DITHERED GAUSSIAN MECHANISM FOR RANDOMNESS-EFFICIENT DIFFERENTIAL PRIVACY

Anonymous authors
Paper under double-blind review

ABSTRACT

We present the *dithered Gaussian mechanism*, an alternative to the discrete Gaussian mechanism for differential privacy that discretizes the private output rather than the noise distribution itself. By interpreting this discretization as post-processing of the Gaussian mechanism, our construction directly inherits the privacy guarantees of the standard Gaussian mechanism while avoiding vulnerabilities caused by finite-precision floating-point outputs. In addition, the mechanism is provably randomness-efficient: by sampling the discretized output values directly, the number of high-quality random bits required for privacy can be reduced significantly and made independent of the noise level. This is achieved by separating the randomness into two sources: a high-quality source used for the privacy-critical sampling step, and a high-performance public source, possibly known to the adversary, that supplies the additional randomness needed for randomized discretization. This separation enables the use of cryptographically secure randomness without substantial performance loss. As an application, we study model training with DP-SGD and show that cryptographically secure noise generation with reduced exposure to floating-point vulnerabilities can be achieved with modest practical overhead.

1 INTRODUCTION

Modern machine learning increasingly relies on sensitive individual-level data, making formal privacy guarantees essential. Differential privacy provides one such guarantee by limiting how much any single data point can influence the output of an algorithm. The Gaussian mechanism (Dwork, Kenthapadi, McSherry, Mironov, and Naor, 2006) is one of the most widely used primitives for ensuring differential privacy. It is especially central in private machine learning, where it forms the noise-addition step of differentially private stochastic gradient descent (DP-SGD) due to Abadi, Chu, Goodfellow, McMahan, Mironov, Talwar, and Zhang (2016).

Despite its ubiquity, the Gaussian mechanism has important practical limitations. First, it is an idealized continuous mechanism, whereas real implementations run on finite-precision computers. As a result, a faithful implementation of a continuous Gaussian distribution is impossible: naive samplers can leave detectable "holes" in the set of representable floating-point values, creating privacy vulnerabilities of the kind first observed for the Laplace mechanism (Mironov, 2012) and later identified for the Gaussian mechanism itself (Jin et al., 2022). A second limitation is the amount of randomness required to sample Gaussian noise. In the ideal continuous model, exact sampling requires infinitely many random bits, while high-precision implementations can still require many random bits in practice. This cost becomes substantial at scale: recent efforts to train language models entirely under differential privacy, such as the *VaultGemma* 1B model, required quadrillions of Gaussian random draws and petabytes of randomness to obtain a formal privacy guarantee (VaultGemma Team, 2025). At this scale, randomness generation itself can become a bottleneck (Egan, 2024). Several discrete-valued mechanisms, most prominently the discrete Gaussian mechanism of Canonne, Kamath, and Steinke (2020), address finite-precision concerns by adding discrete noise to discretized vectors. However, such mechanisms still requires many privacy-critical random bits. Moreover, because the privacy guarantees do not directly follow from those of the continuous Gaussian mechanism, they require separate privacy analyses for composition and amplification by subsampling, making them difficult to use as drop-in replacements for Gaussian noise in DP-SGD.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Our contributions. We propose a drop-in replacement for the Gaussian mechanism that is cryptographically secure, resistant to floating-point attacks, and time-efficient. We achieve these properties through a new approach to noise addition that we call the *dithered Gaussian mechanism*. Conceptually, the mechanism can be viewed as adding Gaussian noise to the sensitive vector and then rounding the result to a discrete grid shifted by a public random offset, or *dither*. Conditioned on the dither, the resulting discrete output distribution can be written down explicitly, allowing us to sample from it directly rather than sampling the intermediate continuous Gaussian noise. The key observation is that the rounded output is a post-processing of the standard Gaussian mechanism, so it directly inherits the privacy guarantees of the Gaussian mechanism. In particular, we show that the number of private random bits can be made independent of the Gaussian noise scale, without significantly changing the distribution of the perturbation relative to the Gaussian mechanism. This substantially reduces the cost of using cryptographically secure randomness. Since the released output is discrete by construction (conditioned on the public dither), the mechanism avoids finite-precision issues associated with floating-point outputs. Finally, we apply the mechanism to DP-SGD and show that cryptographically secure noise generation can be achieved with a modest runtime overhead of about $3\% - 7\%$ in our experiments.

1.1 Related work

Floating-point vulnerabilities and discrete mechanisms. A subtle but important issue in the implementation of differentially private mechanisms is the use of finite-precision floating-point arithmetic. Mironov (2012) showed that naive implementations of the Laplace mechanism can leave detectable "holes" in the set of representable noisy outputs, which can be sufficient to reconstruct a significant part of the underlying data. Similar finite-precision issues have since been identified for other mechanisms, including the exponential mechanism (Ilvento, 2020), Noisy Max (Ding et al., 2025), and the Gaussian mechanism (Jin et al., 2022). A related class of attacks exploits precision-dependent leakage: the precision of the released noisy value may depend on the exponent of the original value, thereby leaking information about the input (Haney et al., 2022). Several implementation-level mitigations have been proposed (Holohan & Braghin, 2021; Haney et al., 2022) and adopted in practical differential privacy libraries, including diffprivlib (Holohan et al., 2019), OpenDP (OpenDP), and the "secure mode" of Opacus (Yousefpour et al., 2021). These approaches improve the robustness of floating-point implementations, but they do not remove the underlying mismatch between continuous idealized mechanisms and finite-precision computation.

A more direct way to avoid this kind of vulnerability is to work only with values from a discrete domain. Several discrete-valued mechanisms have been proposed for this purpose, most prominently the discrete Gaussian mechanism (Canonne et al., 2020), along with other similar discrete noise mechanisms (Ghosh et al., 2009; Dwork et al., 2006; Agarwal et al., 2018; 2021; Bao et al., 2022; Chen et al., 2022). By adding discrete noise and releasing discrete outputs, these mechanisms avoid the floating point artifacts described above. However, they have two important limitations from the perspective of our work. First, they can still require substantial amounts of randomness for noise generation, unless one uses coarse approximations. Second, mechanisms such as the discrete Gaussian approximate Gaussian noise but do not automatically inherit the privacy guarantees of the continuous Gaussian mechanism. Consequently, analyses for composition, amplification by subsampling, or other accounting procedures generally have to be established separately.

Randomness complexity. From a theoretical standpoint, the large randomness requirements of practical differentially private algorithms raise a natural question at the intersection of privacy and complexity: how much randomness is actually necessary for privacy? Canonne, Su, and Vadhan (2025) initiated the study of the *randomness complexity of differential privacy*, showing that highly accurate private mechanisms can sometimes be implemented with surprisingly few random bits, in some settings as few as logarithmically many in the number of released statistics. This contrasts with the common intuition that the randomness required for privacy should scale with the dimension of the output. Their construction is based on a connection to *secluded partitions* from computational geometry (Vander Woude et al., 2024), but is not computationally efficient. More recently, Ghentiyala (2026) gave a computationally efficient alternative, at the cost of a polylogarithmic loss in the randomness-error trade-off.

DP-SGD and discrete mechanisms. One of the most important applications of Gaussian noise in differential privacy is the training of machine learning models, most commonly through differentially private stochastic gradient descent (DP-SGD) (Song et al., 2013; Bassily et al., 2014; Abadi et al., 2016). The privacy guarantees of DP-SGD are typically established using privacy accountants that exploit the structure of the sampling procedure, including Poisson subsampling (Abadi et al., 2016; Mironov, 2017; Koskela et al., 2020; Zhu et al., 2022; Lebeda et al., 2025), shuffling (Erlingsson et al., 2019; Feldman et al., 2021; 2023; Wang et al., 2023), sampling without replacement (Balle et al., 2018; Schuchardt et al., 2024), and random allocations such as Balls-in-Bins (Chua et al., 2025; Feldman & Shenfeld, 2025). Many of these analyses are developed specifically for the Gaussian mechanism, making it difficult to replace Gaussian noise with a different noise distribution without redoing the privacy analysis. Discrete mechanisms have also been studied in combination with DP-SGD, especially in federated learning with secure aggregation (Agarwal et al., 2018; Kairouz et al., 2021; Agarwal et al., 2021; Bao et al., 2022; Chen et al., 2022). In this setting, the main goals are to obtain integer-valued noisy gradient sums, which integrate naturally with secure aggregation, and to reduce communication complexity.

To the best of our knowledge *tight analyses* of subsampled discrete mechanisms are *not available* even for the standard Poisson subsampling scheme. Existing approaches therefore typically start from concentrated DP or Rényi DP guarantees for the underlying discrete mechanism, combine them with a generic subsampling amplification lemma, and then apply generic composition within the chosen privacy framework, as in (Kairouz et al., 2021; Agarwal et al., 2021). This contrasts with our approach, where the discretized mechanism inherits Gaussian privacy guarantees by post-processing and can therefore reuse Gaussian-based analyses directly.

2 Background

In this section, we first introduce the notions of sensitivity, randomness complexity, and privacy that will be used throughout the paper. Second, we present the properties of the discrete Gaussian mechanism with respect to these notions.

Sensitivity. Consider datasets X that are finite subsets of some set $\mathbb{X}$. We work in the add/remove adjacency notion: two datasets $X \sim X'$ are neighbors if one can be obtained from the other by inserting or deleting a single data point. For a dataset of n elements we write its elements as $X = \{x^{(1)}, \dots, x^{(n)}\} \subseteq \mathbb{X}$. For $p \geq 1$, the global ℓ_p-sensitivity of a function $f : \mathbb{X}^* \to \mathbb{R}^d$ is defined as

$$\Delta_p(f) := \sup_{X \sim X'} \|f(X) - f(X')\|_p.$$

Randomness complexity. The *randomness complexity* of a mechanism $\mathcal{M}$ is the expected number of random bits required to produce an output of the mechanism. As noted by Canonne et al. (2025), this quantity is tightly linked, up to an additive constant, to the Shannon entropy of the mechanism. When running a sequence of mechanisms independently (keeping public randomness fixed), the total private randomness complexity can be bounded by the sum of entropies plus an additive constant.

Differential privacy. We can release an estimate of $f(X)$ while satisfying *differential privacy* by adding noise to $f(X)$ scaled according to the sensitivity. Intuitively, differential privacy means that the information released has roughly the same distribution for neighboring datasets X and X'. For example, the Laplace mechanism (for $p = 1$) attains $(\varepsilon, 0)$-differential privacy and the Gaussian mechanism (for $p = 2$) attains (ε, δ)-differential privacy. We refer to Appendix A for definitions of differential privacy and properties of basic noise addition mechanisms.

Discrete Gaussian mechanism. Traditionally, differential privacy has been analyzed under the idealized assumption that mechanisms have access to samples from continuous distributions. Under this viewpoint, the amount of randomness required by a mechanism is typically not accounted for explicitly, and no finite upper bound on the number of random bits is established. In practice, however, implementations must operate on finite-precision computers and therefore rely on discrete distributions or finite procedures for sampling. This issue has been studied, for example, by Balcer & Vadhan (2018) and by Canonne et al. (2020), who show that discrete analogues of standard private mechanisms admit finite bounds on the required randomness. A canonical example is the *discrete Gaussian mechanism* (Canonne et al., 2020), which replaces continuous Gaussian noise with noise drawn from a discrete distribution over the integers with probabilities derived from the density of

a Gaussian. Besides being implementable on finite computers, this mechanism provides a natural starting point for studying the randomness complexity of differentially private algorithms.

Definition 1 (Discrete Gaussian mechanism). *Let $f : \mathcal{X}^n \to \mathbb{Z}^d$ be a function with sensitivity $\Delta_2(f)$. For $\sigma > 0$, the discrete Gaussian mechanism with scale $\sigma\Delta_2(f)$ releases*

$$M(x) = f(x) + Z,$$

where $Z = (Z_1, \ldots, Z_d) \in \mathbb{Z}^d$ has independent coordinates, and each Z_j is sampled from the one-dimensional discrete Gaussian distribution centered at 0 with scale $\sigma\Delta_2(f)$, that is,

$$\mathbb{P}(Z_j = z) = \frac{\exp\left(-\frac{z^2}{2\sigma^2\Delta_2(f)^2}\right)}{\sum_{u\in\mathbb{Z}} \exp\left(-\frac{u^2}{2\sigma^2\Delta_2(f)^2}\right)} \qquad \text{for all } z \in \mathbb{Z}.$$

Discretization. Note that the discrete Gaussian mechanism is defined only for integer-valued functions. To apply it to general real-valued functions, one must first discretize the values of f. Let $f(x) \in \mathbb{R}^d$ have sensitivity $\Delta_2(f)$, and let $h > 0$ be a discretization parameter. If each coordinate is rounded to the nearest integer multiple of h, then the sensitivity of the discretized function $\tilde{f}$ satisfies

$$\Delta_2(\tilde{f}) \le \Delta_2(f) + \frac{h\sqrt{d}}{2}. \tag{1}$$

However, such deterministic rounding introduces bias. *Randomizing* the decision whether to round up or down removes this bias, but increases the sensitivity to $\Delta_2(f) + h\sqrt{d}$. A better trade-off can be obtained by allowing a resampling probability $\beta \in (0, 1)$, resulting in a small bias, as described by Kairouz et al. (2021). They use a random rotation technique followed by randomized rounding to ensure that, with high probability, the sensitivity of the rounded vector is bounded by

$$\tilde{\Delta}_2(f) \le \min\left\{\Delta_2(f) + h\sqrt{d},\ \sqrt{\Delta_2^2(f) + \frac{dh^2}{4} + \sqrt{2\ln(1/\beta)}\left(\Delta_2(f)h + \frac{\sqrt{d}h^2}{2}\right)}\right\}. \tag{2}$$

To guarantee privacy, the rounding must be repeated until the required condition on the ℓ_2 norm is satisfied. (This is the case with high probability, so the privacy cost of the rounding is low.)

Randomness complexity of the discrete Gaussian. The choice of discretization parameter h creates a tension for the discrete Gaussian mechanism because the noise must be generated on the lattice $h\mathbb{Z}^d$. Equivalently, one may first scale the function by $1/h$, round and apply the discrete Gaussian mechanism on $\mathbb{Z}^d$, and then rescale the output by multiplying with h. If $\sigma > 0$ is the noise level required by the Gaussian mechanism for differential privacy, then the corresponding discrete mechanism uses scale $\tau = \sigma\Delta_2(\tilde{f})/h$, before the resulting noise is ultimately rescaled by h. The variance of the resulting mechanism is close to that of the continuous Gaussian mechanism provided that $h \ll \Delta_2(f)/\sqrt{d}$. However, even in the regime where the variance increase is small, the entropy increases significantly as h approaches 0. In particular, the binary entropy of Z_j sampled from the one-dimensional discrete Gaussian distribution with scale τ is

$$H(Z_j) = \frac{1}{2}\log_2(2\pi e\,\tau^2) + o_\tau(1), \tag{3}$$

see Lemma 6 of Ling & Belfiore (2014). Since the coordinates of $Z = (Z_1, \ldots, Z_d)$ are independent, it follows that the binary entropy of the d-dimensional discrete Gaussian is

$$H(Z) = \sum_{j=1}^{d} H(Z_j) = \frac{d}{2}\log_2(2\pi e\,\tau^2) + o_\tau(d). \tag{4}$$

With $\tau = \sigma\Delta_2(\tilde{f})/h$ this is approximately $d\log_2\left(\frac{\sigma\Delta_2(\tilde{f})}{h}\right) + 2d$ for large enough τ.

3 Dithered Gaussian Mechanism

We now present an alternative to the discrete Gaussian that avoids the need to discretize the input. Unlike the methods described in Section 2, it does not incur any increase in sensitivity. Moreover,

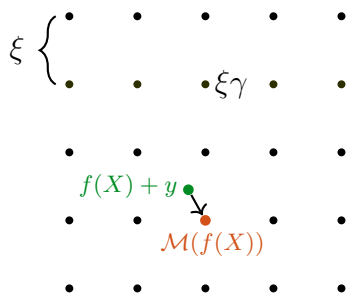


Figure 1: From a privacy perspective, the dithered Gaussian mechanism can be thought of as mapping the Gaussian mechanism output $f(X) + y$ to the nearest point on a randomly shifted axis-aligned grid $\{\xi(\gamma + z) \mid z \in \mathbb{Z}^d\}$. From an implementation perspective, we directly sample from a discrete distribution over the grid.

our method uses provably fewer random bits, making it practical even when noise must be generated using slow cryptographically secure randomness. In addition, the method directly inherits the privacy guarantees of the (continuous) Gaussian mechanism, and therefore does not require separate proofs for composition or amplification by subsampling.

