h = 10^4$ data points from a multivariate Gaussian $\mathcal{N}(0, I_d)$ with $d = 10^3$, normalizing each feature to $[0, 1]$. For discrete scenarios, we quantize the synthetic data using fixed-point with a scale factor of 2^4 for all dimensions. To evaluate utility, we measure the mean squared error (MSE) between the certified DP mean estimation and the ground truth mean. To evaluate efficiency, we measure the runtime and communication costs of Noisette in the WAN setting. We primarily evaluate *continuous* and *discrete* Gaussian mechanisms given their widespread adoption; however, these findings generalize to any additive noise DP mechanism.

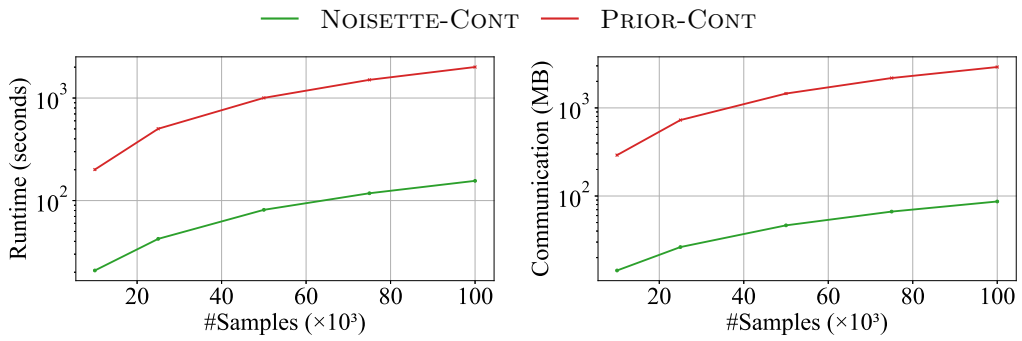
Scalability w.r.t. noise sample numbers. The total number of noise samples, $s \times d$, is a key factor influencing sampling efficiency. We evaluate this by measuring runtime and communication costs across different s. We generate discrete and continuous Gaussian noise to ensure an (ε, δ)-DP guarantee, using $\varepsilon = 0.5, \delta = 10^{-6}$. We compare the costs against SOTA baselines Prior-Disc [CsU19] and Prior-Cont [STC+24], with results presented in Figure 11.

In Figure 11a and Figure 11b, we compare the runtime and communication costs of Noisette against the prior SOTA baseline for discrete Gaussian sampling [CsU19]. We set the staircase bucket size to $m = 16$ (selection of hyper-parameters is discussed in the following section). Consistent with § 6.1, with a larger number of samples, our protocols are significantly more efficient than prior work,



(a) Discrete Gaussian time.

(b) Discrete Gaussian comm.



(c) Continuous Gaussian time.

(d) Continuous Gaussian comm.

Figure 11: Runtime and communication vs. sample numbers for discrete and continuous Gaussian in mean estimation.

achieving approximately 7× faster runtime and 17× lower communication. For small sample sizes, the gain of our protocol becomes smaller, because the cost is dominated by the CDF table size and the per-sample cost is not well-amortized; in the large sample regime, our techniques efficiently batch large numbers of samples. We also observe that the staircase optimization further improves performance, particularly when the sample number is small. As expected, with increasing sample numbers, the savings of staircase optimizations are less significant, as the sample number dominates the overall costs.

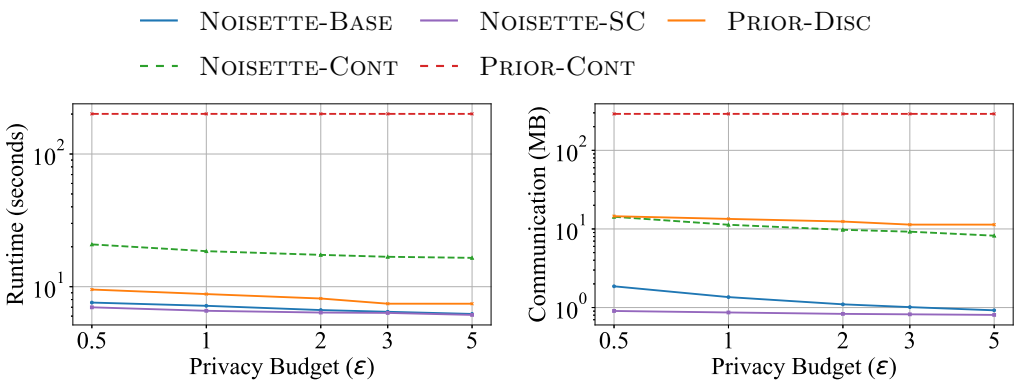
Continuous Gaussian sampling shows similar trends (Figure 11c and Figure 11d): with $m = 2^{-6}$, Noisette-Cont is consistently ∼13× faster than Prior-Cont with 34× less communication.

Scalability w.r.t. privacy budget ε. ε controls the utility, privacy, and efficiency tradeoff. A smaller ε implies stronger privacy protection, requiring a larger noise scale and CDF table, and increases sampling costs. We analyze the sampling efficiency and utility of our protocols under various ε, with 10^4 noise samples.

As expected, increasing ε reduces the CDF table size, thereby lowering the cost of our sampling protocols (Figure 12). While the sampling cost of prior work remains relatively consistent regardless of noise scale, our protocols demonstrate even more significant savings as ε increases.

We further evaluate the utility of our protocols in Table 3. We achieve utility comparable to DP noise without certification, with less than 1% deviation for Noisette-SC and less than 3% for Noisette-Cont across different privacy regimes ε.

Performance w.r.t. staircase bucket width m. As discussed in § 5, the bucket width m



(a) Runtime Cost (b) Communication Cost

Figure 12: Runtime and communication cost vs. ε for discrete and continuous Gaussian in mean estimation.

Mechanism	MSE ($\times 10^{-4}$)		
	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$
Discrete Gaussian	7.58	2.06	0.565
w/ Noisette-Base	7.52	2.05	0.567
w/ Noisette-SC (m=8)	7.49	2.07	0.569
Continuous Gaussian	7.53	2.05	0.568
w/ Noisette-Cont (m=2^{-7})	7.73	2.07	0.577

Table 3: Utility (MSE) of mean estimation for different DP mechanisms under various ε with $\delta = 10^{-6}$.

balances the utility and efficiency tradeoff. With $\varepsilon = 0.1$, $\delta = 10^{-6}$, and 10^3 certified noise samples, we measure the cost (Figure 13) and the MSE (Table 4) under different m. As expected, increasing m significantly reduces runtime and communication, because a larger m reduces the CDF table size, making lookups more efficient. For Noisette-SC, increasing m has minimal influence on utility, yielding results comparable to discrete Gaussian without staircase optimizations. For Noisette-Cont, the staircase modifies the distribution more as m increases, and we can choose a slightly smaller m to achieve utility comparable to the original Gaussian without certification. In practice, m can be tuned to specific applications to achieve the desired balance.

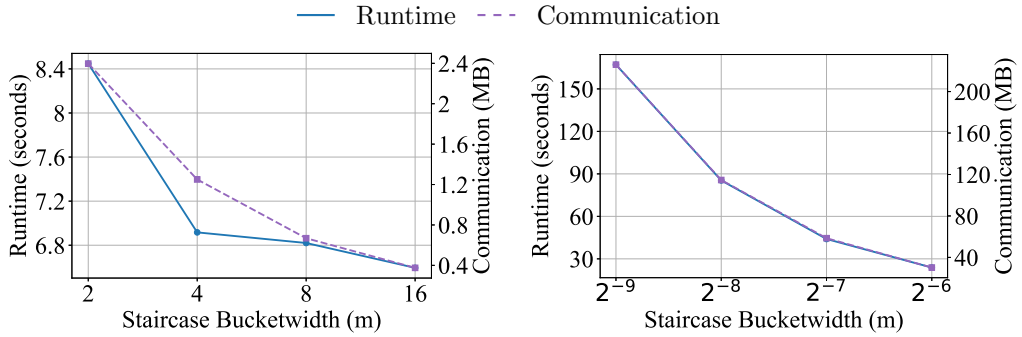
6.2.2 Differentially Private Federated Learning

DPFL has emerged as a crucial real-world application of DP, with widespread adoption by industry leaders such as Apple and Google for protecting user data privacy [HRM+18, PSM+21]. Modern DPFL increasingly fine-tunes pretrained language models with low-rank adapters such as LoRA [HysW+22], which reduces both the optimization cost and the DP noise dimension. We evaluate Noisette by certifying the Gaussian noise sampling during DPFL-LoRA fine-tuning of a Transformer language model.

