# Private Lossless Multiple Release

**Joel Daniel Andersson** [1] [2]   **Lukas Retschmeier** [1] [2]   **Boel Nelson** [2]   **Rasmus Pagh** [1] [2]

## Abstract

Koufogiannis et al. (2016) showed a *gradual release* result for Laplace noise-based differentially private mechanisms: given an $\varepsilon$-DP release, a new release with privacy parameter $\varepsilon' > \varepsilon$ can be computed such that the combined privacy loss of both releases is at most $\varepsilon'$ and the distribution of the latter is the same as a single release with parameter $\varepsilon'$. They also showed gradual release techniques for Gaussian noise, later also explored by Whitehouse et al. (2022). In this paper, we consider a more general *multiple release* setting in which analysts hold private releases with different privacy parameters corresponding to different access/trust levels. These releases are determined one by one, with privacy parameters in arbitrary order. A multiple release is *lossless* if having access to a subset $S$ of the releases has the same privacy guarantee as the least private release in $S$, and each release has the same distribution as a single release with the same privacy parameter. Our main result is that lossless multiple release is possible for a large class of additive noise mechanisms. For the Gaussian mechanism we give a simple method for lossless multiple release with a short, self-contained analysis that does not require knowledge of the mathematics of Brownian motion. We also present lossless multiple release for the Laplace and Poisson mechanisms. Finally, we consider how to efficiently do gradual release of sparse histograms, and present a mechanism with running time independent of the number of dimensions.

## 1. Introduction

*Differential privacy* (Dwork et al., 2006) is a statistical notion that provides provable privacy guarantees. Differentially private (DP) algorithms typically introduce inaccuracy through noise to achieve privacy, and the resulting privacy-accuracy trade-off is the key object of study in the area. Of specific interest is the *privacy budget* that determines how much information the output of a differentially private algorithm may reveal about its inputs—a smaller budget means more private but less accurate results.

**Motivation.** In deployments of differential privacy, it may be hard to determine the appropriate privacy budget to grant an analyst since it depends on trust and accuracy assumptions that may change over time. A system may also have different security clearance levels—a data analyst might have a higher clearance level than a developer, but both might require access to *some* statistics. Similarly, a company that wants to release, for example, user statistics could make an accurate release for their own data analysts, a less accurate release for external consultants, and include an even less accurate release in a report for shareholders or other external actors. A different example setting is users that want to sell their data on data markets: users could let the accuracy of the release depend on how much they are paid, and use different budgets for different releases.

Another usage scenario, relevant in distributed or federated settings, is *local differential privacy* where the data owner could be sharing a differentially private function of their data with multiple servers, and the set of servers might change over time. Here the privacy budget might depend on the server: for example, a patient may trust their local hospital more than their national hospitals, but might not trust the hospitals not to collude by sharing data among themselves.

**Multiple releases.** These scenarios motivate creating *multiple releases* with different privacy budgets aimed at different analysts. However, these releases should be *coordinated* such that a group of analysts who combine their information do not gain more knowledge about the input than the most knowledgeable member of the group. This kind of collusion resilience was first studied by Xiao et al. (2009) with a non-DP privacy objective—we refer to their work for additional motivation.

A related aspect is that we may want to provide an analyst with a less private, more accurate release after the trust we place in them increases. In this case, we want the accuracy of the latest release to match the accuracy that can be obtained, given the combined privacy budget of both releases. That is, no additional cost should be incurred for

---

[1]Basic Algorithms Research Copenhagen (BARC), Denmark [2]University of Copenhagen, Denmark. Correspondence to: Joel Daniel Andersson <jda@di.ku.dk>, Lukas Retschmeier <lure@di.ku.dk>, Boel Nelson <bn@di.ku.dk>, Rasmus Pagh <pagh@di.ku.dk>.

making two releases rather than one. For example, an external consultant that later gets employed directly by the company should be able to get access to the more accurate release without constituting a privacy violation or requiring an increased privacy budget to reach the same accuracy.

**Baseline.** It is possible to create multiple releases for *any* differentially private mechanism if we are willing to increase the privacy budget by a constant factor. In particular, we can create a sequence of independent releases with geometrically increasing privacy parameters, referred to as "$\varepsilon$-doubling" by Ligett et al. (2017), and provide each analyst with the most accurate release they are entitled to. Using composition results, the combined privacy budget for the information given to a set of analysts is a constant factor from the highest budget of a single analyst in the group. In this paper, we study how to make multiple releases in a *lossless* way without accuracy or privacy penalty.

## 1.1. Related Work

Koufogiannis, Han, and Pappas (2016) introduced the concept of (lossless) *gradual release*, which makes it possible to increase the privacy budget with *no loss* in accuracy. They also considered privacy tightening for the Laplace mechanism, where successive releases are increasingly private. In the journal version of the paper, they also introduce gradual release for the Gaussian mechanism under approximate DP which is based on the machinery of Brownian motion.

This technique was later applied in a noise reduction framework by Ligett, Neel, Roth, Waggoner, and Wu (2017) with the goal of producing mechanisms with ex-post privacy, i.e., where the privacy budget is set based on what is required to achieve a desired accuracy level. Follow-up work by Whitehouse, Ramdas, Wu, and Rogers (2022) gave results for noise reduction under approximate DP using Brownian motion. These works were motivated by work on privacy filters and odometers (Rogers et al., 2016), keeping track of privacy budgets over time, rather than multiple release settings. More recently, Pan (2024) demonstrated lossless gradual release for randomized response.

Girgis et al. (2020) studied privacy-utility-randomness trade-offs for a multi-analyst setting under local differential privacy. Here each user has a sample $X_i \sim D$ and makes a single public release of $X_i$. Each analyst then utilizes the public release (together with shared randomness among the users) to estimate $D$ at a more accurate level than what the public release alone would allow. Later Zhang and He (2023) also used lossless gradual release techniques for the Gaussian mechanism in designing a database system supporting analysts with different privilege levels.

Similar research questions have been investigated outside of the DP literature. Xiao, Tao, and Chen (2009) studied
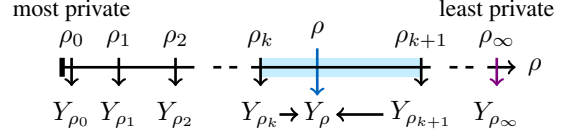


*Figure 1.* The idea behind *lossless multiple release*. For concreteness we consider additive noise mechanism and zero-concentrated differential privacy. Each $Y_{\rho_i}$ denotes a noisy estimate, and to release a new estimate with $\rho > 0$, we can combine the adjacent estimates for $\rho_k$ and $\rho_{k+1}$ together with some fresh noise to obtain a new release $Y_\rho$ that is exactly $\rho$-zCDP. Note that these estimates do not need to be strictly increasing (or decreasing) but can be released in *any order*. Furthermore, releasing any subset of these estimates is exactly $\max(S)$-zCDP, where $S$ is the set of privacy parameters.

releasing a sensitive dataset where each element is kept with probability $p$, and otherwise sampled uniformly from the universe. Li, Chen, Li, and Zhang (2012) considered privatizing data by additive Gaussian noise of scale $\sigma$. Both works dealt with arbitrary sequences of parameters (probabilities $p$ or noise scalings $\sigma$), and demonstrated how to correlate releases to guarantee that (1) each release matches the single-release case and, (2) limiting the sensitive information derived from combining releases. Our method for adaptively producing Gaussian releases deviates from (Li et al., 2012), in that ours does not need to maintain any covariance matrix for past releases. This makes our approach more time- and space-efficient.

Finally, we mention the classic work of Equitz and Cover (1991) who studied *successive refinement of information* where a source is encoded once and then refined in later stages without decreasing the rate compared to a one-shot optimal code. They showed that this is possible precisely when the sequence of reconstructions can be written as a Markov chain. While their work did not focus on privacy, successive refinement is conceptually related to lossless gradual release. Indeed, parts of the mathematical machinery was independently re-discovered by subsequent work on lossless gradual release.

## 1.2. Basic Technique

We illustrate our framework in the setting of additive Gaussian noise and $\rho$-zero-concentrated differential privacy ($\rho$-zCDP)[1]. We want to release a private estimate of a real-valued query $f : \mathcal{X} \to \mathbb{R}$ with $\ell_2$-sensitivity $\Delta_2 = 1$ using the Gaussian mechanism. Now consider the simple case where one wants to privately release an estimate with two privacy levels $\rho < \rho'$. Then releasing $Y_\rho = f(\mathbf{x}) + \mathrm{N}(0, \frac{1}{2\rho})$ together with $Y_{\rho'-\rho} = f(\mathbf{x}) + \mathrm{N}(0, \frac{1}{2(\rho'-\rho)})$ is $\rho'$-zCDP by compo-

---

[1]see definitions in Appendix D

sition. Now observe that we can combine these estimates to produce a better estimate using inverse variance weighting: $Y = \frac{\rho}{\rho + (\rho' - \rho)} Y_\rho + \frac{\rho' - \rho}{\rho + (\rho' - \rho)} Y_{\rho' - \rho}$ yields *exactly* the same utility as a single release under $\rho'$-zCDP. That is, instead of using a privacy budget of $\rho + \rho'$ for independently releasing both estimates, the overall budget spent is just the maximum of both, which is $\rho'$.

To do multiple lossless releases for more general additive noise mechanisms, it turns out that one can always combine existing releases with fresh noise as illustrated in Figure 1. In fact, to make any new release with privacy parameter $\rho$, it suffices to have saved the two "adjacent" releases whose privacy parameters are closest to $\rho$. Our approach applies more generally to a class of (independent) additive noise mechanisms that support gradual release, but the releases can be made in any order, and we do not require the set of releases to be known in advance.

**Our contributions.** We introduce a framework for *lossless multiple release*, generalizing past work on gradual release. In addition to allowing for making multiple private releases and having the privacy loss only scale with the least private release, we impose no specific ordering on the privacy parameters used. Furthermore, we formalize a general theorem (Section 4.2) showing that lossless multiple release is possible for a large class of mechanisms based on adding i.i.d. noise to each coordinate whose distribution satisfies a *convolution preorder*, as defined next.

**Definition 1.1** (Convolution preorder). A family of real-valued distributions $\mathcal{D}(\rho)$ parameterized by $\rho \in \mathbb{R}_+$ is said to satisfy a *convolution preorder*, if given $\mathcal{D}(\rho_1)$ and $\mathcal{D}(\rho_2)$ for $\rho_1 < \rho_2$, there exists a distribution $\mathcal{C}(\rho_2, \rho_1)$ such that $\mathcal{D}(\rho_2) * \mathcal{C}(\rho_2, \rho_1) = \mathcal{D}(\rho_1)$.

This relation is a natural way of stating that the distributions become more noisy as the parameter $\rho$ gets smaller. We are unaware of any existing term in the literature but the definition is closely related to *convolution order* (Shaked & Shanthikumar, 2007). Examples of noise distributions satisfying Definition 1.1 include the Laplace, Poisson, and the Gaussian mechanisms, parameterized by a decreasing function of their variance.

**Theorem 1.2** (Meta theorem, informal). *Let $\mathcal{A}_{f,\rho} : \mathcal{X} \to \mathbb{R}^d$ be a mechanism that adds independent, identically distributed noise to coordinates of a function $f : \mathcal{X} \to \mathbb{R}^d$. If the noise distribution of $\mathcal{A}_{f,\rho}$ satisfies* convolution preorder*, then there exists an algorithm enabling* lossless multiple release*. Also, any invertible post-processing of $\mathcal{A}_{f,\rho}$ preserves this property.*

For the Gaussian mechanism in particular, we show how to get lossless multiple release by only using basic properties of Gaussians, and we show similar results for the Laplace and Poisson mechanisms from first principles. We give concrete

instantiations of *Gaussian sparse histograms* (Section 5.2) and *factorization mechanisms* (Section 5.1).

## 2. Threat Model and Goals

Our setting is a multi-user system with a set of analysts/users $S$, all able to query the same dataset using a differentially private mechanism. User $s$ has their own security clearance level with corresponding privacy budget $\rho_s$. This setting allows for multi-level security where each user belongs to a security clearance level, like in the classic Bell-LaPadula model (Bell & La Padula, 1976), where users with high clearance levels have higher values of $\rho$. A common example of such clearance levels is using increasing levels with labels such as *public*, *restricted*, *confidential*, and *top secret*. The user with the highest clearance in the system has a privacy budget of $\max_{s \in S}(\rho_s)$, which we denote $\rho_{\max}$.

We consider an adversary who gains partial knowledge, that is, an adversary that sees *some* of the releases. Our goal is to design a differentially private mechanism such that an adversary with access to releases from some users $S' \subseteq S$, can learn *at most* what they could have learned from a release with privacy parameter $\max_{s' \in S'}(\rho_{s'})$. This means that even by compromising or colluding with more users, the adversary's knowledge may not increase. In case an adversary observes all releases (e.g., by compromising all users), the privacy loss would be bounded by $\rho_{\max}$.

However, our goal is to design a mechanism where the releases are *lossless* in the sense that the noise distribution from multiple releases with a combined budget $\rho_{\max}$ would be indistinguishable from one single release with the privacy budget $\rho_{\max}$. In other words, the privacy loss should be determined by $\rho_{\max}$, while independent releases would usually have privacy parameter $\sum_{s \in S} \rho_s$ due to composition.

## 3. Gaussian Lossless Multiple Release

Extending the work in (Koufogiannis et al., 2016; Li et al., 2012), we demonstrate next that in the case of the Gaussian mechanism, providing lossless multiple release is clean and follows immediately from simple properties of Gaussians. Throughout the paper, we use the symbol $Y$ to be a random variable that depends on the private dataset and $Z$ to be one that does not.

We first consider the gradual release setting where an increasingly accurate estimate is released, and the overall privacy loss is determined solely by the latest, least private release. In Lemma 3.3, we drop this restriction, allowing releases in any order while still guaranteeing that the overall privacy guarantee is the maximum $\rho$ value provided. For simplicity, we consider the one-dimensional case, where we

want to release some query $f : \mathcal{X} \to \mathbb{R}$. The $d$-dimensional setting is handled by sampling each coordinate independently.