Conceptual description. Our method, the *dithered Gaussian mechanism*, is illustrated in Figure 1. It can be thought of as a post-processing of the Gaussian mechanism with noise scale σ, i.e., the mechanism that outputs $f(X)+y$, where $y \sim \mathcal{N}(0, \sigma^2 I_d)$. For a parameter $\xi > 0$ the post-processing simply rounds each coordinate to an axis-aligned grid of points with distance ξ between consecutive points along all axes. Performing this rounding deterministically would introduce bias, but we avoid this with a random shift of the grid: Sample public randomness $(a, b) \sim \mathrm{Uniform}([0, 1)^2)$ and define the coordinate-dependent dither by $\gamma_i = (ai + b) \bmod 1$ for each $i \in [d]$. Given $f(X) \in \mathbb{R}^d$ and $y \sim \mathcal{N}(0, \sigma^2 I_d)$, the mechanism outputs $\mathcal{M}(f(X))$ where, for $i = 1, \ldots, d$,

$$\mathcal{M}(f(X))_i = \xi \left(\left\lfloor \frac{f(X)_i + y_i}{\xi} - \gamma_i + \frac{1}{2} \right\rfloor + \gamma_i \right). \tag{5}$$

Direct sampling from a discrete distribution. To avoid generating the Gaussian vector y we observe that it is possible to sample the integer-valued random variable

$$Z_i := \left\lfloor \frac{f(X)_i + y_i}{\xi} - \gamma_i + \frac{1}{2} \right\rfloor$$

directly without first sampling y_i. Indeed, for each $k \in \mathbb{Z}$, the event $Z_i = k$ is equivalent to

$$k \le \frac{f(X)_i + y_i}{\xi} - \gamma_i + \tfrac{1}{2} < k + 1 \Leftrightarrow \frac{\xi\left(k + \gamma_i - \frac{1}{2}\right) - f(X)_i}{\sigma} \le \frac{y_i}{\sigma} < \frac{\xi\left(k + \gamma_i + \frac{1}{2}\right) - f(X)_i}{\sigma}.$$

Denoting the cumulative distribution function of $\frac{y_i}{\sigma} \sim \mathcal{N}(0, 1)$ by Φ we thus have:

$$\Pr[Z_i = k] = \Phi\left(\frac{\xi\left(k + \gamma_i + \frac{1}{2}\right) - f(X)_i}{\sigma} \right) - \Phi\left(\frac{\xi\left(k + \gamma_i - \frac{1}{2}\right) - f(X)_i}{\sigma} \right), \qquad k \in \mathbb{Z}. \tag{6}$$

Since $\mathcal{M}(f(X))_i = \xi(Z_i + \gamma_i)$, this shows that to compute $\mathcal{M}(f(X))_i$ it suffices to sample directly from a discrete distribution over $\mathbb{Z}$. The crucial property of the proposed distribution is that its cumulative probabilities coincide with the Gaussian distribution function evaluated at the corresponding quantization boundaries. Indeed, the differences in equation 6 telescope, giving

$$\Pr[Z_i \le k] = \Phi\left(\frac{\xi\left(k + \gamma_i + \frac{1}{2}\right) - f(X)_i}{\sigma} \right). \tag{7}$$

Consequently, inverse-transform sampling reduces to evaluating the inverse Gaussian distribution function Φ^{-1}, allowing us to sample Z_i accurately and efficiently. Note that this convenient representation is not available for the discrete Gaussian distribution, whose cumulative probabilities involve a normalized sum of Gaussian weights over the integer lattice.

Algorithm 1 Dithered Gaussian mechanism

Require: Function value $f(X) \in \mathbb{R}^d$, parameters $\xi > 0, \sigma > 0$
1: Sample public randomness $(a, b) \sim \mathrm{Uniform}([0,1)^2)$
2: **for** $i = 1, \ldots, d$ **do**
3: $\quad \gamma_i \leftarrow (a \cdot i + b) \bmod 1$
4: $\quad$ Sample an integer $Z_i \in \mathbb{Z}$ such that, for every $k \in \mathbb{Z}$,

$$\Pr[Z_i = k] = \Phi\left(\frac{\xi\left(k + \gamma_i + \frac{1}{2}\right) - f(X)_i}{\sigma}\right) - \Phi\left(\frac{\xi\left(k + \gamma_i - \frac{1}{2}\right) - f(X)_i}{\sigma}\right)$$

5: $\quad \mathcal{M}(f(X))_i \leftarrow \xi(Z_i + \gamma_i)$
6: **end for**
7: **return** $\mathcal{M}(f(X))$

Algorithm details. We summarize the mechanism and its sampling distribution in Algorithm 1. Conceptually, Z_i can be sampled by inverse transform sampling using ideal, real-valued arithmetic, we outline the procedure next. Instead of drawing an infinite-precision uniform variable in $[0, 1)$, we progressively generate random bits. After finitely many bits, the sampled prefix determines an interval $[L_i, R_i) \subset [0, 1)$ containing the underlying uniform variable. We use the midpoint of this interval to obtain a candidate k_i and compute its two corresponding probability boundaries. If the entire interval $[L_i, R_i)$ lies between these boundaries, then all possible continuations of the random bit sequence produce the same outcome, and we return $Z_i = k_i$. Otherwise, additional random bits are generated and the interval is refined. In practice, the procedure is implemented in a vectorizable form in PyTorch. Most coordinates are resolved using float64 arithmetic. Coordinates that are too close to a probability boundary, or for which the float64 computation is not sufficiently reliable, are instead evaluated using arbitrary-precision arithmetic provided by the Python library `mpmath`. Additional random bits are generated only for the remaining active coordinates. This makes the procedure efficient on GPUs while avoiding an explicit truncation of the support. Details can be found in Appendix B. We provide an implementation of Algorithm 1 via an anonymous link.[1]

Noise distribution. We characterize the distribution of $\mathcal{M}(X)$ over the random choice of γ and Z:

Lemma 1. *Over the randomness of the shift γ_i and index Z_i, $\mathcal{M}(f(X))_i$ is identically distributed to*

$$f(X)_i + y_i + u_i, \tag{8}$$

where $u_i \sim \mathrm{Uniform}(-\xi/2, \xi/2)$ and $y_i \sim \mathcal{N}(0, \sigma^2)$ are independent.

The lemma is a consequence of the fact that the grid shift γ_i is uniformly distributed and independent of the Gaussian noise y_i, which makes the perturbation caused by rounding to the nearest grid point follow a uniform distribution on $(-\xi/2, \xi/2)$. The full proof is provided in Appendix E.

From a utility perspective, the dithered Gaussian mechanism is therefore equivalent to adding Gaussian noise $\mathcal{N}(0, \sigma^2)$ followed by uniform noise $\mathrm{Uniform}(-\xi/2, \xi/2)$. The variance of the resulting noise is

$$\mathrm{Var}(y_i + u_i) = \sigma^2 + \xi^2/12. \tag{9}$$

We can also quantify directly how close this noise distribution is to the Gaussian distribution used by the standard Gaussian mechanism in terms of the total variation distance.

Proposition 2. *Let $G_\sigma \sim \mathcal{N}(0, \sigma^2)$, and let $U_\xi \sim \mathrm{Uniform}(-\xi/2, \xi/2)$ be independent of G_σ. Then*

$$d_{\mathrm{TV}}(G_\sigma + U_\xi, G_\sigma) \leq 0.0202 \left(\frac{\xi}{\sigma}\right)^2. \tag{10}$$

Thus the distributional error introduced by dithering is quadratic in the relative grid width ξ/σ.

Densities for dithered Gaussians for various choices of ξ can be seen in Figure 2. For $\xi = \sigma$ we see that the density is nearly indistinguishable from that of the standard Gaussian, while larger values of ξ somewhat flatten the distribution.

[1] `https://anonymous.4open.science/r/Dithered-Gaussian-Mechanism-2416/`

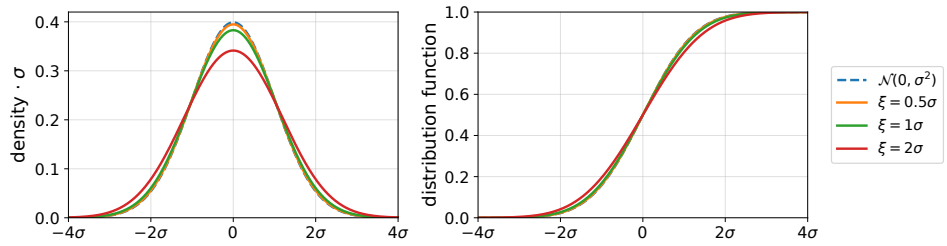


Figure 2: Density and cumulative distribution function of the noise distribution of the dithered Gaussian mechanism for various choices of ξ, compared to the Gaussian mechanism with the same privacy guarantee. Note that $\xi = \sigma$ provides a good approximation of the Gaussian distribution.

Randomness complexity. Next, we analyze the randomness complexity of the dithered Gaussian mechanism. We distinguish between the private (high-quality) randomness used for discrete sampling and the public randomness used to generate the pair (a, b) of uniform random variables defining the grid shift γ. The value γ may be shared with the adversary without compromising privacy, its purpose is only to ensure unbiasedness, and to improve utility and expected randomness complexity.

We bound the entropy of the dithered Gaussian mechanism in the following lemma:

Lemma 3. *The worst-case private binary entropy of the dithered Gaussian mechanism is bounded by*

$$H(Z \mid \gamma) \leq \frac{d}{2} \log_2 \left(2\pi e \left[\left(\frac{\sigma}{\xi} + \frac{1}{2} \right)^2 + \frac{1}{12} \right] \right). \tag{11}$$

Moreover, as $\sigma/\xi \to 0$, the following tighter conditional entropy bound holds:

$$\mathbb{E}_\gamma \left[H(Z \mid \gamma) \right] = O\left(\frac{d\sigma}{\xi} \log \frac{\xi}{\sigma} \right). \tag{12}$$

The full proof is given in Appendix E.

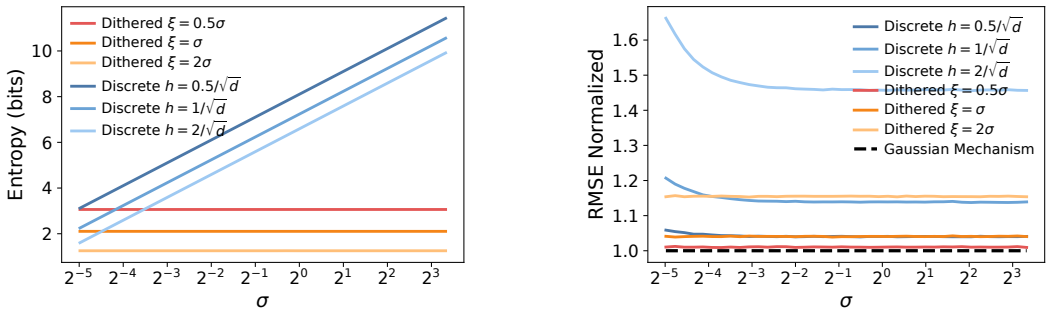
As noted above one can choose $\xi \approx \sigma$ and get an error distribution that is close to Gaussian. In this case, the lemma says that the entropy per coordinate is a small constant. Intuitively, the mechanism has low entropy for $\xi = \Omega(\sigma)$ since the random noise y_i is unlikely to make $\mathcal{M}(f(X))_i$ differ significantly from the output with zero noise, $y_i = 0$. Thus most probability mass for each coordinate is on a few output values, resulting in low entropy. For $\xi \gg \sigma$ the output distribution is almost deterministic for most values of γ_i, resulting in even smaller expected entropy. This argument extends and refines similar observations by Ghentiyala (2026).

In theory, using techniques for sampling from low-entropy distributions based on explicit, ordered sampling probabilities (Knuth & Yao, 1976; Devroye, 1986), multiple runs of the mechanism can be implemented in time near-linear in d using an expected number of random bits that matches the sum of all private entropy bounds plus $O(\log(1/\beta))$ bits with probability $1 - \beta$. As mentioned above, details of the implementation of our actual sampling mechanism can be found in Appendix B.

Lower Bound. In Appendix D, we prove the following lower bound on the private entropy of any differentially private mechanism for releasing a sensitivity-C statistic. To make the binary entropy well-defined we consider the special case where the statistic is integer-valued. Inputs z and z' are considered neighboring if $|z - z'| \leq C$ where $C \in \mathbb{N}$. Let U denote the public randomness, and write $\mathcal{M}_U$ for the mechanism obtained after conditioning on U.

Theorem 4. *There are absolute constants $c, c_0 > 0$ such that for $\varepsilon \in (0, 1)$ and $\delta \leq c_0 \varepsilon$ the following holds. Suppose $\mathcal{M}_U : \mathbb{Z} \to \mathbb{Z}$ is an (ε, δ)-differentially private mechanism under sensitivity C, for every fixed value of public randomness U. If, for some $\alpha \in \mathbb{N}$ and $\beta \in (0, \frac{1}{10})$, for every $z \in \mathbb{Z}$ it holds that $\Pr_{U,R}[|\mathcal{M}_U(z) - z| > \alpha] \leq \beta$ (where R is the private randomness) then there exists $z \in \mathbb{Z}$ such that*

$$\mathbb{E}_U\big[\, H(\mathcal{M}_U(z))\,\big] \geq c\, \frac{C}{\varepsilon\, \alpha}.$$



(a) Entropy in bits.

(b) Normalized RMSE.

Figure 3: Comparison of the dithered and discrete Gaussian mechanisms as a function of the noise scale σ. The left panel reports the entropy per coordinate. The right panel reports the RMSE on vectors in dimension $d = 1000$ sampled uniformly from the unit sphere. For each value of σ, we run each mechanism 1000 times and compute the average reconstruction error. The RMSE is normalized by the Gaussian mechanism baseline, which is equivalent to dividing by σ. The proposed dithered Gaussian mechanism requires a much smaller number of bits, which is independent of σ.

The proof proceeds by fixing a suitable value of the public randomness and using a graph-theoretic argument based on component stability and barrier structure to derive a lower bound on the achievable accuracy. Since the Gaussian noise scale needed for privacy is proportional to the clipping radius C (up to the usual dependence on ε, δ), this lower bound shows that the relevant scale for private entropy is the ratio between the privacy noise scale and the accuracy scale. This is consistent with Lemma 3: the private entropy of the dithered Gaussian mechanism depends on the dimensionless ratio σ/ξ, where σ is the Gaussian noise scale and ξ is the discretization width. In particular, choosing $\xi = \Theta(\sigma)$ gives constant private entropy per coordinate, while taking a finer grid increases the entropy only logarithmically in σ/ξ.

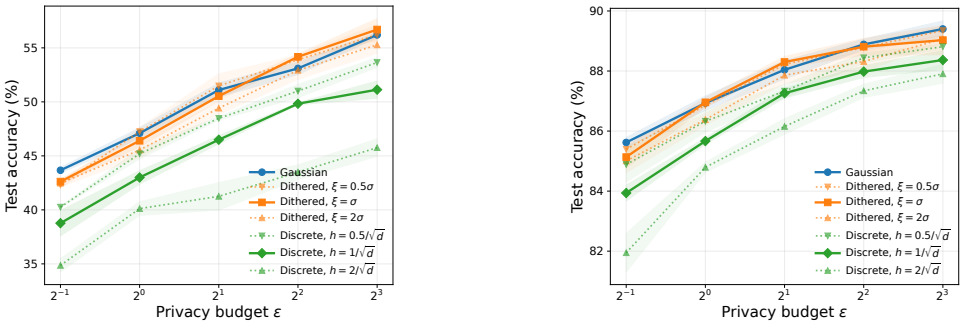
Dithered Laplace mechanism. The proposed dithering scheme also extends to the Laplace mechanism. Appendix F gives the corresponding analysis, including an improved randomness complexity bound over Canonne et al. (2025); see Theorem 9.

4 Experiments

Entropy and RMSE. Assume that the sensitive vectors are drawn from the unit sphere in dimension $d = 1000$. In the case of the discrete Gaussian mechanism, these vectors are further discretized with discretization levels $h \in \{\frac{0.5}{\sqrt{d}}, \frac{1}{\sqrt{d}}, \frac{2}{\sqrt{d}}\}$, roughly preserving vector norms. We set the resampling parameter to $\beta = e^{-1/2}$, as suggested by Kairouz et al. (2021). For the dithered Gaussian mechanism, we vary $\xi \in \{0.5\sigma, \sigma, 2\sigma\}$. In Figure 3, we show that, for a sufficiently large noise level σ, our method has much lower entropy and therefore requires fewer random bits to sample. In terms of utility, measured by the root mean squared error normalized by σ, the dithered Gaussian mechanism is much closer to the Gaussian mechanism, partly because it does not require discretizing the vectors.