Application setup. We simulate FL with 20 clients training locally on private data and a central server responsible for aggregation and DP noise sampling. The clients fine-tune RoBERTa-

Discrete		Continuous	
m	MSE ($\times 10^{-2}$)	m	MSE ($\times 10^{-2}$)
—	1.56	—	1.72
2	1.56	2^{-9}	1.73
4	1.56	2^{-8}	1.73
8	1.55	2^{-7}	1.75
16	1.56	2^{-6}	1.80

Table 4: Utility (MSE) of mean estimation using NOISETTE-SC and NOISETTE-CONT under varying staircase bucket width m, with $\varepsilon = 0.1$, $\delta = 10^{-6}$, $s = 1$, and $d = 10^3$. "—" indicates the original discrete and continuous Gaussian mechanism.



(a) NOISETTE-SC (b) NOISETTE-CONT

Figure 13: Runtime and communication cost of NOISETTE-SC and NOISETTE-CONT vs. staircase bucket width m with $\varepsilon = 0.1$, $\delta = 10^{-6}$, and 10^3 noise samples.

base [LOG+19] with standard LoRA: the backbone is frozen, and only the rank-r LoRA adapters and the classification head are trained. The trainable dimension is 887,042 at the default rank $r = 8$ (739,586 at $r = 4$, and 1,181,954 at $r = 16$). We use the sentence sentiment dataset SST-2 [WSM+19], randomly distributed across the clients (3,367 samples each); all clients participate every round.

Each client computes per-sample clipped gradients on its local batch, commits to their sum, and sends both the commitment and the sum to the server. The server aggregates the per-client sums into a single gradient vector, whose L_2 norm is bounded, adds DP Gaussian noise via NOISETTE, and broadcasts the averaged update; the regulator verifies the proof of correct noise sampling. We train for 500 rounds.

Utility results. We evaluate the model accuracy on the test dataset under different privacy budgets for floating-point Gaussian and NOISETTE-CONT with various parameters m and LoRA rank $r = 8$ in Table 5; all results are averaged over three independent runs. Across different privacy budgets, the accuracy of NOISETTE-CONT is comparable with floating-point Gaussian, and choosing a smaller m increases accuracy because the staircase distortion is smaller. NOISETTE-CONT thus achieves negligible utility drop while preserving a rigorous end-to-end DP guarantee, whereas the baseline Gaussian noise sampled in a floating-point circuit has precision leakage. Similarly, Table 6 evaluates the model accuracy w.r.t. the LoRA rank r (the model size) with $m = 2^{-13}$ under $\varepsilon = 3$ and $\delta = 10^{-6}$; NOISETTE-CONT consistently achieves utility comparable to floating-point Gaussian.

Cost results. Table 7 reports the per-round ZK cost of sampling and applying FP Gaussian noise

Mechanism	Accuracy (%)		
	$\varepsilon = 1$	$\varepsilon = 3$	$\varepsilon = 5$
Continuous Gaussian	89.3±0.7	89.9±0.4	90.4±0.3
Noisette-Cont			
$m = 2^{-13}$	88.8±0.4	89.8±0.8	90.3±0.6
$m = 2^{-11}$	88.2±0.2	89.5±0.8	89.8±0.7
$m = 2^{-9}$	87.4±0.3	88.7±0.3	89.3±0.7

Table 5: Model accuracy under different DP privacy budgets ε for floating-point Gaussian and Noisette-Cont Gaussian with different hyper-parameters m.

Mechanism	Accuracy (%)		
	$r = 4$	$r = 8$	$r = 16$
Continuous Gaussian	90.4±0.4	89.9±0.4	89.3±0.5
Noisette-Cont	90.0±0.5	89.8±0.8	89.3±1.3

Table 6: Model accuracy under different adapter ranks for floating-point Gaussian and Noisette-Cont Gaussian with $m = 2^{-13}$ under $(\varepsilon = 3, \delta = 10^{-6})$-DP.

to the committed gradient aggregate under different ε with LoRA rank $r = 8$. Across different privacy budgets, the cost of Prior-Cont stays constant because its Box-Muller sampler draws one floating-point sample per coordinate regardless of σ; the cost of Noisette-Cont is also nearly invariant, as the large sample number dominates the small table-size changes. Noisette-Cont reduces both the per-round runtime and communication by 28–29× over Prior-Cont across all (ε, m) cells. As expected, choosing a larger m further reduces the cost of Noisette-Cont, as the CDF table shrinks.

Similarly, in Table 8, we evaluate the per-round cost w.r.t. the LoRA rank r (the model size) for Prior-Cont and Noisette-Cont with $m = 2^{-13}$ under $\varepsilon = 3$ and $\delta = 10^{-6}$. Both protocols scale linearly with the number of trainable parameters, and Noisette-Cont remains 27–28× faster and ∼28× smaller in communication across all three ranks. Concretely, at $r = 16$ Noisette-Cont completes one ZK round in ∼211 seconds with 1.4 GB of communication, versus ∼1.7 hours and ∼40 GB for Prior-Cont. These results show that Noisette-Cont achieves orders-of-magnitude improvements in runtime and communication while preserving a rigorous end-to-end DP guarantee and utility comparable to the original continuous Gaussian.

7 Related Work

Our work intersects with several areas of research in cryptography and privacy-preserving machine learning. We organize the related work into two main categories: certifying differential privacy and differential privacy in MPC.

Certifying Differential Privacy. Prior work has explored the intersection of differential privacy and cryptographically verifiable computation. [STC+24] proposed a framework for certifying the training of DPSGD using the Box-Muller transform [BM58] to sample floating-point Gaussian noise, which involves costly computations like square root and log functions. [NFPH15] also certifies

	Time (s)			Comm. (MB)		
	$\varepsilon = 1$	$\varepsilon = 3$	$\varepsilon = 5$	$\varepsilon = 1$	$\varepsilon = 3$	$\varepsilon = 5$
Prior-Cont	4493.4			30834		
Noisette-Cont						
m=2^{-13}	162.4	162.0	161.7	1119.6	1099.0	1092.9
m=2^{-11}	160.9	155.2	157.0	1077.4	1062.4	1060.7
m=2^{-9}	154.9	157.3	155.8	1057.6	1055.2	1054.5

Table 7: Per-round ZK cost of Noisette-Cont Gaussian under different privacy budgets ε and hyper-parameters m. Cost of prior continuous Gaussian is ε invariant.

Protocol	Metric	$r = 4$	$r = 8$	$r = 16$
Prior-Cont	Time (s)	3746.5	4493.4	5987.3
	Comm. (MB)	25709	30834	41086
Noisette-Cont	Time (s)	138.0	161.7	210.9
	Comm. (MB)	929.8	1099.0	1446.1

Table 8: Per-round ZK cost under different LoRA adapter ranks r for Noisette-Cont Gaussian with $m = 2^{-13}$ under $(\varepsilon = 3, \delta = 10^{-6})$-DP.

Laplace noise sampling, which involves evaluating complex log functions in ZK.

Recent work [BGKW24, BAT24] proposed protocols for certifying simple discrete noise distributions like Binomial, but left the certification of more complex distributions like discrete Laplace and Gaussian as an open challenge. Noisette addresses this gap by providing a generic protocol for certifying arbitrary discrete noise sampling.

Another line of work [NST+21, NHS+23] focuses on analyzing the tightness and correctness of the DP algorithms, especially for DP-SGD. Their methods can quantify the robustness of the DP algorithms and the correctness of the implementations, however, their results are not cryptographically provable. Their verification usually considers a strong adversary that can perform membership inference attacks and assumes access to sensitive datasets, making the auditing methods inapplicable to privacy-sensitive scenarios.