**Getting started.** Foreshadowing the usage of $\rho$-zCDP, we initially use $1/(2\rho)$ for denoting the variance of a Gaussian. Our inquiry starts with a basic observation about inverse-variance weighting of Gaussians.

**Lemma 3.1.** *Let $Y_\rho \sim \mathrm{N}(\beta, \frac{1}{2\rho})$ and $Y_{\rho'} \sim \mathrm{N}(\beta, \frac{1}{2\rho'})$ where $\beta \in \mathbb{R}$ and $\rho, \rho' > 0$. Then*

$$Y = \tfrac{\rho}{\rho+\rho'}Y_\rho + \tfrac{\rho'}{\rho+\rho'}Y_{\rho'} \sim \mathrm{N}\left(\beta, \tfrac{1}{2(\rho+\rho')}\right).$$

*Proof.* The mean of $Y$ is immediate by the fact that $Y$ is a weighted-average of $Y_\rho$ and $Y_{\rho'}$, both of mean $\beta$. For the variance, direct computation yields:

$$\mathrm{Var}[Y] = \frac{\rho^2/(2\rho) + \rho'^2/(2\rho')}{(\rho_1 + \rho')^2} = \frac{1}{2(\rho + \rho')}.$$

As the sum of two Gaussians is itself Gaussian, we are done. $\square$

The essence of what is being claimed is that a Gaussian of variance $\frac{1}{2\rho}$ and another Gaussian of variance $\frac{1}{2\rho'}$ with the same mean can be combined into a new Gaussian with variance $\frac{1}{2(\rho+\rho')}$ and the same mean. Inspired by this, we can repeatedly invoke the lemma for the following result.

**Lemma 3.2.** *Given $0 < \rho_1 < \cdots < \rho_m$ and $\beta \in \mathbb{R}$ define $Y_{\rho_1}, \ldots, Y_{\rho_m} \in \mathbb{R}$ where $Y_{\rho_1} \sim \mathrm{N}(\beta, \frac{1}{\rho_1})$, and for $k > 1$:*

$$Y_{\rho_{k+1}} = \tfrac{\rho_k}{\rho_{k+1}}Y_{\rho_k} + \tfrac{\rho_{k+1}-\rho_k}{\rho_{k+1}} \cdot \mathrm{N}\left(\beta, \frac{1}{2(\rho_{k+1}-\rho_k)}\right).$$

*Then for any $i \in [m] : Y_{\rho_i} \sim \mathrm{N}(\beta, \frac{1}{2\rho_i})$ and for any $j \in [m] : \mathrm{Cov}(Y_{\rho_i}, Y_{\rho_j}) = \frac{1}{2\max(\rho_i, \rho_j)}$.*

*Proof.* We first show the distribution of $Y_{\rho_k}$ by induction on $k$. Assume that $Y_{\rho_k} \sim \mathrm{N}(\beta, \frac{1}{\rho_k})$, which is true for the base case of $k = 1$. Note that the inductive step follows immediately from invoking Lemma 3.1. For the covariance, note that we can expand the expressions inside the covariance and "throw away" the independent noise added in each recurrence. Assuming $i \le j$ we get $\mathrm{Cov}(Y_{\rho_i}, Y_{\rho_j}) = \mathrm{Cov}(Y_{\rho_i}, \frac{\rho_i}{\rho_j}Y_{\rho_i}) = \frac{\rho_i}{\rho_j}\mathrm{Var}[Y_{\rho_i}] = \frac{1}{2\rho_j}$, implying the stated covariance. $\square$

**Releases in arbitrary order.** The Gaussian sequence in Lemma 3.2 will be the basis for our lossless multiple release version of the Gaussian mechanism. What the lemma does not address is generating the sequence in arbitrary order: Statically, given the full sequence $(\rho_k)_{k\in[m]}$, we can generate the Gaussians, but what if we receive them one-by-one and in arbitrary order? We will address this next.

**Lemma 3.3.** *For $\rho_\infty \in \mathbb{R}_+$ (possibly $\rho_\infty = \infty$) and $\beta \in \mathbb{R}$, let $M = \{(0, \infty), (\rho_\infty, Y_{\rho_\infty})\}$ where $Y_{\rho_\infty} \in \mathrm{N}(\beta, \frac{1}{2\rho_\infty})$. Consider a finite subset $S \subset (0, \rho_\infty)$ and the following process that runs for $|S|$ iterations:*

1. *Pick an arbitrary $\rho \in S$ and delete it from $S$;*

2. *Let $(\rho_l, Y_{\rho_l}), (\rho_r, Y_{\rho_r}) \in M$ where $\rho \in (\rho_l, \rho_r)$ and $\rho_r - \rho_l$ is minimal;*

3. *Sample $Z \sim \mathrm{N}\left(0, \frac{(1-\rho_l/\rho)(1/\rho-1/\rho_r)}{2(1-\rho_l/\rho_r)}\right)$, let*

$$Y_\rho = \tfrac{1-\rho_l/\rho}{1-\rho_l/\rho_r}Y_{\rho_r} + \tfrac{\rho_l/\rho-\rho_l/\rho_r}{1-\rho_l/\rho_r}Y_{\rho_l} + Z,$$

   *and add $(\rho, Y_\rho)$ to $M$.*

*Then the sequence of random variables generated by the process has the same distribution as described in Lemma 3.2.*

*Proof sketch.* The argument is inductive. Under the hypothesis that all values generated up to the given point have the distribution described by Lemma 3.2, we argue that the newly generated value does too. The argument considers the four different cases for $\rho_l, \rho_r$, e.g., $\rho_l = 0, \rho_r \ne \rho_\infty$. As the proof involves tedious computation we refer the reader to Appendix B for the formal proof. $\square$

**Formalizing lossless multiple release.** Lemma 3.3 will constitute the basis for our implementation of lossless multiple release, but we have yet to formally define this notion. We do so next.

**Definition 3.4** (Lossless multiple release)**.** Let $\mathcal{M}_\rho : \mathcal{X} \to \mathcal{Y}$ be a family of mechanisms on a domain $\mathcal{X}$, indexed by a privacy parameter $\rho \in \mathbb{R}_+$. We say that $\mathrm{M} : \mathcal{X} \times \mathbb{R}_+ \to \mathcal{Y}$ implements $\mathcal{M}_\rho$ with *lossless multiple release* if for every $x \in \mathcal{X}$ it satisfies:

1. $\forall \rho$: $\mathrm{M}(x, \rho)$ and $\mathcal{M}_\rho(x)$ are identically distributed.

2. For every finite subset $S \subset \mathbb{R}_+$, processed in arbitrary order by M, and $y \in \mathcal{Y}$, conditioned on $\mathrm{M}(\mathbf{x}, \max(S)) = y$ the joint distribution of $(\mathrm{M}(\mathbf{x}, \rho))_{\rho \in S}$ is uniquely determined by $y$ and $S$.

Functionally, a mechanism meets the definition if its outputs can be correlated such that for any set of outputs, their joint distribution can be viewed as (randomized) post-processing of the least private release. An implementation necessarily has to store information about releases that have been made, i.e., the sequence of inputs to M and the corresponding outputs, to fulfill the requirement that releases for different privacy parameters are correlated. When M only supports outputting releases for a sequence of *increasing* privacy parameters, we call it *lossless gradual release*.

**Algorithm 1** GaussianMultipleRelease
**Parameters:** $\ell_2$ sensitivity $\Delta_2$
**Inputs:** Set of releases $M$, privacy parameter $\rho$
1:   Find $(\rho_k, Y_{\rho_k}), (\rho_{k+1}, Y_{\rho_{k+1}}) \in M$ such that
2:     $\rho \in [\rho_k, \rho_{k+1}]$ and $\forall (\rho', \cdot) \in M : \rho' \notin (\rho_k, \rho_{k+1})$
3:   Sample $Z_\rho \sim \mathrm{N}(0, \Delta_2^2 \cdot \frac{(1 - \rho_k/\rho) \cdot (1/\rho - 1/\rho_{k+1})}{2(1 - \rho_k/\rho_{k+1})})$
4:   Let $Y_\rho := Z_\rho + \frac{(1 - \rho_k/\rho) Y_{\rho_{k+1}} + (\rho_k/\rho - \rho_k/\rho_{k+1}) Y_{\rho_k}}{1 - \rho_k/\rho_{k+1}}$
5:   Add $M = M \cup \{(\rho, Y_\rho)\}$
6:   **Return** $Y_\rho$

Having stated the definition for lossless multiple release, consider Algorithm 1. It implements Lemma 3.3, and the idea is visualized in Figure 1.

**Corollary 3.5.** *With $M$ initialized as $M = \{(0, \infty), (\infty, f(\mathbf{x}))\}$, Algorithm 1 implements the Gaussian mechanism with lossless multiple release.*

*Proof.* Let $\{Y_\rho\}_{\rho \in S}$ be the set of outputs produced by Algorithm 1 on receiving the set $S$ of privacy parameters in arbitrary order. Observe that the algorithm is implementing the (adaptive) sampling in Lemma 3.3, and so it produces outputs with the same distribution as Lemma 3.2. Property 1 of Definition 3.4 follows immediately from observing that $Y_\rho \sim \mathrm{N}(f(x), \frac{1}{2\rho})$. For property 2, note that every release in Lemma 3.2 can be viewed as randomized post-processing of the least private release. One way to see this is to note that for any two releases $Y_\rho$ and $Y_{\rho'}$ where $\rho < \rho'$, we have that $\mathrm{Cov}(Y_\rho, Y_{\rho'}) = \mathrm{Var}[Y_{\rho'}]$, implying that $Y_\rho = Y_{\rho'} + Z$ for aptly scaled zero-mean Gaussian noise $Z$. $\square$

# 4. Extending to Independent Additive Noise

We will next show that lossless multiple release holds for a larger class of mechanisms. To proceed we introduce the notion of an *independent additive noise mechanism*.

**Definition 4.1** (Independent Additive Noise Mechanism). We define an *independent additive noise mechanism* $\mathcal{A}_{f,\rho} : \mathcal{X} \to \mathbb{R}^d$ as a mechanism of the form

$$\mathcal{A}_{f,\rho}(\mathbf{x}) = f(\mathbf{x}) + Z, \quad Z \sim \mathcal{D}(\rho),$$

where $\mathcal{D}(\rho)$ is a probability distribution parameterized in $\rho$, that draws a $d$-dimensional vector with i.i.d. samples.

It turns out that *any* independent additive noise mechanism $\mathcal{A}_{f,\rho}$ satisfying Definition 1.1 supports lossless multiple release.

**Lemma 4.2.** *Any independent additive noise mechanism $\mathcal{A}_{f,\rho}$ with noise distribution $\mathcal{D}(\rho)$ satisfying convolution preorder can be implemented with lossless multiple release.*

*Proof.* The proof is for the one-dimensional case, but the same argument can be invoked for the multidimensional

case as each coordinate has independent noise. We begin by proving that given $S = \{\rho_k : k \in [m]\} \subset \mathbb{R}^+$, where $\rho_1 < \cdots < \rho_m$, it is possible to construct a set of releases $\{Y_{\rho_k} : k \in [m]\}$ that satisfy Definition 3.4. By Definition 1.1, it is possible to express each $Y_{\rho_k}$ as

$$Y_{\rho_k} = f(\mathbf{x}) + Z_{\rho_m} + \sum_{j=k}^{m-1} W_{\rho_{j+1}, \rho_j}, \qquad (1)$$

where $Z_{\rho_m} \sim \mathcal{D}(\rho_m)$ and $W_{\rho_{j+1}, \rho_j} \sim \mathcal{C}(\rho_{j+1}, \rho_j)$ are sampled independently. To prove that $Y_{\rho_k} \overset{d}{=} \mathcal{A}_{f, \rho_k}(\mathbf{x})$, observe that $Y_{\rho_k}$ is distributed as $f(\mathbf{x})$ plus a random variable drawn from $\mathcal{D}(\rho_m) * \mathcal{C}(\rho_m, \rho_{m-1}) * \cdots * \mathcal{C}(\rho_{k+1}, \rho_k) = \mathcal{D}(\rho_k)$, as needed. For the second property in Definition 3.4, observe that conditioning on $Y_{\rho_m} = y$ we can express every other $Y_{\rho_k}$ for $k \in [m-1]$ as

$$Y_{\rho_k} = y + \sum_{j=k}^{m-1} W_{\rho_{j+1}, \rho_j}.$$

As a result, the joint distribution on $(Y_{\rho_1}, \ldots, Y_{\rho_m})_{|Y_{\rho_m} = y}$ is uniquely determined by $y$, proving the second property.

We will now argue (inductively) that we can produce these releases adaptively. Let $S_1 \subset S_2 \cdots \subset S_m = S$ be a sequence of subsets of $S$ where $|S_i| = i$. For the base case of $S_1$ we can make a single release using $\mathcal{A}_{f, \rho}$. For our inductive hypothesis, assume we have produced releases corresponding to all the privacy parameters in $S_n$. Now, for $S_{n+1}$, we re-label the privacy parameters such that $S_{n+1} = \{\rho_k\}_{k \in [n+1]}$ where $\rho_1 < \cdots < \rho_{n+1}$. For the unique $\rho_i \in S_{n+1} \setminus S_n$, we can conditionally sample it from the joint distribution over $(Y_{\rho_k})_{k \in [n+1]}$, conditioned on the value of each release made in the previous round. By induction, it follows that we can *adaptively* release $S$. $\square$

### 4.1. Sampling in Concrete Settings

Lemma 4.2 proves the existence of a sampling procedure for lossless multiple release. In Section 3 we showed a sampling procedure in the particular case of Gaussian noise. In Appendix A we show that the structure of Algorithm 1 holds for general independent additive noise mechanisms. Namely, if the privacy parameters we support come from a bounded range $(\rho_0, \rho_\infty) \subset \mathbb{R}_+$, then the corresponding algorithm has a similar structure (see Algorithm 5).