Time Comparison. We compare the runtime of the dithered and discrete Gaussian mechanisms for dimensions $d \in \{10^3, 10^4, 10^5, 10^6, 10^7\}$, using the same parameter choices as above. Our main baseline is the discrete Gaussian implementation in `jax_privacy` (Balle et al., 2025), the fastest implementation we found. Across the considered hardware and randomness settings, the dithered Gaussian mechanism is between $2\times$ and $500\times$ faster, with the largest gains in high dimensions; see Tables 1–4. We additionally consider discrete inputs in $[-100, 100]$, a setting favorable to the discrete Gaussian mechanism, and still observe speedups of up to $10\times$ for $d \geq 10^4$; see Table 6. Full experimental and implementation details are provided in Appendix C.

Model Training. We next evaluate the effect of the dithered Gaussian mechanism in an end-to-end DP-SGD training. We train a ConvNet on CIFAR-10 and fine-tune BERT-Base on the IMDb dataset using Opacus (Yousefpour et al., 2021) for privacy budgets $\varepsilon \in \{0.5, 1, 2, 4, 8\}$ with $\delta = 10^{-5}$,



(a) Test accuracy on CIFAR-10.

(b) Test accuracy on IMDb.

Figure 4: Comparison of Gaussian, discrete and dithered Gaussian mechanisms across privacy budgets ε with $\delta = 10^{-5}$ for ConvNet training on CIFAR-10 and BERT-Base fine-tuning on IMDb.

and compare the Gaussian, discrete Gaussian, and our dithered Gaussian mechanism. As shown in Figure 4 and Tables 8 and 9, the dithered Gaussian mechanism closely matches the accuracy of the Gaussian baseline. The accuracy degradation of the discrete Gaussian mechanism is more pronounced, although it is reduced by using a finer discretization grid.

The runtime comparisons in Tables 7 and 10 shows that the dithered Gaussian mechanism with cryptographically secure noise incurs only a 3%–7% runtime overhead relative to standard Gaussian DP-SGD for batch size 512. For the discrete Gaussian mechanism, we follow the implementation in `jax_privacy` and use rounding to neighboring integers, without random rotations, for computational efficiency. Nevertheless, this implementation still incurs approximately a 45% runtime overhead for ConvNet and 320% for BERT-Base. We also compare with Opacus' "secure mode", which is slightly slower than our proposed mechanism, mainly because it requires sampling a larger number of random bytes. However, in terms of protection against floating-point vulnerabilities and privacy accounting, secure mode should be viewed as a heuristic rather than as providing a formal guarantee. Further details on the experimental setup, privacy accounting, and implementations are provided in Appendix C.

The overhead is indicative of what one can expect in general, though the figure will depend on how much time is spent on randomness generation, which in turn depends on factors such as batch size. For instance, even when using only cryptographically secure noise, much larger overheads were observed in the work of Egan (2024), ranging from 74% to over 400%. In that work, a rather small batch size of 32 is used, making the time spent on noise generation relatively larger. It should therefore be expected that, with a much larger batch size, the overhead would drop.

5 Conclusion and Future Directions

We have introduced the dithered Gaussian mechanism, an alternative to previous discrete noise mechanisms, which protects against floating-point vulnerabilities without requiring separate privacy accounting. Instead, it directly inherits the privacy guarantees of the Gaussian mechanism via post-processing. We prove that it requires fewer random bits than the discrete Gaussian mechanism and only moderately increases the time needed to sample the noise compared with a naive, non-secure implementation.

A promising direction for future work is to generalize the proposed dithered Gaussian mechanism to the distributed differential privacy setting by integrating it with secure aggregation in federated learning. The main obstacle is that the dithered Gaussian mechanism is not additive, which prevents its noise from being straightforwardly decomposed across clients. A separate vulnerability, shared by all of the methods considered here, is exposure to timing attacks. We also note that the mechanism would likely benefit from dedicated hardware support to directly represent vectors in discretized form, rather than converting to a floating point representation, and to speed up the sampling step. Conceivably this could make cryptographically secure randomness available in DP-SGD with a truly negligible performance overhead.

AI USE STATEMENT

In this work, we used generative AI tools for code assistance, grammar editing and proof assistance with Lemma 7. We have not used generative AI tools for the proof of Lemma 3. Additionally, we used generative AI tools for proofreading and editorial suggestions before submission. We have reviewed all AI-assisted work. We proofread and tested the code for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide a detailed description of the experimental setup, hardware, and hyperparameters in Appendix C. To support reproducibility, we also provide an implementation of the dithered Gaussian mechanism via an anonymous link.

REFERENCES

Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *ACM SIGSAC Conference on Computer and Communications Security*, pp. 308–318, 2016. doi: 10.1145/2976749.2978318. URL https://doi.org/10.1145/2976749.2978318.

Naman Agarwal, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Brendan McMahan. cpSGD: communication-efficient and differentially-private distributed SGD. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 7575–7586, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/21ce689121e39821d07d04faab328370-Abstract.html.

Naman Agarwal, Peter Kairouz, and Ziyu Liu. The Skellam mechanism for differentially private federated learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5052–5064, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/285baacbdf8fda1de94b19282acd23e2-Abstract.html.

Victor Balcer and Salil Vadhan. Differential privacy on finite computers. In *Innovations in Theoretical Computer Science Conference (ITCS)*, volume 94 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 43:1–43:21, Dagstuhl, Germany, 2018. ISBN 978-3-95977-060-6. doi: 10.4230/LIPIcs.ITCS.2018.43. URL https://doi.org/10.4230/LIPIcs.ITCS.2018.43.

Borja Balle and Yu-Xiang Wang. Improving the Gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning (ICML)*, 2018.

Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: tight analyses via couplings and divergences. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 6280–6290, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/3b5020bb891119b9f5130f1fea9bd773-Abstract.html.

Borja Balle, Leonard Berrada, Zachary Charles, Christopher A Choquette-Choo, Soham De, Vadym Doroshenko, Dj Dvijotham, Andrew Galen, Arun Ganesh, Sahra Ghalebikesabi, Jamie Hayes, Peter Kairouz, Ryan McKenna, Brendan McMahan, Aneesh Pappu, Natalia Ponomareva, Mikhail Pravilov, Keith Rush, Samuel L Smith, Robert Stanforth, and Chaitanya Mishra, 2025. URL http://github.com/google-deepmind/jax_privacy. JAX-Privacy: Algorithms for Privacy-Preserving Machine Learning in JAX.

Ergute Bao, Yizheng Zhu, Xiaokui Xiao, Yin Yang, Beng Chin Ooi, Benjamin Hong Meng Tan, and Khin Mi Mi Aung. Skellam mixture mechanism: a novel approach to federated learning with differential privacy. *Proc. VLDB Endow.*, 15(11):2348–2360, 2022. doi: 10.14778/3551793.3551798. URL https://www.vldb.org/pvldb/vol15/p2348-bao.pdf.

Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: efficient algorithms and tight error bounds. In *Symposium on Foundations of Computer Science (FOCS)*,

pp. 464–473. IEEE, 2014. doi: 10.1109/FOCS.2014.56. URL https://doi.org/10.1109/FOCS.2014.56.

Clément L. Canonne, Gautam Kamath, and Thomas Steinke. The discrete Gaussian for differential privacy. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/b53b3a3d6ab90ce0268229151c9bde11-Abstract.html.

Clément L. Canonne, Francis E. Su, and Salil P. Vadhan. The randomness complexity of differential privacy. In *16th Innovations in Theoretical Computer Science Conference (ITCS)*, volume 325 of *LIPIcs*, pp. 27:1–27:21, 2025. doi: 10.4230/LIPIcs.ITCS.2025.27. URL https://doi.org/10.4230/LIPIcs.ITCS.2025.27.

J. Lawrence Carter and Mark N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18(2):143–154, 1979. doi: 10.1016/0022-0000(79)90044-8. URL https://doi.org/10.1016/0022-0000(79)90044-8.

Wei-Ning Chen, Ayfer Ozgur, and Peter Kairouz. The Poisson binomial mechanism for unbiased federated learning with secure aggregation. In *International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pp. 3490–3506, 2022. URL https://proceedings.mlr.press/v162/chen22s.html.

Tasos C Christofides and Eutichia Vaggelatou. Bounds for the distance between the distributions of sums of absolutely continuous iid convex-ordered random variables with applications. *Journal of applied probability*, 46(1):255–271, 2009.

Lynn Chua, Badih Ghazi, Charlie Harrison, Ethan Leeman, Pritish Kamath, Ravi Kumar, Pasin Manurangsi, Amer Sinha, and Chiyuan Zhang. Balls-and-bins sampling for DP-SGD. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 258 of *Proceedings of Machine Learning Research*, pp. 946–954, 2025. URL https://proceedings.mlr.press/v258/chua25a.html.

Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley, 2001. ISBN 9780471062592. doi: 10.1002/0471200611. URL https://doi.org/10.1002/0471200611.

Luc Devroye. *Non-uniform random variate generation*. Springer, New York, 1986. ISBN 978-1-4613-8645-2. doi: 10.1007/978-1-4613-8643-8. URL https://doi.org/10.1007/978-1-4613-8643-8.

Zeyu Ding, John Durrell, Daniel Kifer, Prottay Protivash, Guanhong Wang, Yuxin Wang, Yingtai Xiao, and Danfeng Zhang. Avoiding floating-point side channels in the report noisy max with gap mechanism. *Journal of Privacy and Confidentiality*, 15(3), 2025. doi: 10.29012/jpc.894. URL https://journalprivacyconfidentiality.org/index.php/jpc/article/view/894.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, aug 2014. ISSN 1551-305X. doi: 10.1561/0400000042. URL https://doi.org/10.1561/0400000042.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: privacy via distributed noise generation. In *Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, volume 4004 of *Lecture Notes in Computer Science*, pp. 486–503. Springer, 2006. doi: 10.1007/11761679_29. URL https://doi.org/10.1007/11761679_29.

Shannon Egan. High-speed random number generator co-processors for machine learning and AI acceleration. In *NeurIPS 2024 Workshop Machine Learning with New Compute Paradigms*, 2024. URL https://openreview.net/forum?id=8oraBCaYbm.

Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. Amplification by shuffling: from local to central differential privacy via anonymity. In *Symposium on Discrete Algorithms (SODA)*, pp. 2468–2479, 2019. doi: 10.1137/1.9781611975482.151. URL https://doi.org/10.1137/1.9781611975482.151.

Vitaly Feldman and Moshe Shenfeld. Privacy amplification by random allocation, 2025. URL https://doi.org/10.48550/arXiv.2502.08202. arXiv preprint arXiv:2502.08202.

Vitaly Feldman, Audra McMillan, and Kunal Talwar. Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling. In *Symposium on Foundations of Computer Science (FOCS)*, pp. 954–964. IEEE, 2021. doi: 10.1109/FOCS52979.2021.00096. URL https://doi.org/10.1109/FOCS52979.2021.00096.

Vitaly Feldman, Audra McMillan, and Kunal Talwar. Stronger privacy amplification by shuffling for Rényi and approximate differential privacy. In *Symposium on Discrete Algorithms (SODA)*, pp. 4966–4981. SIAM, 2023. doi: 10.1137/1.9781611977554.ch181. URL https://doi.org/10.1137/1.9781611977554.ch181.

Quan Geng and Pramod Viswanath. The optimal noise-adding mechanism in differential privacy. *IEEE Transactions on Information Theory*, 62(2):925–951, 2016. doi: 10.1109/TIT.2015.2504967. URL https://doi.org/10.1109/TIT.2015.2504967.

Surendra Ghentiyala. Efficient derandomization of differentially private counting queries. In *Symposium on Simplicity in Algorithms (SOSA)*, 2026. URL https://doi.org/10.1137/1.9781611978964.14.

Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. Universally utility-maximizing privacy mechanisms. In *Symposium on Theory of Computing (STOC)*, pp. 351–360, 2009. doi: 10.1145/1536414.1536464. URL https://doi.org/10.1145/1536414.1536464.

Samuel Haney, Damien Desfontaines, Luke Hartman, Ruchit Shrestha, and Michael Hay. Precision-based attacks and interval refining: how to break, then fix, differential privacy on finite computers, 2022. arXiv preprint arXiv:2207.13793.

Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In *Symposium on Theory of Computing (STOC)*, pp. 705–714, 2010. doi: 10.1145/1806689.1806786. URL https://doi.org/10.1145/1806689.1806786.

Naoise Holohan and Stefano Braghin. Secure random sampling in differential privacy. In *European Symposium on Research in Computer Security (ESORICS)*, volume 12973 of *Lecture Notes in Computer Science*, pp. 523–542. Springer, 2021. doi: 10.1007/978-3-030-88428-4_26. URL https://doi.org/10.1007/978-3-030-88428-4_26.

Naoise Holohan, Stefano Braghin, Pól Mac Aonghusa, and Killian Levacher. Diffprivlib: the IBM differential privacy library. *CoRR*, abs/1907.02444, jul 2019. doi: 10.48550/arXiv.1907.02444. URL http://arxiv.org/abs/1907.02444.

Christina Ilvento. Implementing the exponential mechanism with base-2 differential privacy. In *Conference on Computer and Communications Security (CCS)*, pp. 717–742, 2020. doi: 10.1145/3372297.3417269. URL https://doi.org/10.1145/3372297.3417269.

Jiankai Jin, Eleanor McMurtry, Benjamin IP Rubinstein, and Olga Ohrimenko. Are we there yet? timing and floating-point attacks on differential privacy systems. In *Symposium on Security and Privacy (SP)*, pp. 473–488. IEEE, 2022. doi: 10.1109/SP46214.2022.9833672. URL https://doi.org/10.1109/SP46214.2022.9833672.

Peter Kairouz, Ziyu Liu, and Thomas Steinke. The distributed discrete Gaussian mechanism for federated learning with secure aggregation. In *International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5201–5212. PMLR, 2021. URL http://proceedings.mlr.press/v139/kairouz21a.html.

Donald E. Knuth and Andrew C. Yao. The complexity of nonuniform random number generation. In *Algorithms and Complexity: New Directions and Recent Results*, pp. 357–428. Academic Press, 1976.

Antti Koskela, Joonas Jälkö, and Antti Honkela. Computing tight differential privacy guarantees using FFT. In *Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2560–2569, 2020. URL http://proceedings.mlr.press/v108/koskela20b.html.

Christian Janos Lebeda, Matthew Regehr, Gautam Kamath, and Thomas Steinke. Avoiding pitfalls for privacy accounting of subsampled mechanisms under composition. In *Conference on Secure and Trustworthy Machine Learning (SaTML)*, 2025. doi: 10.1109/SaTML64287.2025.00060. URL https://doi.org/10.1109/SaTML64287.2025.00060.

Cong Ling and Jean-Claude Belfiore. Achieving AWGN channel capacity with lattice Gaussian coding. *IEEE Transactions on Information Theory*, 60(10):5918–5929, 2014. doi: 10.1109/TIT.2014.2332343. URL https://doi.org/10.1109/TIT.2014.2332343.

Ilya Mironov. On significance of the least significant bits for differential privacy. In *ACM Conference on Computer and Communications Security (CCS)*, pp. 650–661, 2012. doi: 10.1145/2382196.2382264. URL https://doi.org/10.1145/2382196.2382264.

Ilya Mironov. Rényi differential privacy. In *IEEE Computer Security Foundations Symposium (CSF)*, pp. 263–275. IEEE, 2017. doi: 10.1109/CSF.2017.11. URL https://doi.org/10.1109/CSF.2017.11.

OpenDP. The OpenDP project, 2021. https://opendp.org.

Tiberiu Popoviciu. Sur les équations algébriques ayant toutes leurs racines réelles. *Mathematica*, pp. 129–145, 1935.

Python Software Foundation. *secrets – Generate secure random numbers for managing secrets*, 2015. https://docs.python.org/3/library/secrets.html.

Jan Schuchardt, Mihail Stoian, Arthur Kosmala, and Stephan Günnemann. Unified mechanism-specific amplification by subsampling and group privacy amplification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/8f9a3fdb29de4c3fd030a18a505223c6-Abstract-Conference.html.

Shuang Song, Kamalika Chaudhuri, and Anand D. Sarwate. Stochastic gradient descent with differentially private updates. In *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 245–248, 2013. doi: 10.1109/GlobalSIP.2013.6736861. URL https://doi.org/10.1109/GlobalSIP.2013.6736861.

TensorFlow Privacy, 2024. URL https://github.com/tensorflow/privacy. TensorFlow Privacy.

Jason Vander Woude, Peter Dixon, A. Pavan, Jamie Radcliffe, and N. V. Vinodchandran. Replicability in learning: geometric partitions and KKM-Sperner lemma. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/8ff87c96935244b63503f542472462b3-Abstract-Conference.html.

