Ripple Mechanisms for Discrete and Private Statistics

Matthew Joseph MTJOSEPH@GOOGLE.COM
Alex Kulesza KULESZA@GOOGLE.COM
Yuyan Wang WANGYY@GOOGLE.COM
Alexander Yu ALEXJYU@GOOGLE.COM
Google Research.

Editors: Steve Hanneke and Tor Lattimore

Abstract

We study *ripple mechanisms* for pure differentially private computation of discrete statistics. For each of three natural statistics – sum, count, and vote – we construct an efficient instance of the ripple mechanism and show that it is often more accurate than the previous state of the art. We also prove that ripple mechanisms are, in some settings, optimal among all discrete pure differentially private additive noise mechanisms.

1. Introduction

Differential privacy (Dwork et al., 2006b) (DP) incorporates randomness into the computation of a statistic T to obscure the presence or absence of any contributing data point. A data point's possible influence is measured by the statistic's sensitivity, which bounds how a statistic can change when a data point is added to or removed from the underlying dataset. For example, ℓ_1 sensitivity bounds the change $\|T(X) - T(X')\|_1$ in the statistic's ℓ_1 norm between neighboring databases $X \sim X'$, and the canonical private algorithm for ℓ_1 sensitivity is adding Laplace noise. More generally, any sensitivity bound $\|T(X) - T(X')\|$ with norm $\|\cdot\|$ leads to an ε-DP noise distribution using the K-norm mechanism (Hardt and Talwar, 2010; Awan and Slavković, 2021).

The K-norm mechanism's generality makes it possible to tailor the norm to the sensitivity space $\{T(X) - T(X') \mid X \sim X'\}$ of the statistic at hand. As one example, Joseph and Yu (2024) studied "contribution-bounded" sums where each data point $x \in \mathbb{R}^d$ has $\|x\|_0 \leq k$ and $\|x\|_\infty \leq 1$. They showed that the K-norm mechanism using the unique norm induced by these constraints can be sampled efficiently and yields lower error than generic ℓ_p-sensitivity-based additive noise. This is because the tightest possible ℓ_p sensitivity bound $\|T(X) - T(X')\|_1 \leq k^{1/p}$ is loose in the sense of needlessly protecting points like $(k^{1/p}, 0, \ldots, 0)$ that the problem constraints rule out. In contrast, the unique norm protects exactly the set of possible points.

A separate line of work has focused on *discrete* additive noise. Motivated by potential vulnerabilities in the floating-point accuracy of continuous mechanisms (Mironov, 2012) and the discreteness of many real-world datasets, discrete analogues of both Laplace (Ghosh et al., 2009) and Gaussian (Canonne et al., 2020) have been developed. Depending on the level of privacy desired, these mechanisms achieve moderately better utility than their continuous counterparts and are at least typically no worse.

We join these two approaches to study problem-specific discrete noise mechanisms.

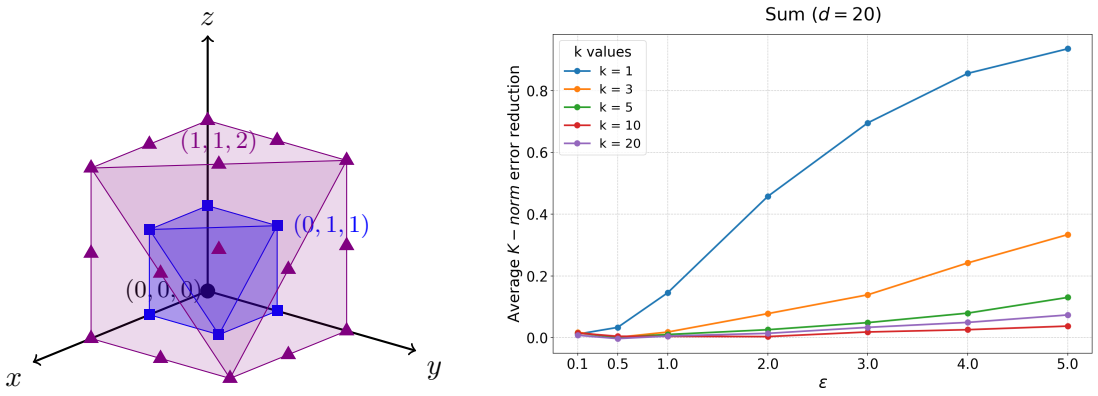

Figure 1: **Left**: An illustration of the first three "ripples" L_0 (black circular vertices), L_1 (blue rectangular vertices), and L_2 (purple triangular vertices) of $\mathcal{R}_{\text{SUM}}$ with $d = 3$ and $k = 2$, restricted to the positive orthant. The probability mass on a vertex in L_n is proportional to $e^{-\varepsilon n}$. **Right**: A plot of the error reduction achieved by $\mathcal{R}_{\text{SUM}}$ mechanism over the Sum K-norm mechanism at $d = 20$; for example, a value of 0.2 means our mechanism adds 20% less error (see Section 7).

1.1. Contributions

We first describe a general discrete noise mechanism, the *ripple mechanism*, whose level sets are discrete "ripples" arising from repeated Minkowski sums over a sensitivity set (see Figure 1 for an illustration). As grounding examples, we consider discrete analogues of the three problems studied by Joseph and Yu (2024): Sum, Count, and Vote. Throughout, the statistic is a sum over points in the database, with different problem-specific data domains:

1. **Sum.** Each data point lies in $\{-1, 0, 1\}^d$ and has at most k nonzero entries.
2. **Count.** Each data point lies in $\{0, 1\}^d$ and has at most k 1s.
3. **Vote.** Each data point x_i is a permutation of $(0, 1, \ldots, d-1)$.

It is easy to specify an instance of the ripple mechanism for each of the three problems above; the technical work arises in sampling them efficiently. Indeed, as the n^{th} "ripple" of an instance is defined by the Minkowski sum over n copies of the original sensitivity set, it is not obvious that the resulting distribution can be sampled in polynomial time at all.

As was the case for the continuous K-norm mechanisms for these problems (Joseph and Yu, 2024), each sampler will rely on an individual analysis of the combinatorial structures arising from the specific data domain. Additionally, and perhaps surprisingly, the techniques used to study these structures have little overlap with those of their continuous counterparts. Summaries of our sampler guarantees appear below. Since the samplers all use reservoir sampling, we bound their expected runtimes, assuming $\varepsilon = O(1)$. We shorthand the Sum, Count, and Vote ripple mechanisms respectively as $\mathcal{R}_{\text{SUM}}$, $\mathcal{R}_{\text{COUNT}}$, and $\mathcal{R}'_{\text{VOTE}}$.[1]

Theorem 1 (Section 4) *$\mathcal{R}_{\text{SUM}}$ can be sampled in expected time $O(d^3k^2)$.*

1. We use $\mathcal{R}'_{\text{VOTE}}$ rather than $\mathcal{R}_{\text{VOTE}}$ because our mechanism will be a slight modification of the true Vote ripple mechanism. See Section 6 for details.

Theorem 2 (Section 5) *$\mathcal{R}_{\text{COUNT}}$ can be sampled in expected time $O(d^4k^2)$.*

Theorem 3 (Section 6) *$\mathcal{R}'_{\text{VOTE}}$ can be sampled in expected time $O(d^2 \log(d))$.*

We primarily evaluate the utility of these mechanisms using simulations. We measure error using the norm induced by the sensitivity set (as done by both Geng et al. (2015) and Kulesza et al. (2025); see Section 2 for details) and take the K-norm mechanism as the baseline for each problem. For Sum and Count, our mechanisms achieve the largest improvement in the sparse-contribution setting where $k \ll d$, and for Vote, we find improvement across most parameter settings. A plot of the simulations for Sum appears in Figure 1, and a full discussion appears in Section 7.

Our final results investigate when the ripple mechanism is optimal among pure DP and discrete additive noise mechanisms (Section 8). We identify a sensitivity set for which the ripple mechanism is provably suboptimal (Section 8.1) and then establish a general set of conditions on the sensitivity polytope under which the ripple mechanism is optimal (Section 8.2), one of which is the well-studied normalized matching property from polytope geometry (and other fields). As a consequence of this result, $\mathcal{R}_{\text{SUM}}$ (and $\mathcal{R}_{\text{COUNT}}$) is optimal at $k = 1$ for arbitrary d, generalizing previous optimality results for the 1-dimensional geometric mechanism (Ghosh et al., 2009; Geng and Viswanath, 2014).

1.2. Related Work

Ripple mechanisms can be viewed as instances of a general class of pure DP mechanisms — including such canonical mechanisms as the exponential mechanism (McSherry and Talwar, 2007) — in which the probability of an output y on database X is proportional to $e^{-d(X,y)}$ where $d(X, y)$ is the number of changes to X needed to make y a "best answer". We view the primary contributions of this paper as the efficient samplers and utility results rather than the general ripple mechanism itself.

Other discrete private mechanisms include binomial (Dwork et al., 2006a) and Skellam (Agarwal et al., 2021) noise. Both mechanisms add more error than discrete Laplace and Gaussian noise, but they are closed under summation and thus well-suited to distributed settings that obtain privacy by combining individually privatized messages. Geng and Viswanath (2014) constructed the optimal discrete mechanism for a specific one-dimensional sensitivity set, which we discuss in Section 8.

A discussion of the relationship between the K-norm mechanism and other private mechanisms appears in the work of Joseph and Yu (2024). At a high level, their work (and ours) constructs efficient mechanisms tailored exactly to specific problems, whereas previous work is either significantly slower or approximate. Subsequent work (Kulesza et al., 2025) combined the K-norm and staircase (Geng et al., 2015) mechanisms and showed that these staircase K-norm mechanisms are optimal when error is measured using the problem's underlying norm. However, the gap between staircase K-norm and K-norm mechanisms is negligible outside the low-privacy regime ($\varepsilon \approx d$).