More precisely, consider two neighboring releases $Y_{\rho_k}$ and $Y_{\rho_{k+1}}$ with noise parameters $\rho_k$ and $\rho_{k+1}$, respectively, and a new lossless release $Y_\rho$ with parameter $\rho \in [\rho_k, \rho_{k+1}]$. We can use (1) on this set of $m + 1$ releases and condition on the values of the $m$ previous releases $Y_{\rho_1}, \ldots, Y_{\rho_m}$. Next, write $W_{\rho_{k+1}, \rho_k} = W_1 + W_2$ where $W_1 = Y_\rho - Y_{\rho_{k+1}}$ and $W_2 = Y_{\rho_k} - Y_\rho$. In Appendix A, we show that sampling

$Y_\rho$ can be reduced to the following task:

Sample $W_1$ conditioned on $W_1 + W_2 = Y_{\rho_k} - Y_{\rho_{k+1}}$. (2)

**Example: Poisson mechanism.** Consider the independent additive noise mechanism using the Poisson distribution, $\mathrm{Poi}(\lambda)$. This mechanism has the property that noise is always a non-negative integer, making it a natural noise distribution for integer vectors in settings where negative noise is undesirable. Appendix E states some basic privacy properties of the Poisson mechanism.

The family of Poisson distributions parameterized by $\rho = 1/\lambda$ satisfies convolution preorder since for any $\lambda_1 > \lambda_2$, $\mathrm{Poi}(\lambda_1) = \mathrm{Poi}(\lambda_1 - \lambda_2) * \mathrm{Poi}(\lambda_2)$. To adaptively perform private lossless multiple release of a value $f(\mathbf{x})$ using the Poisson mechanism and parameters $\lambda_1 > \cdots > \lambda_m$ we notice that the "bridging" distribution in the $k^{\text{th}}$ term of the sum in (1) has distribution $\mathrm{Poi}(\lambda_k - \lambda_{k+1})$. Note that without conditioning on $Y_{\rho_1}, \ldots, Y_{\rho_m}$, $W_1 \sim \mathrm{Poi}(\lambda - \lambda_{k+1})$ and $W_2 \sim \mathrm{Poi}(\lambda_k - \lambda)$. By Lemma E.4 in the appendix we have that the sampling in (2) reduces to $W_1 \sim \mathrm{Binomial}(N, p)$ for $N = Y_{\rho_k} - Y_{\rho_{k+1}}$ and $p = (\lambda - \lambda_{k+1})/(\lambda_k - \lambda_{k+1})$.

**Example: Laplace Mechanism.** Koufogiannis et al. (2016) have already shown that the Laplace distribution parameterized by $\rho = 1/b$ satisfies convolution preorder for any scale parameters $b_1 > b_2$. Let $\mathrm{LapBridge}(b_2, b_1)$ be the probability distribution that draws $0$ with probability $b_2^2/b_1^2$ and from $\mathrm{Lap}(0, b_1)$ with the remaining probability. Then $\mathrm{Lap}(0, b_2) * \mathrm{LapBridge}(b_2, b_1) = \mathrm{Lap}(0, b_1)$ exactly. To implement lossless release via the sampling in (2), it turns out that the conditional distribution is a mixture $W_1 \sim \mathrm{LapBridge}(b, b_2)$ of three distributions: With some probability $W_1$ is equal to either $0$ or $Y_{\rho_k} - Y_{\rho_{k+1}}$, and otherwise it is sampled from the convolution of two Laplace distributions. We also show that the related exponential distribution satisfies convolution preorder, see Appendix F for the full details.

## 4.2. Lossless Multiple Release as a Blackbox

Inspired by Corollary 15 in Koufogiannis et al. (2016), we provide a meta theorem for lossless multiple release based on independent additive noise mechanisms. The central component is the following lemma, showing that the class of lossless multiple release mechanisms is closed under invertible post-processing.

**Lemma 4.3.** *Let $\mathcal{M}_\rho : \mathcal{X} \to \mathcal{Y}$ satisfy lossless multiple release, and let $H : \mathcal{Y} \to \mathcal{Y}'$ be an invertible function. Then $\mathcal{M}'_\rho = H \circ \mathcal{M}_\rho$ also satisfies lossless multiple release.*

*Proof.* Let $\mathrm{M} : \mathcal{X} \times \mathbb{R}_+ \to \mathcal{Y}$ be an implementation of $\mathcal{M}_\rho$ satisfying lossless multiple release, and let $\mathrm{M}' = H \circ \mathrm{M}$, which we will argue implements lossless multiple release. The only property that does not

trivially hold for $\mathrm{M}'$ is the second property of Definition 3.4. For a set $S \subset \mathbb{R}_+$, note that to condition on $\mathrm{M}'(\mathbf{x}, \max(S)) = y$ is equivalent to conditioning on $H^{-1}(\mathrm{M}'(\mathbf{x}, \max(S))) = \mathrm{M}(\mathbf{x}, \max(S)) = H^{-1}(y) \in \mathcal{Y}$. We thus have that conditioning $\mathrm{M}'(\mathbf{x}, \max(S)) = y$ implies that the joint distribution over $(\mathrm{M}(\mathbf{x}, \rho))_{\rho \in S}$ is fully determined, and consequently so is $(H(\mathrm{M}(\mathbf{x}, \rho)))_{\rho \in S} = (\mathrm{M}'(\mathbf{x}, \rho))_{\rho \in S}$, and we are done. $\square$

**Theorem 4.4** (Meta theorem). *Let $\mathcal{M}_\rho : \mathcal{X} \to \mathcal{Y}$ be a family of mechanisms on a domain $\mathcal{X}$, indexed by a privacy parameter $\rho \in \mathbb{R}_+$. Furthermore, assume $\mathcal{M}_\rho$ can be decomposed as $\mathcal{M}_\rho = H \circ \mathcal{A}_{f,\rho}$ where*

- *$\mathcal{A}_{f,\rho} : \mathcal{X} \to \mathbb{R}^d$ is an independent additive noise mechanism for releasing $f : \mathcal{X} \to \mathbb{R}^d$ with privacy parameter $\rho$ and with noise distribution satisfying a convolution preorder (Definition 1.1);*

- *$H : \mathbb{R}^d \to \mathcal{Y}$ is an invertible post-processing step;*

*Then $\mathcal{M}_\rho$ can be implemented with lossless multiple release.*

*Proof.* The theorem follows from invoking Lemma 4.2 together with Lemma 4.3. $\square$

**Supporting non-invertible post-processing.** To support non-invertible post-processing, we also introduce a weaker notion: *weakly lossless multiple release*.

**Definition 4.5** (Weakly Lossless Multiple Release). A mechanism $\mathcal{M}_\rho$ supports *weakly lossless multiple release* if it can be written as $\mathcal{M}_\rho = H \circ \mathcal{M}'_\rho$ where $\mathcal{M}'_\rho$ supports lossless multiple release and $H$ is an arbitrary function (possibly chosen from some distribution).

We define *weakly lossless gradual release* analogously. It follows that these classes of mechanisms are closed under all post-processing. While this notion is indeed weaker, any algorithm $\mathrm{M}(\mathbf{x}, \rho)$ implementing weakly lossless multiple release for a $\rho$-private mechanism $\mathcal{M}_\rho(\mathbf{x})$, will have the property that the set of releases $(\mathrm{M}(\mathbf{x}, \rho))_{\rho \in S}$ are $\max(S)$-private, if $\rho$-privacy is closed under post-processing. Examples of such privacy notions are $\varepsilon$-DP and $\rho$-zCDP. We get the following immediate corollary to Theorem 4.4.

**Corollary 4.6.** *If the function $H$ in Theorem 4.4 is not invertible, then $\mathcal{M}_\rho = H \circ A_{f,\rho}$ supports weakly lossless multiple release.*

Weakly lossless multiple release will play a role in the next section, where we consider non-invertible post-processing such as truncation.

# 5. Applications

In this section, we describe two applications supported by our framework for lossless multiple release:

- Factorization mechanisms (Li et al., 2015), where we want to privately release a linear query $A\mathbf{x}$, and

- Sparse Gaussian histograms, also known as stability histograms (Wilkins et al., 2024; Google Anonymization Team, 2020).

We will make use of both *lossless* multiple release (Definition 3.4), and its weaker variant (Definition 4.5). This will be necessary since, e.g., the truncation used for sparse Gaussian histograms is not an invertible function, and so not covered by Theorem 4.4.

Because the noise generation for histograms on large domains is expensive, we give a dimension-independent algorithm that works in the gradual release setting. Throughout, we state our results as post-processings of the Gaussian mechanism, but analogous results hold for any other independent additive noise mechanism meeting Definition 1.1.

## 5.1. Lossless Multiple Release Factorization Mechanism

Let $A$ be a query matrix and fix a public factorization $A = LR$. Denote the *factorization mechanism* (Li et al., 2015) as $\mathcal{F}_\rho(\mathbf{x}) = A\mathbf{x} + Lz = L(R\mathbf{x} + Z)$ on some private dataset $\mathbf{x}$ where $Z$ is a vector drawn from a distribution $\mathcal{D}(\rho)$ satisfying Definition 1.1 parameterized by $\rho$, typically Gaussian or Laplace noise.

**Lemma 5.1.** *$\mathcal{F}_\rho$ can be implemented with weakly lossless multiple release, and lossless multiple release if $L$ has a left-inverse.*

*Proof.* Observe that $\mathcal{F}_\rho(\mathbf{x}) = L \circ \mathcal{A}_{R,\rho}$ for an IAN mechanism $\mathcal{A}_{R,\rho}(\mathbf{x}) = R\mathbf{x} + Z$ where $Z \sim \mathcal{D}(\rho)$ for a $\mathcal{D}(\rho)$ satisfying Definition 1.1. The result follows from Theorem 4.4 and Corollary 4.6, and noting that the linear map $L$ is invertible exactly when $L$ has a left-inverse. $\square$

Besides proving existence in Lemma 5.1, we also give an explicit instantiation in Algorithm 2 using the Gaussian mechanism.

**Lemma 5.2.** *With initialization $M = \{(0, \infty), (\infty, A\mathbf{x})\}$, Algorithm 2 implements the Gaussian noise factorization mechanism $\mathcal{F}_\rho$ with (weakly) lossless multiple release.*

*Proof sketch.* Note that the algorithm is practically a copy of Algorithm 1, but for the specific sensitive function $R\mathbf{x}$. The only structural difference is that the linearity of the post-processing $L$ allows for storing correlated Gaussian noise directly in $M$. $\square$

---

**Algorithm 2** FactorizationMultipleRelease

**Parameters:** Factorization $A = LR$, $\ell_2$ sensitivity $\Delta_2$
**Inputs:** Set of releases $M$, privacy parameter $\rho$
1: Find $(\rho_k, Y_{\rho_k}), (\rho_{k+1}, Y_{\rho_{k+1}}) \in M$ such that
2: $\quad \rho \in [\rho_k, \rho_{k+1})$ and $\forall (\rho', \cdot) \in M : \rho' \notin (\rho_k, \rho_{k+1})$
3: $Z_\rho \sim \mathrm{N}\left(0, \Delta_2^2 \cdot \frac{(1-\rho_k/\rho)\cdot(1/\rho-1/\rho_{k+1})}{2(1-\rho_k/\rho_{k+1})}\right)^d$
4: Let $Y_\rho := L \cdot Z_\rho + \frac{(1-\rho_k/\rho)Y_{\rho_{k+1}} + (\rho_k/\rho-\rho_k/\rho_{k+1})Y_{\rho_k}}{1-\rho_k/\rho_{k+1}}$
5: Set $M = M \cup \{(\rho, Y_\rho)\}$
6: **Return** $Z$

---

## 5.2. Weakly Lossless Multiple Release of Histograms

We will now describe an example where the post-processing is neither linear nor invertible: releasing sparse (Gaussian) histograms (Korolova et al., 2009; Google Anonymization Team, 2020; Balle & Wang, 2018). Let $\mathbf{x} = (X_i)^n$ be a dataset of $n$ users, and we want to privately release the histogram $H(\mathbf{x}) = \sum_{i=1}^n X_i$ where $X_i \in \{0, 1\}^d$ and a user can contribute to $l$ distinct counts and the Gaussian mechanism which scales proportional to $\sqrt{l}$ is preferable. In many natural settings, the domain size $d$ can be very large, and therefore, the resulting histogram is usually very sparse, $k = \|H(\mathbf{x})\|_0 \ll d$. Releasing a noisy histogram where noise has been added to *each* coordinate would destroy the sparsity. One can cope by introducing a threshold $\tau > 0$ where only noisy counts that exceed $\tau$ are released. $\tau$ is usually set high enough such that the noise to a zero count will (after thresholding with high probability) still be zero. We denote the *support* of the histogram as $U(H(\mathbf{x})) = \{i \in [d] : H(\mathbf{x})_i \neq 0\}$ as the index of the non-zero coordinates. Define the thresholding function $T_\tau : \mathbb{R}^d \to \mathbb{R}^d$ as:

$$T_\tau(x) = (y_i)_{i=1}^d, \quad \text{where} \quad y_i = \begin{cases} x_i, & \text{if } x_i \geq \tau \\ 0, & \text{otherwise} \end{cases}.$$

Denote the (independent additive noise) sparse histogram mechanism as $\mathcal{H}_\rho(\mathbf{x}) = T_\tau(H(\mathbf{x}) + Z)$ where $Z \sim \mathcal{D}(\rho)$ is a distribution satisfying a convolution preorder (Definition 1.1).