VaultGemma Team. VaultGemma: a differentially private Gemma model. *arXiv preprint arXiv:2510.15001*, 2025. doi: 10.48550/arXiv.2510.15001. URL https://arxiv.org/abs/2510.15001. Version 2, 22 Oct 2025.

Shaowei Wang, Yun Peng, Jin Li, Zikai Wen, Zhipeng Li, Shiyu Yu, Di Wang, and Wei Yang. Privacy amplification via shuffling: unified, simplified, and tightened. *CoRR*, abs/2304.05007, 2023. doi: 10.48550/arXiv.2304.05007. URL https://doi.org/10.48550/arXiv.2304.05007. arXiv preprint arXiv:2304.05007.

Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, et al. Opacus: user-friendly differential privacy library in PyTorch, 2021. URL https://arxiv.org/abs/2109.12298. arXiv preprint arXiv:2109.12298.

Yuqing Zhu, Jinshuo Dong, and Yu-Xiang Wang. Optimal accounting of differential privacy via characteristic function. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 151 of *Proceedings of Machine Learning Research*, pp. 4782–4817. PMLR, 2022. URL https://proceedings.mlr.press/v151/zhu22c.html.

A Differential Privacy Basics

In this appendix we recall the standard definitions of differential privacy, see Dwork & Roth (2014) for details and a historical overview. We adopt the standard *unbounded* (add/remove-one) adjacency model: two datasets $X, X' \subseteq \mathbb{X}$ are said to be *neighbors*, written $X \sim X'$, if one can be obtained from the other by either adding or removing a single element. Formally,

$$X \sim X' \quad \Longleftrightarrow \quad |\, X \triangle X' \,| = 1,$$

where $\triangle$ denotes the symmetric difference. This model corresponds to protecting the participation of a single individual, and is the default unless otherwise stated.

Definition 2 (Pure and approximate differential privacy)**.** *Let $\varepsilon \geq 0$ and $\delta \in [0, 1)$. A randomized mechanism $\mathcal{M}$ with range $\mathcal{Y}$ is* (ε, δ)-differentially private *if for all neighboring datasets $X \sim X'$ and all measurable $S \subseteq \mathcal{Y}$,*

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

When $\delta = 0$ we say that $\mathcal{M}$ is ε-differentially private *(*pure *differential privacy).*

Lemma 5 (Post-processing)**.** *If a mechanism $\mathcal{M} : \mathbb{X}^* \to \mathcal{Y}$ is (ε, δ)-differentially private and K is any (possibly randomized) mapping $K : \mathcal{Y} \to \mathcal{Z}$ whose internal randomness is independent of the input dataset X, then the composed mechanism $K \circ \mathcal{M}$ is also (ε, δ)-differentially private.*

Definition 3 (ℓ_p-sensitivity)**.** *For a function $f : \mathbb{X}^* \to \mathbb{R}^d$ and $p \in [1, \infty]$, the ℓ_p-sensitivity is*

$$\Delta_p(f) \;:=\; \sup_{X \sim X'} \|f(X) - f(X')\|_p.$$

Definition 4 (Laplace mechanism)**.** *For $\lambda > 0$ and a function $f : \mathbb{X}^* \to \mathbb{R}^d$, the* Laplace mechanism *is defined as $\mathcal{M}(X) \;=\; f(X) \;+\; y$ where $y_i \sim \mathrm{Laplace}(\lambda)$ independently for $i = 1, \ldots, d$.*

The Gaussian mechanism adds isotropic Gaussian noise to a vector-valued function. Namely, for $\sigma > 0$ and $f : \mathbb{X}^* \to \mathbb{R}^d$, it is given by $\mathcal{M}(X) = f(X) + z$, where $z \sim \mathcal{N}(0, \sigma^2 I_d)$. The following result of Balle & Wang (2018) gives an exact characterization of its privacy guarantee in terms of the ℓ_2-sensitivity of f.

Theorem 6 (Analytic Gaussian mechanism)**.** *Let $f : \mathbb{X}^* \to \mathbb{R}^d$ have ℓ_2-sensitivity $\Delta_2(f) > 0$, and let $\mathcal{M}(X) = f(X) + z$ with $z \sim \mathcal{N}(0, \sigma^2 I_d)$. Then, for $\varepsilon \geq 0$ and $\delta \in [0, 1)$, the mechanism $\mathcal{M}$ is (ε, δ)-differentially private if and only if*

$$\delta \;\geq\; \Phi\!\left(\frac{\Delta_2(f)}{2\sigma} - \frac{\varepsilon\sigma}{\Delta_2(f)}\right) - e^{\varepsilon}\Phi\!\left(-\frac{\Delta_2(f)}{2\sigma} - \frac{\varepsilon\sigma}{\Delta_2(f)}\right),$$

where Φ denotes the cumulative distribution function of the standard Gaussian distribution.

B Sampling Algorithm

In this section, we present Algorithm 2 for sampling from the dithered Gaussian distribution (equation 6). The critical property of the dithered Gaussian distribution is that, although it is discrete, its cumulative distribution function can be expressed directly in terms of the Gaussian CDF as $F_i(k) \;=\; \Phi\big((\xi(k + \gamma_i + \frac{1}{2}) - f(X)_i)/\sigma\big)$. Consequently, for any $U_i \in (0, 1)$, applying the Gaussian inverse CDF yields a value whose quantization matches the corresponding interval of the dithered distribution. In particular, given $U_i \sim \mathrm{Unif}(0, 1)$, the candidate $k_i \;=\; \lceil (f(X)_i + \sigma\Phi^{-1}(U_i))/\xi - \gamma_i - \frac{1}{2} \rceil$ is precisely the integer whose CDF interval contains U_i, that is, $U_i \in [F_i(k_i - 1), F_i(k_i))$. Hence, k_i is a valid sample from the desired dithered Gaussian distribution.

Sampling a true uniform random variable requires an infinite number of bits. Therefore, instead, we sample a finite number $r_i \times t$ of bits, structured into r_i blocks of t bits for efficiency, since sampling one extra bit at a time is impractical. The sampled bits determine an interval $[L_i, H_i)$ containing the uniform random variable that would be obtained if sampling were continued indefinitely. We use the midpoint $U_i = (L_i + H_i)/2$ to obtain a candidate k_i through the inverse CDF. If the entire interval $[L_i, H_i)$ is contained in the corresponding CDF interval $[F_i(k_i - 1), F_i(k_i))$, then all possible

Algorithm 2 Dithered Gaussian Sampling

Require: $f(X)_i \in \mathbb{R}$ and $\gamma_i \in [0, 1)$ for $i \in [d]$, parameters $\xi > 0$, $\sigma > 0$, block size $t \geq 1$
Ensure: Samples $Z_1, \ldots, Z_d$

Initialize active set $A \leftarrow [d]$; for each i, no outcome has yet been assigned
while $A \neq \emptyset$ **do**
  Generate a block $B_i \in \{0, \ldots, 2^t - 1\}$ of t random bits for each $i \in A$
  for each $i \in A$ **do**
    Let $B_{i,1}, \ldots, B_{i,r_i}$ be all blocks generated so far for row i
    $L_i \leftarrow \sum_{s=1}^{r_i} B_{i,s} 2^{-ts}, \quad R_i \leftarrow L_i + 2^{-tr_i}$
    $U_i \leftarrow (L_i + R_i)/2$
    $k_i \leftarrow \left\lfloor \dfrac{f(X)_i + \sigma\Phi^{-1}(U_i)}{\xi} - \gamma_i + \frac{1}{2} \right\rfloor$
    $F_i(k_i - 1) \leftarrow \Phi\left(\dfrac{\xi(k_i + \gamma_i - \frac{1}{2}) - f(X)_i}{\sigma} \right)$
    $F_i(k_i) \leftarrow \Phi\left(\dfrac{\xi(k_i + \gamma_i + \frac{1}{2}) - f(X)_i}{\sigma} \right)$
    if $[L_i, R_i) \subseteq [F_i(k_i - 1), F_i(k_i))$ **then**
      $Z_i \leftarrow k_i$
      Remove i from A
    end if
  end for
end while
return $Z_1, \ldots, Z_d$

continuations of the sampled bits lead to the same value k_i, and we output $Z_i = k_i$. Otherwise, we sample another block of t bits and refine the interval. We maintain a set of indices A for which the sampled value has not yet been determined; we call this the active set. In practice, we choose the block size so that most values are determined on the first attempt, and additional bits are sampled only for the remaining, smaller active set.

Finite precision in practical implementation. Being implemented in PyTorch, the efficiency of the sampler depends on resolving most coordinates using float64 arithmetic. For each coordinate, we first use `torch.special.ndtri` to obtain a candidate value k_i from the midpoint of the current random interval $[L_i, R_i)$, and then verify the candidate by evaluating the two corresponding CDF boundaries $F_i(k_i - 1)$ and $F_i(k_i)$ using `torch.special.ndtr`. To account for finite-precision error, we only accept a value when the interval lies inside the corresponding CDF cell with a conservative margin $\epsilon = 10^{-12}$, i.e., when $L_i \geq F_i(k_i - 1) + \epsilon$ and $R_i \leq F_i(k_i) - \epsilon$. Coordinates that cannot be certified are kept in the active set and sampled with additional random bits, which shrinks $[L_i, R_i)$ while leaving the target distribution unchanged.

After a fixed number of such fast refinement steps, the small number of unresolved coordinates are handled using arbitrary-precision arithmetic with the Python library `mpmath`. The original float64 parameters are converted to their exact rational representations, and the CDF boundaries are recomputed at a precision that increases with the number of sampled random bits. If the current random interval is still too close to a CDF boundary to be resolved at that precision, another block of random bits is sampled and the procedure is repeated. Thus, unlike the truncated implementation, the practical sampler introduces no truncation of the support: finite precision only determines whether a coordinate can be safely resolved in the fast float64 path or must be deferred to the high-precision fallback.

To make the procedure efficient and prevent float64 overflow or loss of precision, we also normalize the vector $f(X)$ by prescaling it by the clipping norm before clipping, thereby guaranteeing that its coordinates are bounded by 1 in absolute value. After secure discrete sampling, we rescale the vector by multiplying it by the clipping norm back. We may further require that the noise multiplier itself be represented by a reasonable float64 value, say $\sigma \in [10^{-5}, 10^2]$, to avoid loss of precision during the computation. These are, however, not binding assumptions, since we can always rely on arbitrary precision computation, albeit at the cost of reduced efficiency.

We provide an implementation of the proposed method in an anonymous repository

`https://anonymous.4open.science/r/Dithered-Gaussian-Mechanism-2416`

C Experimental Details

Randomness Efficiency. Our implementation of the dithered Gaussian mechanism takes 2 bytes of randomness per coordinate on average. Most of the coordinates need even less than this, but we take this overconservative estimate to gain efficiency. An efficient implementation of the discrete Gaussian mechanism in `jax_privacy` (Balle et al., 2025) takes on average 40 bytes per coordinate, so 20 times more randomness than we do. In particular, they sample 3 random variables of size 8 bytes and use oversampling of size

$$m = \left\lfloor \frac{5d}{3} \right\rfloor + 2\lfloor \sqrt{d} \rfloor + 4, \tag{13}$$

resulting in roughly $5/3 \times 24 = 40$ bytes per coordinate for sufficiently large d. Numerical experiments under the same settings as those presented in the tables confirm this finding.

Time Measurements. We measure the runtime required to privatize a d-dimensional vector with unit ℓ_2-sensitivity, using a fixed Gaussian noise multiplier of $\sigma = 0.01$. Our primary baseline is the discrete Gaussian mechanism implemented in `jax_privacy` (Balle et al., 2025), which is the most efficient implementation we found. A comparison with other, slower implementations, including OpenDP (OpenDP) and diffprivlib (Holohan et al., 2019), is reported in Table 5. We do not include TensorFlow Privacy (TensorFlow Privacy, 2024), since, to the best of our knowledge, it provides only an approximate implementation of the discrete Gaussian distribution and does not support cryptographically secure noise generation.

For the discrete Gaussian mechanism, we use the random-rotation procedure of Kairouz et al. (2021), with sensitivity given by equation 2 and resampling parameter $\beta = e^{-1/2}$. This procedure additionally requires padding the input dimension to the next power of two and applying a random rotation before discretization, but in return gives lower error in the privatized vector.

We consider dimensions $d \in \{10^3, 10^4, 10^5, 10^6, 10^7\}$, dithering parameters $\xi \in \{0.5\sigma, \sigma, 2\sigma\}$, and discrete-Gaussian discretization levels $h \in \{0.5/\sqrt{d}, 1/\sqrt{d}, 2/\sqrt{d}\}$. For both mechanisms, we consider cryptographically secure randomness generated using the Python standard-library `secrets` module, as well as faster PyTorch non-cryptographic pseudorandom generation. Our implementation of the dithered Gaussian mechanism is designed to run efficiently on GPUs. Implementations of the discrete Gaussian distribution are either designed primarily for CPUs or, as in the case of `jax_privacy`, benefit less from GPU acceleration because the main bottleneck is random-number generation, which is performed on the CPU. Overall, the dithered Gaussian mechanism is approximately 2–500$\times$ faster than the `jax_privacy` discrete Gaussian implementation across the tested settings, with the largest speedups occurring in high dimensions.

Tables 1, 2, 3, and 4 report the resulting noise-generation runtimes under the different hardware and randomness settings. CPU experiments use 16 cores of an AMD EPYC 9654 processor, while GPU experiments use a single NVIDIA A100 GPU together with 16 CPU cores. All reported runtimes are averaged over 100 runs.

We also report a runtime comparison for discrete inputs in the range $[-100, 100]$ with noise multiplier $\sigma = 10$. This setting favors the discrete Gaussian mechanism, as it no longer requires an additional discretization step. The results are reported in Table 6. In particular, for sufficiently large dimensions, $d \geq 10^4$, our dithered Gaussian mechanism is still significantly faster than the considered implementations of the discrete Gaussian mechanism.

Model Training Experiments. We compare the Gaussian, discrete Gaussian, and proposed dithered Gaussian mechanisms in the setting of DP-SGD training. We use the Opacus-based (Yousefpour et al., 2021) implementation of DP-SGD to train a ConvNet ($d \approx 3 \times 10^5$) on CIFAR-10 and fine tune BERT-Base ($d \approx 10^8$) on IMDb. In particular, for CIFAR-10 we use a batch size of 512, a clipping norm of 1, and train for 10 epochs under privacy budgets $\varepsilon \in \{0.5, 1, 2, 4, 8\}$ with $\delta = 10^{-5}$. The optimal learning rate was the same across the $2\times$ search grid and equal to 0.002 for all runs.

For IMDb experiments we use a batch size 1024, a clipping norm of 1, fine tune for 10 epochs with the same privacy budgets. The optimal learning rate was 0.0005. For the Gaussian and dithered Gaussian mechanisms, we use the PRV accountant provided by Opacus to account for subsampling. For the discrete Gaussian mechanism, following the implementation in `jax_privacy`, we apply deterministic nearest-integer rounding to each gradient and use the zCDP analysis of the discrete Gaussian privacy profile together with the RDP accountant from the `dp_accounting` library. This results in a slightly larger noise level than that required by the Gaussian mechanism. We note that using the algorithm of Kairouz et al. (2021) for the discrete Gaussian mechanism requires applying a random rotation and randomized rounding to each per-example gradient. In our experiments, this incurs approximately a $20\times$ runtime overhead, making this approach impractical for centralized learning.

We observe no significant accuracy degradation when using the dithered Gaussian mechanism; see Figure 4 and Tables 8 and 9. For the discrete Gaussian mechanism, the accuracy degradation is more pronounced, although it is reduced when using a finer discretization grid.

We report the runtime comparison 4b in Tables 7 and 10, using a single NVIDIA A100 GPU and 16 CPU cores. For noise generation, we use cryptographically secure randomness from the Python standard-library `secrets` module (Python Software Foundation, 2015). Our proposed dithered Gaussian mechanism incurs only a 3%–7% runtime overhead, whereas the discrete Gaussian mechanism is approximately 45% slower for CIFAR-10 and 3.2 times slower for IMDb experiments. We note that the current training implementation in `jax_privacy` does not include this randomized rotation, resulting in a larger sensitivity but a faster training. We also compare against the "secure mode" of Opacus, which uses cryptographically secure noise generation and heuristics to reduce floating-point vulnerabilities. This mode is slower than our proposed mechanism, mainly because of the larger amount of randomness required.

Table 1: Sampling runtime on CPU in milliseconds with non-secure noise (mean $\pm$ standard deviation) over 100 runs.