Differential Privacy in MPC. Local DP requires each client to add some noise while central DP requires the clients to trust the server for adding the noise. To get the best of both worlds, several works propose the use of MPC for sampling noise from specific distributions. Dwork et al. [DKM+06] describe an efficient method for sampling from a geometric distribution using securely sampled biased coins. Champion et al. [CsU19] propose a method for securely sampled biased coins based on an efficient oblivious stack protocol. Wei et al. [WYF+23] design a rejection sampling protocol for the discrete Gaussian distribution based on the method from [CKS20], using geometric sampling from [DKM+06]. Fu and Wang [FW24] present a comprehensive study of differential privacy techniques in MPC. They also evaluate the rejection sampling protocol of [WYF+23] with oblivious-stack biased-coin sampling [CsU19].

Keller et al. [KMS+24] propose algorithms for sampling from discrete Laplace and Gaussian distributions, using the snapping mechanism to handle privacy issues caused by floating-point operations and integer scaling mechanisms for both distributions. Olive [FFR+25] proposes a generic MPC protocol for DP mechanisms using lookup tables, where the distribution is truncated,

represented as a table, and sampled by choosing a random table entry with equal probability. However, they use threshold homomorphic encryption to realize their algorithm efficiently in MPC, making it non-trivial to adapt their approach to a ZK setting.

The approaches from prior work that use an MPC interface to evaluate circuits for efficient sampling can be adapted to the ZK setting to achieve certifiable DP by using a ZK interface to prove circuit satisfiability. However, these circuits are typically large and thus, do not scale well for a large number of samples. The closest related work is the independent preprint of Meisingseth et al. [MRS25], which also proposes lookup-table-based DP noise sampling but targets MPC instead of ZKP. There are two structural differences. First, they approximate the table-index distribution by Bernoulli draws inside MPC, whereas Noisette samples directly via inverse transform on the CDF; this lets us batch a single set-inclusion check across many samples and amortize the per-sample cost. Second, they handle only discrete distributions, while Noisette supports continuous FP32 sampling with a rigorous DP guarantee.

8 Conclusion

We propose Noisette, a suite of powerful protocols for certifying DP noise sampling that support any additive continuous or discrete DP distribution. In contrast to prior work, which focuses on optimizing specific mechanisms, Noisette provides a general and unifying framework that achieves order-of-magnitude improvements in both computational efficiency and communication overhead. For the continuous case, Noisette-Cont provides a rigorous end-to-end (ε, δ)-DP guarantee under standard FP32 arithmetic, eliminating the precision-induced privacy loss that prior certifiable continuous DP samplers ignored. We validate its practicality on real-world DP applications, including mean estimation and federated learning for privacy-preserving language model training. Our results establish Noisette as the first efficient, scalable, and general-purpose solution for certifiable DP noise sampling.

References

[Abo16] John M Abowd. The challenge of scientific reproducibility and privacy protection for statistical agencies. *Census Scientific Advisory Committee*, pages 1011–1020, 2016.

[ACG+16a] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[ACG+16b] Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *ACM Conf. on Computer and Communications Security (CCS) 2016*, pages 308–318. ACM Press, 2016.

[AKL21] Naman Agarwal, Peter Kairouz, and Ziyu Liu. The skellam mechanism for differentially private federated learning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 5052–5064, 2021.

[BAT24] Tariq Bontekoe, Hassan Jameel Asghar, and Fatih Turkmen. Efficient verifiable differential privacy with input authenticity in the local and shuffle model. Cryptology ePrint Archive, Report 2024/1042, 2024.

[BBMH+21] Carsten Baum, Lennart Braun, Alexander Munch-Hansen, Benoît Razet, and Peter Scholl. Appenzeller to brie: Efficient zero-knowledge proofs for mixed-mode arithmetic and Z2k. In *ACM Conf. on Computer and Communications Security (CCS) 2021*, pages 192–211. ACM Press, 2021.

[BBMHS22] Carsten Baum, Lennart Braun, Alexander Munch-Hansen, and Peter Scholl. Moz$\mathbb{Z}_{2^k}$arella: Efficient vector-OLE and zero-knowledge proofs over $\mathbb{Z}_{2^k}$. In *Advances in Cryptology—Crypto 2022, Part IV*, LNCS, pages 329–358, 2022.

[BC23] Ari Biswas and Graham Cormode. Interactive proofs for differentially private counting. In *ACM Conf. on Computer and Communications Security (CCS) 2023*, pages 1919–1933. ACM Press, 2023.

[BGKW24] Zoë Ruha Bell, Shafi Goldwasser, Michael P Kim, and Jean-Luc Watson. Certifying private probabilistic mechanisms. In *Annual International Cryptology Conference*, pages 348–386. Springer, 2024.

[BM58] George EP Box and Mervin E Muller. A note on the generation of random normal deviates. *The annals of mathematical statistics*, 29(2):610–611, 1958.

[BMRS21] Carsten Baum, Alex J. Malozemoff, Marc B. Rosen, and Peter Scholl. Mac'n'cheese: Zero-knowledge proofs for boolean and arithmetic circuits with nested disjunctions. In *Advances in Cryptology—Crypto 2021, Part IV*, LNCS, pages 92–122, 2021.

[BW18] Borja Balle and Yu-Xiang Wang. 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*, volume 80 of *Proceedings of Machine Learning Research*, pages 403–412. PMLR, 2018.

[CBJX26] Rishav Chourasia, Ergute Bao, Uzair Javaid, and Xiaokui Xiao. Auditing Apple's DifferentialPrivacy.framework: Implementation Bugs, Misconfigurations, and Practical Risks . In *2026 IEEE Symposium on Security and Privacy (SP)*, pages 1815–1834, Los Alamitos, CA, USA, May 2026. IEEE Computer Society.

[CFG24] Matteo Campanelli, Dario Fiore, and Rosario Gennaro. Natively compatible super-efficient lookup arguments and how to apply them. Cryptology ePrint Archive, Report 2024/1058, 2024.

[Che52] Herman Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, pages 493–507, 1952.

[CHM+20] Alessandro Chiesa, Yuncong Hu, Mary Maller, Pratyush Mishra, Psi Vesely, and Nicholas P. Ward. Marlin: Preprocessing zkSNARKs with universal and updatable SRS. In *Advances in Cryptology—Eurocrypt 2020, Part I*, volume 12105 of *LNCS*, pages 738–768, 2020.

[CKS20] Clément L Canonne, Gautam Kamath, and Thomas Steinke. The discrete gaussian for differential privacy. *Advances in Neural Information Processing Systems*, 33:15676–15688, 2020.

[CsU19] Jeffrey Champion, abhi shelat, and Jonathan Ullman. Securely sampling biased coins with applications to differential privacy. In *ACM Conf. on Computer and Communications Security (CCS) 2019*, pages 603–614. ACM Press, 2019.

[DILO22] Samuel Dittmer, Yuval Ishai, Steve Lu, and Rafail Ostrovsky. Improving line-point zero knowledge: Two multiplications for the price of one. In *ACM Conf. on Computer and Communications Security (CCS) 2022*, pages 829–841. ACM Press, 2022.

[DKM+06] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology—Eurocrypt 2006*, LNCS, pages 486–503, 2006.

[DMNS06a] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *3rd Theory of Cryptography Conference—TCC 2006*, volume 3876 of *LNCS*, pages 265–284, 2006.

[DMNS06b] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[DPT] Apple Differential Privacy Team. Apple differential privacy technical overview. `https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf`.

[DPT17] Apple Differential Privacy Team. Learning with privacy at scale. `https://machinelearning.apple.com/research/learning-with-privacy-at-scale`, 2017.

[DR+14] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and trends® in theoretical computer science*, 9(3–4):211–407, 2014.

[FFR+25] Olive Franzese, Congyu Fang, Radhika, Xiao Wang, Somesh Jha, Nicolas Papernot, and Adam Dziedzic. Secure noise sampling for differentially private collaborative learning. In *Proceedings of the 32nd ACM Conference on Computer and Communications Security*, 2025.

[ftc11] Ftc charges deceptive privacy practices in google's rollout of its buzz social network. https://www.ftc.gov/news-events/news/press-releases/2011/03/ftc-charges-deceptive-privacy-practices-googles-rollout-its-buzz-social-network, 2011.

[FW24] Yucheng Fu and Tianhao Wang. Benchmarking secure sampling protocols for differential privacy. In *ACM Conf. on Computer and Communications Security (CCS) 2024*, pages 318–332. ACM Press, 2024.

[goo] Google differential privacy library. `https://github.com/google/differential-privacy`.