2. Preliminaries

Definition 4 (Dwork et al. (2006b)) *Databases X, X' from data domain $\mathcal{X}$ are* neighbors $X \sim X'$ *if they differ in the presence or absence of a single record. A randomized mechanism $\mathcal{M} : \mathcal{X} \to \mathcal{O}$ is* ε-differentially private *(DP) if for all $X \sim X' \in \mathcal{X}$ and any $S \subseteq \mathcal{O}$, $\mathbb{P}_{\mathcal{M}}[\mathcal{M}(X) \in S] \leq e^{\varepsilon}\mathbb{P}_{\mathcal{M}}[\mathcal{M}(X') \in S]$.*

Definition 5 (Kattis and Nikolov (2017); Awan and Slavković (2021)) *The* sensitivity space *of statistic T and data domain $\mathcal{X}$ is $S(T, \mathcal{X}) = \{T(X) - T(X') \mid X \sim X' \text{ and } X, X' \subset \mathcal{X}\}$. When $S(T, \mathcal{X})$ is discrete, we call it a* sensitivity set.

For each problem in this paper, the statistic is simply a sum over points in the database, but the underlying data domain $\mathcal{X}$ varies.[2] The resulting sensitivity set is straightforward.

Lemma 6 *Define statistic $\sum(X) = \sum_{x \in X}$. Then $S(\sum, \mathcal{X}) = \pm\mathcal{X}$.*

We measure both empirical (Section 7) and theoretical (Section 8) error of the ripple mechanisms using the sensitivity polytope norms induced by each problem. These are identical to the norms used by Joseph and Yu (2024) for their K-norm mechanisms, and we adapt their exposition.

Definition 7 *The* sensitivity polytope $P = CH(S(T, \mathcal{X}))$ *is the convex hull of the sensitivity set.*

Note that P contains the origin since sensitivity spaces are symmetric.

Lemma 8 *If a sensitivity polytope P is bounded, absorbing (for every $u \in \mathbb{R}^d$, there exists $c > 0$ such that $u \in cP$), and symmetric around 0 ($u \in P \Leftrightarrow -u \in P$), then the function $\|\cdot\|_P \colon \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ given by $\|u\|_P = \inf\{c \in \mathbb{R}_{\geq 0} \mid u \in cP\}$ is a norm, and we call it the norm induced by P.*

3. Ripple Mechanisms

The structure and analysis of the general ripple mechanism is simple; the main technical work of this paper is constructing and efficiently sampling its problem-specific instances. Given statistic T and database X from domain $\mathcal{X}$, the mechanism outputs $T(X) + Z$ where Z is drawn from a distribution characterized by level sets (i.e., ripples) induced by $S(T, \mathcal{X})$. The level sets arise from Minkowski sums of $S(T, \mathcal{X})$.

Definition 9 *The* Minkowski sum *of two sets X, Y is the set $X \boxplus Y = \{x + y : x \in X, y \in Y\}$.*

Definition 10 *Given statistic T and domain $\mathcal{X}$, define $L_0^{\leq} = L_0 = \{0\}$. For $n \geq 1$, define $L_n^{\leq} = L_{n-1}^{\leq} \boxplus S(T, \mathcal{X})$ and the n^{th}-level set is $L_n = L_n^{\leq} - \cup_{i=0}^{n-1} L_i^{\leq}$.*

If $0 \in S(T, \mathcal{X})$, then we always have $L_{n-1}^{\leq} \subseteq L_n^{\leq}$ and so $L_n = L_n^{\leq} - L_{n-1}^{\leq}$. This applies to both Sum and Count, but it does not apply to Vote, where the level sets have more complicated geometry. The general mechanism outputs a point from from a level set with probability proportional to its level set index n.

Definition 11 *Given privacy parameter $\varepsilon > 0$, statistic T, domain $\mathcal{X}$, and database X, define the* ripple mechanism *$R(T, \mathcal{X}, X)$ to output $T(X) + Z_T$ where Z_T is a random variable with support $\cup_{i=0}^{\infty} L_i$ and mass $f_Z(y) \propto e^{-\varepsilon \cdot g(y)}$ where $g(y) = n$ if $y \in L_n$ and $g(y) = +\infty$ otherwise.*

It is easy to show that the ripple mechanism is ε-DP. A proof appears in Section 9 (Lemma 59).

2. Note, however, that the ripple mechanism can still be defined for general statistics T.

3.1. Sampling

All of our applications of the ripple mechanism use a general reservoir sampling approach. The proof is straightforward and can be found in Section 9.

Lemma 12 *Suppose we have an instance $R(T, \mathcal{X}, X)$ of the ripple mechanism with oracle access to compute arbitrary $|L_n|$ as well as known $N = \sum_{i=0}^{\infty} e^{-\varepsilon i}|L_i|$. Define $\{w_i\}_{i=0}^{\infty}$ by $w_i = \frac{e^{-\varepsilon i}|L_i|}{N}$. Let S be the random variable for the point obtained by flipping a sequence of coins with bias $\{b_n\}_{n=0}^{\infty}$ where $b_n = \frac{w_n}{1-\sum_{i=0}^{n-1} w_i}$, then uniformly sampling L_n for the first index n whose coin flip comes up heads. Then $S \sim Z_T$, as defined in Definition 11.*

The technical work for each mechanism therefore consists of 1) computing the normalizing constant N, and 2) showing how to compute $|L_n|$ and uniformly sample from L_n, for arbitrary n. Once these components are in place, we apply Lemma 12.

For all three mechanisms, we compute the normalizing constant N using Z-transforms.

Definition 13 *Define the* unilateral Z-transform *of a sequence $\{x_n\}_{n=0}^{\infty}$ by $\mathcal{Z}(x_n) = \sum_{n=0}^{\infty} x_n z^{-n}$. We will write $\mathcal{Z}(x_n)[z_0]$ to denote evaluating $\mathcal{Z}(x_n)$ at $z = z_0$.*

It is easy to check that $N = \mathcal{Z}(|L_n|)[e^{\varepsilon}]$. Although $\mathcal{Z}(|L_n|)[e^{\varepsilon}]$ is an infinite series, in each problem we will show that it can be reduced to a finite one that can be evaluated efficiently.

4. Sum Ripple Mechanism

We first recall the data domain $\mathcal{X}_{\text{SUM}}$ from Section 1: each data point lies in $\{-1, 0, 1\}^d$ and has at most k nonzero entries. By Lemma 6, $S(\sum, \mathcal{X}_{\text{SUM}}) = \mathcal{X}_{\text{SUM}}$. The proof of this (and other results in this section) appears in Section 10.

4.1. Computing N

We start with some results about the different "ripples" L_n. Each orthant of $\mathcal{X}_{\text{SUM}}$ is identical, so we start with the positive orthant, though some care is necessary to properly account for boundary points. Let O_+ be the positive orthant in $\mathbb{R}^d$, i.e. the space of all non-negative linear combinations of the standard bases vectors. We first describe the lattice points in $L_n \cap O_+$ with linear constraints. The points in other orthants are symmetric.

Lemma 14 *$L_n^{\leq} \cap O_+$ is the set of points that can be written as the summation of n binary vectors in $\{0, 1\}^d$ each with support of size at most k.*

Notice that every point in L_n can be mapped to O_+ by taking the absolute value of each coordinate. This map induces equivalence classes of points where the points in $L_n \cap O_+$ are representatives of each class. In particular, if $v \in L_n \cap O_+$ has support size s, then v represents a class of size 2^s since we can independently set the sign of each of the coordinates in the support. The following lemmas characterize the points in $L_n \cap O_+$.

Lemma 15 *$L_n \cap O_+$ is the set of lattice points $v \in O_+$ where $\max(\lceil \frac{\|v\|_1}{k} \rceil, \|v\|_\infty) = n$.*

Corollary 16 *$L_n^{\leq} \cap O_+$ is the set of lattice points $v \in O_+$ where $\max(\lceil \frac{\|v\|_1}{k} \rceil, \|v\|_\infty) \leq n$.*

The definition below further partitions the point set $L_n \cap O_+$ according to the size of support.

Definition 17 *Fix $s \in \{0, ..., d\}$. Define $L_{n,s} \subset L_n \cap O_+$ to be the points with support size s, and define $L_{n,s}^{\leq} \subset L_n^{\leq} \cap O_+ = \cup_{i=0}^{n} L_{i,s}$. Similarly, for $J \subset \{1, ..., d\}$ where $|J| = s$ define $L_{n,s,J} \subset L_{n,s}$ to be the subset of points with support exactly at the indices of J, and define $L_{n,s,J}^{\leq} = \cup_{i=0}^{n} L_{i,s,J}$.*

Notice that, $|L_{n,s,J}|$ is a constant for all J such that $|J| = s$, thus for the sake of computing cardinality one can assume $J = [s]$. The following result shows that we can compute N using a Z-transform based on $|L_{n,s,[s]}^{\leq}|$.

Lemma 18 *The normalizing constant $N = \sum_{n \geq 0} |L_n| e^{-n\varepsilon}$ is equal to $\sum_{s=0}^{d} 2^s (1-z) \binom{d}{s} \mathcal{Z}(|L_{n,s,[s]}^{\leq}|)[z^{-1}]$ evaluated at $z = e^{-\varepsilon}$.*

In order to compute $\mathcal{Z}(|L_{n,s,[s]}^{\leq}|)$, we will use *Ehrhart polynomials* to count $|L_{n,s,[s]}^{\leq}|$ and use known results about h^* polynomials to reduce this infinite series into a closed form finite sum.