	$d = 10^3$	$d = 10^4$	$d = 10^5$	$d = 10^6$	$d = 10^7$
Dithered ($\xi = 0.5\sigma$)	0.35 ± 0.19	0.49 ± 0.08	1.51 ± 0.06	13.35 ± 0.29	$\mathbf{105 \pm 4}$
Dithered ($\xi = \sigma$)	0.17 ± 0.07	0.42 ± 0.08	$\mathbf{1.49 \pm 0.04}$	$\mathbf{13.25 \pm 0.24}$	127 ± 19
Dithered ($\xi = 2\sigma$)	$\mathbf{0.14 \pm 0.04}$	$\mathbf{0.37 \pm 0.11}$	1.50 ± 0.04	13.35 ± 0.22	131 ± 19
Discrete ($h = 0.5/\sqrt{d}$)	0.39 ± 0.01	2.55 ± 0.16	12.94 ± 0.91	79.2 ± 2.2	1868 ± 22
Discrete ($h = 1/\sqrt{d}$)	0.37 ± 0.01	2.34 ± 0.01	12.46 ± 0.09	79.8 ± 2.5	1866 ± 20
Discrete ($h = 2/\sqrt{d}$)	0.37 ± 0.01	2.16 ± 0.01	12.56 ± 0.31	78.6 ± 1.7	1872 ± 23

Table 2: Sampling runtime on GPU in milliseconds with non-secure noise (mean $\pm$ standard deviation) over 100 runs.

	$d = 10^3$	$d = 10^4$	$d = 10^5$	$d = 10^6$	$d = 10^7$
Dithered ($\xi = 0.5\sigma$)	0.80 ± 0.44	1.40 ± 0.30	1.52 ± 0.29	2.19 ± 0.36	3.38 ± 0.39
Dithered ($\xi = \sigma$)	0.62 ± 0.30	1.22 ± 0.35	$\mathbf{1.43 \pm 0.15}$	1.92 ± 0.43	$\mathbf{3.14 \pm 0.09}$
Dithered ($\xi = 2\sigma$)	$\mathbf{0.54 \pm 0.21}$	$\mathbf{1.09 \pm 0.45}$	1.43 ± 0.17	$\mathbf{1.75 \pm 0.41}$	3.16 ± 0.25
Discrete ($h = 0.5/\sqrt{d}$)	1.50 ± 0.03	3.52 ± 0.05	13.22 ± 0.44	104.4 ± 0.8	1635 ± 7
Discrete ($h = 1/\sqrt{d}$)	1.52 ± 0.39	3.49 ± 0.06	13.18 ± 0.14	104.3 ± 0.6	1639 ± 5
Discrete ($h = 2/\sqrt{d}$)	1.47 ± 0.04	3.20 ± 0.13	13.19 ± 0.10	104.3 ± 0.8	1636 ± 5

Table 3: Sampling runtime on CPU in milliseconds with cryptographically secure noise (mean $\pm$ standard deviation) over 100 runs.

	$d = 10^3$	$d = 10^4$	$d = 10^5$	$d = 10^6$	$d = 10^7$
Dithered ($\xi = 0.5\sigma$)	0.28 ± 0.16	0.72 ± 0.37	1.39 ± 0.07	12.66 ± 0.28	$\mathbf{105 \pm 1}$
Dithered ($\xi = \sigma$)	0.18 ± 0.07	0.41 ± 0.09	1.37 ± 0.06	12.56 ± 0.25	126 ± 21
Dithered ($\xi = 2\sigma$)	$\mathbf{0.17 \pm 0.06}$	$\mathbf{0.37 \pm 0.10}$	$\mathbf{1.36 \pm 0.03}$	$\mathbf{12.40 \pm 0.41}$	125 ± 18
Discrete ($h = 0.5/\sqrt{d}$)	0.50 ± 0.01	3.46 ± 0.01	27.04 ± 1.8	175.0 ± 2.3	3609 ± 39
Discrete ($h = 1/\sqrt{d}$)	0.50 ± 0.01	3.47 ± 0.03	26.13 ± 1.5	175.9 ± 2.9	3574 ± 18
Discrete ($h = 2/\sqrt{d}$)	0.51 ± 0.12	3.50 ± 0.21	25.66 ± 1.4	176.1 ± 3.1	3574 ± 16

Table 4: Sampling runtime on GPU in milliseconds with cryptographically secure noise (mean $\pm$ standard deviation) over 100 runs.

	$d = 10^3$	$d = 10^4$	$d = 10^5$	$d = 10^6$	$d = 10^7$
Dithered ($\xi = 0.5\sigma$)	0.72 ± 0.90	1.48 ± 0.20	2.24 ± 0.28	9.12 ± 0.42	74.66 ± 0.55
Dithered ($\xi = \sigma$)	0.60 ± 0.25	1.38 ± 0.36	2.26 ± 0.29	8.98 ± 0.44	$\mathbf{74.43 \pm 0.45}$
Dithered ($\xi = 2\sigma$)	$\mathbf{0.58 \pm 0.23}$	$\mathbf{1.16 \pm 0.43}$	$\mathbf{2.19 \pm 0.19}$	$\mathbf{8.75 \pm 0.43}$	74.80 ± 0.48
Discrete ($h = 0.5/\sqrt{d}$)	1.76 ± 0.04	6.64 ± 0.64	35.79 ± 2.1	300.6 ± 5.6	4793 ± 16
Discrete ($h = 1/\sqrt{d}$)	1.75 ± 0.04	6.37 ± 0.59	35.39 ± 0.31	299.4 ± 6.2	4797 ± 14
Discrete ($h = 2/\sqrt{d}$)	1.76 ± 0.03	6.24 ± 0.82	35.58 ± 0.68	299.1 ± 5.6	4796 ± 15

Table 5: CPU sampling runtime with cryptographically secure noise (mean $\pm$ standard deviation) over 100 runs, in milliseconds. We set $\xi = \sigma$ for the dithered Gaussian mechanism and $h = 1/\sqrt{d}$ for the discrete Gaussian mechanism. All methods are evaluated using 16 cores of an AMD EPYC 9654 CPU.

	$d = 10^3$	$d = 10^4$	$d = 10^5$	$d = 10^6$
Dithered Gaussian (ours)	$\mathbf{0.18 \pm 0.07}$	$\mathbf{0.41 \pm 0.09}$	$\mathbf{1.37 \pm 0.06}$	$\mathbf{12.56 \pm 0.25}$
Discrete Gaussian (jax_privacy)	0.50 ± 0.01	3.47 ± 0.03	26.13 ± 1.5	175.9 ± 2.9
Discrete Gaussian (OpenDP)	30.82 ± 0.86	342.1 ± 5.5	2193 ± 13	17230 ± 130
Discrete Gaussian (diffprivlib)	24.67 ± 0.76	267.4 ± 6.1	1906 ± 5	25320 ± 90

Table 6: Sampling runtime comparison with cryptographically secure noise (mean $\pm$ standard deviation) over 100 runs, in milliseconds, for discrete inputs with coordinates bounded to $[-100, 100]$ and noise multiplier $\sigma = 10$. All measurements are performed using 16 cores of an AMD EPYC 9654 CPU.

	$d = 10^3$	$d = 10^4$	$d = 10^5$	$d = 10^6$
Dithered Gaussian (ours)	0.36 ± 0.13	$\mathbf{0.46 \pm 0.09}$	$\mathbf{1.32 \pm 0.03}$	$\mathbf{11.51 \pm 0.81}$
Discrete Gaussian (jax_privacy)	$\mathbf{0.16 \pm 0.002}$	1.24 ± 0.01	12.27 ± 0.30	132.4 ± 2.0
Discrete Gaussian (OpenDP)	14.18 ± 0.45	140.1 ± 1.1	1400 ± 13	13970 ± 14
Discrete Gaussian (diffprivlib)	22.93 ± 0.71	210.3 ± 1.9	2113 ± 7	21160 ± 111

Table 7: Runtime on CIFAR-10 over 10 epochs in seconds across privacy levels (mean ± sample standard deviation) over 3 runs.

	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = 4$	$\varepsilon = 8$
Gaussian (Opacus)	$\mathbf{126.3 \pm 1.5}$	$\mathbf{128.9 \pm 5.2}$	129.2 ± 4.6	$\mathbf{127.4 \pm 3.3}$	$\mathbf{126.4 \pm 0.9}$
Opacus Secure Mode	135.5 ± 3.4	136.1 ± 1.1	133.5 ± 1.5	136.2 ± 1.5	134.3 ± 1.1
Dithered ($\xi = 0.5\sigma$)	141.8 ± 20.9	130.3 ± 2.9	133.4 ± 9.2	128.1 ± 1.1	128.8 ± 1.5
Dithered ($\xi = \sigma$)	138.0 ± 14.2	129.1 ± 2.0	129.6 ± 2.2	129.1 ± 1.6	131.9 ± 7.9
Dithered ($\xi = 2\sigma$)	129.7 ± 2.1	139.0 ± 21.3	$\mathbf{128.2 \pm 1.5}$	130.0 ± 4.1	149.4 ± 35.8
Discrete ($h = 0.5/\sqrt{d}$)	190.8 ± 2.4	191.1 ± 2.6	190.7 ± 1.9	193.6 ± 6.8	193.3 ± 2.4
Discrete ($h = 1/\sqrt{d}$)	189.2 ± 0.3	191.4 ± 1.9	192.7 ± 4.7	193.2 ± 4.0	191.0 ± 1.2
Discrete ($h = 2/\sqrt{d}$)	190.4 ± 2.7	190.1 ± 1.1	189.1 ± 1.0	190.4 ± 1.9	192.8 ± 2.9

Table 8: Test accuracy on CIFAR-10 in percent across privacy levels (mean ± sample standard deviation) over 3 runs.

	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = 4$	$\varepsilon = 8$
Gaussian (Opacus)	$\mathbf{43.7 \pm 0.3}$	47.1 ± 0.6	51.1 ± 0.8	53.1 ± 0.7	56.2 ± 0.3
Dithered ($\xi = 0.5\sigma$)	42.3 ± 0.4	$\mathbf{47.2 \pm 0.4}$	$\mathbf{51.5 \pm 1.3}$	53.9 ± 0.5	56.2 ± 1.1
Dithered ($\xi = \sigma$)	42.6 ± 0.3	46.4 ± 0.7	50.5 ± 0.5	$\mathbf{54.2 \pm 0.3}$	$\mathbf{56.7 \pm 1.2}$
Dithered ($\xi = 2\sigma$)	42.5 ± 0.4	45.5 ± 0.8	49.4 ± 0.7	52.9 ± 1.0	55.3 ± 1.0
Discrete ($h = 0.5/\sqrt{d}$)	40.3 ± 0.1	45.2 ± 0.6	48.5 ± 0.2	51.0 ± 0.7	53.6 ± 0.5
Discrete ($h = 1/\sqrt{d}$)	38.8 ± 1.4	43.0 ± 0.8	46.5 ± 0.5	49.8 ± 0.1	51.1 ± 1.0
Discrete ($h = 2/\sqrt{d}$)	34.9 ± 0.7	40.1 ± 0.8	41.3 ± 1.5	43.5 ± 0.8	45.8 ± 1.0

Table 9: Test accuracy of BERT-Base on IMDb in percent across privacy levels (mean ± sample standard deviation) over 3 runs.

	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = 4$	$\varepsilon = 8$
Gaussian (Opacus)	$\mathbf{85.6 \pm 0.2}$	86.9 ± 0.3	88.0 ± 0.4	$\mathbf{88.9 \pm 0.2}$	$\mathbf{89.4 \pm 0.3}$
Dithered ($\xi = 0.5\sigma$)	85.4 ± 0.3	86.9 ± 0.1	88.2 ± 0.1	88.8 ± 0.2	89.4 ± 0.2
Dithered ($\xi = \sigma$)	85.1 ± 0.4	$\mathbf{87.0 \pm 0.2}$	$\mathbf{88.3 \pm 0.2}$	88.8 ± 0.3	89.0 ± 0.6
Dithered ($\xi = 2\sigma$)	85.0 ± 0.5	86.4 ± 0.2	87.9 ± 0.3	88.3 ± 0.4	89.1 ± 0.2
Discrete ($h = 0.5/\sqrt{d}$)	84.9 ± 0.7	86.3 ± 0.4	87.3 ± 0.3	88.4 ± 0.2	88.8 ± 0.1
Discrete ($h = 1/\sqrt{d}$)	83.9 ± 0.3	85.7 ± 0.1	87.3 ± 0.2	88.0 ± 0.2	88.4 ± 0.4
Discrete ($h = 2/\sqrt{d}$)	82.0 ± 0.8	84.8 ± 0.2	86.2 ± 0.3	87.3 ± 0.3	87.9 ± 0.4

Table 10: Runtime of BERT-Base on IMDb over 10 epochs in seconds across privacy levels (mean ± sample standard deviation) over 3 runs.

	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = 4$	$\varepsilon = 8$
Gaussian (Opacus)	$\mathbf{3896 \pm 14}$	$\mathbf{3896 \pm 13}$	$\mathbf{3895 \pm 14}$	$\mathbf{3895 \pm 14}$	$\mathbf{3895 \pm 14}$
Opacus Secure Mode	4762 ± 10	4757 ± 6	4766 ± 9	4755 ± 4	4752 ± 8
Dithered ($\xi = 0.5\sigma$)	4167 ± 13	4158 ± 14	4158 ± 14	4158 ± 13	4157 ± 14
Dithered ($\xi = \sigma$)	4152 ± 14	4153 ± 15	4153 ± 15	4152 ± 14	4154 ± 14
Dithered ($\xi = 2\sigma$)	4149 ± 14	4149 ± 13	4149 ± 13	4148 ± 12	4149 ± 14
Discrete ($h = 0.5/\sqrt{d}$)	12680 ± 112	12534 ± 33	12583 ± 62	12540 ± 20	12537 ± 20
Discrete ($h = 1/\sqrt{d}$)	12556 ± 14	12594 ± 81	12517 ± 8	12606 ± 69	12544 ± 33
Discrete ($h = 2/\sqrt{d}$)	12608 ± 75	12600 ± 55	12560 ± 36	12557 ± 44	12545 ± 62

D LOWER BOUND

Canonne, Su, and Vadhan study the randomness complexity of differentially private mechanisms for the summation problem, where the input X is a dataset of n vectors in $[0,1]^d$. In this setting, the function has ℓ_1-sensitivity $\Delta_1(f) = d$. They prove a lower bound on the amount of randomness required by accurate differentially private mechanisms for this problem (Canonne et al., 2025, Cor. 4.2). Specifically, for an (ε, δ)-differentially private mechanism with Hamming adjacency and parameters $\varepsilon \leq 1/d$ and $\delta \leq 1/(6d^2)$, any mechanism that retains nontrivial accuracy (say $\beta \leq 1/d$ and $\alpha \leq n/2 - 1$ on inputs of length n) has expected randomness complexity

$$\mathbb{E}[H(\mathcal{M}(X))] \;\geq \log_2 d \;-\; O(1). \tag{14}$$

This holds even under *bounded differential privacy* where the data size n is public information.

In this section, we show an improved lower bound for approximate differential privacy by separating the public randomness U from the private randomness needed to ensure differential privacy. We consider the following setting. Given a clipping radius $C \in \mathbb{N}$, the function we consider is the identity function $f(z) = z$ defined on $\mathbb{Z}$, where integers z and z' are neighboring if and only if $|z - z'| \leq C$, so that the ℓ_1-sensitivity is C. A mechanism for f implies a mechanism for summation of integers in $\{0, \ldots, C\}$. We compute the private entropy of a mechanism $\mathcal{M}$ defined on $\mathbb{Z}$ as

$$H(\mathcal{M}) \;:=\; \sup_{z \in \mathbb{Z}} \mathbb{E}_U\big[\, H(\mathcal{M}_U(f(z)))\,\big],$$

where the expectation is over public randomness U and $H(\mathcal{M}_U(f(z)))$ is the Shannon entropy of the output distribution of $\mathcal{M}$ on input $f(z)$ given U.

Theorem 4. *There are absolute constants $c, c_0 > 0$ such that for $\varepsilon \in (0,1)$ and $\delta \leq c_0\varepsilon$ the following holds. Suppose $\mathcal{M}_U : \mathbb{Z} \to \mathbb{Z}$ is an (ε, δ)-differentially private mechanism under sensitivity C, for every fixed value of public randomness U. If, for some $\alpha \in \mathbb{N}$ and $\beta \in (0, \frac{1}{10})$, for every $z \in \mathbb{Z}$ it holds that $\Pr_{U,R}[|\mathcal{M}_U(z) - z| > \alpha] \leq \beta$ (where R is the private randomness) then there exists $z \in \mathbb{Z}$ such that*

$$\mathbb{E}_U\big[\, H(\mathcal{M}_U(z))\,\big] \geq c\,\frac{C}{\varepsilon\,\alpha}.$$

The proof proceeds by fixing a suitable value of the public randomness and using a graph-theoretic argument based on component stability and barrier structure to derive a lower bound on the achievable accuracy.