**Lemma 5.3.** *$\mathcal{H}_\rho$ can be implemented with weakly lossless multiple release.*

*Proof.* Observe that $\mathcal{H}_\rho$ can be expressed as a composite function $T_\tau \circ \mathcal{A}_{\rho,f}(\mathbf{x})$ for an independent additive noise mechanism $\mathcal{A}_{\rho,f}(\mathbf{x})$ meeting Definition 1.1. The statement thus follows from Corollary 4.6. $\square$

It is straightforward to implement weakly lossless gradual release for stability histograms using our framework with time and space complexity linear in dimension $d$; see Algorithm 3. The following lemma is proved in Appendix C.

---

**Algorithm 3** HistogramGradualRelease

**Parameters:** $\ell_2$ sensitivity $\Delta_2$

**Inputs:** histogram $H(\mathbf{x})$, set of releases $M$,
    privacy parameter $\rho$, threshold $\tau$

1: Extract the single element $(\rho', Z') \in M$
2: Sample $\tilde{Z} \sim \mathrm{N}\left(0, \frac{1}{2}\Delta_2^2(\rho - \rho')\right)^d$
3: Let $Z = \dfrac{\rho'}{\rho}Z' + \dfrac{1}{\rho}\tilde{Z}$
4: **for** each $i \in [d]$ **do**
5:    **if** $H(\mathbf{x})_i + Z_i > \tau$ **then** $Y_i = H(\mathbf{x})_i + Z_i$
6:    **else** $Y_i = 0$
7: **end for**
8: Set $M = \{(\rho, Z)\}$
9: **Return** $Y$

---

**Lemma 5.4.** *With initialization $M = \{(0,0)\}$, Algorithm 3 implements $\mathcal{H}_\rho$ with weakly lossless gradual release.*

**Improving efficiency for weakly lossless gradual release.** Korolova, Kenthapadi, Mishra, and Ntoulas (2009) showed that under approximate differential privacy, one can skip the noise generation for zero counts if thresholding is enabled, adding only a small probability for infinite privacy loss if this coordinate is non-zero in a neighboring dataset. Unfortunately, this approach does not fit our meta theorem, so instead, we apply a technique due to Cormode, Procopiuc, Srivastava, and Tran (2012) that efficiently simulates the noise distribution of those zero counts that would exceed the threshold and thus gives an identical distribution to the truncated noisy histogram. The procedure is described in the following lemma.

**Lemma 5.5.** *For a fixed noise distribution $\mathcal{D}$, the output distribution of releasing $\mathcal{T}_\tau(H(\mathbf{x}) + Z)$ with $Z \sim \mathcal{D}$ are independent noise samples is equal to the following process: (a) add noise to every non-zero coordinate in $H(\mathbf{x})_i$ for $i \in U(H(\mathbf{x}))$ and then (b) draw $q \sim \mathrm{Binomial}(d-k, p)$ where $p = \Pr_{Z \sim \mathcal{D}}[Z > \tau]$. (c) Sample a subset $Q \subseteq [d] \setminus U(H(\mathbf{x}))$ of size $|Q| = q$ uniformly at random and (d) set the noise for $H(\mathbf{x})$ to $Z \sim \mathcal{D}$ conditioned on being above the threshold $\tau$.*

*Proof sketch.* A formal argument can be found in Cormode et al. (2012), here we just provide a sketch. It is clear that every entry in the support gets the correct output distribution. Furthermore, the number of zero counts $q$ whose noise exceeds $\tau$ follows a binomial distribution, and we can add conditional noise to a uniformly drawn subset of size $q$. $\square$

Algorithm 6 in Appendix C implements the efficient routine described by Lemma 5.5 under weakly lossless gradual release. The key challenge is that the probability for a histogram count to exceed the threshold in a given round now depends on whether it exceeded the threshold in any prior

round. Intuitively, all zero counts that have never exceeded the threshold have an equal probability to exceed the threshold in any given round, and so the simulation technique in Lemma 5.5 can be used for these counts (with different probabilities and conditional distributions). However, once any zero count exceeds the threshold, it will have to be treated the same as non-zero counts in all future rounds.

The utility and privacy of Algorithm 6 is proved by arguing that its output distribution matches that of Algorithm 3. A proof sketch for the following lemma is given in Appendix C.

**Lemma 5.6.** *Let $H(\mathbf{x})$ be a histogram over $[d]$, $\rho_1, \ldots, \rho_m$ be a sequence increasing of privacy budgets, $\tau_1, \ldots, \tau_m$ be a sequence of thresholds and $\Delta_2 > 0$ the $\ell_2$-sensitivity of the histogram. Also let the sequences of outputs $(Y^{(1)}, \ldots, Y^{(m)})$ and $(\hat{Y}^{(1)}, \ldots, \hat{Y}^{(m)})$ be derived from running Algorithm 3 and Algorithm 6 respectively with the preceding parameters as input. Then the sequences $(Y^{(1)}, \ldots, Y^{(m)})$ and $(\hat{Y}^{(1)}, \ldots, \hat{Y}^{(m)})$ are identically distributed.*

The benefit of the efficient sampling technique is that the number of sampled Gaussians will *never* exceed that of the naïve approach and can potentially be in the order of the sparsity depending on the parameter regime.

**Lemma 5.7.** *Let $c$ be the number of zero-counts in the histogram output by Algorithm 6 least once over its execution. Then Algorithm 6 will sample $(k+c)m$ (truncated or otherwise) Gaussians.*

We again refer the reader to Appendix C for a proof.

# 6. Empirical Evaluation

To confirm our theoretical claims, we empirically evaluate the accuracy of lossless multiple release against a baseline algorithm that uses independent releases. We evaluate the impact by focusing on noise in isolation to avoid capturing the effect of specific queries. The baseline algorithm is a simple Gaussian mechanism where noise is drawn independently for each consecutive release. To showcase our algorithm's performance, we demonstrate how the cost incurred by uncoordinated releases grow with the amount of releases, in contrast to the lossless multiple release where there is no additional cost. We repeat our experiments $10^6$ times, and measure the variance of the noise. The plot (Figure 2) shows, as expected, that our mechanism does not lose any utility from making multiple releases. As we can see, there is an expected increase in variance when going from one release to multiple releases—the initial jump is larger the more releases we want to make as the budget used for the second release is the difference between the starting point ($\rho = 0.001$) and a constant increase in budget, whereas the subsequent releases all have the same difference in budget.
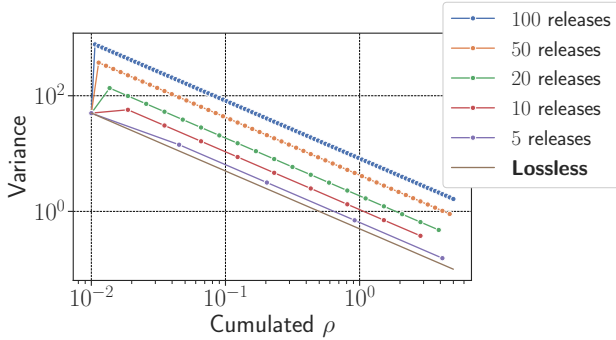
*Figure 2.* Accuracy comparison of multiple uncorrelated releases compared to lossless multiple release. Budgets are spaced evenly on a logarithmic scale between $\rho = 0.001$ and $\rho = 5$ on the x-axis. Creating independent releases with a denser set of privacy parameters comes at the cost of increased variance. In the lossless setting we get the best possible variance with no bound on the number of releases.

## 7. Conclusion and Open Questions

We have initiated a systematic study of differential privacy with multiple releases, motivated by settings in which many levels of privacy or trust may co-exist. The main message is that it is possible to generalize a large class of lossless gradual release techniques to this setting, even when releases are determined online and in no particular order. For mechanisms based on Gaussian, Laplace, or Poisson additive noise we give simple and efficient sampling procedures for creating new lossless releases. In particular, we are able to do lossless multiple release for any factorization mechanism using invertible matrices. Finally, we consider algorithmic challenges related to lossless gradual release and show that private sparse histograms may be computed much more efficiently than what a direct application of our general results would imply.

There are still many open questions concerning mechanisms that do not inherit their privacy guarantee from an independent additive noise mechanism. In particular, it would be interesting to determine under which conditions the exponential mechanism (McSherry & Talwar, 2007) supports lossless multiple release. Koufogiannis et al. (2016) conjecture that this is always possible. Other central private algorithms, such as report noisy max (Dwork et al., 2014) also do not have known lossless multiple release mechanisms, even in the gradual release setting. A final challenge we would like to mention is implementing our mechanisms on a finite computer, e.g., creating a multiple release version of the discrete Gaussian mechanism (Canonne et al., 2020). An appealing approach would be to base this on Poisson noise, which meets the technical conditions for our framework and approaches the Gaussian distribution in the limit.

## Impact Statement

This paper presents work whose goal is to advance the field of private machine learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

## References

Agarwal, N., Suresh, A. T., Yu, F. X., Kumar, S., and McMahan, B. cpSGD: Communication-Efficient and Differentially-Private Distributed SGD. In *Advances in Neural Information Processing Systems*, volume 31, pp. 7575–7586, 2018.

Balle, B. and Wang, Y. Improving the Gaussian Mechanism for Differential Privacy: Analytical Calibration and Optimal Denoising. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 403–412. PMLR, 2018.

Bell, D. E. and La Padula, L. J. Secure Computer System: Unified Exposition and Multics Interpretation:. Technical report, Defense Technical Information Center, Fort Belvoir, VA, mar 1976.

Bun, M. and Steinke, T. Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds. In *Theory of Cryptography*, Lecture Notes in Computer Science, pp. 635–658, Berlin, Heidelberg, 2016. Springer. ISBN 978-3-662-53641-4. doi: 10.1007/978-3-662-53641-4_24.

Canonne, C. L., Kamath, G., and Steinke, T. The Discrete Gaussian for Differential Privacy. In *Advances in Neural Information Processing Systems*, volume 33, pp. 15676–15688. Curran Associates, Inc., 2020.

Cormode, G., Procopiuc, C. M., Srivastava, D., and Tran, T. T. L. Differentially Private Summaries for Sparse Data.

In *15th International Conference on Database Theory*, pp. 299–311, New York, NY, USA, 2012. ACM. ISBN 9781450307918. doi: 10.1145/2274576.2274608.

Ding, Z., Kifer, D., Saghaian N. E., S. M., Steinke, T., Wang, Y., Xiao, Y., and Zhang, D. The Permute-and-Flip Mechanism is Identical to Report-Noisy-Max with Exponential Noise. *CoRR*, abs/2105.07260, 2021.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography*, Lecture Notes in Computer Science, pp. 265–284. Springer Berlin Heidelberg, 2006. ISBN 978-3-540-32732-5. doi: 10.1007/11681878_14.

Dwork, C., Roth, A., et al. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014. doi: 10.1561/0400000042.

Equitz, W. H. R. and Cover, T. M. Successive Refinement of Information. *IEEE Trans. Inf. Theory*, 37(2):269–275, 1991. doi: 10.1109/18.75242.

Girgis, A. M., Data, D., Chaudhuri, K., Fragouli, C., and Diggavi, S. N. Successive refinement of privacy. *IEEE Journal on Selected Areas in Information Theory*, 1(3): 745–759, 2020. doi: 10.1109/JSAIT.2020.3040403.

Google Anonymization Team. Delta for Thresholding. https://github.com/google/differential-privacy/blob/a7cc26ad91f74756fbe39bab44af6d655b37cc61/common_docs/Delta_For_Thresholding.pdf, 2020.

Korolova, A., Kenthapadi, K., Mishra, N., and Ntoulas, A. Releasing Search Queries and Clicks Privately. In *Proceedings of the 18th International Conference on World Wide Web*, pp. 171–180, 2009. doi: 10.1145/1526709.1526733.

Kotz, S., Kozubowski, T. J., and Podgórski, K. *The Laplace Distribution and Generalizations*. Birkhäuser Boston, 2001. ISBN 978-1-4612-6646-4 978-1-4612-0173-1. doi: 10.1007/978-1-4612-0173-1.

Koufogiannis, F., Han, S., and Pappas, G. J. Gradual Release of Sensitive Data under Differential Privacy. *Journal of Privacy and Confidentiality*, 7(2), 2016. ISSN 2575-8527. doi: 10.29012/jpc.v7i2.649. Number: 2.

Li, C., Miklau, G., Hay, M., McGregor, A., and Rastogi, V. The Matrix Mechanism: Optimizing Linear Counting Queries under Differential Privacy. *VLDB J.*, 24(6):757–781, 2015. doi: 10.1007/s00778-015-0398-x.

Li, Y., Chen, M., Li, Q., and Zhang, W. Enabling Multilevel Trust in Privacy Preserving Data Mining. *IEEE Trans. Knowl. Data Eng.*, 24(9):1598–1612, 2012. doi: 10.1109/TKDE.2011.124.

Ligett, K., Neel, S., Roth, A., Waggoner, B., and Wu, Z. S. Accuracy First: Selecting a Differential Privacy Level for Accuracy Constrained ERM. In *Advances in Neural Information Processing Systems*, volume 30, pp. 2566–2576, 2017.

McSherry, F. and Talwar, K. Mechanism Design via Differential Privacy. In *IEEE Symposium on Foundations of Computer Science*, pp. 94–103. IEEE, 2007. doi: 10.1109/FOCS.2007.41.

Pan, M. Randomized Response with Gradual Release of Privacy Budget. *CoRR*, abs/2401.13952, 2024. doi: 10.48550/arXiv.2401.13952.

Rogers, R. M., Vadhan, S. P., Roth, A., and Ullman, J. R. Privacy Odometers and Filters: Pay-as-you-Go Composition. In *Advances in Neural Information Processing Systems*, volume 29, pp. 1921–1929, 2016.

Shaked, M. and Shanthikumar, J. G. (eds.). *Univariate Stochastic Orders*, pp. 3–79. Springer New York, New York, NY, 2007. ISBN 978-0-387-34675-5. doi: 10.1007/978-0-387-34675-5_1.