Definition 19 *Let $X \in \mathbb{R}^d$ be a convex polytope with integral vertices. The* Ehrhart polynomial *$E(X, t)$ for any positive integer t is a polynomial in t that counts the number of lattice points (i.e., points in $\mathbb{Z}^d$) in tX. The h^** polynomial *of X is defined as $E^*(X, z) = (1 - z)^{d+1} \sum_{t \geq 0} E(X, t) z^t$.*

We will show that $L_{n,s,J}^{\leq}$ is congruent to the lattice points in the set $\{v \in (0, n]^s : 0 < \sum_i v_i \leq nk\}$ which in turn can be decomposed as the disjoint union of a certain number of shapes called *open hypersimplices*. This lets us import results by Han and Josuat-Vergès (2016) on the h^* polynomial of the open hypersimplex. Consequently, we will show that we can express the $\mathcal{Z}(|L_{n,s,[s]}^{\leq}|)$ series in terms of the h^* polynomial of open hypersimplices $\mathcal{A}_{d,k}^{r} = \{v \in [0, r)^d : k - 1 \leq \sum_i v_i < k\}$.

Lemma 20 *For any $s \in \{0, ..., d\}$, the number of lattice points in $L_{n,s,[s]}^{\leq}$ is equal to the number of lattice points in $\cup_{i=s}^{s-k+1} n\mathcal{A}_{s,i}^{1}$.*

By the definition of Erhart polynomial, the number of lattice points in $n\mathcal{A}_{s,i}^{1}$ is $E(\mathcal{A}_{s,i}^{1} n)$, so

$$\mathcal{Z}(|L_{n,s,[s]}^{\leq}|)[z^{-1}] = \sum_{n \geq 0} |L_{n,s,[s]}^{\leq}| z^n = \sum_{n \geq 0} \sum_{i=s-k+1}^{s} E(\mathcal{A}_{s,i}^{1}, n) z^n = (1-z)^{-s-1} \sum_{i=s-k+1}^{s} E^*(\mathcal{A}_{s,i}^{1}, z).$$

Plugging this into Lemma 18 yields N. The h^* polynomial $E^*(\mathcal{A}_{s,i}^{1}, z)$ was computed by Han and Josuat-Vergès (2016). See Section 10 for more details on its closed-form expression.

Corollary 21 *The normalizing constant N can be computed in time $O(d^3)$.*

4.2. Computing $|L_n|$ and Sampling Uniformly From L_n

Recall from Lemma 12 that each step of reservoir sampling requires computing a "heads" probability $w_n = \frac{e^{-\varepsilon n}|L_n|}{N}$. The previous section showed how to compute N, so the next step is computing $|L_n|$. We have already partitioned the set $L_n \cap O_+$ into disjoint subsets $L_{n,s,J}$ where the support index set is J with size s. Given support size s, L_n contains $2^s\binom{d}{s}$ disjoint subsets of cardinality $|L_{n,s,J}|$, up to changing the signs of s non-zero coordinates and picking an index set of size s from $[d]$. Thus $|L_n| = \sum_{s=0}^{d} 2^s\binom{d}{s}|L_{n,s,J}|$, and it suffices to compute the different $|L_{n,s,J}|$.

Definition 22 *Define $X_J = \{v \in L_{n,s,J} \mid \text{some } v_j = n\}$, i.e., the set of points in $L_{n,s,J}$ with ℓ_∞ norm n, and $X_J^c = L_{n,s,J} - X_J$. For $1 \le i \le k$, define $X_{J,i} = \{v \in L_{n,s,J} \mid$ exactly i coordinates $v_j = n\}$, so $X_J = \cup_{i=1}^{k} X_{J,i}$.*

Note that the upper limit of $\cup_{i=1}^{k} X_{J,i}$ does not extend past k since $\|v\|_1 \le nk$ for any $v \in L_{n,s,J}$. We will compute $|L_{n,s,J}| = |X_J| + |X_J^c|$ by expressing both summands in terms of certain quantities related to ordered partitions of integers.

Definition 23 *Let $C(d, t, m)$ be the number of ordered sequences of d integers in $\{1, \ldots, m\}$ that sum to t, and let $C([a, b], [c, d], e)$ denote the the 2-dimensional table of entries $\{C(x, y, e) \mid a \le x \le b, c \le y \le d\}$.*

It is possible to efficiently compute the relevant entries of C using dynamic programming.

Claim 24 *We can compute $C([0, d], [0, nk], n - 1)$ in $O(dnk)$.*

Given C, we can compute $|L_n|$ by expressing its components in terms of C entries.

Claim 25 *After pre-computing $C([0, d], [0, nk], n - 1)$, we can compute:*

- *$|X_J|$ in time $O(nk^2)$ using the equation $|X_{J,i}| = \binom{s}{i}\sum_{t=0}^{nk-ni} C(s - i, t, n - 1)$*
- *$|X_J^c|$ in time $O(k)$ using the equation $|X_J^c| = \sum_{i=0}^{k-1} C(s, nk - i, n - 1)$*
- *$|L_{n,s,J}|$ and $|L_{n,s}|$ in time $O(nk^2)$ using $|L_{n,s,J}| = |X_J| + |X_J^c|$ and $|L_{n,s}| = \binom{d}{s}|L_{n,s,J}|$*
- *$|L_n|$ in time $O(dnk^2)$ using $|L_n| = \sum_{s=0}^{d} 2^s|L_{n,s}|$.*

The last step in our sampler is uniformly sampling the layer L_n whose coin flip comes up "heads". We follow the decomposition logic of L_n above, sampling an $L_{n,s,J}$ and then X_J or X_J^c.

If we sample X_J, then we use the first bullet point of Claim 25 to sample a set $X_{j,i}$. Finally, we sample a point from $X_{J,i}$ by observing that the sum of the $s - i$ coordinates bounded by $n - 1$ is $t \in [0, nk - ni]$ and that the probability of selecting each t is proportional to $C(s - i, t, n - 1)$. Once we select t, we can step through the recursive computation of $C(s - i, t, n - 1)$ to iteratively construct a sequence of $s - i$ integers in $[n - 1]$ summing to t. Once such a sequence v is constructed, the sample of $L_{n,s,J}$ is then formed by starting with the all 0s vector of length d, setting s of the J indices to n uniformly at random, and embedding v into the remaining indices of J in left to right order. If we instead sample X_J^c, then we use the second bullet point of Claim 25 and then use a sequence construction similar to the one used with X_J.

Claim 26 *We can uniformly sample a point from L_n in time $O(dnk^2)$.*

4.3. Sampler Runtime

We start with a general bound on the expected hitting time of reservoir sampling.

Lemma 27 *In the ripple mechanism, if $\mathcal{Z}(|L_n|)[z]$ can be expressed as a summation of terms of bounded power of z and $1 - z^{-1}$*

$$\mathcal{Z}(|L_n|)[z] = \sum_{s=0}^{d} \sum_{w=0}^{d+1} B_{s,w} z^{-w} (1 - z^{-1})^{-s}$$

where $B_{s,w}$ are non-negative coefficients, then the expected hitting time T of the reservoir sampling is bounded as $\mathbb{E}[T] = O(\frac{d}{1-e^{-\varepsilon}})$.

We apply Lemma 27 to $\mathcal{R}_{\text{SUM}}$ (and will do this for all three ripple mechanisms).

Corollary 28 *Let T be the hitting time random variable for reservoir sampling for Sum. Then $\mathbb{E}[T] = O(\frac{d}{1-e^{-\varepsilon}})$.*

The overall runtime combines Corollary 28 with the runtimes from the previous section.

Corollary 29 *When $\varepsilon = O(1)$, the expected running time of $\mathcal{R}_{\text{SUM}}$ is $O(d^3k^2)$.*

5. Count Ripple Mechanism

The data domain $\mathcal{X}_{\text{COUNT}}$ consists of data points lying in $\{0,1\}^d$ with at most k nonzero entries. By Lemma 6, $S(\sum, \mathcal{X}_{\text{COUNT}}) = \pm\mathcal{X}_{\text{COUNT}}$. Proofs for the results in this section appear in Section 11.

5.1. Computing N

In Count, we use $\{K_n\}_{n=0}^{\infty}$ instead of $\{L_n\}_{n=0}^{\infty}$ for layer variables to avoid naming collisions when using results from Sum.

Definition 30 *Define $K_0 = \{0\}$. Define B_k be the set of binary vectors with at most k ones. Define $K_1 = B_k \cup \{-v : v \in B_k\}$. Recall that K_n is the set of sums of n points from K_1 that cannot be expressed as a sum of fewer points from K_1. Given a sign vector $J \in \{-1,0,1\}^d$, let $K_{n,J} \subset K_n$ be the set of points $v \in K_n$ such that $(\mathsf{sign}(v_1), ...\mathsf{sign}(v_n)) = J$.*

This partitions each K_n into sets $K_{n,J}$ indexed by sign vectors. Lemma 32 expresses these $K_{n,J}$ sets in terms of $L_{n,s,X}$ sets from Sum. In particular, a point $p \in K_{n,J} \Leftrightarrow p = x - y$ where $x \in L_{a,p,P}$ and $y \in L_{n-a,m,M}$ where P is the positive support indices of J, M is the negative support indices of J, and $a \in [0,n]$. This will enable us to reuse many of the results from our Sum analysis.

Definition 31 *Given sign vector $J \in \{-1,0,1\}^d$, define the (p,m)* signature *of J to be the pair of integers such that J has exactly p positive entries and m negative entries. If $v \in K_{n,J}$ where J has signature (p,m), we will also say that v has sign vector J and v has signature (p,m). Define $J_{p,m} \subset \{-1,0,1\}^d$ to be the set of sign vectors with signature (p,m). Define $K_{n,p,m} \subset K_n$ to be the subset of points with signature (p,m).*

We can now give the exact decomposition of each $K_{n,J}$.

Lemma 32 *Suppose $J \in J_{p,m}$. Let P be the positive support indices of J, let M be the negative support indices of J. Then we have the internal direct sum decomposition $K_{n,J} = \sqcup_{a=0}^{n}(L_{a,p,P} \oplus -L_{n-a,m,M})$ where $L_{a,p,P}, L_{n-a,m,M}$ are defined in Definition 17 and $-L_{n-a,m,M}$ denotes the set $\{-v : v \in L_{n-a,m,M}\}$.*

This decomposition also yields the desired $|K_n|$.