Proof. Write $H = H(\mathcal{M})$. We also use the standard group-privacy consequence of (ε, δ)-differential privacy: if two inputs are connected by t neighboring steps, then for every event E,

$$\Pr[\mathcal{M}_U(z') \in E] \geq e^{-t\varepsilon}\left(\Pr[\mathcal{M}_U(z) \in E] - \delta \sum_{j=0}^{t-1} e^{j\varepsilon}\right) \tag{15}$$

for every fixed value U of the public randomness.

We first record a simple consequence of accuracy. For a sufficiently small absolute constant $a > 0$, no mechanism satisfying the hypotheses can have $\alpha < aC/\varepsilon$. Indeed, let $z' = z + 2\alpha + 1$, so the intervals

$$I_z = [z - \alpha, z + \alpha], \qquad I_{z'} = [z' - \alpha, z' + \alpha]$$

are disjoint. Accuracy gives $\mathbb{E}_U \Pr[\mathcal{M}_U(z) \in I_z] \geq 1 - \beta$ and $\mathbb{E}_U \Pr[\mathcal{M}_U(z') \in I_{z'}] \geq 1 - \beta$. Since $I_z \cap I_{z'} = \emptyset$, the second inequality implies $\mathbb{E}_U \Pr[\mathcal{M}_U(z') \in I_z] \leq \beta$. On the other hand, averaging equation 15 over U and using $\delta \leq c_0\varepsilon$, with a and c_0 small enough, gives a constant lower bound larger than β on $\mathbb{E}_U \Pr[\mathcal{M}_U(z') \in I_z]$ whenever $\alpha < aC/\varepsilon$, a contradiction. Thus, in the non-vacuous case we may assume

$$\alpha \geq aC/\varepsilon. \tag{16}$$

Fix n and write $Q_n = \{1, \ldots, n\}$. Since the mechanism satisfies the stated error guarantee on every input,

$$\mathbb{E}_U \Pr_{z \sim Q_n}[|\mathcal{M}_U(z) - z| > \alpha] \leq \beta.$$

Also, by the definition of H,

$$\mathbb{E}_U \mathbb{E}_{z\sim Q_n}[H(\mathcal{M}_U(z))] \le H.$$

Markov's inequality therefore gives a value U^* such that

$$\Pr_{z\sim Q_n}[|\mathcal{M}_{U^*}(z) - z| > \alpha] \le 2\beta, \tag{17}$$

$$\mathbb{E}_{z\sim Q_n}[H(\mathcal{M}_{U^*}(z))] \le 3H. \tag{18}$$

Let

$$S = \{z \in Q_n : H(\mathcal{M}_{U^*}(z)) \ge 1/2\}.$$

By equation 18, $|S| \le 6Hn$. For every $z \notin S$, the output distribution of $\mathcal{M}_{U^*}(z)$ has an atom m_z of probability at least $7/8$: otherwise its Shannon entropy would be at least the binary entropy $h_2(1/8) > 1/2$.

Choose a sufficiently small absolute constant $b > 0$, and set

$$k = \max\{1, \lfloor bC/\varepsilon \rfloor\}.$$

If $z, z' \notin S$ and $|z - z'| \le k$, then $m_z = m_{z'}$. To see this, let $t = \lceil |z - z'|/C \rceil$. By the choice of k, $t\varepsilon \le b + \varepsilon$, and the additive term in equation 15 is at most an absolute constant that can be made smaller than $1/4$ by choosing c_0 small enough. Applying equation 15 to the event $\{\mathcal{M}_{U^*}(\cdot) = m_z\}$ gives probability greater than $1/8$ under input z'. Since $z' \notin S$, all atoms other than its atom of mass at least $7/8$ have probability at most $1/8$, so $m_z = m_{z'}$.

Build a graph G whose vertices are $Q_n \setminus S$, connecting two vertices when their distance is at most k. Each connected component is an interval in the line Q_n, and consecutive components are separated by a run of at least k points of S. Moreover, by the stability just proved, all vertices in a component D share a common modal output m_D.

For a component D, at most $2\alpha + 1$ of its vertices satisfy $|z - m_D| \le \alpha$. Thus, if $|D| > 4\alpha$, then for uniformly random $z \in D$,

$$\Pr[|\mathcal{M}_{U^*}(z) - z| > \alpha] \ge \frac{7}{8}\left(1 - \frac{2\alpha + 1}{|D|}\right) \ge \frac{7}{20}.$$

Together with equation 17, this implies that the total number of vertices lying in components of size larger than 4α is at most $6\beta n$. Consequently the total size of components of size at most 4α is at least

$$(1 - 6\beta)n - |S|.$$

On the other hand, the barrier property bounds the number of components by $|S|/k + 1$. Hence the total size of components of size at most 4α is at most

$$4\alpha\left(\frac{|S|}{k} + 1\right) \le 4\alpha\left(\frac{6Hn}{k} + 1\right).$$

Combining the last two displays and dividing by n, then letting $n \to \infty$, gives

$$1 - 6\beta \le 6H + \frac{24\alpha H}{k}.$$

By equation 16 and the definition of k, the ratio α/k is bounded below by an absolute constant. Since $\beta < 1/10$, this implies

$$H \ge c'\frac{k}{\alpha} \ge c\,\frac{C}{\varepsilon\alpha}$$

for absolute constants $c', c > 0$. This is the desired lower bound. □

E Properties of the Dithered Gaussian Mechanism

This section contains full proofs of properties of the dithered Gaussian mechanism that were sketched in the main body.

Lemma 1. *Over the randomness of the shift γ_i and index Z_i, $\mathcal{M}(f(X))_i$ is identically distributed to*

$$f(X)_i + y_i + u_i, \tag{8}$$

where $u_i \sim \mathrm{Uniform}(-\xi/2, \xi/2)$ and $y_i \sim \mathcal{N}(0, \sigma^2)$ are independent.

Proof. The dithered Gaussian mechanism is defined in equation 5:

$$\mathcal{M}(f(X))_i = \xi \left(\left\lfloor \frac{f(X)_i + y_i}{\xi} - \gamma_i + \frac{1}{2} \right\rfloor + \gamma_i \right), \tag{19}$$

where $\gamma_i = (ia + b) \bmod 1$ with $a, b \sim \mathrm{Uniform}([0,1)^2)$ independent of X and y. Consider the value

$$\frac{u_i}{\xi} := \left\lfloor t_i - \gamma_i + \frac{1}{2} \right\rfloor - t_i + \gamma_i, \qquad \text{where} \qquad t_i = \frac{f(X)_i + y_i}{\xi}. \tag{20}$$

We will show that $\frac{u_i}{\xi}$ is uniformly distributed on $[-1/2, 1/2]$ independently of t_i, which is sufficient for the proof of the lemma. Let $\{\cdot\}$ denote the fractional part. Then

$$t_i - \gamma_i + \frac{1}{2} = \left\lfloor t_i - \gamma_i + \frac{1}{2} \right\rfloor + \left\{ t_i - \gamma_i + \frac{1}{2} \right\}. \tag{21}$$

Therefore,

$$\frac{u_i}{\xi} = \frac{1}{2} - \left\{ t_i - \gamma_i + \frac{1}{2} \right\}. \tag{22}$$

It remains to show that the fractional part is uniformly distributed on $[0, 1)$ and independent of t_i. Fix arbitrary values of t_i and a. Conditional on these values, the quantity $t_i - ia + 1/2$ is a constant, while b remains uniform on $[0, 1)$. Hence, for any measurable set $\mathcal{U} \subseteq [0, 1)$,

$$\Pr\left(\left\{ t_i - ia + \frac{1}{2} - b \right\} \in \mathcal{U} \,\middle|\, t_i, a \right) = \lambda(\mathcal{U}), \tag{23}$$

where λ denotes Lebesgue measure on $[0, 1)$. This follows because the map

$$b \mapsto \left\{ t_i - ia + \frac{1}{2} - b \right\} \tag{24}$$

is a measure-preserving shift and reflection modulo 1. Therefore,

$$\left\{ t_i - \gamma_i + \frac{1}{2} \right\} \sim \mathrm{Uniform}([0, 1)) \tag{25}$$

independently of t_i. □

Proposition 2. *Let $G_\sigma \sim \mathcal{N}(0, \sigma^2)$, and let $U_\xi \sim \mathrm{Uniform}(-\xi/2, \xi/2)$ be independent of G_σ. Then*

$$d_{\mathrm{TV}}(G_\sigma + U_\xi, G_\sigma) \leq 0.0202 \left(\frac{\xi}{\sigma} \right)^2. \tag{10}$$

Proof. The proof is based on Lemma 2 of Christofides & Vaggelatou (2009), together with an explicit computation of $\|\phi''\|_1$ and substitution of the variance of the uniform random variable. For completeness, we provide the full proof here.

By scaling all random variables by $1/\sigma$, it suffices to compare the standard Gaussian $G \sim \mathcal{N}(0,1)$ to $G + U_r$, where $U_r \sim \mathrm{Uniform}(-r/2, r/2)$ and $r = \xi/\sigma$. The density of $G + U_r$ is

$$q_r(x) = \frac{1}{r}\int_{x-r/2}^{x+r/2} \phi(t)\,dt = \frac{\Phi(x+r/2) - \Phi(x-r/2)}{r}.$$

Thus

$$d_{\mathrm{TV}}(G_\sigma + U_\xi, G_\sigma) = \frac{1}{2}\int_{-\infty}^{\infty} \left| \frac{\Phi(x+r/2) - \Phi(x-r/2)}{r} - \phi(x) \right| dx,$$

since total variation distance between two distributions with densities is one half of the L^1-distance between the densities.

We next prove the bound. Using $\mathbb{E}[U_r] = 0$, we can write

$$q_r(x) - \phi(x) = \mathbb{E}[\phi(x + U_r) - \phi(x) - U_r\phi'(x)]\,.$$

For a fixed $v \in [-r/2, r/2]$, Taylor's formula with integral remainder gives

$$\phi(x+v) - \phi(x) - v\phi'(x) = \int_0^v (v-s)\phi''(x+s)\,ds.$$

Taking L^1-norms and using translation-invariance of the Lebesgue integral,

$$\|\phi(\cdot + v) - \phi - v\phi'\|_1 \le \frac{v^2}{2}\,\|\phi''\|_1. \tag{26}$$

Therefore

$$\|q_r - \phi\|_1 \le \frac{\mathbb{E}[U_r^2]}{2}\,\|\phi''\|_1 = \frac{r^2}{24}\,\|\phi''\|_1.$$

Since $\phi''(x) = (x^2 - 1)\phi(x)$, we have

$$\begin{aligned}\|\phi''\|_1 &= \int_{-\infty}^{\infty} |x^2 - 1|\phi(x)\,dx \\ &= 2\int_{-1}^{1} (1 - x^2)\phi(x)\,dx = 4\phi(1),\end{aligned} \tag{27}$$

where the last equality follows from $(x\phi(x))' = (1 - x^2)\phi(x)$. Thus

$$d_{\mathrm{TV}}(G + U_r, G) = \frac{1}{2}\|q_r - \phi\|_1 \le \frac{\phi(1)}{12} r^2,$$

and $\phi(1)/12 < 0.0202$, which gives equation 10. □

We remark that the constant $\phi(1)/12$ is sharp for a bound of this form.

Before proving Lemma 3, we show an auxiliary upper bound on the entropy of an integer-valued random variable in terms of its variance.

Lemma 7 (Discrete entropy bound (Cover & Thomas, 2001, Theorem 9.7.1))**.** *Let Z be an integer-valued random variable with finite variance. Then its Shannon binary entropy satisfies*

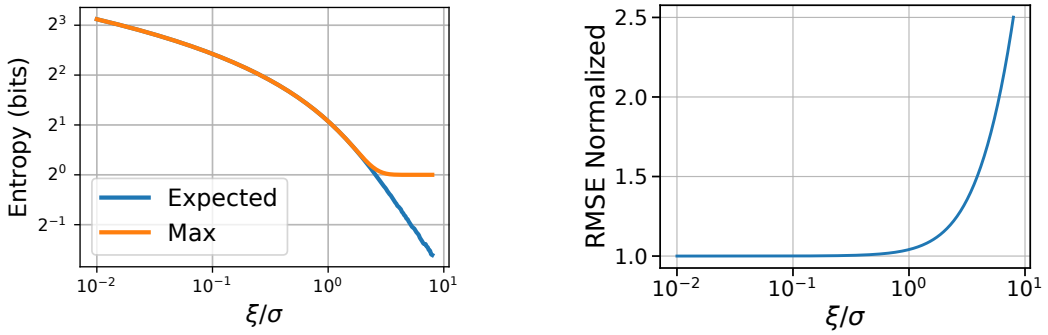
$$H(Z) \le \frac{1}{2}\log_2\left(2\pi e\left(\mathrm{Var}(Z) + \frac{1}{12}\right)\right). \tag{28}$$

Proof. We provide a more detailed, self-contained proof. Let $U \sim \mathrm{Uniform}(-1/2, 1/2]$ be independent of Z. Then $Z + U$ has density

$$\rho(x) := \rho_{Z+U}(x) = \sum_{k=-\infty}^{+\infty} \Pr[Z = k]\mathbf{1}_{\{k-\frac{1}{2} < x \le k+\frac{1}{2}\}}. \tag{29}$$

The density ρ is nonnegative and pointwise bounded by $\max_k \Pr[Z = k] \le 1$. Therefore, if μ denotes the distribution of $Z + U$, then for every measurable set $\mathcal{A}$,

$$\mu(\mathcal{A}) = \int_{\mathcal{A}} \rho(x)\,dx \le \lambda(\mathcal{A}). \tag{30}$$



(a) Entropy of the dithered Gaussian distribution.

(b) Root mean squared error normalized by σ.

Figure 5: The expected and maximum entropy (a) and normalized RMSE of the dithered Gaussian mechanism (b) as functions of ξ/σ.

In particular, if $\lambda(\mathcal{A}) = 0$, then $\mu(\mathcal{A}) = 0$. Hence μ is absolutely continuous with respect to the Lebesgue measure λ. Therefore, we can compute the differential entropy of $Z + U$:

$$h(Z+U) = -\int_{-\infty}^{+\infty} \rho(x) \log_2 \rho(x)\, dx = \sum_{k=-\infty}^{+\infty} -\int_{k-\frac{1}{2}}^{k+\frac{1}{2}} \rho(x) \log_2 \rho(x)\, dx \tag{31}$$

$$= \sum_{k=-\infty}^{+\infty} -\int_{k-\frac{1}{2}}^{k+\frac{1}{2}} \Pr[Z=k] \log_2 \Pr[Z=k]\, dx \tag{32}$$

$$= -\sum_{k=-\infty}^{+\infty} \Pr[Z=k] \log_2 \Pr[Z=k] = H(Z). \tag{33}$$

Thus, the differential entropy of $Z + U$ coincides with the discrete entropy of Z. Since the Gaussian distribution maximizes differential entropy among all absolutely continuous random variables with fixed variance,

$$h(Z+U) \le \frac{1}{2} \log_2 \left(2\pi e \operatorname{Var}(Z+U)\right). \tag{34}$$

Using independence of Z and U,

$$\operatorname{Var}(Z+U) = \operatorname{Var}(Z) + \operatorname{Var}(U) = \operatorname{Var}(Z) + \frac{1}{12}. \tag{35}$$

Combining the above identities yields

$$H(Z) \le \frac{1}{2} \log_2 \left(2\pi e \left(\operatorname{Var}(Z) + \frac{1}{12}\right)\right), \tag{36}$$

which completes the proof. □

Lemma 3. *The worst-case private binary entropy of the dithered Gaussian mechanism is bounded by*

$$H(Z \mid \gamma) \le \frac{d}{2} \log_2 \left(2\pi e \left[\left(\frac{\sigma}{\xi} + \frac{1}{2}\right)^2 + \frac{1}{12}\right]\right). \tag{11}$$

Moreover, as $\sigma/\xi \to 0$, the following tighter conditional entropy bound holds:

$$\mathbb{E}_\gamma \left[H(Z \mid \gamma)\right] = O\left(\frac{d\sigma}{\xi} \log \frac{\xi}{\sigma}\right). \tag{12}$$

Proof. Recall from equation 6 that, conditionally on γ_i and $f(X)_i$, the distribution of Z_i is given by

$$\Pr[Z_i = k] = \Phi\left(\frac{\xi}{\sigma}\left(k + \gamma_i + \frac{1}{2} - \frac{f(X)_i}{\xi}\right)\right) - \Phi\left(\frac{\xi}{\sigma}\left(k + \gamma_i - \frac{1}{2} - \frac{f(X)_i}{\xi}\right)\right). \tag{37}$$

Given the grid γ, the random variables Z_i are independent. Therefore,

$$H(Z \mid \gamma) = \sum_{i=1}^{d} H(Z_i \mid \gamma). \tag{38}$$

By Lemma 7, for each coordinate we have

$$H(Z_i \mid \gamma) \le \frac{1}{2}\log_2\left(2\pi e\left(\mathrm{Var}(Z_i \mid \gamma) + \frac{1}{12}\right)\right). \tag{39}$$

Now consider the random variable $Y_i \sim \mathcal{N}\left(\frac{f(X)_i}{\xi} - \gamma_i, \frac{\sigma^2}{\xi^2}\right)$. Then Z_i can be obtained by rounding Y_i to the nearest integer, since

$$\Pr\left(k - \frac{1}{2} \le Y_i < k + \frac{1}{2}\right) = \Pr[Z_i = k]. \tag{40}$$

Using the Cauchy-Schwarz inequality, we get

$$\mathrm{Var}(Z_i \mid \gamma_i) = \mathrm{Var}\left(Y_i + (Z_i - Y_i) \mid \gamma_i\right) \le \left(\sqrt{\mathrm{Var}(Y_i \mid \gamma_i)} + \sqrt{\mathrm{Var}(Z_i - Y_i \mid \gamma_i)}\right)^2. \tag{41}$$

Since $\mathrm{Var}(Y_i \mid \gamma_i) = \sigma^2/\xi^2$ and $|Z_i - Y_i| \le 1/2$, Popoviciu's inequality (Popoviciu, 1935) gives

$$\mathrm{Var}(Z_i - Y_i \mid \gamma_i) \le \frac{1}{4}. \tag{42}$$

Hence

$$\mathrm{Var}(Z_i \mid \gamma_i) \le \left(\frac{\sigma}{\xi} + \frac{1}{2}\right)^2. \tag{43}$$

Combining the above inequalities, we obtain

$$H(Z \mid \gamma) \le \frac{d}{2}\log_2\left(2\pi e\left[\left(\frac{\sigma}{\xi} + \frac{1}{2}\right)^2 + \frac{1}{12}\right]\right). \tag{44}$$

The right-hand side does not depend on γ, so taking expectation over γ does not change the bound.