Whitehouse, J., Ramdas, A., Wu, Z. S., and Rogers, R. M. Brownian Noise Reduction: Maximizing Privacy Subject to Accuracy Constraints. In *Advances in Neural Information Processing Systems*, volume 35, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Wilkins, A., Kifer, D., Zhang, D., and Karrer, B. Exact Privacy Analysis of the Gaussian Sparse Histogram Mechanism. *Journal of Privacy and Confidentiality*, 14(1), feb 2024. ISSN 2575-8527. doi: 10.29012/jpc.823.

Xiao, X., Tao, Y., and Chen, M. Optimal Random Perturbation at Multiple Privacy Levels. *Proc. VLDB Endow.*, 2 (1):814–825, 2009. doi: 10.14778/1687627.1687719.

Zhang, S. and He, X. DProvDB: Differentially Private Query Processing with Multi-Analyst Provenance. *Proc. ACM Manag. Data*, 1(4):267:1–267:27, 2023. doi: 10.1145/3626761.

---

**Algorithm 4** GenericMultipleRelease

---

**Parameters:** Noise distribution family $\mathcal{D}(\rho)$ satisfying a convolution preorder,
    bridging noise distribution $\mathcal{C}(\rho, \rho')$ where $\mathcal{D}(\rho') * \mathcal{C}(\rho', \rho) = \mathcal{D}(\rho)$ for $0 < \rho < \rho'$.
**Inputs:** Sensitive value $f(\mathbf{x})$, new privacy parameter $\rho_k$, set of past releases $M = \{(\rho_i, Y_{\rho_i}) : i \in [m] \setminus \{k\}\}$

  1: **if** $M = \emptyset$ **then**
  2:    Sample $Z_{\rho_k}$ from $\mathcal{D}(\rho_{\rho_k})$
  3:    Set $Y_{\rho_k} = f(\mathbf{x}) + Z_{\rho_k}$
  4: **else if** $k = 1$ **then**
  5:    Sample $W_{\rho_{k+1}, \rho_k}$ from $\mathcal{C}(\rho_{k+1}, \rho_k)$
  6:    Set $Y_{\rho_k} = Y_{\rho_{k+1}} + W_{\rho_{k+1}, \rho_k}$
  7: **else if** $k \in (1, m)$ **then**
  8:    Sample $W_{\rho_{k+1}, \rho_k}$ from $\mathcal{C}(\rho_{k+1}, \rho_k)$ conditioned on $W_{\rho_{k+1}, \rho_k} + W_{\rho_k, \rho_{k-1}} = Y_{\rho_{k-1}} - Y_{\rho_{k+1}}$
  9:    Set $Y_{\rho_k} = Y_{\rho_{k+1}} + W_{\rho_{k+1}, \rho_k}$
10: **else if** $k = m$ **then**
11:    Sample $Z_{\rho_k}$ from $\mathcal{D}(\rho_k)$ conditioned on $Z_{\rho_k} + W_{\rho_k, \rho_{k-1}} = Y_{\rho_{k-1}} - f(\mathbf{x})$
12:    Set $Y_{\rho_k} = f(\mathbf{x}) + Z_{\rho_k}$
13: **end if**
14: Set $M = M \cup \{(\rho_k, Y_{\rho_k})\}$
15: **Return** $Y_{\rho_k}$

---

## A. On Lossless Multiple Release Sampling

Lemma 4.2 makes no claim about how easy it is to sample noise from the conditional noise distribution, but guarantees its existence. We will use this section for discussing this sampling in greater detail. Consider the joint distribution of the releases defined by Equation (1) from the proof of Lemma 4.2, re-stated below for convenience:

$$Y_{\rho_k} = f(\mathbf{x}) + Z_{\rho_m} + \sum_{j=k}^{m-1} W_{\rho_{j+1}, \rho_j}, \qquad k = 1, \ldots, m, \tag{1}$$

where $Y_{\rho_k}$ is the $\rho_k$-private release, $Z_{\rho_m} \sim \mathcal{D}(\rho_m)$ and $W_{\rho_{j+1}, \rho_j} \sim \mathcal{C}(\rho_{j+1}, \rho_j)$. Recall that $\mathcal{C}(\rho', \rho)$ is the unique distribution where for any $0 < \rho < \rho' : \mathcal{D}(\rho) = \mathcal{D}(\rho') * \mathcal{C}(\rho', \rho)$. As argued for in the proof of Lemma 4.2, to make a new $\rho$-private release, we consider the joint distribution, and sample the new releases conditioned on all past releases.

Formally, for a set of privacy parameters $\rho_1 < \cdots < \rho_m$, we are interested in releasing $\rho_k$ for a single $k \in [m]$, conditioned on the set of past releases $M = \{(\rho_i, Y_{\rho_i}) : i \in [m] \setminus \{k\}\}$. We give pseudocode for this in Algorithm 4, and a lemma for its correctness.

**Lemma A.1.** *Initializing $M = \emptyset$ and then running Algorithm 4 for a set of privacy parameters $\{\rho_i : i \in [m]\}$ (processed in arbitrary order), will produce a set of outputs $\{Y_{\rho_i} : i \in [m]\}$ with distribution given by Equation (1).*

*Proof.* For $M = \emptyset$, we have that $Y_{\rho_k} = f(\mathbf{x}) + \mathcal{D}(\rho_k) = \mathcal{A}_{f, \rho_k}(\mathbf{x})$ on lines 2-3, as expected, where $\mathcal{A}$ is our independent additive noise noise mechanism releasing $f$ with $\rho$-privacy. For the remaining cases, assume $M$ contains $m - 1$ releases with the correct joint distribution, and we are generating a new release at privacy level $\rho_k$. We will argue that Algorithm 4 produces a $Y_{\rho_k}$ whose joint distribution with the releases in $M$ matches (1). For $k = 1$, observe that $Y_{\rho_1} = Y_{\rho_2} + W_{\rho_2, \rho_1}$, and so lines 5-6 are correct. For $1 < k < m$, note that $Y_{\rho_k} = Y_{\rho_{k+1}} + W_{\rho_{k+1}, \rho_k}$ and $Y_{\rho_{k-1}} = Y_{\rho_k} + W_{\rho_k, \rho_{k-1}}$, which combined allows us to identify the correct conditional distribution. $Y_{\rho_k}$ is given by $Y_{\rho_{k+1}} + W_{\rho_{k+1}, \rho_k}$, conditioned on $W_{\rho_{k+1}, \rho_k} + W_{\rho_k, \rho_{k-1}} = Y_{\rho_{k-1}} - Y_{\rho_{k+1}}$, matching lines 8-9. For the final case of $k = m$, then $Y_{\rho_m} = f(\mathbf{x}) + Z_{\rho_m}$ and we have that $Y_{\rho_{m-1}} = Y_{\rho_m} + W_{\rho_m, \rho_{m-1}}$. It follows that the correct noise distribution is $Y_{\rho_m} = f(\mathbf{x}) + Z_{\rho_m}$, conditioned on $Z_{\rho_m} + W_{\rho_m, \rho_{m-1}} = Y_{\rho_{m-1}} - f(\mathbf{x})$, matching lines 11-12. An inductive argument identical to that given in the proof of Lemma 4.2 completes the proof. $\square$

### A.1. Simplification and Removing Dependency on the Dataset

Note that $f(\mathbf{x})$ is only used in Algorithm 4 when a more accurate release is generated (lines 1 and 10). In the remaining cases we are only adding noise to past releases. This already allows us to simplify the algorithm, and argue for not having to

---

**Algorithm 5** SimplifiedGenericMultipleRelease

---

**Parameters:** Noise distribution family $\mathcal{D}(\rho)$ satisfying a convolution preorder,
   bridging noise distribution $\mathcal{C}(\rho, \rho')$ where $\mathcal{D}(\rho') * \mathcal{C}(\rho', \rho) = \mathcal{D}(\rho)$ for $0 < \rho < \rho'$
**Inputs:** Privacy parameter $\rho$, set of releases $M$
 1: Find $(\rho_k, Y_{\rho_k}), (\rho_{k+1}, Y_{\rho_{k+1}}) \in M$ such that $\rho \in [\rho_k, \rho_{k+1})$ and $\forall (\rho', \cdot) \in M : \rho' \notin (\rho_k, \rho_{k+1})$
 2: Sample $W_{\rho_{k+1}, \rho}$ from $\mathcal{C}(\rho_{k+1}, \rho)$ conditioned on $W_{\rho_{k+1}, \rho} + W_{\rho, \rho_k} = Y_{\rho_k} - Y_{\rho_{k+1}}$
 3: Set $Y_\rho = Y_{\rho_{k+1}} + W_{\rho_{k+1}, \rho}$
 4: Set $M = M \cup \{(\rho, Y_\rho)\}$
 5: **Return** $Y_\rho$

---

store $f(\mathbf{x})$ in memory indefinitely. The algorithm reduces to the case (see line 7 of Algorithm 4) where only the two closest releases are combined into a new one. Such an algorithm is given in Algorithm 5, together with Lemma A.2.

**Lemma A.2.** *Let* $M = \{(\rho_0, Y_{\rho_0}), (\rho_\infty, Y_{\rho_\infty})\}$ *for* $Y_{\rho_\infty} = \mathcal{A}_{\rho_\infty, f}(x)$, *and* $Y_{\rho_0} = Y_{\rho_\infty} + W_{\rho_\infty, \rho_0}$. *Then running Algorithm 5 with input* $M$ *and a set of privacy parameters* $\{\rho_i : i \in [m]\} \subset (\rho_0, \rho_\infty)$ *(processed in arbitrary order), will produce a set of outputs* $\{Y_{\rho_i} : i \in [m]\}$ *with distribution given by Equation (1). Moreover, the set* $M$ *will at all times satisfy* $\rho_\infty$*-privacy.*

*Proof.* The statement follows from carefully comparing to Algorithm 4. Let $S = (\rho_1, \ldots, \rho_m)$ be a sequence of noise values from the lemma statement, and let $O = (Y_{\rho_1}, \ldots, Y_{\rho_m})$ be the outputs. Consider a second sequence $S' = (\rho_\infty, \rho_0, \rho_1, \ldots, \rho_m)$, and consider the corresponding sequence of outputs $O' = (Y'_{\rho_\infty}, Y'_{\rho_0}, Y'_{\rho_1}, \ldots, Y'_{\rho_m})$ from inputting $S'$ and $M' = \emptyset$ to Algorithm 4. We can directly check that $Y_{\rho_0} \stackrel{d}{=} Y'_{\rho_0}$ and $Y_{\rho_\infty} \stackrel{d}{=} Y'_{\rho_\infty}$. For the remaining outputs, note that just before each algorithm is called, their internal states $M$ and $M'$ are also identically distributed. Now, the fact that each $\rho_i \in (\rho_0, \rho_\infty)$ implies that the remaining outputs $Y'_{\rho_1}, \ldots, Y'_{\rho_m}$ are generated from lines 10-12 in Algorithm 4. Checking carefully, lines 1-3 in Algorithm 5 are implementing the same routine, and since $M \stackrel{d}{=} M'$, it follows that $(Y_{\rho_1}, \ldots, Y_{\rho_m}) \stackrel{d}{=} (Y'_{\rho_1}, \ldots, Y'_{\rho_m})$, and so the statement follows from Lemma A.1. The last statement on the $\rho_\infty$-privacy of $M$ follows from Algorithm 5 implements Equation (1), and so each release in $M$ can at any time during execution be viewed as randomized post-processing of $Y_\infty$. □

Essentially, if we commit to supporting a bounded range of privacy parameters then we get a simpler algorithm. Lemma A.2 also says something more: If we commit to supporting a lowest level of privacy $\rho_\infty$, then Algorithm 5 can be implemented in such a way that its internal state is $\rho_\infty$-private. After initializing $M$ using the sensitive function $f(\mathbf{x})$, we can erase $f(\mathbf{x})$ from memory and $Y_\infty$ will contain enough private information to generate all future releases. This could prove useful in settings where the time between releases is large, and we want to limit the private information leaked if the state of the algorithm were to be compromised. This can be compared with the gradual release setting, where natural implementations would require consistent access to $f(\mathbf{x})$.

## B. Omitted Proof for Gaussian Lossless Multiple Release

*Proof of Lemma 3.3.* Throughout the proof we will ignore the factor 2 in the denominator of the variance. To argue that the values generated by the process match the distribution Lemma 3.2, we will argue for increasing subsets of releases. Let $S_n = \{\rho_k : k \in [n-1]\}$ where $0 < \rho_1 < \cdots < \rho_n < \rho_\infty$ be the set of values in $S$ for which we have generated Gaussians at the start of the $n^{\text{th}}$ round of the process. Note that we re-label the $\rho$'s between the rounds such that $\rho_k \in S_n$ always is the $k^{\text{th}}$ smallest value in $S_n$. Our argument will proceed by induction: assume that all the Gaussians $\{Y_{\rho_k} : k \in [n]\}$ generated after $n$ rounds have the distribution given by Lemma 3.2. Then we will show that $\{Y_{\rho_k} : k \in [n-1]\} \cup \{Y_\rho\}$ has the same distribution as predicted by $S_n \cup \{\rho\}$ from invoking Lemma 3.2.

We begin with our base case. For $n = 1$, we have that $\rho_l = 0$ and $\rho_r = \rho_\infty$, and so $Y_\rho = Y_{\rho_\infty} + \mathrm{N}(0, 1/\rho - 1/\rho_\infty)$. Since $Y_{\rho_\infty} \sim \mathrm{N}(\beta, 1/\rho_\infty)$, we have that $Y_\rho \sim \mathrm{N}(\beta, 1/\rho)$, as expected, and so the base case passes.

For $n \geq 2$, we first consider the following cases.