Corollary 33 *Suppose $J \in J_{p,m}$. Let P be the positive support indices of J, let M be the negative support indices of J. Then $|K_{n,J}| = \sum_{a=0}^{n} |L_{a,p,P}||L_{n-a,m,M}|$, $|K_{n,p,m}| = \binom{d}{p}\binom{d-p}{m}|K_{n,J}|$, and $|K_n| = \sum_{p=0}^{d}\sum_{m=0}^{d-p}|K_{n,p,m}|$.*

Recall that the normalizing constant N can be expressed as $N = \mathcal{Z}(|K_n|)$ evaluated at $z = e^{\varepsilon}$. Roughly speaking, we will use the direct sum decomposition $K_{n,J} = \sqcup_{a=0}^{n}(L_{a,p,P} \oplus -L_{n-a,m,M})$ to express the power series $\mathcal{Z}(|K_n|)[z^{-1}]$ as the convolution of the two power series $\mathcal{Z}(|L_{n,p,P}|)[z^{-1}]$ and $\mathcal{Z}(|L_{n,m,M}|)[z^{-1}]$ from the Sum section.

Lemma 34 *The normalizing constant $N = \sum_{n\geq 0}|K_n|e^{-n\varepsilon} = \sum_{p=0}^{d}\sum_{m=0}^{d-p}\sum_{n\geq 0}|L_{n,p,m}|e^{-n\varepsilon}$, and N can be computed in time $O(d^3)$.*

5.2. Computing $|K_n|$ and Sampling Uniformly From K_n

We start by computing the collections $\{|K_{n,p,m}|\}_{p+m=0}^{d}$ and $\{|L_{a,p,[p]}|\}_{p=0,a=0}^{d,n}$.

Lemma 35 *We can compute both $\{|K_{n,p,m}|\}_{p+m=0}^{d}$ and $\{|L_{a,p,[p]}|\}_{p=0,a=0}^{d,n}$ in time $O(dn^2k^2) + O(d^2n)$.*

These collections will be used both to compute the $|K_n|$ for reservoir sampling and to uniformly sample from K_n.

Lemma 36 *After pre-computing $\{|K_{n,p,m}|\}_{p+m=0}^{d}$, we can compute $|K_n|$ in $O(d^2)$.*

The overall strategy for sampling K_n is as follows. First, we will sample signature (p, m) according to the weights $\{|K_{n,p,m}|\}_{0\leq p+m\leq d}$, which will specify a single $J_{p,m}$. Second, we uniformly sample some $J \in J_{p,m}$ with positive support P and negative support M, which will specify a single $K_{n,J}$. As $K_{n,J} = \sqcup_{a=0}^{n}(L_{a,p,P} \oplus -L_{n-a,m,M})$, we next sample some index a with weights proportional to $|(L_{a,p,P} \oplus -L_{n-a,m,M})|$. Finally, we sample $v_p \in L_{a,p,P}$ and $v_m \in L_{n-a,m,M}$ and define vector $v \in \mathbb{Z}^d$ to be the vector with support J whose P indices contains v_p and whose M indices contains $-v_m$.

Lemma 37 *After pre-computing $\{|K_{n,p,m}|\}_{0\leq p+m\leq d}$ and $\{|L_{a,p,[p]}|\}_{p=0,a=0}^{d,n}$, we can sample a point uniformly from K_n in time $O(dnk)$.*

5.3. Runtime

We again use Lemma 27 to show that the expected hitting time for $\mathcal{R}_{\text{COUNT}}$'s reservoir sampling step is $O(d[1 - e^{-\varepsilon}]^{-1})$ (Lemma 66). The final result collects the runtimes of the subroutines.

Corollary 38 *The expected running time of $\mathcal{R}_{\text{COUNT}}$ is $O(d^4k^2)$ when $\varepsilon = O(1)$.*

6. Vote Ripple Mechanism

Recall the data domain $\mathcal{X}_{\text{VOTE}}$ from Section 1: each data point x_i is a permutation of $(0, 1, \ldots, d-1)$. By Lemma 6, $S(\sum, \mathcal{X}_{\text{VOTE}}) = \pm\mathcal{X}_{\text{VOTE}}$. Unlike the previous two mechanisms, our ripple mechanism for Vote will start from an enlarged version of this true sensitivity set, consisting of the lattice points in $\pm CH(\mathcal{X}_{\text{VOTE}})$, where $CH(\cdot)$ denotes the convex hull. At a high level, this sacrifices some of the exactness of the ripple mechanism—each layer is "bigger than it needs to be"—for a tractable analysis of the layers that will eventually enable efficient sampling. We note that the Vote K-norm mechanism constructed by Joseph and Yu (2024) includes the same extra points, since it used the norm defined by unit ball $CH(\Pi_d \cup -\Pi_d)$.

We first relate our mechanism to the well-studied *standard permutohedron*.

Definition 39 *Let* $\Pi_d = CH(S_d(\{0, 1, ..., d-1\})) = CH(\mathcal{X}_{\text{VOTE}})$ *denote the* standard permutohedron.

We now define the level sets P_n as the lattice points of convex sets $\mathcal{L}_n$. The resulting P_1 will contain $S(\sum, \mathcal{X}_{\text{VOTE}})$, which suffices for the privacy proof of Lemma 59 to still apply. We will also be able to show how to decompose $\mathcal{L}_n^{\leq}$, and this will give a similar decomposition for $P_n^{\leq}$ that will be key to our efficient sampler.

Definition 40 *Define* $\mathcal{L}_0^{\leq} = \{0\}$. *For* $n \geq 1$, *define* $\mathcal{L}_n^{\leq} = \cup_{b \in B}(\boxplus_{i=1}^{n} b_i \Pi_d)$ *where* B *is the set of vectors in* $\{-1, 1\}^n$ *which are some -1's (possibly 0) followed by all 1s. For* $n \in \{0, 1\}$, *define* $\mathcal{L}_n = \mathcal{L}_n^{\leq}$. *For* $n \geq 2$, *define* $\mathcal{L}_n = \mathcal{L}_n^{\leq} - \cup_{i=0}^{n-1} \mathcal{L}_i^{\leq}$. *Define* $P_n^{\leq}$ *to be the lattice points in* $\mathcal{L}_n^{\leq}$ *and define* P_n *to be the lattice points in* $\mathcal{L}_n$.

It is immediate that P_1 is the set of all lattice points in $\Pi_d \cup -\Pi_d$, which contains $\pm\mathcal{X}_{\text{VOTE}}$. Since P_1 strictly contains L_1 as defined in Definition 10, we use P_n instead of L_n for the level sets of the Vote mechanism. However, the remaining structure and sampling of the ripple mechanism persists unchanged.

6.1. Computing N and $|P_n|$

We first examine the structure of $\mathcal{L}_n^{\leq}$ by first decomposing it (Lemma 70) into zonotopes arising from a certain matrix A.

Definition 41 *Let* A *be the* $d \times \binom{d}{2}$ *matrix whose columns are* $e_i - e_j$ *for* $1 \leq i < j \leq d$. *Let* $v = (0, 1, ..., d-1)$.

We then further decompose those zonotopes into paralellotopes (Lemma 74). This yields a similar decomposition of $P_n^{\leq}$ into lattice points in these paralellotopes (Corollary 75). One application shows that each point in $P_n^{\leq}$ is a sum of lattice points from Π_d and $-\Pi_d$.

Lemma 42 $P_n^{\leq}$ *is the set of lattice points that can be reached by summing* n *lattice points in* $\Pi_d \cup -\Pi_d$.

This is similar, to the property $L_n^{\leq} = L_{n-1}^{\leq} \boxplus S(T, \mathcal{X})$ described in Definition 10, though $P_n^{\leq}$ does not contain 0. In order to understand P_n, we then establish some properties relating different $P_n^{\leq}$. The proof of this result relies heavily on the aforementioned parallelotope decomposition.

Lemma 43 *$P^{\leq}_{n-2} \subset P^{\leq}_n$ for all $n \geq 2$. Moreover, $P^{\leq}_{n-1} \cap P^{\leq}_n = \emptyset$ for all $n \geq 1$. Consequently, we have the two nested towers $P^{\leq}_1 \subset P^{\leq}_3 \subset ...$ and $P^{\leq}_0 \subset P^{\leq}_2 \subset ...$ and $P_n = P^{\leq}_n - P^{\leq}_{n-2}$.*

We next show that we can count points in $P^{\leq}_n$ by counting forests on d vertices.

Definition 44 *Define Q_k to be the set of forests on d vertices $\{1, ..., d\}$ with k edges.*

At a high level, we use the decomposition of $\mathcal{L}^{\leq}_n$ into zonotopes (Lemma 70) to reduce the problem of computing $|P^{\leq}_n|$ to computing the Ehrhart polynomial of a certain zonotope involving A and then further reduce this (Lemma 80) to computing the GCD of certain minors of matrices arising from $iA \oplus -(n-i)A$. It turns out that each minor can be interpreted as a graph on d vertices, and some further work connects these graphs to the Q_k.

Lemma 45 *$|P^{\leq}_n| = (n+1)\sum_{k=0}^{d-1} |Q_k| n^k$ where $0^0 = 1$ for $n = 0$.*

The next step extends our work on $|P^{\leq}_n|$ to $|P_n|$. It mostly consists of repeated application of the reasoning from Lemma 45.

Lemma 46 *$|P_0| = |P^{\leq}_0| = 1$, $|P_1| = |P^{\leq}_1| = 2\sum_{k=0}^{d-1} |Q_k|$, and for $n \geq 2$, $|P_n| = \sum_{k=0}^{d-1} |Q_k|((n+1)n^k - (n-1)(n-2)^k)$.*

We can now combine these results to get an expression for N, the normalizing constant for our mechanism's output distribution.