[Gro16] Jens Groth. On the size of pairing-based non-interactive arguments. Cryptology ePrint Archive, Report 2016/260, 2016.

[GRS08] Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. Universally utility-maximizing privacy mechanisms. *CoRR*, abs/0811.2841, 2008.

[GV15] Quan Geng and Pramod Viswanath. The optimal noise-adding mechanism in differential privacy. *IEEE Transactions on Information Theory*, 62(2):925–951, 2015.

[GW20] Ariel Gabizon and Zachary J Williamson. plookup: A simplified polynomial protocol for lookup tables. *Cryptology ePrint Archive*, 2020.

[GWC19] Ariel Gabizon, Zachary J. Williamson, and Oana Ciobotaru. PLONK: Permutations over Lagrange-bases for oecumenical noninteractive arguments of knowledge. Cryptology ePrint Archive, Report 2019/953, 2019.

[HRM+18] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

[HysW+22] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[KMS+24] Hannah Keller, Helen Möllering, Thomas Schneider, Oleksandr Tkachenko, and Liang Zhao. Secure noise sampling for dp in mpc with finite precision. In *Proceedings of the 19th International Conference on Availability, Reliability and Security*, pages 1–12, 2024.

[LOG+19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[LPDV23] Johan Lokna, Anouk Paradis, Dimitar I Dimitrov, and Martin Vechev. Group and attack: Auditing differential privacy. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 1905–1918, 2023.

[LV06] Friedrich Liese and Igor Vajda. On divergences and informations in statistics and information theory. *IEEE Transactions on Information Theory*, 52(10):4394–4412, 2006.

[met23] Meta required to report platform statistics to third-party auditors. `https://www.justice.gov/crt/case/united-states-v-meta-platforms-inc-fka-facebook-inc-sdny`, 2023.

[Mir12] Ilya Mironov. On significance of the least significant bits for differential privacy. In *ACM Conf. on Computer and Communications Security (CCS) 2012*, pages 650–661. ACM Press, 2012.

[MKA+08] Ashwin Machanavajjhala, Daniel Kifer, John Abowd, Johannes Gehrke, and Lars Vilhuber. Privacy: Theory meets practice on the map. In *2008 IEEE 24th international conference on data engineering*, pages 277–286. IEEE, 2008.

[MRS25] Fredrik Meisingseth, Christian Rechberger, and Fabian Schmid. Accelerating multiparty noise generation using lookups. *Cryptology ePrint Archive*, 2025.

[NFPH15] Arjun Narayan, Ariel Feldman, Antonis Papadimitriou, and Andreas Haeberlen. Verifiable differential privacy. In *Proceedings of the Tenth European Conference on Computer Systems*, pages 1–14, 2015.

[NHS+23] Milad Nasr, Jamie Hayes, Thomas Steinke, Borja Balle, Florian Tramèr, Matthew Jagielski, Nicholas Carlini, and Andreas Terzis. Tight auditing of differentially private machine learning. pages 1631–1648. USENIX Association, 2023.

[NRS07] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 75–84, 2007.

[NST+21] Milad Nasr, Shuang Song, Abhradeep Thakurta, Nicolas Papernot, and Nicholas Carlini. Adversary instantiation: Lower bounds for differentially private machine learning. In *IEEE Symp. Security and Privacy 2021*, pages 866–882. IEEE, 2021.

[otC21] United States. Bureau of the Census. *Disclosure Avoidance for the 2020 Census: An Introduction*. Department of Commerce, U.S. Census Bureau, 2021.

[PFM+22] Luke Pearson, Joshua Fitzgerald, Héctor Masip, Marta Bellés-Muñoz, and Jose Luis Muñoz-Tapia. PlonKup: Reconciling PlonK with plookup. Cryptology ePrint Archive, Report 2022/086, 2022.

[PSM+21] Matthias Paulik, Matt Seigel, Henry Mason, Dominic Telaar, Joris Kluivers, Rogier Van Dalen, Chi Wai Lau, Luke Carlson, Filip Granqvist, Chris Vandevelde, et al. Federated evaluation and tuning for on-device personalization: System design & applications. *arXiv preprint arXiv:2102.08503*, 2021.

[STC+24] Ali Shahin Shamsabadi, Gefei Tan, Tudor Ioan Cebere, Aurélien Bellet, Hamed Haddadi, Nicolas Papernot, Xiao Wang, and Adrian Weller. Confidential-dpproof: Confidential proof of differentially private training. In *International Conference on Learning Representations (ICLR)*, 2024.

[WMK16] Xiao Wang, Alex J. Malozemoff, and Jonathan Katz. EMP-toolkit: Efficient MultiParty computation toolkit. `https://github.com/emp-toolkit`, 2016.

[WSM+19] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019.

[WYF+23] Chengkun Wei, Ruijing Yu, Yuan Fan, Wenzhi Chen, and Tianhao Wang. Securely sampling discrete gaussian noise for multi-party differential privacy. In *ACM Conf. on Computer and Communications Security (CCS) 2023*, pages 2262–2276. ACM Press, 2023.

[WYY+22] Chenkai Weng, Kang Yang, Zhaomin Yang, Xiang Xie, and Xiao Wang. AntMan: Interactive zero-knowledge proofs with sublinear communication. In *ACM Conf. on Computer and Communications Security (CCS) 2022*, pages 2901–2914. ACM Press, 2022.

[YLBC25] Chhavi Yadav, Evan Laufer, Dan Boneh, and Kamalika Chaudhuri. Expproof : Operationalizing explanations for confidential models with ZKPs. In *Forty-second International Conference on Machine Learning*, 2025.

[YSWW21] Kang Yang, Pratik Sarkar, Chenkai Weng, and Xiao Wang. QuickSilver: Efficient and affordable zero-knowledge proofs for circuits and polynomials over any field. In *ACM Conf. on Computer and Communications Security (CCS) 2021*, pages 2986–3001. ACM Press, 2021.

[ZBK+22] Arantxa Zapico, Vitalik Buterin, Dmitry Khovratovich, Mary Maller, Anca Nitulescu, and Mark Simkin. Caulk: Lookup arguments in sublinear time. In *ACM Conf. on Computer and Communications Security (CCS) 2022*, pages 3121–3134. ACM Press, 2022.

A DP Mechanisms

We summarize the commonly used additive DP mechanisms [DR+14, CKS20, GRS08, BW18]; throughout, f has sensitivity Δ and the mechanism outputs $\mathcal{M}(x) = f(x) + Y$.
Laplace [DR+14]: $Y \sim Lap(t)$ with density $\frac{1}{2t}e^{-|y|/t}$, $y \in \mathbb{R}$; it satisfies ε-DP with $\varepsilon = \Delta/t$.
Gaussian [BW18]: $Y \sim \mathcal{N}(0, \sigma^2)$ with density $\frac{1}{\sqrt{2\pi}\sigma}e^{-y^2/(2\sigma^2)}$, $y \in \mathbb{R}$; it satisfies (ε, δ)-DP with $\delta = \Phi(\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}) - e^{\varepsilon}\Phi(-\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta})$, where Φ denotes the standard normal CDF.
Discrete Laplace [GRS08]: $Y \sim Lap_{\mathbb{Z}}(t)$ with PMF $\frac{e^{1/t}-1}{e^{1/t}+1} \cdot e^{-|k|/t}$, $k \in \mathbb{Z}$; it satisfies ε-DP with $\varepsilon = \Delta/t$.
Discrete Gaussian [CKS20]: $Y \sim \mathcal{N}_{\mathbb{Z}}(0, \sigma^2)$ with PMF $e^{-k^2/(2\sigma^2)} / \sum_{i\in\mathbb{Z}} e^{-i^2/(2\sigma^2)}$, $k \in \mathbb{Z}$; it satisfies (ε, δ)-DP with $\delta = \Pr[Y > \frac{\varepsilon\sigma^2}{\Delta} - \frac{\Delta}{2}] - e^{\varepsilon}\Pr[Y > \frac{\varepsilon\sigma^2}{\Delta} + \frac{\Delta}{2}]$.

B Table Construction.