*Case 1*: $\rho_l = 0, \rho_r = \rho_1 \in S_n$. In this case, $Y_\rho = Y_{\rho_1} + \mathrm{N}(1/\rho - 1/\rho_1)$, and so $Y_\rho \sim \mathrm{N}(\beta, 1/\rho)$. Since $\rho < \rho_1$, we have that $\forall i \in [n-1] : \mathrm{Cov}(Y_\rho, Y_{\rho_i}) = \mathrm{Cov}(Y_{\rho_1}, Y_{\rho_i}) = 1/\rho_i$, as expected for the release with the smallest value in $\{\rho\} \cup S_n$, and so the case is complete.

*Case 2*: $\rho_l = \rho_{n-1}, \rho_r = \rho_\infty = \infty$. We have to deal with the case where we have set $\rho_\infty = \infty$ separately, as it is a bit of a trick. Note that we get $Y_\rho = \frac{(1-\rho_{n-1})\beta + \rho_{n-1}Y_{\rho_{n-1}}}{\rho} + \mathrm{N}(0, \frac{1-\rho_{n-1}/\rho}{\rho})$, since $Y_{\rho_\infty} = \beta$ in this case. The end result is once more a sum of Gaussians, with mean $\beta$, and for the variance we can explicitly compute that $\mathrm{Var}[Y_\rho] = \frac{\rho_{n-1}^2}{\rho_{n-1}\rho^2} - \frac{\rho - \rho_{n-1}}{\rho^2} = 1/\rho$, as expected. For the covariance, for $i \in [n-1] : \mathrm{Cov}(Y_\rho, Y_{\rho_i}) = \frac{\rho_{n-1}}{\rho}\mathrm{Cov}(Y_{\rho_{n-1}}, Y_{\rho_i}) = 1/\rho$, as expected for the largest value in $\{\rho\} \cup S_n$, and so we are done. Now we consider the most general case.

*Case 3*: $\rho_l \in S_{n-1}$ and $\rho_r \neq \infty$. We begin with computing the variance of $Y$ using the hypothesis $\mathrm{Var}[Y_{\rho_r}] = 1/\rho_r$ and $\mathrm{Var}[Y_{\rho_l}] = 1/\rho_l$

$$\mathrm{Var}[Y_\rho] = \frac{(\rho_l^{-1} - \rho^{-1})(\rho^{-1} - \rho_r^{-1})}{\rho_l^{-1} - \rho_r^{-1}} + \frac{(\rho_l^{-1} - \rho^{-1})^2\,\mathrm{Var}[Y_{\rho_r}] + (\rho^{-1} - \rho_r^{-1})^2\,\mathrm{Var}[Y_{\rho_l}]}{(\rho_l^{-1} - \rho_r^{-1})^2}$$

$$= \frac{1}{(\rho_l^{-1} - \rho_r^{-1})^2}\left[(\rho_l^{-1} - \rho^{-1})(\rho^{-1} - \rho_r^{-1})(\rho_l^{-1} - \rho_r^{-1}) + \rho_r^{-1}(\rho_l^{-1} - \rho^{-1})^2 + \rho_l^{-1}(\rho^{-1} - \rho_r^{-1})^2\right]$$

$$= \frac{1}{(\rho_l^{-1} - \rho_r^{-1})^2} \cdot \rho^{-1}(\rho_l^{-1} - \rho_r^{-1})^2 = \frac{1}{\rho},$$

where the third equality follows from applying the identity

$$(a-b)(b-c)(a-c) + c(a-b)^2 + a(b-c)^2 = b(a-c)^2,$$

to the expression in the brackets for $a = \rho_l^{-1}, b = \rho^{-1}$ and $c = \rho_r^{-1}$, proving the correctness of the variance. Furthermore, $Y_\rho$ is a sum of Gaussians and a convex combination of two Gaussians with mean $\beta$, so $Y_\rho \sim \mathrm{N}(\beta, 1/\rho)$. What remains to show is that the covariances match up, which we do next.

We start with the case where $\rho_r \neq \rho_\infty$, and so there exists $\rho_k, \rho_{k+1} \in S_n$ such that $\rho \in (\rho_k, \rho_k + 1)$ For $j \in [n-1]$, we therefore have that

$$\mathrm{Cov}(Y_\rho, Y_{\rho_j}) = \mathrm{Cov}\left(\frac{(\rho_l^{-1} - \rho^{-1})Y_{\rho_r} + (\rho^{-1} - \rho_r^{-1})Y_{\rho_l}}{\rho_l^{-1} - \rho_r^{-1}}, Y_{\rho_j}\right)$$

$$= \frac{1}{\rho_l^{-1} - \rho_r^{-1}}\left[(\rho_l^{-1} - \rho^{-1})\mathrm{Cov}(Y_{\rho_r}, Y_{\rho_j}) + (\rho^{-1} - \rho_r^{-1})\mathrm{Cov}(Y_{\rho_l}, Y_{\rho_j})\right]$$

$$= \frac{1}{\rho_l^{-1} - \rho_r^{-1}}\left[(\rho_l^{-1} - \rho^{-1})\min(\rho_r^{-1}, \rho_j^{-1}) + (\rho^{-1} - \rho_r^{-1})\min(\rho_l^{-1}, \rho_j^{-1})\right].$$

We consider the cases of $j \leq k$ and $j > k$ separately. For $j \leq k$, we have that

$$\mathrm{Cov}(Y_\rho, Y_{\rho_j}) = \frac{1}{\rho_l^{-1} - \rho_r^{-1}}\left[(\rho_l^{-1} - \rho^{-1})\rho_r^{-1} + (\rho^{-1} - \rho_r^{-1})\rho_l^{-1}\right] = \frac{1}{\rho},$$

and similarly for $j \geq k+1$ we get

$$\mathrm{Cov}(Y_\rho, Y_{\rho_j}) = \frac{1}{\rho_k^{-1} - \rho_{k+1}^{-1}}\left[(\rho_k^{-1} - \rho^{-1})\rho_j^{-1} + (\rho^{-1} - \rho_{k+1}^{-1})\rho_j^{-1}\right] = \frac{1}{\rho_j}.$$

It follows that $\mathrm{Cov}(Y_\rho, Y_{\rho_j}) = \frac{\Delta_2^2}{2\max(\rho, \rho_j)}$ for $j \in [m]$, and so we are done with this part of the covariances.

For the last step, we consider what happens when $\rho_r = \rho_\infty < \infty$. In this case, $\rho_l = \rho_{n-1}$ and so for $j \in [n-1]$:

$$\mathrm{Cov}(Y_\rho, Y_{\rho_j}) = \mathrm{Cov}\left(\frac{(1 - \rho_{n-1}/\rho)Y_{\rho_\infty} + (\rho_{n-1}/\rho - \rho_{n-1}/\rho_\infty)Y_{\rho_{n-1}}}{1 - \rho_{n-1}/\rho_\infty}, Y_{\rho_j}\right)$$

$$= \frac{1 - \rho_{n-1}/\rho}{1 - \rho_{n-1}/\rho_\infty}\mathrm{Cov}(Y_{\rho_\infty}, Y_{\rho_j}) + \frac{\rho_{n-1}/\rho - \rho_{n-1}/\rho_\infty}{1 - \rho_{n-1}/\rho_\infty}\mathrm{Cov}(Y_{\rho_{n-1}}, Y_{\rho_j})$$

$$= \frac{1 - \rho_{n-1}/\rho}{\rho_\infty - \rho_{n-1}} + \frac{1/\rho - 1/\rho_\infty}{1 - \rho_{n-1}/\rho_\infty} = \frac{1 - \rho_{n-1}/\rho + \rho_\infty/\rho - 1}{\rho_\infty - \rho_{n-1}} = 1/\rho,$$

and now we are done. $\square$

---

**Algorithm 6** EfficientHistogramGradualRelease

---

**Parameters:** $\ell_2$-sensitivity $\Delta_2$
**Inputs:** Private histogram $H(\mathbf{x})$, privacy budgets $\rho_1 < \cdots < \rho_m$, thresholds $\tau_1, \ldots, \tau_m$
 1: Let $S^{(0)} = U(H(\mathbf{x}))$ **// Tracking counts that have already been released**
 2: Also let $Z^{(0)} = \{0\}^d$ and $\rho_0 = 0$.
 3: $\forall r \in [m]$ : let distribution $\mathcal{D}^{(r)} = \mathrm{N}\left(0, \frac{1}{2}\Delta_2^2(\rho_r - \rho_{r-1})\right)$
 4: **for** each round $r \in [m]$ **do**
 5:     Initialize $Y^{(r)} = \{0\}^d$
 6:     **for** each tracked count in preceding round $j \in S^{(r-1)}$ **do**
 7:         Draw fresh noise $\tilde{Z}_j^{(r)} \sim \mathcal{D}^{(r)}$
 8:         Update aggregate noise $Z_j^{(r)} = \frac{\rho_{r-1}}{\rho_r}Z_j^{(r-1)} + \frac{1}{\rho_r}\tilde{Z}_j^{(r)}$
 9:         **if** $H(\mathbf{x})_j + Z_j^{(r)} > \tau_r$ **then**
10:             $Y_j^{(r)} = H(\mathbf{x})_j + Z_j^{(r)}$
11:         **end if**
12:     **end for**
        **// Simulating Noise for zero counts**
13:     Let $\tilde{Z}^{(k)} \sim \mathcal{D}^{(k)}$ for $k \in [r]$ be random variables.
14:     Compute $p^{(r)} = \Pr\left[\sum_{k=1}^{r}\tilde{Z}^{(k)} > \tau_r\rho_r \mid \{\forall \ell \in [r-1] : \sum_{k=1}^{\ell}\tilde{Z}^{(k)} \le \rho_\ell\tau_\ell\}\right]$
15:     Draw $q \sim \mathrm{Binomial}\left(d - |S^{(r-1)}|, p^{(r)}\right)$.
16:     Select a subset $Q \subseteq [d] \setminus S^{(r-1)}$ uniformly at random of size $q$.
17:     **for** each index $j \in Q$ **do**
18:         Initialize $Z_j^{(r)} = 0$
19:         **for** $k \in [r-1]$ **do**
20:             Draw fresh noise $\tilde{Z}_j^{(k)} \sim \mathcal{D}^{(k)}$ conditioned on $\tilde{Z}_j^{(k)} \le \rho_k\tau_k - Z_j^{(r)}$
21:             Update aggregate noise $Z_j^{(r)} = Z_j^{(r)} + \tilde{Z}_j^{(k)}$
22:         **end for**
23:         Draw fresh noise $\tilde{Z}_j^{(k)} \sim \mathcal{D}^{(k)}$ conditioned on $\tilde{Z}_j^{(k)} > \rho_k\tau_k - Z_j^{(r)}$
24:         Update aggregate noise $Z_j^{(r)} = Z_j^{(r)} + \tilde{Z}_j^{(k)}$
25:         $Y_j^{(r)} = H(\mathbf{x})_j + Z_j^{(r)}$ **// Guaranteed to be above the threshold.**
26:     **end for**
27:     Set $S^{(r)} = S^{(r-1)} \cup Q$
28:     **output** private histogram $Y^{(r)}$
29: **end for**

---

## C. Details on (Efficient) Weakly Lossless Gradual Release of Sparse Gaussian Histograms

Algorithm 3 implements *weakly lossless gradual release* of private histograms. We prove this next.

*Proof of Lemma 5.4.* Observe that for any increasing sequence $(\rho_k)_{k \in m}$, the corresponding sequence of variables $Z$ on line 3 produced by the algorithm, have the same distribution as in Lemma 3.1 for $\beta = 0$. The $Y$ that is ultimately returned, however, is a post-processing of $H(\mathbf{x}) + Z$, which has the same distribution as Lemma 3.1 for $\beta = H(\mathbf{x})$. Therefore Algorithm 3 is implementing lossless gradual release for the Gaussian mechanism applied to $H(\mathbf{x})$, combined with a non-invertible post-processing. The lemma statement follows from Corollary 4.6. $\quad\square$

Note that one *only* has to store the noisy terms from the preceding round to implement Algorithm 3. Nevertheless, it might be infeasible, say, when the domain is really large, to sample the noise for the zero coordinates. Algorithm 6 uses the computational trick in Lemma 5.5 to speed up this computation. The algorithm is static, where the privacy budgets are fixed upfront but can easily be converted to an online algorithm. Recall that $U(H(\mathbf{x})) = \{i \in [d] : H(\mathbf{x})_i \ne 0\}$ is the support of the histogram.

We proceed to give proofs for Lemma 5.6 and 5.7.

*Proof sketch for Lemma 5.6.* The claim directly follows an inductive argument. For the base case, note that the first iteration of Algorithm 6 is running the same routine as described by Lemma 5.5, and so $\hat{Y}^{(1)}$ must be identically distributed to $Y^{(1)}$. For our inductive hypothesis, assume that the subsequences $(Y^{(1)}, \ldots, Y^{(k)})$ and $(\hat{Y}^{(1)}, \ldots, \hat{Y}^{(k)})$ are identically distributed. Note that for the $(k+1)^{\text{st}}$ release, Algorithm 6 handles all the true nonzero counts and every zero count that has ever exceeded the threshold in a past round in the same way. These counts in $\hat{Y}^{(k+1)}$ and $Y^{(k+1)}$ clearly have the same distribution.

Now, note that each of the remaining zero counts have up to and including the $k^{\text{th}}$ round been reported as zero in each round, and so their probability of exceeding the threshold, $p^{(k+1)}$, should be equal across all of them.

The sampling performed by Algorithm 6 is structured in the same manner as Lemma 5.5, but with sampling probability $p^{(k+1)}$, and the noise terms added for any zero count exceeding the threshold, are different. For the distributions to match, the probability of exceeding the threshold and the noise distribution for an element chosen to exceed the threshold are more complex. The probability of exceeding the threshold is conditioned on prefixes of Gaussians, which correctly simulates the probability of not exceeding the threshold at any prior round until first doing so in the $(k+1)^{\text{st}}$ one. The noise added once selected to exceed the threshold is a sum of truncated Gaussians, which simulate the same event. As the principle is built on the same logic as Lemma 5.5, we have that $(Y^{(1)}, \ldots, Y^{(k+1)})$ and $(\hat{Y}^{(1)}, \ldots, \hat{Y}^{(k+1)})$ are identically distributed, and so the claim is true by induction. □

*Proof of Lemma 5.7.* Note that each of the $k$ non-zero true counts will, in each round, have fresh noise added on line 7. If a non-zero count is selected by the binomial sampling in the $k^{\text{th}}$ round, the for-loop starting on line 17 will result in sampling $k$ noise terms, after which for future rounds it will get treated as a non-zero count. It follows that non-zero entries contribute $km$ samples, and zero counts exceeding the threshold contribute $cm$ samples. □

## D. Definitions

We begin with the most common version of differential privacy.