Lemma 47 *$N = (1 - e^{-2\varepsilon}) Z(|P^{\leq}_n|)[e^{\varepsilon}]$ where*

$$\mathcal{Z}(|P^{\leq}_n|)[z^{-1}] = 1 + \frac{z}{(1-z)}\left(1 + \frac{1}{1-z}\right) + \sum_{k=1}^{d-1} |Q_k|(1-z)^{-k-2}\left[\sum_{j=0}^{k} \mathcal{A}(k+1, j) z^{j+1} + (1-z)\sum_{j=0}^{k-1} \mathcal{A}(k, j) z^{j+1}\right]$$

and $\mathcal{A}(k, j)$ is the Eulerian number defined as the number of permutations on $\{1, ..., k\}$ with j descents.

6.2. Sampling Uniformly From a Layer P_n

A few pieces of the sampler remain. First, the preceding results express various quantities of interest in terms of $|Q_k|$, the number of forests on d labelled vertices with k edges. We compute the $|Q_k|$ by computing a table F where $F_{d,m}$ is the number of forests on d labelled vertices with m components. We have $|Q_k| = F_{d,d-k}$, and F will be useful in other sampling steps as well.

Lemma 48 *We can compute the $F_{d,m}$ table for all $m \in \{1, ...d\}$ in $O(d^2 \log(d))$.*

We compute F by expressing the entries of F as coefficients of a generating function and then showing that each column can be computed as a convolution using FFTs in time $O(d \log(d))$.

Our final result verifies that, after computing F, each $|P_n|$ can be computed efficiently. It also describes how to sample P_n efficiently. The sampler relies on decomposing P_n into intersections with carefully indexed and weighted parallelotopes, as sketched in the previous section.

Lemma 49 *Given $F_{d,m}$ for $1 \leq m \leq d$, for any $n \geq 0$, we can compute $|P_n|$ in time $O(d)$ and sample a point uniformly from P_n in time $O(d^2 \log(d))$.*

6.3. Runtime

The main technical work in bounding the Vote ripple mechanism's expected runtime is bounding the expected number of reservoir sampling coin flips. We again use Lemma 27.

Lemma 50 *The hitting time random variable for reservoir sampling, T, satisfies $\mathbb{E}[T] = O(\frac{d}{1-e^{-\varepsilon}})$.*

The overall runtime simply applies Lemma 50 and observes that each failed coin flip takes time $O(d)$, while the single successful coin flip takes further time $O(d^2 \log(d))$ (Lemma 49).

Corollary 51 *When $\varepsilon = O(1)$, the expected running time of $\mathcal{R}_{\text{VOTE}}$ is $O(d^2 \log(d))$.*

7. Simulations

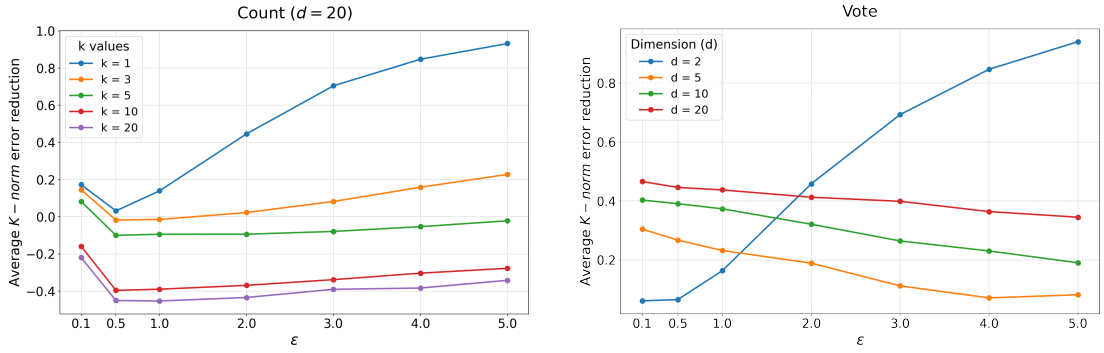

Figure 2: Relative norm error for Count and Vote (see Figure 1 for Sum). Letting E_K (E_R) denote the mean norm error of the K-norm (ripple) mechanism, the plot shows $\frac{E_K - E_R}{E_K}$ at various settings of ε. A positive value indicates lower error for the ripple mechanism. For Sum and Count, each line corresponds to a different k at a fixed $d = 20$. For Vote, each line is a different setting of d.

We compare $\mathcal{R}_{\text{SUM}}$, $\mathcal{R}_{\text{COUNT}}$, and $\mathcal{R}'_{\text{VOTE}}$ against their respective K-norm mechanisms from Joseph and Yu (2024). The staircase K-norm mechanism of Kulesza et al. (2025) dominates the K-norm mechanism, but this difference is only noticeable in the very low-privacy regime $\varepsilon \approx d$. Since we mostly do not consider this setting in our experiments, we use the simpler K-norm mechanism.

For each problem, we sample 5,000 mechanism outputs and measure average error using the norm induced by the convex hull of the sensitivity set (Definition 7). This is the same error used by Kulesza et al. (2025) as well as in our optimality results. We provide efficient algorithms for computing each of the norms in Section 13. Plots of the results appear in Figure 1 (Sum) and Figure 2 (Count and Vote). Experiment code can be found on Github (Google, 2026).

A few general trends emerge. $\mathcal{R}_{\text{SUM}}$ and $\mathcal{R}_{\text{COUNT}}$ add significantly less error than their counterpart K-norm mechanisms in the sparse-contribution setting where k is small. For $\mathcal{R}_{\text{SUM}}$, this advantage shrinks to near 0 as k approaches the dimension d; for $\mathcal{R}_{\text{COUNT}}$, the effect eventually reverses, and K-norm is superior for large k. Both mechanisms' relative performance generally increases with ε.

Turning to $\mathcal{R}_{\text{VOTE}}$, when $d \geq 5$ we see consistent error improvements across ε, and the improvements appear to increase with d. All but the $d = 2$ setting exhibit a slight decrease relative to the K-norm mechanism as ε increases. The unusual curve plotted at $d = 2$ is likely a consequence of its unusual sensitivity set. Unlike $d > 2$, the sensitivity set for $d = 2$ has no interior points, and is in fact equivalent to the Count sensitivity set at $k = 1$. Similar plots using $d = 10$ appear in Section 14.

We therefore suggest that $\mathcal{R}_{\text{SUM}}$ and $\mathcal{R}_{\text{VOTE}}$ are most useful across parameter settings, while $\mathcal{R}_{\text{COUNT}}$ is best in the sparse-contribution setting where k is small. We also note that, for a specific parameter setting, a practitioner can empirically test and identify the better of the ripple and K-norm mechanisms before interacting with the data at all.

8. Optimality of Ripple Mechanism

This section analyzes when the ripple mechanism is optimal among ε-DP additive noise mechanisms with range $\mathbb{Z}^d$.[3] A mechanism uses additive noise if, to compute statistic $T(X)$, it outputs $T(X)+Z$, where Z is sampled independently of $T(X)$, and the optimal mechanism minimizes $\mathbb{E}[\|Z\|_P]$.

Existing work on optimal additive noise mechanisms largely assumes that the mechanism output has support $\mathbb{R}^d$ and thus does not apply to our setting. One exception is a result of Geng and Viswanath (2014) obtained for a specific sensitivity set, which we discuss in the next section.

8.1. A Sub-Optimal Ripple Mechanism

As alluded to in Section 1, Geng and Viswanath (2014) considered the specific one-dimensional sensitivity set $S = \{-\Delta, -\Delta+1, \ldots, 0, \ldots, \Delta-1, \Delta\}$ for an integer $\Delta \geq 1$. When $\Delta = 1$, the ripple mechanism on S is identical to the geometric mechanism, which is optimal (Ghosh et al., 2009). However, when $\Delta \geq 2$, Geng and Viswanath (2014) show that a different "discrete staircase" mechanism is optimal. This implies that the ripple mechanism is not optimal for S in general.

8.2. Optimal Ripple Mechanisms

We provide a sufficient set of conditions under which the ripple mechanism is optimal. The first collection of conditions describes when the sensitivity polytope is *well-behaved*.

Definition 52 *Consider a sensitivity space $S(T, \mathcal{X})$ with associated sensitivity polytope P and ripple mechanism $\mathcal{R}_{T,\mathcal{X}}$. We say P is well-behaved if these satisfy:*

1. Self-contained*: $L_1^{\leq} = S(T, \mathcal{X}) = P \cap \mathbb{Z}^d$, which implies $O \in S(T, \mathcal{X})$.*
2. Generates the whole space*: each point in $\mathbb{Z}^d$ belongs to $L_n^{\leq}$ for some n.*
3. Integer-decomposition Property (IDP)*: for any integral $k \geq 1$, $(kP) \cap \mathbb{Z}^d = \boxplus_{i=1}^{k}(P \cap \mathbb{Z}^d)$.*
4. Reflexive*: every $x \in \mathbb{Z}^d$ lies in ∂kP for some integral k, where ∂ denotes the boundary/surface.*

A self-contained $S(T, \mathcal{X})$ has no "hole" in the sense that it contains every lattice point in its convex hull, including the origin O. Recall that this guarantees $L_{n-1}^{\leq} \subset L_n^{\leq}$, as the Minkowski sums for L_n can include 0. When P is IDP, we know that $L_n = nP \setminus (n-1)P \cap \mathbb{Z}^d$; hence the ripple mechanism assigns a probability $\mathbb{P}(x) \propto e^{-n\varepsilon}$ to it. Finally, if P is reflexive then for all $x \in L_n$, $\|x\|_P = n$.

Sensitivity polytopes are not in general well-behaved. The Vote polytope, for example, is not self-contained. However, the Sum and Count polytopes are well-behaved for all $k \leq d$.

We will show that a ripple mechanism is optimal when its polytope is well-behaved and its level sets satisfy the *Normalized Matching Property*. We define this property using adjacent level sets.