The deterministic procedure $\Pi_{\mathsf{TConstruct}}(\mathcal{D}', N)$ constructs the public CDF table as follows. We first determine the truncation bound M from Table 2 and compute the PMF $p_i = \mathcal{D}'(\boldsymbol{x}_i)$ for $i \in [0, M]$; for symmetric distributions, we set $p_i = 2p_i$ for $i \in [1, M]$ and normalize $p_i = p_i / \sum_{j=0}^{M} p_j$. We then take partial sums $p_i^{max} = \sum_{j=0}^{i} p_j$ and $p_i^{min} = p_i^{max} - p_i$, and quantize the probabilities to n-bit fixed point (Lemma 2). The output is the public table $T = \{p_i^{min}, p_i^{max}, \boldsymbol{x}_i\}_{i\in[0,M]}$, whose intervals $(p_i^{min}, p_i^{max}]$ partition the fixed-point grid; for continuous distributions, the PMF is replaced by the PDF integral.

C The Plookup Protocol

For completeness, we provide the plookup set-inclusion protocol [GW20], which Noisette uses for the set inclusion check (§ 2 and § 4). The prover commits to an auxiliary input $C = (A||B)$ sorted according to A, and both parties check the identity $f(a, b) = g(a, b)$ at a random point for the polynomials in Eq. (1). Figure 14 presents the full protocol Π_{plookup}.

Protocol Π_{plookup}

Input: $\mathcal{P}$ holds sets A and B; both parties hold A and $[\![B]\!]$, with $|A| = M$ and $|B| = N$.

Protocol:

1. $\mathcal{P}$ computes $C = A||B$ and sorts the elements in C according to A, i.e., for $i \in [N+M-1]$, either $C[i] = C[i+1]$, or $C[i] = A[j]$ and $C[i+1] = A[j+1]$ for some j. $\mathcal{P}$ sends $[\![C]\!]$ to $\mathcal{V}$.
2. $\mathcal{P}$ defines the polynomials f and g from [GW20], as given in Eq. 1.
3. $\mathcal{V}$ samples $a, b \leftarrow \mathbb{F}_{2^k}$ and sends them to $\mathcal{P}$; both parties compute $[\![f(a,b)]\!]$ and $[\![g(a,b)]\!]$.
4. $\mathcal{P}$ opens $[\![f(a,b)]\!]$ and $[\![g(a,b)]\!]$, and $\mathcal{V}$ outputs $f(a,b) \stackrel{?}{=} g(a,b)$.

Figure 14: Protocol for checking set-inclusion.

D Proof of Theorem 1

Proof. Fix $\mathcal{D}'$ and N, and let $T \leftarrow \Pi_{\mathsf{TConstruct}}(\mathcal{D}', N)$ be the public table $T = \{(p_i^{\min}, p_i^{\max}, x_i)\}_{i\in[M]}$ such that the intervals $\{(p_i^{\min}, p_i^{\max}]\}_{i\in[M]}$ form a partition of the fixed-point grid $G_n := \{2^{-n}, 2 \cdot 2^{-n}, \ldots, 1\}$ of $(0,1]$. Define the deterministic map $g_T : G_n \to \{x_1, \ldots, x_M\}$ by $g_T(u) = x_j$ for the unique j satisfying $p_j^{\min} < u \le p_j^{\max}$, and let $\widehat{\mathcal{D}'}$ be the induced distribution of $g_T(u)$ for u uniform on G_n, i.e., $u = (U+1)/2^n$ for a uniform n-bit string U (§ 4.2); $\widehat{\mathcal{D}'}$ is the distribution $\mathcal{D}''$ of Lemma 2.

Case 1 (Both parties honest / completeness). By the specification of $\mathcal{F}_{\mathsf{rand}}$, the prover receives independent uniform coins $U_1, \ldots, U_N \in \{0,1\}^n$, encoding values $u_1, \ldots, u_N \in G_n$, and both parties obtain commitments $[\![U_i]\!]$. For each $i \in [N]$, the honest prover chooses the unique row $S_i := T_j$ such that $p_j^{\min} < u_i \le p_j^{\max}$, and uses $\mathcal{F}_{\mathsf{ZK}}$ to prove the relations in Step 5. Completeness of $\mathcal{F}_{\mathsf{ZK}}$ implies $\mathcal{V}$ outputs true. Moreover, $\mathcal{P}$ outputs $S_{i,2} = g_T(u_i)$, hence $\{S_{i,2}\}_{i\in[N]}$ are i.i.d. from $\widehat{\mathcal{D}'}$, and $\mathcal{V}$ receives the corresponding commitments $\{[\![S_{i,2}]\!]\}_{i\in[N]}$. Finally, since T is obtained from n-bit fixed-point CDF probabilities for $\mathcal{D}'$, then by Lemma 2 the joint distribution satisfies $\Delta(\widehat{\mathcal{D}'}^N, (\mathcal{D}')^N) \le \eta_{\mathsf{cdf}}$. Equivalently, the outputs match N ideal samples from $\mathcal{D}'$ except with probability at most η_{cdf}.

Case 2 (Corrupted prover / soundness). Let $\mathcal{P}^\star$ be any PPT prover and suppose $\mathcal{V}$ outputs true. By soundness of $\mathcal{F}_{\mathsf{ZK}}$, except with probability $\mathsf{negl}(\kappa)$, Step 5a implies $S_{i,0} < u_i \le S_{i,1}$ for all $i \in [N]$. For Step 5b, if $\mathcal{P}^\star$ causes acceptance while committing to some triple $(S_{i,0}, S_{i,1}, S_{i,2})$ that is not a row of T, then the compressed inclusion check requires a collision under the degree-2 map $(a_0, a_1, a_2) \mapsto a_0 + \alpha a_1 + \alpha^2 a_2$ with $\alpha \in \mathbb{F}_{\mathsf{ZK}}$. A fixed bad triple collides with a given row of T with probability at most $2/|\mathbb{F}_{\mathsf{ZK}}|$; a union bound over the M rows and N samples gives $2MN/|\mathbb{F}_{\mathsf{ZK}}|$. Thus, except with probability $\mathsf{negl}(\kappa) + 2MN/|\mathbb{F}_{\mathsf{ZK}}|$, acceptance implies that each committed S_i is a table row consistent with u_i, and therefore $S_{i,2} = g_T(u_i)$ for all i. Since $\{u_i\}$ are uniform on G_n by $\mathcal{F}_{\mathsf{rand}}$, the accepted outputs are distributed as $\widehat{\mathcal{D}'}^N$ (and are η_{cdf}-close to $(\mathcal{D}')^N$ with Lemma 2), as required by $\mathcal{F}_{\mathsf{sample}}$.

Case 3 (Corrupted verifier / zero-knowledge). A corrupted verifier's view consists only of commitments and the $\mathcal{F}_{\mathsf{ZK}}$ transcripts for the asserted relations, and it never receives openings nor the sample values. By the zero-knowledge property of $\mathcal{F}_{\mathsf{ZK}}$, there exists a PPT simulator $\mathcal{S}$ that, on

input the public values (including T and N) and the verifier's randomness (including α), produces a transcript computationally indistinguishable from the real view.

Combining the three cases, Π_{sample} realizes $\mathcal{F}_{\mathsf{sample}}$ in the $(\mathcal{F}_{\mathsf{ZK}}, \mathcal{F}_{\mathsf{rand}})$ hybrid model, with the η_{cdf} deviation of Lemma 2. □

E Proof of Theorem 2

Throughout, p denotes the PMF of $\mathcal{D}$ (symmetric and non-increasing in $|x|$) and $q := p_s$ the staircase PMF of Definition 4, which is symmetric and non-increasing in $|x|$ as noted in § 5.1.