Conditional Entropy Bound. In the regime $\sigma/\xi \to 0$, the Gaussian noise Y_i affects the rounded value only with small probability. For each coordinate i, let $t^i_{x,\gamma} := \left\lfloor \frac{f(X)_i}{\xi} - \gamma_i + \frac{1}{2} \right\rfloor$ denote the rounded value in the absence of Gaussian noise, and let $r^i_{x,\gamma} := \left\{ \frac{f(X)_i}{\xi} - \gamma_i + \frac{1}{2} \right\}$ denote the corresponding fractional part. We define $p^i_\gamma := \mathbb{P}\left(Z_i = t^i_{x,\gamma} \mid \gamma\right)$ as the probability that the Gaussian perturbation does not change the rounded value. This probability is given explicitly by

$$\begin{aligned} p^i_\gamma &= \Phi\left(\frac{\xi\left(t^i_{x,\gamma} + \gamma_i + \frac{1}{2}\right) - f(X)_i}{\sigma}\right) - \Phi\left(\frac{\xi\left(t^i_{x,\gamma} + \gamma_i - \frac{1}{2}\right) - f(X)_i}{\sigma}\right) \\ &= \Phi\left(\frac{\xi\left(1 - r^i_{x,\gamma}\right)}{\sigma}\right) - \Phi\left(-\frac{\xi r^i_{x,\gamma}}{\sigma}\right), \end{aligned} \tag{45}$$

where the second equality follows from $t^i_{x,\gamma} + r^i_{x,\gamma} = \frac{f(X)_i}{\xi} - \gamma_i + \frac{1}{2}$.

Since γ_i is uniformly distributed on $[0,1]$, $r^i_{x,\gamma}$ is also uniformly distributed on $[0,1]$. Therefore,

$$\begin{aligned} \mathbb{E}_\gamma\left[p^i_\gamma\right] &= \int_0^1 \left[\Phi\left(\frac{\xi(1-r)}{\sigma}\right) - \Phi\left(-\frac{\xi r}{\sigma}\right)\right] dr \\ &= \frac{\sigma}{\xi}\int_0^{\xi/\sigma} (2\Phi(r) - 1)\, dr \\ &= 2\Phi\left(\frac{\xi}{\sigma}\right) - 1 + \frac{2\sigma}{\xi}\phi\left(\frac{\xi}{\sigma}\right) - \sqrt{\frac{2}{\pi}}\frac{\sigma}{\xi}. \end{aligned} \tag{46}$$

Since $\phi(\xi/\sigma)$ and $2\Phi(\xi/\sigma) - 2$ decay exponentially as $\sigma/\xi \to 0$, we have

$$\mathbb{E}_\gamma\left[p_\gamma^i\right] = 1 - \sqrt{\frac{2}{\pi}}\frac{\sigma}{\xi} + o\left(\frac{\sigma}{\xi}\right). \tag{47}$$

We next separate the event $Z_i = t_{x,\gamma}^i$ in the definition of the entropy. For $k \neq t_{x,\gamma}^i$, define the conditional probabilities

$$q_{k,\gamma}^i := \mathbb{P}\left(Z_i = k \mid \gamma,\, Z_i \neq t_{x,\gamma}^i\right) = \frac{\mathbb{P}(Z_i = k \mid \gamma)}{1 - p_\gamma^i}. \tag{48}$$

Then

$$\begin{aligned} H(Z_i \mid \gamma) &= -p_\gamma^i \log_2 p_\gamma^i - \sum_{k \neq t_{x,\gamma}^i} \mathbb{P}(Z_i = k \mid \gamma) \log_2 \mathbb{P}(Z_i = k \mid \gamma) \\ &= -p_\gamma^i \log_2 p_\gamma^i - (1 - p_\gamma^i) \sum_{k \neq t_{x,\gamma}^i} q_{k,\gamma}^i \log_2\left((1 - p_\gamma^i) q_{k,\gamma}^i\right) \\ &= -p_\gamma^i \log_2 p_\gamma^i - (1 - p_\gamma^i)\log_2(1 - p_\gamma^i) - (1 - p_\gamma^i) \sum_{k \neq t_{x,\gamma}^i} q_{k,\gamma}^i \log_2 q_{k,\gamma}^i \\ &= h_2\left(p_\gamma^i\right) + (1 - p_\gamma^i) H\left(Z_i \mid \gamma,\, Z_i \neq t_{x,\gamma}^i\right), \end{aligned} \tag{49}$$

where $h_2(p) := -p\log_2 p - (1-p)\log_2(1-p)$ is the binary entropy function.

By Lemma 7,

$$H\left(Z_i \mid \gamma,\, Z_i \neq t_{x,\gamma}^i\right) \leq \frac{1}{2}\log_2\left(2\pi e\left[\mathrm{Var}\left(Z_i \mid \gamma,\, Z_i \neq t_{x,\gamma}^i\right) + \frac{1}{12}\right]\right). \tag{50}$$

By the conditional law of total variance,

$$\begin{aligned} \mathrm{Var}(Z_i \mid \gamma) = {} & \left(1 - p_\gamma^i\right) \mathrm{Var}\left(Z_i \mid \gamma,\, Z_i \neq t_{x,\gamma}^i\right) \\ & + p_\gamma^i \,\mathrm{Var}\left(Z_i \mid \gamma,\, Z_i = t_{x,\gamma}^i\right) \\ & + p_\gamma^i\left(1 - p_\gamma^i\right)\left(\mathbb{E}\left[Z_i \mid \gamma,\, Z_i = t_{x,\gamma}^i\right] - \mathbb{E}\left[Z_i \mid \gamma,\, Z_i \neq t_{x,\gamma}^i\right]\right)^2. \end{aligned} \tag{51}$$

Therefore,

$$\mathrm{Var}\left(Z_i \mid \gamma,\, Z_i \neq t_{x,\gamma}^i\right) \leq \frac{\mathrm{Var}(Z_i \mid \gamma)}{1 - p_\gamma^i} \leq \frac{1}{1 - p_\gamma^i}\left(\frac{\sigma}{\xi} + \frac{1}{2}\right)^2, \tag{52}$$

where the last inequality follows from equation 43. Consequently,

$$H(Z_i \mid \gamma) \leq h_2\left(p_\gamma^i\right) + \frac{1 - p_\gamma^i}{2}\log_2\left(2\pi e\left[\frac{\left(\frac{\sigma}{\xi} + \frac{1}{2}\right)^2}{1 - p_\gamma^i} + \frac{1}{12}\right]\right). \tag{53}$$

To take the expectation over γ, we use Jensen's inequality. The binary entropy function is concave because

$$h_2''(x) = -\frac{1}{x(1-x)\ln 2} < 0. \tag{54}$$

Moreover, for any $a, b > 0$, the function

$$g(x) := (1 - x)\log_2\left(\frac{a}{1 - x} + b\right), \qquad g''(x) = -\frac{a^2}{(1-x)\left(a + b(1-x)\right)^2 \ln 2} < 0, \tag{55}$$

is also concave. Therefore, the right-hand side of equation 53 is concave in p_γ^i. Jensen's inequality gives

$$\mathbb{E}_\gamma\left[H(Z_i \mid \gamma)\right] \leq h_2\left(\mathbb{E}_\gamma\left[p_\gamma^i\right]\right) + \frac{1 - \mathbb{E}_\gamma\left[p_\gamma^i\right]}{2}\log_2\left(2\pi e\left[\frac{\left(\frac{\sigma}{\xi} + \frac{1}{2}\right)^2}{1 - \mathbb{E}_\gamma\left[p_\gamma^i\right]} + \frac{1}{12}\right]\right). \tag{56}$$

By equation 47, $1 - \mathbb{E}_\gamma[p_\gamma^i] = \Theta(\sigma/\xi)$. Consequently, $h_2(\mathbb{E}_\gamma[p_\gamma^i]) = O\left(\frac{\sigma}{\xi}\log\frac{\xi}{\sigma}\right)$. Since $\left(\frac{\sigma}{\xi}+\frac{1}{2}\right)^2 = \Theta(1)$, the second term has the same asymptotic order. Hence,

$$\mathbb{E}_\gamma\left[H(Z_i \mid \gamma)\right] = O\left(\frac{\sigma}{\xi}\log\frac{\xi}{\sigma}\right). \tag{57}$$

Finally, by subadditivity of entropy, $H(Z \mid \gamma) \le \sum_{i=1}^d H(Z_i \mid \gamma)$. Taking expectations and summing over i yields

$$\mathbb{E}_\gamma\left[H(Z \mid \gamma)\right] = O\left(\frac{d\sigma}{\xi}\log\frac{\xi}{\sigma}\right), \tag{58}$$

which concludes the proof. □

F Dithered Laplace Mechanism

The proposed scheme can be generalized to other types of additive-noise mechanisms. In this section, we discuss the implications of our dithering scheme for the Laplace mechanism. In a recent paper (Canonne et al., 2025), Canonne, Su, and Vadhan showed that the randomness complexity of differentially private mechanisms for vector summation can be drastically reduced. When the statistic f has ℓ_1-sensitivity $\Delta_1(f)$, there exists an ε-differentially private mechanism that for $\alpha = \Omega(\Delta_1(f)\log(d)/\varepsilon)$ is $(\alpha, \mathrm{poly}(1/d))$-accurate with expected randomness complexity, as stated in the following corollary.

Corollary 8. *Let $\varepsilon > 0$, $\beta \in (0,1)$, and suppose the accuracy target α satisfies*

$$\alpha \ge C_0\,\frac{\Delta_1(f)}{\varepsilon}\,\log(2d/\beta)$$

for a sufficiently large universal constant C_0. Then there exists an ε-differentially private mechanism $\mathcal{M}$ that is (α,β)-accurate and whose expected randomness complexity is

$$R(\mathcal{M}) = O\!\left(\frac{d\,\Delta_1(f)}{\alpha\,\varepsilon}\cdot\log(d)\cdot\log\frac{\alpha\,\varepsilon}{\Delta_1(f)\log d} + \log d\right).$$

Choosing error parameter $\alpha = d^2\log^2(d)/\epsilon$ minimizes randomness complexity to $O(\log d)$ bits. At the other end of the trade-off, error $\alpha = O(\Delta_1(f)\log(d)/\epsilon)$ is asymptotically optimal (Hardt & Talwar, 2010; Geng & Viswanath, 2016) among ϵ-differentially private algorithms for $\beta = 1/\mathrm{poly}(d)$ and results in randomness complexity $\mathbb{E}[H(\mathcal{M}(f(X)))] = O(d)$ bits. While the mechanism of Canonne et al. (2025) is not computationally efficient, recent work by Ghentiyala (2026) provides a computationally efficient alternative (with slightly higher randomness complexity). By replacing the cumulative distribution function in the dithered Gaussian algorithm with that of the Laplace distribution with parameter λ (which controls the privacy level), we obtain the dithered Laplace mechanism; see Algorithm 3. On the upper bound side we show an improvement over the bound of Canonne et al. (2025) in the following theorem.

Theorem 9. *Suppose $f : \mathbb{X}^* \to \mathbb{R}^d$ has sensitivity $\Delta_1(f)$. For every $\varepsilon > 0$, $\beta \in (0,1)$, and*

$$\alpha > \frac{2\Delta_1(f)\log(d/\beta)}{\varepsilon},$$

there exists a random variable U and mechanisms $\mathcal{M}_U : \mathbb{X}^* \to \mathbb{R}^d$, defined by post-processing the Laplace mechanism on $f(X)$, that are ε-differentially private for every fixed value of U. Moreover, $\mathcal{M}_U(f(X))$ is an (α,β)-accurate approximation of $f(X)$, and its expected private randomness complexity satisfies

$$\mathbb{E}_U\big[H(\mathcal{M}_U(f(X)))\big] = O\!\left(\frac{d\,\Delta_1(f)}{\alpha\varepsilon}\log\!\left(\frac{\alpha\varepsilon}{\Delta_1(f)}\right)\right).$$

Compared with Corollary 8, our bound removes the additive $\log(d)$ term and improves the leading term by a factor of $\log(d)$. For error tolerance $\alpha = \omega(d\Delta_1(f)\log(d)/\epsilon)$, the private randomness complexity becomes $\mathbb{E}_U\big[H(\mathcal{M}_U(f(X)))\big] = o(1)$ independent of X. That is, the mechanism requires

Algorithm 3 Dithered Laplace Mechanism

Require: Function value $f(X) \in \mathbb{R}^d$, parameters $\xi > 0$, $\lambda > 0$

1: Sample public randomness $(a, b) \sim \mathrm{Uniform}([0,1)^2)$
2: **for** $i = 1, \ldots, d$ **do**
3: $\quad \gamma_i \leftarrow (a \cdot i + b) \bmod 1$
4: $\quad$ Sample an integer $Z_i \in \mathbb{Z}$ such that, for every $k \in \mathbb{Z}$,

$$\Pr[Z_i = k] = F_\lambda\left(\xi\left(k + \gamma_i + \frac{1}{2}\right) - f(X)_i\right) - F_\lambda\left(\xi\left(k + \gamma_i - \frac{1}{2}\right) - f(X)_i\right),$$

$$\text{where } F_\lambda(t) = \begin{cases} \frac{1}{2}e^{t/\lambda}, & t < 0, \\ 1 - \frac{1}{2}e^{-t/\lambda}, & t \geq 0, \end{cases}$$

5: $\quad \mathcal{M}(f(X))_i \leftarrow \xi(Z_i + \gamma_i)$
6: **end for**
7: **return** $\mathcal{M}(f(X))$

only a negligible amount of fresh, private randomness to run. For error $\alpha = \Theta(\Delta_1(f)\log(d/\beta)/\epsilon)$, matching the ℓ_∞-error of the standard Laplace mechanism, the private randomness complexity is $\mathbb{E}_U[H(\mathcal{M}_U(f(X)))] = O(d \log\log(d)/\log d)$ which is $o(d)$. The mechanism's public randomness U can be sampled using $O(\log(d))$ random bits with high probability, matching the lower bound equation 14 on total randomness. We prove Theorem 9, together with a bound on public randomness complexity, in the following lemma.