3. A similar analysis is possible for other discrete lattices, but we use $\mathbb{Z}^d$ for simplicity.

Definition 53 *In adjacent level sets L_{k-1} and L_k, if two lattice points $x \in L_{k-1}$ and $y \in L_k$ have $y - x \in S(T, \mathcal{X})$, then we say x, y are* neighbors.

We define the Normalized Matching Property using a graph G_k on adjacent level sets in which each node corresponds to a lattice point in L_{k-1} or L_k and there are edges between neighboring points.

Definition 54 *In the context above, we say that P's two adjacent level sets, L_{k-1} and L_k, have the Normalized Matching Property (NMP) if there exists a flow from L_{k-1} to L_k in graph G_k, such that every point in L_{k-1} has source of 1 unit flow, and every point in L_k is a sink with flow $\frac{|L_{k-1}|}{|L_k|}$; in other words the source flows can be distributed evenly among the sinks. If it holds for all integral k's, we say P has the NMP.*

This property is related to the fractional variant of *Hall's marriage theorem*, which arises in contexts across fields like combinatorics, graph theory and algebraic geometry. In *Sperner theory*, it is related to *Erhart unimodality* (Engel, 1997). It is also closely related to fundamental properties such as log-concavity of poset sizes and the LYM property in *poset theory*. Deriving general classes of level sets with the Normalized Matching Property is an open problem.

In our context, this property implies that the probability mass can be "averaged out" within each level set, and still preserve ε-DP.

Lemma 55 *If P has NMP, take any probability mass function $f(\cdot)$ that is ε-DP for P, and average the mass over every level set L_k: $\forall v \in L_k$, $f_{avg}(v) = \frac{\sum_{u \in L_k} f(u)}{|L_k|}$, the resulting $f_{avg}(\cdot)$ is still ε-DP.*

The proof of this result constructs a linear combination of the pair-wise ε-DP constraints, where the flow value on each edge is used as a coefficient for the corresponding constraint, to establish the implied ε-DP constraint on averaged masses.

Define the class $\mathcal{M}_{S(T,\mathcal{X}),\varepsilon}$ to be all ε-DP additive noise mechanisms with its support being $\mathbb{Z}^d$. We can take any ε-DP mechanism from $\mathcal{M}_{S(T,\mathcal{X},\varepsilon)}$, collect the probability mass within each level set L_n and distribute it evenly on all points in L_n to get another ε-DP mechanism whose probability on any $x \in \mathbb{Z}^d$ is always proportional to $e^{-\varepsilon \|x\|_P}$, while maintaining the same expected error value, where $\|x\|_P \in \mathbb{Z}^+$. The optimization problem is then reduced to 1D, with variables being the *constant* probabilities for each level set. The following theorem shows that it is optimal to have that probability decline by $e^{-\varepsilon}$ from every L_n to L_{n+1}.

Theorem 56 *For any problem whose convex hull of sensitivity space is a well-behaved polytope P with NMP, the ripple mechanism has minimum expected $\|\cdot\|_P$ error within class $\mathcal{M}_{S(T,\mathcal{X}),\varepsilon}$.*

Note that this optimality result covers all private discrete mechanisms with support on $\mathbb{Z}^d$. This can also be viewed an extension of the one-dimensional optimality result for the *geometric mechanism* when the sensitivity space $\Delta = \{\pm 1, 0\}$, proved by Geng and Viswanath (2014).

The following result shows that in the Sum problem, when $k = 1$, the sensitivity polytope has NMP, which implies strong optimality of the ripple mechanism in this case.

Lemma 57 *When the sensitivity polytope P is the ℓ_1 unit ball: $P = \{x \in \mathbb{Z}^d : \|x\|_1 \leq 1\}$, it has NMP.*

Since the Sum and Count sensitivity sets are identical at $k = 1$, we get the following corollary.

Corollary 58 *Let B_1^d be the ℓ_1 unit ball in $\mathbb{R}^d$. Then for the d-dimensional Sum and Count problems, when $k = 1$, the ripple mechanism is the $\|\cdot\|_{B_1^d}$-optimal discrete additive-noise ε-DP mechanism with support $\mathbb{Z}^d$.*

References

Naman Agarwal, Peter Kairouz, and Ziyu Liu. The skellam mechanism for differentially private federated learning. *Neural Information Processing Systems*, 2021.

Federico Ardila, Mariel Supina, and Andrés Vindas-Meléndez. The equivariant Ehrhart theory of the permutahedron. *Proceedings of the American Mathematical Society*, 2020.

Jordan Awan and Aleksandra Slavković. Structure and sensitivity in differential privacy: Comparing k-norm mechanisms. *Journal of the American Statistical Association*, 2021.

Clément L Canonne, Gautam Kamath, and Thomas Steinke. The discrete gaussian for differential privacy. *Neural Information Processing Systems (NeurIPS)*, 2020.

Munther Dahleh. Convolutions, Laplace & Z-Transforms, 2003.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, 2006a.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference (TCC)*, 2006b.

Richard Ehrenborg and Margaret Readdy. A poset view of the major index. *Advances in Applied Mathematics*, 2015.

Konrad Engel. *Sperner theory*. Cambridge University Press, 1997.

AR Forouzan. Region of convergence of derivative of z transform. *Electronics Letters*, 2016.

Quan Geng and Pramod Viswanath. The optimal mechanism in differential privacy. In *International Symposium on Information Theory (ISIT)*, 2014.

Quan Geng, Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The staircase mechanism in differential privacy. *Journal of Selected Topics in Signal Processing*, 2015.

Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. Universally utility-maximizing privacy mechanisms. In *Symposium on Theory of Computing (STOC)*, 2009.

Google. dp_ripple. `https://github.com/google-research/google-research/tree/master/dp_ripple`, 2026.

Guo-Niu Han and Matthieu Josuat-Vergès. Flag statistics from the Ehrhart h*-polynomial of multi-hypersimplices. *The Electronic Journal of Combinatorics*, 2016.

Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In *Symposium on the Theory of Computing (STOC)*, 2010.

W N Hsieh and Daniel J Kleitman. Normalized matching in direct products of partial orders. *Studies in Applied Mathematics*, 1973.

Matthew Joseph and Alexander Yu. Some Constructions of Private, Efficient, and Optimal K-Norm and Elliptic Gaussian Noise. In *Conference on Learning Theory (COLT)*, 2024.

Assimakis Kattis and Aleksandar Nikolov. Lower bounds for differential privacy from gaussian width. In *Symposium on Computational Geometry (SOCG)*, 2017.

Alex Kulesza, Ananda Theertha Suresh, and Yuyan Wang. General Staircase Mechanisms for Optimal Differential Privacy. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2025.

Frank ZK Li and Xunhao Liu. Combinatorial proofs on the joint distribution of descents and inverse descents. *arXiv preprint arXiv:2212.05307*, 2022.

Peter Luschny. Eulerian Polynomials, 2010.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Foundations of Computer Science (FOCS)*, 2007.

Ilya Mironov. On significance of the least significant bits for differential privacy. In *Computer and Communications Security (CCS)*, 2012.

Online Encyclopedia of Integer Sequences (OEIS). A105599, 2025.

Alexander Postnikov. Permutohedra, associahedra, and beyond. *International Mathematics Research Notices*, 2009.

Geoffrey C Shephard. Combinatorial properties of associated zonotopes. *Canadian Journal of Mathematics*, 1974.

Richard Stanley. A Zonotope Associated with Graphical Degree Sequences. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science, The Victor Klee Festschrift*, 1983.

Xiaodong Wang, Lei Wang, and Yingjie Wu. An Optimal Algorithm for Prufer Codes. *Journal of Software Engineering Applications*, 2009.

Günter M Ziegler. *Lectures on Polytopes*. Springer-Verlag, 1995.

9. Proofs From Section 3

Lemma 59 *The ripple mechanism is ε-DP.*

Proof Let X and X' be neighboring databases and fix some output y. Then $\mathbb{P}\left[R(T,X)=y\right] = \mathbb{P}\left[Z_T = y - T(X)\right]$. If $y - T(X) \notin \cup_{i=0}^{\infty} L_i$, then $\mathbb{P}\left[Z_T = y - T(X)\right] = 0$. By the definition of $S(T)$ (Definition 5), there exists $s \in S(T)$ such that $T(X) = T(X') + s$. Since $y - T(X) \notin \cup_{i=0}^{\infty} L_i$, neither is $y - T(X) + s$, so

$$0 = \mathbb{P}\left[Z_T = y - T(X) + s\right] = \mathbb{P}\left[Z_T = y - T(X')\right] = \mathbb{P}\left[R(T,X') = y\right].$$

If instead $y - T(X)$ is contained in some L_n, then $y - T(X') = y - T(X) + s$ is in L_{n-1}, L_n, or L_{n+1}. In any case, by Definition 11,

$$\frac{\mathbb{P}\left[R(T,X)=y\right]}{\mathbb{P}\left[R(T,X')=y\right]} = \frac{\mathbb{P}\left[Z_T = y - T(X)\right]}{\mathbb{P}\left[Z_T = y - T(X'\right]} = \frac{\mathbb{P}\left[Z_T = y - T(X)\right]}{\mathbb{P}\left[Z_T = y - T(X) + s\right]} \in [e^{-\varepsilon}, e^{\varepsilon}].$$

■

Lemma 12 *Suppose we have an instance $R(T, \mathcal{X}, X)$ of the ripple mechanism with oracle access to compute arbitrary $|L_n|$ as well as known $N = \sum_{i=0}^{\infty} e^{-\varepsilon i}|L_i|$. Define $\{w_i\}_{i=0}^{\infty}$ by $w_i = \frac{e^{-\varepsilon i}|L_i|}{N}$. Let S be the random variable for the point obtained by flipping a sequence of coins with bias $\{b_n\}_{n=0}^{\infty}$ where $b_n = \frac{w_n}{1-\sum_{i=0}^{n-1} w_i}$, then uniformly sampling L_n for the first index n whose coin flip comes up heads. Then $S \sim Z_T$, as defined in Definition 11.*

Proof For arbitrary i, every point in L_i has the same probability in both S and Z_T, so it suffices to show that each L_i has the same probability of being chosen in S and Z_T. We use induction, and the base case is immediate. If the inductive hypothesis holds through L_n, then $\mathbb{P}\left[S \in \cup_{i=0}^{n} L_i\right] = \sum_{i=0}^{n} w_i$. Thus

$$\mathbb{P}\left[S \in L_{n+1}\right] = (1 - \sum_{i=0}^{n} w_i) \cdot \frac{w_{n+1}}{1 - \sum_{i=0}^{n} w_i} = w_{n+1}.$$

■

10. Proofs From Section 4

This section provides the full details for $\mathcal{R}_{\text{SUM}}$. Sampler pseudocode appears in Algorithm 1.