Lemma 3 (Shift monotonicity)**.** *Let q be a symmetric PMF that is non-increasing in $|x|$. For every $\alpha > 1$, $D_\alpha(q \,\|\, q + s)$ is non-decreasing in $|s|$.*

Proof. By symmetry, take $s \geq 0$. Let $Z = \sum_x q(x)^\alpha$ and let R have PMF q^α/Z, so that $e^{(\alpha-1)D_\alpha(q\|q+s)} = \sum_x q(x)^\alpha q(x-s)^{1-\alpha} = Z\,\mathbb{E}\big[g(q(R-s))\big]$ with $g(y) = y^{1-\alpha}$ decreasing. Each level set $\{x : q(x) \geq y\}$ is a symmetric interval $[-\beta, \beta]$, so $\Pr[q(R-s) \geq y] = \Pr[R \in [s-\beta, s+\beta]]$. Increasing s by one exchanges the window's endpoint $s-\beta$ for $s+1+\beta$, which is farther from the origin ($s+1+\beta > |s-\beta|$); since the PMF of R is symmetric and non-increasing in $|x|$, this never increases the probability. Hence $q(R-s)$ becomes stochastically smaller as s grows, $\mathbb{E}[g(\cdot)]$ is non-decreasing in s, and so is D_α. □

Lemma 4 (Profile dominance at aligned shifts)**.** *For every shift $\sigma = tm$, $t \in \mathbb{Z}_{>0}$, and every $\gamma \geq 0$,*

$$\sum_{x\in\mathbb{Z}} \big(q(x) - \gamma\, q(x-\sigma)\big)_+ \;\leq\; \sum_{x\in\mathbb{Z}} \big(p(x) - \gamma\, p(x-\sigma)\big)_+.$$

Thus, $D_\alpha(q \,\|\, q+\sigma) \leq D_\alpha(p \,\|\, p+\sigma)$ for every $\alpha > 1$.

Proof. Write $d_f(x) := f(x) - \gamma f(x-\sigma)$ and split $\mathbb{Z}$ into the crossing segment $S = \{0, 1, \ldots, \sigma\}$ and the remaining buckets.

1) Outside S. Every bucket maps onto a bucket under the shift: $C_k - \sigma = C_{k-t}$ for $k \geq t+1$, and $-C_k - \sigma = -C_{k+t}$ for $k \geq 1$. Since q is constant on buckets and the bucket masses of q and p coincide,

$$\sum_{x\in C} \big(d_q(x)\big)_+ = \Big(\sum_{x\in C} d_p(x)\Big)_+ \leq \sum_{x\in C} \big(d_p(x)\big)_+$$

for each such bucket C.

2) On S. By symmetry, $q(x-\sigma) = q(\sigma - x)$, so d_q is non-increasing on S ($q(x)$ is non-increasing and $q(\sigma - x)$ non-decreasing in x), and therefore

$$\sum_{x\in S} \big(d_q(x)\big)_+ = \max\Big(0,\; \max_{0\leq y\leq\sigma} \big[Q_y - \gamma\widetilde{Q}_y\big]\Big),$$

where $Q_y := \sum_{x=0}^{y} q(x)$ and $\widetilde{Q}_y := \sum_{x=\sigma-y}^{\sigma} q(x)$; define $P_y, \widetilde{P}_y$ analogously. Within each bucket, p is non-increasing and q is its average, so every prefix sum of $p - q$ is non-negative (the average of the j largest of m non-increasing values is at least their overall average) and vanishes at bucket ends; the zero atom contributes 0. Hence $Q_y \leq P_y$ for all $y \in S$, with equality at $y = \sigma$, which also yields $\widetilde{Q}_y = Q_\sigma - Q_{\sigma-y-1} \geq P_\sigma - P_{\sigma-y-1} = \widetilde{P}_y$. Thus, for every y,

$$Q_y - \gamma\widetilde{Q}_y \;\leq\; P_y - \gamma\widetilde{P}_y \;=\; \sum_{x\leq y} d_p(x) \;\leq\; \sum_{x\in S} \big(d_p(x)\big)_+,$$

which proves the claim on S and, with the aligned buckets, the hockey-stick inequality.

Finally, every f-divergence between probability distributions is a non-negative mixture of the hockey-stick divergences $\sum_x (\mu(x) - \gamma\nu(x))_+$ over $\gamma \geq 0$, taken in both orientations [LV06]; here the reversed orientation reduces to the stated one, since $\sum_x (q(x-\sigma) - \gamma q(x))_+ = \sum_x (q(x) - \gamma q(x-\sigma))_+$ by the substitution $x \mapsto \sigma - x$ and symmetry (likewise for p). Applying this with $f(u) = u^\alpha - 1$ gives $\sum_x q(x)^\alpha q(x-\sigma)^{1-\alpha} \leq \sum_x p(x)^\alpha p(x-\sigma)^{1-\alpha}$, i.e., the Rényi comparison. □

Proof of Theorem 2. For any $|s| \leq \Delta = tm$ and $\forall \alpha > 1$,

$$D_\alpha(\mathcal{D}_s \| \mathcal{D}_s + s) \leq D_\alpha(\mathcal{D} \| \mathcal{D} + \Delta) \leq \tfrac{1}{2}\varepsilon^2 \alpha,$$

by Lemma 3, Lemma 4 (with $\sigma = \Delta$), and the $\frac{1}{2}\varepsilon^2$-CDP hypothesis on $\mathcal{D}$, respectively. □

Remark 1 (Continuous case)**.** *Both lemmas carry over verbatim to the continuous staircase noise of Definition 5, with sums replaced by integrals and arbitrary real shifts in Lemma 3: there is no zero atom, every length-m cell of the grid maps onto a cell under shifts tm, and the prefix-sum and unimodality arguments are unchanged. This is used in Appendix G and in the multivariate analysis in Appendix F.*

F Multivariate Staircase DP Noise

Our analysis in § 5.1 covers one-dimensional noise and multidimensional noise with a per-dimension sensitivity bound (e.g., feature vectors where each dimension has its own bound). Other applications bound only the joint L_p norm (e.g., DPSGD bounds the L_2 norm of gradients for the Gaussian mechanism), so the per-dimension shifts can be arbitrary integers and no single m divides every shift.

Theorem 5 (Privacy Guarantee of Multivariate Discrete Staircase DP Noise)**.** *Let the original discrete DP noise $\mathcal{D}$ satisfy $\frac{1}{2}\varepsilon^2$-CDP under the L_p sensitivity bound Δ_p, with PMF symmetric and monotone, i.e., $p(x) = p(-x)$ and $p(x) \geq p(x+1)$, $\forall x \in \mathbb{Z}_{\geq 0}$, and define*

$$\beta^2 := \sup_{\alpha > 1} \max_{t \in \mathbb{Z},\, 1 \leq |t| \leq \Delta_p} \frac{D_\alpha(\mathcal{D}_s \| \mathcal{D}_s + t)}{D_\alpha(\mathcal{D} \| \mathcal{D} + t)}.$$

Then the staircase noise $\mathcal{D}_s$ satisfies $\frac{1}{2}(\beta\varepsilon)^2$-CDP.

Proof. Consider any shift vector $\mathbf{t} \in \mathbb{Z}^d$ with $\|\mathbf{t}\|_p \leq \Delta_p$; every nonzero coordinate satisfies $1 \leq |t_i| \leq \Delta_p$, and zero coordinates contribute no divergence. By the additivity of Rényi divergence over product distributions and the definition of β, for all $\alpha > 1$:

$$\sum_{i=1}^{d} D_\alpha(\mathcal{D}_s \| \mathcal{D}_s + t_i) \leq \beta^2 \sum_{i=1}^{d} D_\alpha(\mathcal{D} \| \mathcal{D} + t_i) \leq \frac{1}{2}(\beta\varepsilon)^2 \alpha.$$

□

The constant β is finite and can be bounded analytically. By Lemma 3 and Lemma 4, $D_\alpha(\mathcal{D}_s \| \mathcal{D}_s + t) \leq D_\alpha(\mathcal{D} \| \mathcal{D} + t')$ for $|t'| = \lceil |t|/m \rceil m$; for noise whose Rényi divergence grows at most quadratically in the shift (discrete Gaussian, whose divergence is $\alpha t^2/(2\sigma^2)$ up to negligible corrections [CKS20], and discrete Laplace), this yields

$$\beta^2 \;\leq\; \max_{t \in \mathbb{Z},\, 1 \leq |t| \leq \Delta_p} \left(\frac{\lceil |t|/m \rceil\, m}{|t|} \right)^2,$$

which equals m^2 when unit shifts are admissible and is at most $(1 + m/t_{\min})^2$ when every admissible shift satisfies $|t| \geq t_{\min} \geq m$. In practice, we recommend computing β exactly by sweeping the admissible one-dimensional shifts t and Rényi orders α (the ratio converges as $\alpha \to \infty$); for discrete Gaussian the bound above is tight and independent of σ, giving $\beta = m$ with unit shifts and $\beta = 10/9 \approx 1.11$ for $m = 10$ when every admissible shift is at least $8m$, so m should stay small relative to the smallest admissible shift. Alternatively, the rounding argument of Theorem 3 (Appendix G) applies in the discrete case: since nonzero integer shifts have magnitude at least 1, at most $\min\{d, \Delta_p^p\}$ coordinates differ, so $\mathcal{D}_s$ inherits the original guarantee at the inflated sensitivity $\Delta_p + m \min\{d, \Delta_p^p\}^{1/p} \leq (1 + m)\Delta_p$, preferable when $\Delta_p \gg m \min\{d, \Delta_p^p\}^{1/p}$.