**Definition D.1** (($\varepsilon, \delta$)-Differential Privacy (Dwork et al., 2014)). A randomized algorithm $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ is ($\varepsilon, \delta$)-differentially private if for all $S \subseteq \text{Range}(\mathcal{M})$ and all pairs of neighboring inputs $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$, it holds that

$$\Pr[\mathcal{M}(\mathbf{x}) \in S] \leq \exp(\varepsilon) \Pr[\mathcal{M}(\mathbf{x}') \in S] + \delta,$$

where $(\varepsilon, 0)$-DP is referred to as $\varepsilon$-DP.

Zero-Concentrated Differential Privacy (zCDP) is a notion of differential privacy that provides a simple but accurate analysis of privacy loss, particularly under composition.

**Definition D.2** (Bun & Steinke (2016), $\rho$-zCDP). Let $\rho > 0$. An algorithm $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ satisfies $\rho$-zCDP, if for all $\alpha > 1$ and all pairs of neighboring inputs $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$, it holds that

$$D_\alpha\left(\mathcal{M}(\mathbf{x}) \,\|\, \mathcal{M}(\mathbf{x}')\right) \leq \rho\alpha,$$

where $D_\alpha\left(\mathcal{M}(\mathbf{x}) \,\|\, \mathcal{M}(\mathbf{x}')\right)$ denotes the $\alpha$-Rényi divergence between two output distributions of $\mathcal{M}(\mathbf{x})$ and $\mathcal{M}(\mathbf{x}')$.

**Lemma D.3** (Bun & Steinke (2016), Composition). *If $\mathcal{M}_1(\mathbf{x})$ and $\mathcal{M}_2(\mathbf{x})$ satisfy $\rho_1$-zCDP and $\rho_2$-zCDP, respectively, then $(\mathcal{M}_1(\mathbf{x}), \mathcal{M}_2(\mathbf{x}))$ satisfies $(\rho_1 + \rho_2)$-zCDP.*

**Lemma D.4** (Bun & Steinke (2016), Gaussian Mechanism). *Let $f : \mathcal{X} \to \mathbb{R}^d$ be a query with $\ell_2$-sensitivity $\Delta_2$. Consider the mechanism $\mathcal{G}_{f,\rho} : \mathcal{X} \to \mathbb{R}^d$ that, on private input $\mathbf{x}$, releases a sample from $f(\mathbf{x}) + \text{N}\left(0, \frac{\Delta_2^2}{2\rho}\right)^d$. Then $\mathcal{G}_{f,\rho}$ satisfies $\rho$-zCDP.*

One way to uniquely describe probability distributions is via their *Characteristic Functions*.

**Definition D.5** (Characteristic Function). The characteristic function of a random variable $X$ is defined as $\varphi_X(t) = \mathbb{E}[e^{itX}]$.

For proving Lemma F.4 and Claim F.7, we will use a nice property of CFs for the convolution of two random variables:

**Lemma D.6** (Convolution of Characteristic Functions). *Let $X, Y$ be two independent RVs with CFs $\varphi_X(t)$ and $\varphi_Y(t)$ respectively, then $\varphi_{X+Y}(t) = \varphi_X(t) \cdot \varphi_Y(t)$.*

*Proof.* Furthermore, by linearity of expectation, we have for the sum of $X + Y$:

$$\varphi_{X+Y}(t) = \mathbb{E}[e^{it(X+Y)}] = \mathbb{E}[e^{itX}] + \mathbb{E}[e^{itY}] = \varphi_X(t) \cdot \varphi_X(t). \qquad \square$$

# E. The Poisson Mechanism

We are not aware of any explicit statements in the literature on the privacy guarantees obtained by adding Poisson distributed noise to a $d$-dimensional vector of integers. However, since the Poisson distribution is the limiting distribution of binomial distributions with the same mean $\lambda = Np$, where $N$ is the number of trials, such bounds can be derived from existing bounds on the binomial mechanism. For the sake of completeness, we include such statements based on the following theorem from (Agarwal et al., 2018):

**Theorem E.1** (Agarwal et al. (2018)). *For any $\delta$, parameters $N, p$ and sensitivity bounds $\Delta_1, \Delta_2, \Delta_\infty$ such that*

$$Np(1-p) \geq \max\left(23 \log\left(\frac{10d}{\delta}\right), 2\Delta_\infty\right),$$

*the $d$-dimensional Binomial mechanism is $(\varepsilon, \delta)$-differentially private for*

$$\varepsilon = \frac{\Delta_2 \sqrt{2 \log \frac{1.25}{\delta}}}{\sqrt{Np(1-p)}} + \frac{\Delta_2 c_p \sqrt{\log \frac{10}{\delta}} + \Delta_1 b_p}{Np(1-p)(1-\delta/10)} + \frac{\frac{2}{3}\Delta_\infty \log \frac{1.25}{\delta} + \Delta_\infty d_p \log \frac{20d}{\delta} \log \frac{10}{\delta}}{Np(1-p)}.$$

*where*

$$d_p \triangleq \frac{4}{3} \cdot \left(p^2 + (1-p)^2\right), \quad b_p \triangleq \frac{2(p^2 + (1-p)^2)}{3} + (1-2p), \quad c_p \triangleq \sqrt{2}\left(3p^3 + 3(1-p)^3 + 2p^2 + 2(1-p)^2\right).$$

Setting $p = \lambda/N$ and considering the limiting bound when $N \to \infty$ we get:

**Theorem E.2** (Privacy guarantees of the Poisson Mechanism). *The $d$-dimensional Poisson mechanism with parameter $\lambda > \max(23 \log(10d/\delta), 2\Delta_\infty)$, is $(\varepsilon, \delta)$-differentially private with*

$$\varepsilon = \frac{\Delta_2 \sqrt{2 \log \frac{1.25}{\delta}}}{\sqrt{\lambda}} + \frac{5\sqrt{2}\,\Delta_2 \sqrt{\log \frac{10}{\delta}} + \frac{5}{3}\Delta_1}{\lambda(1-\delta/10)} + \frac{\frac{2}{3}\Delta_\infty \log \frac{1.25}{\delta} + \frac{4}{3}\Delta_\infty \log \frac{20d}{\delta} \, \log \frac{10}{\delta}}{\lambda}.$$

A simpler expression can be derived for unit sensitivities by relaxing the constants and assuming that $\delta$ is not too large:

**Corollary E.3** (Simplified upper bound with unit sensitivities). *Assume that $\Delta_1 = \Delta_2 = \Delta_\infty = 1$. Then for $\delta < 1/100$, the $d$-dimensional Poisson mechanism with parameter $\lambda > 23 \log(10d/\delta)$ is $(\varepsilon, \delta)$-differentially private for*

$$\varepsilon = \frac{\sqrt{2 \log \frac{1.25}{\delta}}}{\sqrt{\lambda}} + \frac{2 \log \frac{20d}{\delta} \, \log \frac{10}{\delta}}{\lambda}.$$

We will need the following lemma that determines the sampling step of private lossless multiple release for the Poisson mechanism:

**Lemma E.4.** *Let $X_1 \sim \text{Poi}(\lambda_1)$ and $X_2 \sim \text{Poi}(\lambda_2)$ be independent Poisson random variables. Then, for any nonnegative integer $k$, the conditional distribution of $X_1$ given $X_1 + X_2 = k$ is*

$$X_1 \mid (X_1 + X_2 = k) \sim \text{Binomial}\left(k, \frac{\lambda_1}{\lambda_1 + \lambda_2}\right).$$

*Proof.* Since $X_1$ and $X_2$ are independent, their joint probability mass function is for all $x = 0, 1, \cdots, k$

$$P[X_1 = x, \, X_2 = k - x] = e^{-(\lambda_1 + \lambda_2)} \frac{\lambda_1^x}{x!} \frac{\lambda_2^{k-x}}{(k-x)!}.$$

Moreover, the sum $X_1 + X_2$ is Poisson with parameter $\lambda_1 + \lambda_2$, so that

$$\Pr[X_1 + X_2 = k] = e^{-(\lambda_1 + \lambda_2)} \frac{(\lambda_1 + \lambda_2)^k}{k!} \, .$$

Thus, by the definition of conditional probability,

$$
\begin{aligned}
\Pr[X_1 = x \mid X_1 + X_2 = k] &= \frac{\Pr[X_1 = x, \, X_2 = k - x]}{\Pr[X_1 + X_2 = k]} \\
&= \frac{e^{-(\lambda_1 + \lambda_2)} \dfrac{\lambda_1^x}{x!} \dfrac{\lambda_2^{k-x}}{(k-x)!}}{e^{-(\lambda_1 + \lambda_2)} \dfrac{(\lambda_1 + \lambda_2)^k}{k!}} \\
&= \binom{k}{x} \left( \frac{\lambda_1}{\lambda_1 + \lambda_2} \right)^x \left( \frac{\lambda_2}{\lambda_1 + \lambda_2} \right)^{k-x} .
\end{aligned}
$$

This is precisely the probability mass function of a Binomial random variable with parameters $k$ and $\frac{\lambda_1}{\lambda_1 + \lambda_2}$. $\qquad \square$

## F. The Laplace Mechanism

Koufogiannis et al. (2016) showed how to do lossless *gradual* releases for the Laplace mechanism and supports either tightening the privacy guarantees or loosening them. We will now strengthen this result by exactly showing how to do them in arbitrary order, similar to what was done in Section 4 for the Gaussian mechanism. We will first show that the Laplace distribution satisfies Definition 1.1. This already implies the existence of an algorithm supporting gradual lossless release via Lemma 4.2. After, we will derive how to sample a new release with scale parameter $b$ given two distinct releases with scaling $b_2 < b$ and $b < b_1$.

**Definition F.1** (Laplace distribution). The *zero-centered Laplace distribution* $\mathrm{Lap}(0, b)$ with scale parameter $b > 0$ has probability density function $f_b(x) = \frac{1}{2b} \exp(-|x|/b)$ for all $x \in \mathbb{R}$.

**Lemma F.2** ((Kotz et al., 2001) Characteristic function of Laplace). *Let $X$ be Laplace random variable with probability density function as in Definition F.1, then for all $t \in \mathbb{R}$, the characteristic function of $X$ is*

$$\varphi_X(t) = \mathbb{E}[e^{itX}] = \int_{-\infty}^{\infty} e^{itx} \frac{1}{2b} e^{-|x|/b} dx = \frac{1}{1 + b^2 t^2} \, .$$

*Proof.* By a simple integration:

$$
\begin{aligned}
\mathbb{E}[e^{itX}] &= \frac{1}{2b} \int_{-\infty}^{\infty} e^{itx - |x|/b} dx = \frac{1}{2b} \left( \int_{-\infty}^{0} e^{(1/b + ti)x} dx + \int_{0}^{\infty} e^{(-1/b + ti)x} dx \right) \\
&= \frac{1}{2b} \left( \frac{1}{1/b + ti} + \frac{1}{1/b - ti} \right) = \frac{1}{1 + b^2 t^2} \, . \qquad \square
\end{aligned}
$$

We will now show that the zero-centered Laplace distribution with scale parameter $b$ satisfies convolution preorder (Definition 1.1). Note that a larger value of $b$ corresponds to a more private release by definition. Building on this, we will condition on the two closest releases to create a new one in the middle, as done in Lemma 3.3 for the Gaussian. The following fact was already shown in Koufogiannis et al. (2016), but we give a proof here for completeness using a technique based on characteristic functions as done by Equitz & Cover (1991).

**Claim F.3** (Convolution preorder: Laplace). *Fix $b_2 < b_1 \in \mathbb{R}^+$ let $X \sim \mathrm{Lap}(0, b_2)$ and draw*

$$
W = \begin{cases} 0 & \text{with probability } b_2^2/b_1^2 \\ \mathrm{Lap}(0, b_1) & \text{otherwise} \end{cases} , \quad \text{then } X + W \sim \mathrm{Lap}(0, b_1) \, .
$$

*Proof.* We know that the characteristic function of $X$ is $\varphi_X(t) = \frac{1}{1+b_2^2 t^2}$ and for the convolution $\varphi_{X+W} = \frac{1}{1+b_1^2 t^2}$. Because of independence, the convolution is defined for all $t$ as:

$$\varphi_X(t)\varphi_W(t) = \varphi_{X+W}(t)$$
$$\Leftrightarrow \varphi_W(t) = \frac{1+b_2^2 t^2}{1+b_1^2 t^2} = \frac{b_1^2 + b_1^2 b_2^2 t^2}{b_1^2(1+b_1^2 t^2)} = \frac{b_2^2(1+b_1^2 t^2) + b_1^2 - b_2^2}{b_1^2(1+b_1^2 t^2)} = \frac{b_2^2}{b_1^2} + \left(1 - \frac{b_2^2}{b_1^2}\right) \cdot \frac{1}{1+b_1^2 t^2} .$$

The last expression encodes the convex combination of the claimed mixture distribution because $\frac{1}{1+b_1^2 t^2}$ is again the characteristic function of a zero-centered Laplace distribution with scaling parameter $b_1$. $\square$

We next prove a simple result about the convolution of two Laplace distributions (compare also eq., 2.3.23 of Kotz et al. (2001)).