Lemma 14 *$L_n^{\leq} \cap O_+$ is the set of points that can be written as the summation of n binary vectors in $\{0,1\}^d$ each with support of size at most k.*

Proof Suppose $v \in L_n^{\leq} \cap O_+$. Let $v(i)$ denote the ith coordinate of v. Write $v(i) = \sum_{j=1}^{n} w_j(i)$ where each $w_j \in L_1 \cup \{0\}$ and $w_j(i) \in \{-1, 0, 1\}$. We can replace pairs of coordinates $w_a(i)$ and $w_b(i)$ that equal 1 and -1 by two 0s without changing the value of v. When we've exhausted this replacement, $v(i)$ will be a sum of 0s and 1s since $v(i) \geq 0$. Let w_j' be the modified w_j vector and note that $w_j' \in \{0,1\}^d$. Then w_j' has support at most k since replacing coordinates of w_j by 0 only decreases the support size. Now, $v = \sum_{j=1}^{n} w_j'$ so v is indeed a sum of n vectors in $\{0,1\}^d$ each with support of size at most k. Conversely, a sum of n binary vectors in $\{0,1\}^d$ with support of size at most k is in $L_n^{\leq}$ by definition, and clearly also in $L_n^{\leq} \cap O^+$. ■

Lemma 15 *$L_n \cap O_+$ is the set of lattice points $v \in O_+$ where $\max(\lceil \frac{\|v\|_1}{k} \rceil, \|v\|_\infty) = n$.*

Proof For any $n \geq 1$, let the target set of lattice points be $M_n = \{v \in \mathbb{Z}^d \cap O_+ : \max(\lceil \frac{\|v\|_1}{k} \rceil, \|v\|_\infty) = n\}$. We show by strong induction on n that $L_n \cap O_+ = M_n$. In the base case, $L_0 \cap O_+ = \{0\} = M_0$ obviously holds. Suppose we have shown that it holds for L_i where $i < n$, we will prove the equivalence of the two sets by showing $L_n \cap O_+ \subseteq M_n$ and $M_n \subseteq L_n \cap O_+$.

To see that $L_n \cap O_+ \subseteq M_n$, pick any $v \in L_n \cap O_+$. By definition of L_n, v can be written as $v = \sum_{i=1}^{n} x_i$, where $x_i \in S(T)$, so each x_i must satisfy $\|x_i\|_1 \leq k$ and $\|x_i\|_\infty \leq 1$. By triangle inequality we must have $\max(\lceil \frac{\|v\|_1}{k} \rceil, \|v\|_\infty) \leq n$. However, if $\max(\lceil \frac{\|v\|_1}{k} \rceil, \|v\|_\infty) = m < n$ then $v \in L_m$ by induction, a contradiction because all level sets must be disjoint. Hence $v \in M_n$.

Next, we focus on showing $M_n \subseteq L_n \cap O_+$. Take any $v \in M_n$. By the induction assumption $v \notin L_m \cap O_+$ for any $m < n$ since by induction assumption $L_m \cap O_+ = M_m$, which means

$\max(\lceil \frac{\|v\|_1}{k} \rceil, \|v\|_\infty) = m < n$. We will show that $v = v' + x$ for some $v' \in L_{n-1} \cap O_+ = M_{n-1}$ and $x \in S(T)$. Without loss of generality, assume the values in vector v are descending in terms of indices: $v_1 \geq v_2 \geq \ldots \geq v_d \geq 0$. For index $i = 1, \ldots, d$, let $x_i = 1$ if both $v_i \geq 1$ and $i \leq k$; otherwise 0. Obviously $x \in S(T)$ since it only has 1 at indices at most k, so we only have to show $v' = v - x \in L_{n-1} \cap O_+$. Thus v' is non-negative since $v_i \geq x_i$ by construction so $v' \in O_+$.

Case 1: $\|v\|_\infty = n$. There are at most k of n's in v since $\|v\|_1 \leq kn$, so $x_i = 1$ where $v_i = n$, thus $\|v'\|_\infty = n-1$. To show $\|v'\|_1 \leq k(n-1)$, notice that, if v has support at least k, x has support k, hence $\|v'\|_1 = \|v\|_1 - k \leq k(n-1)$; on the other hand if v has support less than k, v' also has support less than k and each coordinate of v' is at most $n-1$, hence $\|v'\|_1 \leq k(n-1)$ always holds, showing $v' \in M_{n-1}$.

Case 2: $\lceil \frac{\|v\|_1}{k} \rceil = n$, and $\|v\|_\infty < n$. v must have support at least k; hence x has k of 1's, and $\|v'\|_1 = \|v\|_1 - k$, so $\lceil \frac{\|v'\|_1}{k} \rceil = \lceil \frac{\|v\|_1}{k} \rceil - 1 = n - 1$, showing $v' \in M_{n-1}$.

Hence we've proved that $M_n \subseteq L_n \cap O_+$.

■

Corollary 16 *$L_n^{\leq} \cap O_+$ is the set of lattice points $v \in O_+$ where $\max(\lceil \frac{\|v\|_1}{k} \rceil, \|v\|_\infty) \leq n$.*

Proof Recall that $L_n \cap O_+ = M_n = \{v : v \in O_+, \max(\lceil \frac{\|v\|_1}{k} \rceil, \|v\|_\infty) = n\}$. Then $L_n^{\leq} \cap O_+ = \cup_{i=0}^n L_i \cap O_+ = \cup_{i=0}^n M_i = \{v : v \in O_+, \max(\lceil \frac{\|v\|_1}{k} \rceil, \|v\|_\infty) \leq n\}$ ■

Lemma 18 *The normalizing constant $N = \sum_{n \geq 0} |L_n| e^{-n\varepsilon}$ is equal to $\sum_{s=0}^d 2^s (1-z) \binom{d}{s} \mathcal{Z}(|L^{\leq}_{n,s,[s]}|)[z^{-1}]$ evaluated at $z = e^{-\varepsilon}$.*

Proof We start with the formulation of N from Lemma 12.

$$\begin{aligned} N &= \sum_{n \geq 0} |L_n| e^{-\varepsilon n} \\ &= \sum_{s=0}^d 2^s \sum_{n \geq 0} |L_{n,s}| e^{-\varepsilon n} \\ &= \sum_{s=0}^d 2^s \mathcal{Z}(|L_{n,s}|)[e^{\varepsilon}]. \end{aligned}$$

Next, we will show that $\mathcal{Z}(|L_{n,s}|)[z^{-1}] = (1-z)\binom{d}{s} \mathcal{Z}(|L^{\leq}_{n,s,[s]}|)[z^{-1}]$.

$$
\begin{aligned}
Z(|L_{n,s}|)[z^{-1}] &= \sum_{n=0}^{\infty} |L_{n,s}| z^n \\
&= |L_{0,s}| + \sum_{n=1}^{\infty} (|L^{\leq}_{n,s}| - |L^{\leq}_{n-1,s}|) z^n \\
&= |L_{0,s}| + \sum_{n=1}^{\infty} |L^{\leq}_{n,s}| z^n - \sum_{n=1}^{\infty} |L^{\leq}_{n-1,s}| z^n \\
&= |L_{0,s}| + (\mathcal{Z}(|L^{\leq}_{n,s}|)[z^{-1}] - |L^{\leq}_{0,s}|) - z \sum_{n=1}^{\infty} |L^{\leq}_{n-1,s}| z^{n-1} \\
&= |L_{0,s}| + (\mathcal{Z}(|L^{\leq}_{n,s}|)[z^{-1}] - |L_{0,s}|) - z \sum_{n=0}^{\infty} |L^{\leq}_{n,s}| z^n \\
&= (1-z)\mathcal{Z}(|L^{\leq}_{n,s}|)[z^{-1}] \\
&= (1-z) \sum_{J \subset \binom{[d]}{s}} \mathcal{Z}(|L^{\leq}_{n,s,J}|)[z^{-1}] \\
&= (1-z) \binom{d}{s} \mathcal{Z}(|L^{\leq}_{n,s,[s]}|)[z^{-1}]
\end{aligned}
$$

■

Lemma 20 *For any $s \in \{0, ..., d\}$, the number of lattice points in $L^{\leq}_{n,s,[s]}$ is equal to the number of lattice points in $\cup_{i=s}^{s-k+1} n\mathcal{A}^1_{s,i}$.*

Proof By Corollary 16, $L^{\leq}_{n,s,[s]}$ is equal to the lattice points in $P_{n,s,[s]} = \{x : x \in O_+, \text{support}(x) = [s], 0 \leq \|x\|_\infty \leq n, 0 \leq \|x\|_1 \leq nk\}$. Projecting the points of $P_{n,s,[s]}$ to its support index set $[s]$ gives the set $\{v \in (0,n]^s : 0 < \sum_i v_i \leq nk\}$. Note that $\cup_{i=s}^{s-k+1} n\mathcal{A}^1_{s,i} = \{v \in [0,n)^s : n(s-k) \leq \sum_i v_i < ns\}$ is congruent to the shape $\{v \in (0,n]^s : 0 < \sum_i v_i \leq nk\}$ under the rotation map $x \to (n, ..., n) - x$ (note: the union indices going backwards from $i = s$ to $s - k + 1$ is written to emphasize the fact that the last index of the map's pre-image corresponds to the first index of the map's image). ■

To understand the closed form finite formula, we need a few auxiliary definitions involving a variant of permutation groups.