G Proof of Theorem 3

In the multivariate *discrete* case, the privacy loss of the staircase noise can be bounded through a per-shift divergence ratio (Appendix F). That approach degenerates in the continuous case: the per-dimension shifts can be arbitrarily small nonzero reals, and the corresponding divergence ratio is unbounded as $|t| \to 0$. We instead prove Theorem 3 by rounding the shifts up to multiples of the bucket width and absorbing the rounding into the inflated sensitivity.

Proof. The hypothesis states that for every shift vector $\mathbf{s} \in \mathbb{R}^d$ with $\|\mathbf{s}\|_p \leq (1 + c)\Delta_p$ and every $\alpha > 1$,

$$\sum_{i=1}^{d} D_\alpha(\mathcal{D} \| \mathcal{D} + s_i) \leq \frac{1}{2}\varepsilon'^2 \alpha,$$

where the sum over dimensions follows from the additivity of Rényi divergence over product distributions. Now consider any shift vector $\mathbf{t} \in \mathbb{R}^d$ with $\|\mathbf{t}\|_p \leq \Delta_p$. By Lemma 3 (which holds for real shifts in the continuous case, Remark 1), $D_\alpha(\mathcal{D}_s \| \mathcal{D}_s + t)$ is non-decreasing in $|t|$, so rounding each shift up to the next multiple of the bucket width, $|t_i'| = \lceil |t_i|/m \rceil \, m$, can only increase each term. Since every t_i' is a multiple of m, the continuous version of Lemma 4 gives

$$D_\alpha(\mathcal{D}_s \| \mathcal{D}_s + t_i') \leq D_\alpha(\mathcal{D} \| \mathcal{D} + t_i'), \quad i \in [d].$$

The rounding increases each coordinate by less than m, so by the Minkowski inequality:

$$\|\mathbf{t}'\|_p \leq \|\mathbf{t}\|_p + \|\mathbf{1}m\|_p \leq \Delta_p + m d^{1/p} = (1 + c)\Delta_p.$$

Applying the hypothesis to $\mathbf{t}'$ yields, for all $\alpha > 1$,

$$\sum_{i=1}^{d} D_\alpha(\mathcal{D}_s \| \mathcal{D}_s + t_i) \leq \sum_{i=1}^{d} D_\alpha(\mathcal{D} \| \mathcal{D} + t_i') \leq \frac{1}{2}\varepsilon'^2 \alpha,$$

i.e., $\mathcal{D}_s$ satisfies $\frac{1}{2}\varepsilon'^2$-CDP under the L_p sensitivity Δ_p. □

For a general noise mechanism, ε' is determined by how its privacy guarantee depends on the sensitivity. For the Gaussian mechanism ($p = 2$), $\frac{1}{2}\varepsilon'^2$-CDP at sensitivity $(1 + c)\Delta_2$ holds with $\varepsilon' = (1 + c)\Delta_2/\sigma = (1 + c)\varepsilon$, where $\frac{1}{2}\varepsilon^2$-CDP is the guarantee at the original sensitivity Δ_2 [CKS20]; for the Laplace mechanism ($p = 1$), the pure-DP guarantee scales the same way, giving $\varepsilon' = (1 + c)\varepsilon$ as well.

H Floating-Point Error Analysis

NOISETTE-CONT in § 5.3 draws a bucket index $k \in \mathbb{Z}$ and an offset $v \sim \mathrm{Unif}[0,1)$ and forms the noise $km + vm$ with $m := 2^{-a}$. A finite-precision implementation departs from this ideal in two places: v is replaced by $U/2^B$ with $U \sim \mathrm{Unif}\{0, \ldots, 2^B - 1\}$, and the additions run in FP32 rather than over $\mathbb{R}$. We bound each error separately.

H.1 Discretizing the uniform sample

All arithmetic in this subsection is exact over $\mathbb{R}$ up to one final FP32 rounding. Let $f \in \mathbb{F}_{32}$ denote the FP32 query result to which noise is added. Let $\mathcal{D}_1$ and $\mathcal{D}_2$ denote the output distributions of $\mathsf{fp}(f + k2^{-a} + v\,2^{-a})$ and $\mathsf{fp}(f + k2^{-a} + U\,2^{-a-B})$, respectively; the bound of Theorem 4 is $\mathrm{TV}(\mathcal{D}_1, \mathcal{D}_2) \le \eta_B$ under the assumption that, for every admissible k, the interval $I_k := [f + k2^{-a},\ f + (k+1)2^{-a})$ stays inside the finite FP32 range.

Proof of Theorem 4. Fix k and set $M := 2^B$, $\Delta := 2^{-a-B}$, $X := f + k2^{-a}$, $I := [X, X + 2^{-a})$, $G := \{X + j\Delta : 0 \le j < M\}$. Let μ and ν be uniform on I and G. For every subinterval $J \subseteq I$, the number of grid points in J differs from $|J|/\Delta$ by at most 1, giving

$$\big|\nu(J) - |J|/|I|\big| \le 1/M, \tag{2}$$

and the same bound holds when boundary points carry weight $1/2$ (the weighted count lies between the open and closed counts).

Split I by resolution: $L := I \cap (-2^{23}\Delta,\, 2^{23}\Delta)$ and $H := I \setminus L$; every point of H lies in an FP32 region with spacing $\ge \Delta$. Let Q be FP32 round-to-nearest and let $\widetilde{Q}$ agree with Q on L but split exact midpoints in H equally between neighbors. Write $P := \mu \circ Q^{-1}$, $S := \nu \circ \widetilde{Q}^{-1}$, $D := \nu \circ Q^{-1}$, so $\mathrm{TV}(P, D) \le \mathrm{TV}(P, S) + \mathrm{TV}(S, D)$.

Low-resolution part. $|L| \le 2^{24}\Delta$, so $\mu(L) \le 2^{24-B}$ and $\nu(L) \le \mu(L) + 2^{-B}$ by Eq. 2. The contribution of L to $\mathrm{TV}(P, S)$ is at most $(\mu(L) + \nu(L))/2 \le 2^{24-B} + 2^{-B-1}$.

High-resolution part. For a connected component $J \subseteq H$ in an FP32 region with spacing $s = q\Delta$, $q \in \mathbb{N}$, every full rounding cell of length s has weighted grid count exactly q under $\widetilde{Q}$, matching its continuous mass; only the two partial cells at the endpoints of J can deviate, each by at most $1/M$ by Eq. 2. The number of FP32 regions met by I on one side is bounded two ways. Since $|I| \le 2^{-a} \le 2^A$, I meets at most $128 + A$ regions. Also, the high-resolution part begins at radius $T := 2^{23}\Delta$: when $T \ge 2^{-126}$, the positive part lies in normal bins between T and $T + 2^{-a}$, giving at most $\lceil \log_2(1 + 2^{B-23}) \rceil + 1 \le B - 21$ bins; otherwise the positive part covers at most the subnormal region together with normal bins up to 2^{-a}, again at most $B - 21$. The per-side count is therefore $\le H_{A,B}$, and the high-resolution contribution to $\mathrm{TV}(P, S)$ is at most $2H_{A,B}\,2^{-B}$.

Midpoint ties. S and D differ only at grid points in H that fall on FP32 midpoints. Each high-resolution region on one side contains at most 2^{23} midpoints, for $\le 2H_{A,B} \cdot 2^{23}$ hits in total. Each carries mass 2^{-B}, and replacing an equal split by a deterministic tie rule shifts TV by at most 2^{-B-1}. Hence $\mathrm{TV}(S, D) \le H_{A,B}\,2^{23-B}$.