**Lemma F.4** (Convolution). *For two fixed scaling parameters $b_1 \neq b_2 \in \mathbb{R}^+$, let $X_1 \sim \mathrm{Lap}(0, b_1)$ and $X_2 \sim \mathrm{Lap}(0, b_2)$. Then the density of $X_1 + X_2$ is given by*

$$(f_{b_1} * f_{b_2})(t) = \frac{1}{2(b_1^2 - b_2^2)} \left( b_1 e^{-|t|/b_1} - b_2 e^{-|t|/b_2} \right) .$$

*Proof.* Assume $t \geq 0$ and note that the other case follows by symmetry. We compute the density of the convolution by a straightforward integration:

$$(f_{b_1} * f_{b_2})(t) = \int_{-\infty}^{\infty} f_{b_1}(x) f_{b_2}(t-x) dx = \frac{1}{4 b_1 b_2} \int_{-\infty}^{\infty} \exp\left(-\frac{|x|}{b_1} - \frac{|t-x|}{b_2}\right) dx$$

$$= \frac{1}{4 b_1 b_2} \cdot \left( \int_{-\infty}^{0} e^{\frac{x}{b_1} - \frac{t-x}{b_2}} dx + \int_{0}^{t} e^{-\frac{x}{b_1} - \frac{t-x}{b_2}} dx + \int_{t}^{\infty} e^{-\frac{x}{b_1} - \frac{x-t}{b_2}} dx \right)$$

$$= \frac{1}{4 b_1 b_2} \cdot \left( \left[ \frac{\exp\left(x/b_1 - (t-x)/b_2\right)}{b_1^{-1} + b_2^{-1}} \right]_{-\infty}^{0} \right.$$

$$\left. + \left[ \frac{\exp(-x/b_1 - (t-x)/b_2)}{b_2^{-1} - b_1^{-1}} \right]_{0}^{t} + \left[ \frac{\exp(-x/b_1 - (x-t)/b_2)}{-b_2^{-1} - b_1^{-1}} \right]_{t}^{\infty} \right)$$

$$= \frac{1}{4} \cdot \left( \frac{\exp(-t/b_2)}{b_1 + b_2} + \frac{\exp(-t/b_1) - \exp(-t/b_2)}{b_1 - b_2} + \frac{\exp(-t/b_1)}{b_1 + b_2} \right)$$

$$= \frac{1}{2(b_1^2 - b_2^2)} \left( b_1 e^{-t/b_1} - b_2 e^{-t/b_2} \right) . \qquad \square$$

Now, we are ready to show that the Laplace mechanism can be implemented with multiple releases.

**Lemma F.5** (Multiple release Laplace). *For fixed $0 < b_2 < b < b_1$, let $\mu_1 = b^2/b_1^2$ and $\mu_2 = b_2^2/b^2$ and $D_1 \sim \mathrm{Ber}(\mu_1)$ and $D_2 \sim \mathrm{Ber}(\mu_2)$. Furthermore, let*

$$X_1 = \begin{cases} 0 & \text{if } D_1 = 1 \\ \mathrm{Lap}(0, b) & \text{otherwise} \end{cases} \quad \text{and} \quad X_2 = \begin{cases} 0 & \text{if } D_2 = 1 \\ \mathrm{Lap}(0, b_2) & \text{otherwise} \end{cases} .$$

*Then we have that $\Pr[X_1 = 0 \mid X_1 + X_2 = 0] = 1$, and for every real number $k \neq 0$ the distribution of $X_1$ conditioned on $X_1 + X_2 = k$ is:*

$$X_1 \mid (X_1 + X_2 = k) \sim \begin{cases} 0 & \text{with probability } \mu_1 \cdot (1 - \mu_2) \cdot \dfrac{\exp(-|k|/b_2)}{2 b_2 \cdot f_{X_1+X_2}(k)} \\[2ex] k & \text{with probability } (1 - \mu_1) \cdot \mu_2 \cdot \dfrac{\exp(-|k|/b)}{2b \cdot f_{X_1+X_2}(k)} \\[2ex] H(b, b_2, k) & \text{with probability } (1 - \mu_1) \cdot (1 - \mu_2) \cdot \dfrac{(f_b * f_{b_2})(k)}{f_{X_1+X_2}(k)} \end{cases} \qquad (3)$$

*where* $0$ *and* $k$ *denote the constant distributions,* $H(b, b_2, k)$ *is the probability distribution with probability density function*

$$h(x) = \frac{f_b(x) f_{b_2}(k - x)}{(f_b * f_{b_2})(k)},$$

$$f_{X_1+X_2}(k) = \frac{\mu_1(1 - \mu_2)}{2b_2} e^{-|k|/b_2} + \frac{(1 - \mu_1)\mu_2}{2b} e^{-|k|/b} \frac{(1 - \mu_1)(1 - \mu_2)}{2(b^2 - b_2^2)} \left( b e^{-|k|/b} - b_2 e^{-|k|/b_2} \right) + \mu_1\mu_2\delta_0(x), \text{ and}$$

$$(f_b * f_{b_2})(k) = \frac{1}{2(b_1^2 - b_2^2)} \left( b_1 e^{-t/b_1} - b_2 e^{-t/b_2} \right),$$

*where* $\delta_0$ *is the Dirac delta function.*

*Proof.* First we consider $\Pr[X_1 = 0 | X_1 + X_2 = 0]$, arguing that

$$\Pr[X_1 = 0 | X_1 + X_2 = 0] = \frac{\Pr[X_1 = 0] \Pr[X_2 = 0]}{\Pr[X_1 + X_2 = 0]} = \frac{\mu_1\mu_2}{\mu_1\mu_2 + 0} = 1.$$

To see why the first equality holds split $\Pr[X_1 + X_2 = 0]$ into each of the four combinations of discrete/continuous for $X_1, X_2$. The only non-zero contribution to the probability mass comes from the discrete/discrete case, as the remaining cases contribute mass proportional to the probability that a continuous distribution assumes an exact value, which is zero.

Next we turn to the distribution of $X_1$ conditioned on $X_1 + X_2 = k$ where $k \neq 0$. Denote by $f_b$ the probability density function of $\mathrm{Lap}(0, b)$. Note that the mixture densities become

$$f_{X_1}(x) = \mu_1\delta_0(x) + (1 - \mu_1)f_b(x),$$
$$f_{X_2}(x) = \mu_2\delta_0(x) + (1 - \mu_2)f_{b_2}(x).$$

Before analyzing the different cases separately, we compute the convolution $X_1 + X_2$.

**Convolution $f_{X_1+X_2}$.**

Note that have to take care of a subtle technicality: The case that both $X_1$ and $X_2$ are zero from the discrete part can only happen when we condition on $X_1 + X_2 = 0$. We can now give the density of the convolution $X_1 + X_2$ at any real point $x$:

$$f_{X_1+X_2}(x) = \sum_{(d_1,d_2)\in\{0,1\}^2} f_{X_1+X_2|D_1=d_1,D_2=d_2}(x) \cdot \Pr[D_1 = d_1 \wedge D_2 = d_2]$$

$$= \mu_1(1 - \mu_2) \cdot f_{b_2}(x) + (1 - \mu_1)\mu_2 \cdot f_b(x) + (1 - \mu_1)(1 - \mu_2) \cdot (f_b * f_{b_2})(x) + \mu_1\mu_2\delta_0(x)$$

$$= \frac{\mu_1(1 - \mu_2)}{2b_2} e^{-|x|/b_2} + \frac{(1 - \mu_1)\mu_2}{2b} e^{-|x|/b} + (1 - \mu_1)(1 - \mu_2)\frac{1}{2(b^2 - b_2^2)} \left( b e^{-|x|/b} - b_2 e^{-|x|/b_2} \right)$$

$$+ \mu_1\mu_2\delta_0(x),$$

where the third term follows from Lemma F.4. Note that the last term only contributes when $x = 0$.

We can now show how the sampling procedure in the claim is justified. We first assume $k \neq 0$ and analyze three possible cases how $X_1 + X_2$ is built up: Either both of them are drawn from the (continuous) Laplace distribution or exactly one. (The case where both are from their respective discrete parts can only happen if $k = 0$, analyzed above.)

**Case 1:** $X_1 = 0$ and $X_2 = k$.
$X_2 = k$ is necessarily from its continuous part. By the definition of conditional probability, we have for $k \neq 0$:

$$\Pr[X_1 = 0 | X_1 + X_2 = k, k \neq 0] = \frac{\Pr[X_1 = 0 \wedge X_1 + X_2 = k]}{\Pr[X_1 + X_2 = k]} = \frac{\Pr[X_1 = 0] \Pr[X_2 = k]}{\Pr[X_1 + X_2 = k]}$$

$$= \mu_1 \frac{f_{X_2}(k)}{f_{X_1+X_2}(k)} = \frac{\mu_1(1 - \mu_2)}{2b_2 f_{X_1+X_2}(k)} \cdot e^{-|k|/b_2} := p_1.$$

For the third equality, we simply used definition of a probability density function via its limit:

$$\lim_{\Delta\to 0^+} \frac{\Pr[X_2 \in [k - \Delta, k + \Delta]]}{\Pr[X_1 + X_2 \in [k - \Delta, k + \Delta]]} = \frac{f_{X_2}(k)}{f_{X_1+X_2}(k)}.$$

19

**Case 2:** $X_1 = k$ and $X_2 = 0$

Now assume the flipped case. By a similar argument:

$$\Pr[X_1 = k | X_1 + X_2 = k] = \frac{\Pr[X_1 = k, X_1 + X_2 = k]}{\Pr[X_1 + X_2 = k]} = \frac{\Pr[X_1 = k]\Pr[X_2 = 0]}{\Pr[X_1 + X_2 = k]} = \frac{(1-\mu_1)\mu_2}{2b f_{X_1+X_2}(k)} \cdot e^{-|k|/b} := p_2 \,.$$

**Case 3:** $X_1 \neq 0, X_2 \neq 0, X_1 + X_2 = k$

In the remaining case, both $X_1$ and $X_2$ are independently sampled from continuous Laplace distributions with probability density functions $f_b(x)$ and $f_{b_2}(x)$. Therefore, with the remaining probability $1 - (p_1 + p_2)$, we know that $X_1$ is sampled according to the following conditional probability density function:

$$f_{X_1}(x)|_{X_1+X_2=k} = \frac{f_{X_1, X_1+X_2}(x, k)}{(f_b * f_{b_2})(k)} = \frac{f_b(x) f_{b_2}(k-x)}{(f_b * f_{b_2})(k)} \,,$$

where the last line follows from the trivial identity $X_1 + X_2 = k \Leftrightarrow X_2 = k - X_1$. This is a valid probability density function because $f$ is trivially non-negative due to its parts being non-negative and furthermore

$$\int_{-\infty}^{\infty} \frac{f_b(x) f_{b_2}(x-k)}{(f_b * f_{b_2})(k)} dx = \frac{1}{(f_b * f_{b_2})(k)} \int_{-\infty}^{\infty} f_b(x) f_{b_2}(k-x) dx = \frac{(f_b * f_{b_2})(k)}{(f_b * f_{b_2})(k)} = 1 \,.$$

What is left is to verify that the probabilities in Equation (3) indeed add up to one for $k \neq 0$:

$$\frac{(1-\mu_1)(1-\mu_2)(f_b * f_{b_2})(k)}{f_{X_1+X_2}(k)} + \frac{(1-\mu_1)\mu_2 \exp(-|k|/b)}{2b f_{X_1+X_2}(k)} + \frac{\mu_1(1-\mu_2)\exp(-|k|/b_2)}{2b_2 f_{X_1+X_2}(k)}$$

$$= \frac{1}{f_{X_1+X_2}(k)} \underbrace{\left( (1-\mu_1)(1-\mu_2)(f_b * f_{b_2})(k) + \frac{(1-\mu_1)\mu_2 \exp(-|k|/b)}{2b} + \frac{\mu_1(1-\mu_2)\exp(-|k|/b_2)}{2b_2} e^{-\lambda_2|k|} \right)}_{f_{X_1+X_2}(k) \text{ for } k \neq 0} = 1$$

$\square$

### F.1. Showing Convolution Preorder for Exponential Noise

The exponential distribution is closely related to the Laplace distribution, but gives poor differential privacy guarantees. Nevertheless, it still serves as a building block for some private mechanisms, e.g., Report-Noisy-Max (Ding et al., 2021). We show next that it also satisfies a convolution preorder.

**Definition F.6** (Exponential distribution). *The* exponential distribution $\mathrm{Exp}(\lambda)$ *with rate parameter* $\lambda > 0$ *has probability density function* $f_\lambda(x) = \lambda \exp(-\lambda x)$ *for all* $x \in \mathbb{R}_+$. *Furthermore, its characteristic function is given by* $\varphi_X(t) = \mathbb{E}[e^{itX}] = \frac{\lambda}{\lambda - it}$

**Claim F.7** (Convolution preorder: Exponential distribution). *Fix* $\lambda_1 < \lambda_2 \in \mathbb{R}_+$, *let* $X \sim \mathrm{Exp}(\lambda_2)$ *and draw*

$$W = \begin{cases} 0 & \text{with probability } \lambda_1/\lambda_2 \\ \mathrm{Exp}(\lambda_1) & \text{otherwise} \end{cases}, \quad \text{then } X + W \sim \mathrm{Exp}(\lambda_1) \,.$$

*Proof.* Using the same trick as in the proof of Claim F.3, we have that

$$\varphi_W(t) = \frac{\varphi_{X+W}(t)}{\varphi_X(t)} = \frac{\lambda_1}{\lambda_2} \cdot \frac{\lambda_2 - it}{\lambda_1 - it} = \frac{\lambda_1}{\lambda_2}\left(1 + \frac{\lambda_2 - \lambda_1}{\lambda_1 - it}\right) = \frac{\lambda_1}{\lambda_2} + \left(1 - \frac{\lambda_1}{\lambda_2}\right)\varphi_{X+W}(t)$$

where the final expression is the characteristic function of the claimed mixture distribution. $\square$