Lemma 10. *Let $f : \mathbb{X}^* \to \mathbb{R}^d$ have ℓ_1-sensitivity $\Delta_1(f)$. Consider the dithered Laplace mechanism $\mathcal{M}$ defined in Algorithm 3 with parameters $\xi > 0$, $\lambda > 0$, and public randomness $U = (a, b)$. Fix $\beta \in (0, 1)$. The mechanism satisfies:*

(a) ***Pure differential privacy:*** *For every choice of $\varepsilon > 0$, if $\lambda = \Delta_1(f)/\varepsilon$, then $\mathcal{M}$ is ε-differentially private and (ξ, β)-accurate with*

$$\xi = 2\,\Delta_1(f)\log(d/\beta)/\varepsilon.$$

(b) ***Private randomness complexity:*** *The mechanism has expected private entropy*

$$\mathbb{E}_U[H(\mathcal{M}_U(f(X)))] \;=\; O\left(\frac{d\,\lambda}{\xi}\,\log\frac{\xi}{\lambda}\right).$$

(c) ***Public randomness complexity:*** *If relaxed to output a special value $\perp$ with probability at most β, the mechanism can be implemented using $O(\log(d/\beta))$ bits of public randomness.*

(d) ***Distributional properties:*** *If relaxed to use an unbounded number of public random bits, the mechanism can be implemented such that it uses $O(\log(d/\beta))$ public random bits with probability $1 - \beta$ and so that*

- *it is unbiased, $\mathbb{E}[\mathcal{M}(f(X))] = f(X)$, and*
- *its coordinates are pairwise independent, $\mathrm{Cov}(\mathcal{M}(f(X))_i, \mathcal{M}(f(X))_j) = 0$ for all $i \neq j$.*

Proof. We split the proof in five parts, covering privacy, error, private randomness complexity, public randomness complexity, and distributional properties, respectively.

Privacy. Conceptually the mechanism ensures ϵ-differential privacy by adding independent Laplace noise $y \sim \mathrm{Laplace}(\Delta_1(f)/\epsilon)^d$ to the value $f(X)$. The ℓ_1 sensitivity of f is $\Delta_1(f)$, so Laplace noise with scale $\lambda = \Delta_1(f)/\epsilon$ ensures the required privacy guarantee for $f(X) + y$. The subsequent rounding step maps the noisy result to the nearest point on an ξ-spaced grid shifted by the public offsets $\xi\,\gamma_i$ with $\gamma_i = (ai + b) \bmod 1$ and $(a, b) \sim \mathrm{Uniform}([0,1)^2)$. The public randomness (a, b) is independent of the data, so post-processing preserves privacy for every fixed choice of (a, b).

Error bound. As for the error incurred by the mechanism we note that

$$\|\mathcal{M}(f(X)) - f(X)\|_\infty \leq \|y\|_\infty + \xi/2,$$

since the error in each coordinate is bounded by the sum of the perturbation given by y and the rounding to the nearest grid point which changes the value by at most $\xi/2$. By a union bound over the coordinates, the probability that the noise exceeds $\xi/2$ is:

$$\Pr[\|y\|_\infty > \xi/2] \leq d \, \exp\left(-\frac{\xi/2}{\lambda}\right),$$

which is less than β when $\xi \geq 2\lambda \log(d/\beta)$. Thus, with probability at least $1-\beta$ the total error is bounded by ξ. Substituting $\lambda = \Delta_1(f)/\epsilon$ completes the computation.

Private randomness complexity. Fix an arbitrary shift vector $\gamma \in \mathbb{R}^d$ and condition on γ throughout this part; we only use its distribution at the end. For a given input $z = f(X)$, define the (deterministic) baseline output

$$\mathcal{M}_{\gamma,0}(z) \;:=\; \xi \cdot \Big(\Big\lfloor \frac{z}{\xi} - \gamma + \tfrac{1}{2} \Big\rfloor + \gamma \Big),$$

obtained by setting the noise to zero. Let $Y = (Y_1, \ldots, Y_d) \sim \mathrm{Laplace}(\lambda)^d$. The private entropy for this input and shift is $H(\mathcal{M}_{\gamma,Y}(z))$, where γ is the public randomness. Since the coordinates of Y are independent and rounding is coordinate-wise, $\mathcal{M}_{\gamma,Y}(z)$ is a deterministic function of independent integer "step counters" $K_i \in \mathbb{Z}$ defined by $\mathcal{M}_{\gamma,Y}(z)_i \;=\; \mathcal{M}_{\gamma,0}(z)_i \;+\; K_i\,\xi$. Hence

$$H(\mathcal{M}_{\gamma,Y}(z)) \;\leq H(K_1, \ldots, K_d \mid \gamma, z) \;=\; \sum_{i=1}^{d} H(K_i \mid \gamma, z), \tag{59}$$

where the last equality uses independence of Y. For $i \in [d]$ let $\tau_i = \tau_i(\gamma, z) \in [0, \xi/2]$ be the distance from z_i to the nearest boundary of the randomly shifted grid on the ith coordinate. That is,

$$\tau_i := \min_{k \in \mathbb{Z}} \{|z_i - \xi(k + \gamma_i + 1/2)|\}.$$

For $p_{0,i}(\gamma, z) :=\; 1 - e^{-\tau_i/\lambda}$ we have the following lower bound on the probability of $K_i = 0$:

$$\Pr[\, K_i = 0 \mid \gamma, z \,] \;\geq \Pr[\, |Y_i| < \tau_i \,] \;=\; p_{0,i}(\gamma, z).$$

For $k \geq 1$, crossing k grid boundaries requires $|Y_i| \geq \tau_i + (k-1)\xi$, so the one-sided tail satisfies

$$\Pr[\, |K_i| \geq k \mid \gamma, z \,] \;\leq e^{-(\tau_i + (k-1)\xi)/\lambda} \;=\; e^{-\tau_i/\lambda}\, q^{\,k-1}, \text{ where } q := e^{-\xi/\lambda}.$$

Thus $|K_i|$ is stochastically dominated by a geometric distribution on $\{1, 2, \ldots\}$ with ratio q, and the signed tail $\{K_i = \pm k : k \geq 1\}$ by a two-sided geometric (discrete Laplace) with the same ratio. Let h_2 be the binary entropy function, and let $H_{\mathrm{geo}}(q) \;:=\; -\log_2(1-q) \;-\; \frac{q}{1-q}\,\log_2 q$ be the entropy of a geometric $(1-q)$ distribution on $\{1, 2, \ldots\}$. Encoding first the event $\{K_i = 0\}$ and then the signed magnitude gives

$$H(K_i \mid \gamma, z) \;\leq h_2\big(p_{0,i}(\gamma, z)\big) \;+\; \big(1 - p_{0,i}(\gamma, z)\big)\,\Big(\,1 \;+\; H_{\mathrm{geo}}(q)\,\Big). \tag{60}$$

Combining equation 60 over i with equation 59 yields a per-input, per-shift entropy bound.

Averaging over the shift. In our implementation the public shift is per-coordinate: $\xi\,\gamma_i$ with $\gamma_i = (ai + b) \bmod 1$ and $(a, b) \sim \mathrm{Uniform}([0,1)^2)$. For each fixed input z and coordinate i, the marginal $\gamma_i \sim \mathrm{Uniform}([0,1))$ is independent of Y_i, so the remainder $(z_i - \xi\gamma_i)/\xi \bmod 1$ is uniform on $[0,1)$. Consequently, the distance $\tau_i(\gamma, z)$ from z_i to the nearest half-integer boundary is uniform on $[0, \xi/2]$. Hence

$$\overline{p}_0 \;:=\; \mathbb{E}_\gamma[\, p_{0,i}(\gamma, z) \,] \;=\; \mathbb{E}_{T \sim \mathrm{Uniform}([0,\xi/2])}[\, 1 - e^{-T/\lambda} \,] \;=\; 1 \;-\; \frac{2\lambda}{\xi}\Big(1 - e^{-\xi/(2\lambda)}\Big),$$

independent of i and z. By concavity of h_2 and linearity of expectation,

$$\mathbb{E}_\gamma\big[\, H(K_i \mid \gamma, z) \,\big] \;\leq h_2(\overline{p}_0) \;+\; (1 - \overline{p}_0)\,\Big(\,1 \;+\; H_{\mathrm{geo}}(q)\,\Big), \qquad q = e^{-\xi/\lambda}. \tag{61}$$

Finally,

$$\mathbb{E}_U\big[H(\mathcal{M}_U(z))\big] \;\le d\,\Big[\, h_2(\overline{p}_0) \;+\; (1-\overline{p}_0)\,\big(1+H_{\mathrm{geo}}(q)\big)\,\Big], \tag{62}$$

with $\overline{p}_0 = 1 - \frac{2\lambda}{\xi}(1-e^{-\xi/(2\lambda)})$ and $q = e^{-\xi/\lambda}$. Combining the above, we obtain an explicit asymptotic bound on the expected private entropy. Under the accuracy condition $\xi \ge 2\lambda\log(d/\beta)$ established above (implying $\xi/\lambda \gg 1$ in the regimes of interest), the geometric ratio $q = e^{-\xi/\lambda}$ satisfies $q \ll 1$, and $1-\overline{p}_0 = \frac{2\lambda}{\xi}\big(1+O(e^{-\xi/(2\lambda)})\big)$. Using $h_2(1-x) \le x\log_2(\frac{e}{x})$ and $H_{\mathrm{geo}}(q) = \Theta(q\log\frac{1}{q})$ for small q, inequality equation 62 yields

$$\mathbb{E}_U[H(\mathcal{M}_U(z))] \;\le C\,d\cdot\frac{\lambda}{\xi}\,\log_2\frac{\xi}{\lambda}$$

for some absolute constant $C > 0$. This establishes the claimed bound in part (b).

Public randomness complexity (for part (c)). Fix $f(X)$ and the private noise y. The output changes only when, for some coordinate $i \in [d]$, the function

$$\gamma_i \mapsto \frac{f(X)_i + y_i}{\xi} - \gamma_i + \tfrac{1}{2} \;=\; \frac{f(X)_i + y_i}{\xi} - \big((a\cdot i + b) \bmod 1\big) + \tfrac{1}{2}$$

crosses an integer. Equivalently, $(a,b) \in [0,1)^2$ lies in a set where $(ai+b) \bmod 1$ is within distance η of a (fixed) boundary point $r_i \in [0,1)$ determined by $f(X)_i$ and y_i. For a precision parameter $\eta \in (0,1)$, let

$$\mathcal{B}_\eta \;:=\; \bigcup_{i=1}^{d} \Big\{(a,b)\in[0,1)^2 \;:\; \mathrm{dist}\,\big((ai+b) \bmod 1,\; r_i\big) \le \eta\Big\}.$$

Each set inside the union is the preimage (under the map $(a,b) \mapsto (ai+b) \bmod 1$) of an interval of length 2η in $[0,1)$, so its total area in $[0,1)^2$ is at most 2η. By the union bound, the probability mass of $\mathcal{B}_\eta$ is at most $2d\,\eta$. Now choose $\eta := \beta/(2d)$. Then $\mu(\mathcal{B}_\eta) \le \beta$. If we quantize a, b to m bits each with

$$m = \left\lceil \log_2 \frac{4d^2}{\beta} \right\rceil,$$

then the rounded pair $(\tilde{a}, \tilde{b})$ satisfies

$$\mathrm{dist}\,\big((\tilde{a}i + \tilde{b}) \bmod 1, (ai+b) \bmod 1\big) \le (i+1)2^{-m} \le \eta$$

for every $i \in [d]$. Thus, for every $(a,b) \notin \mathcal{B}_\eta$, the rounded pair $(\tilde{a}, \tilde{b})$ determines the same output as (a,b). Therefore, an implementation that outputs $\perp$ whenever $(a,b) \in \mathcal{B}_\eta$ uses at most $2m = O(\log(d/\beta))$ public bits and triggers $\perp$ with probability at most β. This proves part (c).

Public randomness complexity and distributional properties (for part (d)). If we allow unbounded public randomness, we can reveal the bits of (a,b) *incrementally* until $(a,b) \notin \mathcal{B}_\eta$ is certified; with probability at least $1-\beta$ this occurs by $m = \lceil \log_2(4d^2/\beta)\rceil$ bits per parameter, so with probability $1-\beta$ the public randomness used is $O(\log(d/\beta))$ bits (and on the rare event we reveal more bits). Since (a,b) are uniform on $[0,1)^2$, the induced shifts $\gamma_i = (ai+b) \bmod 1$ satisfy: (i) for each i, $\gamma_i \sim \mathrm{Uniform}([0,1))$; and (ii) for $i \ne j$, the pair (γ_i, γ_j) is uniform on $[0,1)^2$. This *pairwise independence* property is a variant of classical constructions of pairwise independent hash functions (Carter & Wegman, 1979), see Lemma 11 in Appendix G for a proof. Consequently, with the round-to-nearest construction, the per-coordinate quantization errors $E_i = \mathcal{M}(f(X))_i - (f(X)_i + y_i)$ are pairwise independent and uniform on $[-\xi/2, \xi/2]$, independent of $(f(X), y)$, and in particular have mean zero. This proves part (d). □

G Pairwise Independence of Affine Shifts

We now show the pairwise independence of the shift values γ in the dithered rounding scheme. The proof is analogous to Carter and Wegman's classical proof of pairwise independence of random linear functions over finite fields (Carter & Wegman, 1979); we include it for completeness.

Lemma 11. *Let $a, b \sim \mathrm{Uniform}([0,1))$ be independent, and for each integer t define $\gamma_t := (at + b) \bmod 1 \in [0,1)$. Then for every t we have $\gamma_t \sim \mathrm{Uniform}([0,1))$, and for distinct integers $t \neq s$ the pair (γ_t, γ_s) is uniform on $[0,1)^2$. In particular, γ_t and γ_s are (pairwise) independent.*

Proof. We work on the unit interval $[0,1)$ with addition taken modulo 1. Write $\{x\}$ for the fractional part of a real x. Throughout, a, b are independent and uniform on $[0,1)$. *Uniform marginal.* Fix an integer t. For any measurable set $I \subseteq [0,1)$ and any fixed $a \in [0,1)$,

$$\Pr_b\big[\, \gamma_t \in I \;\big|\; a \,\big] \;=\; \Pr_b\big[\, \{ta + b\} \in I \,\big] \;=\; \Pr_b\big[\, b \in I - \{ta\} \,\big] \;=\; |I|,$$

since $b \mapsto \{b + c\}$ is a shift on the circle and preserves the total length of sets in $[0,1)$. Averaging over a gives $\Pr[\gamma_t \in I] = |I|$, i.e., $\gamma_t \sim \mathrm{Unif}[0,1)$. *Pairwise independence.* Let $t \neq s$ be integers, and let $I, J \subseteq [0,1)$ be measurable. We show

$$\Pr\big[\, \gamma_t \in I,\ \gamma_s \in J \,\big] \;=\; |I|\,|J|.$$

Condition on a. For fixed a, using b uniform and the shift argument as above,

$$\begin{aligned}\Pr_b\big[\, \gamma_t \in I,\ \gamma_s \in J \;\big|\; a \,\big] &= \Pr_b\big[\, \mathrm{frac}(b + ta) \in I \quad \text{and} \quad \mathrm{frac}(b + sa) \in J \,\big] \\ &= \big|\, (I - \mathrm{frac}(ta)) \cap (J - \mathrm{frac}(sa)) \,\big|.\end{aligned}$$

Set $\delta := \mathrm{frac}((t-s)a) \in [0,1)$ and note that $(I - \mathrm{frac}(ta)) \cap (J - \mathrm{frac}(sa)) = J \cap (I - \delta)$ up to a common shift, hence the length above equals $\phi(\delta) := |\, J \cap (I - \delta)\,|$. Therefore

$$\Pr\big[\, \gamma_t \in I,\ \gamma_s \in J \,\big] \;=\; \mathbb{E}_a\big[\, \phi(\{(t-s)a\}) \,\big].$$

We claim that $\{(t-s)a\}$ is uniform on $[0,1)$ when a is uniform (for any nonzero integer $t - s$). Indeed, partition $[0,1)$ into $|t-s|$ intervals of length $1/|t-s|$; multiplication by $|t-s|$ maps each interval linearly onto $[0,1)$, so for any measurable $B \subseteq [0,1)$ the preimage $\{a : \{(t-s)a\} \in B\}$ has size $|B|$. Hence $\{(t-s)a\} \sim \mathrm{Unif}[0,1)$. It remains to compute the average of $\phi(\delta)$ over uniform δ. By averaging over all possible shifts and using that translating a set within [0,1) does not change its total length,

$$\int_0^1 \phi(\delta)\, d\delta \;=\; \int_0^1 \Big(\int_0^1 \mathbf{1}_J(x)\, \mathbf{1}_I(x+\delta)\, dx \Big)\, d\delta \;=\; \int_0^1 \mathbf{1}_J(x) \Big(\int_0^1 \mathbf{1}_I(x+\delta)\, d\delta \Big)\, dx \;=\; |J| \cdot |I|.$$

Therefore $\mathbb{E}_a[\phi(\{(t-s)a\})] = |I|\,|J|$, and the desired equality follows. Since this holds for all measurable I, J, the pair (γ_t, γ_s) is uniform on $[0,1)^2$, and in particular γ_t and γ_s are independent. □