Combining the three terms gives the fixed-k bound

$$\mathrm{TV}(P, D) \;\le\; 2^{24-B} + H_{A,B}\,2^{23-B} + (2H_{A,B} + \tfrac{1}{2})\,2^{-B}.$$

The unconditional $\mathcal{D}_1, \mathcal{D}_2$ are mixtures of P, D under the same index distribution; TV is convex under mixtures, so the bound transfers. □

Algorithm 1 WideIntFP32(f, k, U)

1: $r \leftarrow \max\{149,\ a + B\}$
2: $F \leftarrow$ the unique integer with $f = F\,2^{-r}$
3: $N \leftarrow (k2^B + U)\,2^{r-a-B}$
4: $S \leftarrow F + N$ ▷ exact wide-integer addition
5: **return** the FP32 round-to-nearest, ties-to-even, of $S\,2^{-r}$

H.2 Exact wide-integer implementation

A naive FP32 implementation rounds after every intermediate addition, introducing further error beyond $\mathcal{D}_2$. We avoid this by placing the query value and the noise term $\theta_{k,U} := k\,2^{-a} + U\,2^{-a-B} = (k2^B + U)\,2^{-a-B}$ on a common binary lattice, adding them exactly as wide integers, and rounding only once at the end (Alg. 1): every FP32 value is an integer multiple of 2^{-149} and $\theta_{k,U}$ is one of 2^{-a-B}, so with $r := \max\{149,\ a + B\}$ both lie in $2^{-r}\mathbb{Z}$.

Line 2 in Alg. 1 is computed directly from the FP32 bits of f. If f has sign σ_f, exponent e_f, and fraction m_f, then

$$F = \begin{cases} (-1)^{\sigma_f} m_f\,2^{r-149}, & e_f = 0, \\ (-1)^{\sigma_f}(2^{23} + m_f)\,2^{e_f+r-150}, & 1 \le e_f \le 254. \end{cases}$$

Thus $f = F2^{-r}$ exactly; the final rounding in Line 5 is realized by integer rounding within the target binade, as justified in the proof of Theorem 6.

Theorem 6 (Exactness of WideIntFP32)**.** *Assume the signed accumulator in Alg. 1 does not overflow and $f + k\,2^{-a} + U\,2^{-a-B}$ lies in the finite FP32 range. Then Alg. 1 returns $\mathsf{fp}\big(f + k\,2^{-a} + U\,2^{-a-B}\big)$.*

Proof. By construction of F. $F\,2^{-r} = f$. Also, $N\,2^{-r} = (k2^B + U)\,2^{-a-B} = k\,2^{-a} + U\,2^{-a-B} = \theta_{k,U}$. Since the accumulator does not overflow, the integer sum $S = F + N$ is exact, hence

$$S\,2^{-r} = f + \theta_{k,U}.$$

It remains to justify the last rounding step. If $|S| < 2^{r-126}$, then $S2^{-r}$ lies in the zero/subnormal range, whose FP32 grid is exactly $\{q2^{-149} : 0 \le q \le 2^{23}\}$. Rounding to FP32 is therefore the same as rounding $|S|/2^{r-149}$ to the nearest integer, ties to even, and restoring the sign. If $|S| \ge 2^{r-126}$, let $t = \lfloor \log_2 |S| \rfloor$. Then $S2^{-r}$ lies in the normal binade with unbiased exponent $t - r$, and every FP32 value in that binade has the form $\pm q\,2^{t-r-23}$ with $2^{23} \le q \le 2^{24} - 1$. So FP32 rounding is exactly nearest-even rounding of $|S|/2^{t-23}$, followed by the usual one-step carry if the rounded significand equals 2^{24}. Therefore Line 5 returns $\mathsf{fp}(S2^{-r}) = \mathsf{fp}(f + \theta_{k,U})$. □

Corollary 2 (No additional floating-point error)**.** *The output distribution of Alg. 1 is exactly $\mathcal{D}_2$; the only remaining discrepancy from the continuous sampler is the B-bit discretization bounded in Theorem 4.*

H.3 From TV bound to (ε, δ)-DP

TV closeness shifts δ. If a mechanism $\mathcal{M}$ is (ε, δ)-DP and $\mathrm{TV}(\mathcal{M}(x), \mathcal{M}'(x)) \le \eta$ for every input x, then $\mathcal{M}'$ is $(\varepsilon, \delta + (1 + e^{\varepsilon})\eta)$-DP: for any measurable S, $\Pr[\mathcal{M}'(x) \in S] \le \Pr[\mathcal{M}(x) \in S] + \eta \le e^{\varepsilon} \Pr[\mathcal{M}(x') \in S] + \delta + \eta \le e^{\varepsilon} \Pr[\mathcal{M}'(x') \in S] + \delta + (1 + e^{\varepsilon})\eta$. For N independent noise samples, TV is subadditive over product measures, so the joint TV between the ideal N-sample mechanism and its FP32 implementation is at most $\eta_{\mathsf{fp}} = N\eta_B$, with η_B as in Theorem 4.

Corollary 3 ((ε, δ)-DP of FP32 NOISETTE-CONT)**.** *Suppose the ideal continuous-staircase mechanism is (ε, δ)-DP (Theorem 3). Its FP32 implementation via Alg. 1 with B-bit uniform discretization differs from the ideal by a TV of at most $\eta_{\mathsf{trunc}} + \eta_{\mathsf{cdf}} + \eta_{\mathsf{fp}}$, i.e., the errors of Corollary 1 plus the FP32 discretization error. By the closeness bound above and adding the ZK soundness error, it is*

$$\Big(\varepsilon,\ \delta + \delta_{\mathsf{cert}} + (1 + e^{\varepsilon})\big(\eta_{\mathsf{trunc}} + \eta_{\mathsf{cdf}} + \eta_{\mathsf{fp}}\big)\Big)\text{-}DP,$$

with δ_{cert} the data-independent soundness error of Corollary 1.

Sufficiently large parameters make the error terms negligible: for instance, $B \geq 24 + \log_2\big(100N(1 + e^{\varepsilon})/\delta\big)$ keeps $(1 + e^{\varepsilon})\eta_{\mathsf{fp}} \leq \delta/100$.

I Parameter Setup

We split the overall failure probability of NOISETTE-CONT as

$$\delta = \delta_{\mathsf{main}} + \delta_{\mathsf{trunc}} + \delta_{\mathsf{cdf}} + \delta_{\mathsf{fp}} + \delta_{\mathsf{cert}},$$

following Corollary 1 and Corollary 3. The term δ_{main} is the failure probability of the base Gaussian mechanism, whose noise scale is calibrated at the inflated sensitivity $(1 + c)\Delta_p$ of Theorem 3. The terms δ_{trunc}, δ_{cdf}, and δ_{fp} absorb the total-variation errors of support truncation (Lemma 1), CDF-table quantization (Lemma 2), and B-bit FP32 discretization (Theorem 4); a TV error η increases δ by $(1 + e^{\varepsilon})\eta$, so we target each TV error at its slack divided by $(1 + e^{\varepsilon})$. The term δ_{cert} is the certification soundness error: instantiating $\mathcal{F}_{\mathsf{ZK}}$ with QuickSilver over $\mathbb{F}_{2^{128}}$ gives $\delta_{\mathsf{cert}} \leq (2MN + |\mathcal{C}|)/2^{128}$, with M the CDF table size, N the number of committed samples, and $|\mathcal{C}|$ the circuit size, which stays below 10^{-24} even for our largest instance with $\sim 10^{11}$ field multiplications. These choices require little tuning: at $\delta = 10^{-6}$ we set $\delta_{\mathsf{trunc}} = 10^{-8}$, $\delta_{\mathsf{cdf}} = 10^{-9}$, and $\delta_{\mathsf{fp}} = 10^{-8}$, a total slack of about 2×10^{-8} that leaves $\delta_{\mathsf{main}} \approx \delta$ and changes σ by less than 1% relative to a calibration without certification.

The most demanding configuration in our evaluation combines the largest LoRA adapter rank $r = 16$ (Table 6) with the strictest privacy budget $\varepsilon = 1$ (Table 5): $d = 1{,}181{,}954$ trainable parameters, $T = 500$ training rounds, per-round sampling rate $q \approx 4.75 \times 10^{-3}$, bucket width $m = 2^{-13}$, and clipping norm $\Delta = 1$. The derived parameters are the noise scale $\sigma_{\mathsf{base}} = 1.19$, a truncation bound of 10.79, the CDF table size $M = 88{,}372$, the fixed-point CDF precision $n = 78$ bits, and the uniform-offset width $B = 96$ bits.