Definition 60 *Define S_d to be the group of permutations on the set $\{0, ..., d-1\}$. For any $\sigma \in S_d$, define $des(\sigma) = |\{i : i \in \{0, 1, ..., d-1\}, \sigma_i > \sigma_{i+1}\}|$. Define the set of* colored permutations *S^r_d to be pairs (σ, c) where $\sigma \in S_d$ and $c \in \{0, 1, ..., r-1\}^d$. Define the descent number of a colored permutation as $des(\sigma, c) = |\{i \mid i \in \{0, 1, ..., d-1\}, c_i > c_{i+1}$ or $c_i = c_{i+1}$ and $\sigma_i > \sigma_{i+1}\}|$. Define the flag descent number of a colored permutation as $fdes(\sigma, c) = r \cdot des(\sigma, c) + c_d$.*

The following result is adapted from Proposition 12 of Han and Josuat-Vergès (2016).

Lemma 61 $E^*(\mathcal{A}^r_{d,k}, z) = \sum_{\substack{(\sigma,c)\in S^r_d \\ fdes(\sigma,c)=k-1}} z^{des(\sigma^{-1})+1}$ *with the convention that* $E^*(\mathcal{A}^r_{d,0}, z) = \mathbb{1}_{d=0}$ *and* $E^*(\mathcal{A}^r_{d,k}, z) = 0$ *if* $k < 0$ *or* $k > rd$, *where* σ^{-1} *is the* inverse permutation *of* σ.

Note that we have computed the h^* polynomial of $A^r_{d,k}$ but for our application we only need to consider the case where $r = 1$. This simplifies the closed form formula.

Corollary 62 $E^*(\mathcal{A}^1_{d,k}, z) = \sum_{\substack{\sigma\in S_d \\ des(\sigma)=k-1}} z^{des(\sigma^{-1})+1}$.

Proof When $r = 1$, then $c = (0, ..., 0)$ so we only need to sum over S_d, and $\text{fdes}(\sigma, c) = \text{des}(\sigma)$ since $c_d = 0$. ■

We have shown that the computation of the normalizing constant N reduces to computing $\mathcal{Z}(|L^{\leq}_{n,s,[s]}|)[z^{-1}]$. We can now prove a finite closed form formula for this series using Corollary 62. This finite formula is expressed in terms of the coefficients of the bivariate generating function $A_d(y, z) = \sum_{\pi\in S_d} y^{\text{des}(\pi)+1} z^{\text{des}(\pi^{-1})+1} = \sum_{x,w=1}^{d} A_{d,x,w} y^x z^w$ where $A_{d,x,w}$ counts the number of permutations $\pi \in S_d$ with $x - 1$ descents and and whose inverse permutation π^{-1} has $w - 1$ descents.

Lemma 63 *Fix* $s \in \{0, ..., d\}$. *Then*

$$\mathcal{Z}(|L^{\leq}_{n,s,[s]}|)[z^{-1}] = (1 - z)^{-s-1} \sum_{i=s-k+1}^{s} \sum_{w=0}^{s} A_{s,i,w} z^w \tag{1}$$

where $A_{s,i,w} z^w$ *are coefficients that can computed in run time* $O(d^3)$.

Proof By Lemma 20, $|L^{\leq}_{n,s,[s]}| = |\cup_{i=s}^{s-k+1} n\mathcal{A}^1_{s,i}| = \sum_{i=s-k+1}^{s} E(\mathcal{A}^1_{s,i}, n)$.

$$\begin{aligned}
\mathcal{Z}(|L^{\leq}_{n,s,[s]}|)[z^{-1}] &= \sum_{n\geq 0} |L^{\leq}_{n,s,[s]}| z^n \\
&= \sum_{n\geq 0} \sum_{i=s-k+1}^{s} E(\mathcal{A}^1_{s,i}, n) z^n \\
&= \sum_{i=s-k+1}^{s} \sum_{n\geq 0} E(\mathcal{A}^1_{s,i}, n) z^n \\
&= (1 - z)^{-s-1} \sum_{i=s-k+1}^{s} E^*(\mathcal{A}^1_{s,i}, z) \\
&= (1 - z)^{-s-1} \sum_{i=s-k+1}^{s} \sum_{\substack{\sigma\in S_s \\ \text{des}(\sigma)=i-1}} z^{\text{des}(\sigma^{-1})+1}
\end{aligned}$$

where we have used Corollary 62 in the last equality.

Claim 64

$$\sum_{\substack{\sigma \in S_s \\ des(\sigma)=i-1}} z^{des(\sigma^{-1})+1} = \sum_{w=0}^{s} A_{s,i,w} z^w \tag{2}$$

and we can compute the $A_{s,i,w}$ coefficients in run time $O(d^3)$

Proof Define $A_d(y,z) = \sum_{\pi \in S_d} y^{\text{des}(\pi)+1} z^{\text{des}(\pi^{-1})+1} = \sum_{x,w=1}^{d} A_{d,x,w} y^x z^w$ where $A_{d,x,w}$ sums the coefficients of $y^x z^w$. Define $A_{d,x,w}(y,z) = A_{d,x,w} y^x z^w$. Theorem 1.1 in Li and Liu (2022) has shown how to compute these functions recursively. In the base cases, $A_{1,1,1} = 1$ and $A_{d,x,w} = 0$ when $x = 0$ or $w = 0$. In the general cases, $1 \le x, w \le d$ and $d \ge 2$, we have

$$\begin{aligned} dA_{d,x,w} =& (xw - d - 1)A_{d-1,x,w} + (1 - d + w(d + 1 - x))A_{d-1,x-1,w} + \\ & (1 - d + x(d + 1 - w))A_{d-1,x,w-1} + (d - 1 + (d + 1 - x)(d + 1 - w))A_{d-1,x-1,w-1} \end{aligned}$$

We can see that computing each cell of the dynamic programming table for A is $O(1)$ time by using the above recurrence so the time to compute up to cell $A_{d,x,w}$ is $O(dxw)$.

Then

$$\sum_{\substack{\sigma \in S_s \\ \text{des}(\sigma)=i-1}} z^{\text{des}(\sigma^{-1})+1} = \sum_{w=1}^{s} A_{s,i,w}(1,z) = \sum_{w=1}^{s} A_{s,i,w} z^w = \sum_{w=0}^{s} A_{s,i,w} z^w$$

where the final equality holds since $A_{s,i,0} = 0$. Since we need to compute up to cell $A_{d,d,d}$ (in the case $j = 0, i = d, w = d$), then the total run time of computing this expression is $O(d^3)$. ■

■

Corollary 65

$$N = \sum_{s=0}^{d} 2^s (1 - e^{-\varepsilon})^{-s} \binom{d}{s} \sum_{i=s-k+1}^{s} \sum_{w=0}^{s} A_{s,i,w} e^{-\varepsilon w}$$

and N can be computed in run time $O(d^3)$.

Proof The formula for N follows immediately from combining Lemma 18 and Lemma 63. Computing the $A_{s,i,w}$ coefficients takes time $O(d^3)$ by Lemma 63. Computing binomial coefficients upto the dth row of pascal's triangle takes time $O(d^2)$. After computing the expressions above, the triple summation takes time $O(d^3)$. ■

Claim 25 *After pre-computing $C([0,d],[0,nk],n-1)$, we can compute:*

- *$|X_J|$ in time $O(nk^2)$ using the equation $|X_{J,i}| = \binom{s}{i} \sum_{t=0}^{nk-ni} C(s-i,t,n-1)$*
- *$|X_J^c|$ in time $O(k)$ using the equation $|X_J^c| = \sum_{i=0}^{k-1} C(s, nk-i, n-1)$*

- $|L_{n,s,J}|$ *and* $|L_{n,s}|$ *in time* $O(nk^2)$ *using* $|L_{n,s,J}| = |X_J| + |X_J^c|$ *and* $|L_{n,s}| = \binom{d}{s}|L_{n,s,J}|$
- $|L_n|$ *in time* $O(dnk^2)$ *using* $|L_n| = \sum_{s=0}^{d} 2^s |L_{n,s}|$.

Proof We can view each point of $X_{J,i}$ as first picking i coordinates out of J to be equal to n. Then the possible sums of the rest of the coordinates range in $\{0, 1, ..., nk - ni\}$. Hence $|X_{J,i}| = \binom{s}{i} \sum_{t=0}^{nk-ni} C(s-i, t, n-1)$, where $\binom{s}{i}$ accounts for the placement of the i maximal coordinates with value n. Our goal now is to compute the 2-dimensional table $C(\cdot, \cdot, n-1)$ with upper left hand cell $C(0, 0, n-1)$ and lower right hand cell $C(d-1, nk-ni, n-1)$.

The time to compute the binomial coefficients $\binom{s}{i}$ is $O(d^2)$. The time to compute each $|X_{J,i}|$ is then $O(nk)$, so the time to compute all $\{|X_{J,i}|\}_{i=1}^{k}$ is $O(nk^2)$. The overall running time to compute $|X_J| = \sum_{i=1}^{k} |X_{J,i}|$ is $O(dnk) + O(d^2) + O(nk^2) = O(dnk)$.

For X_J^c, recall that $X_J^c = L_{n,s,J} - X_J$. Suppose $v \in X_J^c$. Each of the s coordinates of v at the J indices is in $\{1, ..., n-1\}$ and the possible values for $\|v\|_1$ range in $\{nk - (k-1), ..., nk\}$ because $\|v\|_\infty < n$ meaning that $\lceil \frac{\|v\|_1}{k} \rceil = n$. We therefore need to compute the 2-dimensional $C(\cdot, \cdot, n-1)$ table with upper left hand cell $C(0, nk-(k-1), n-1)$ and lower right hand cell $C(d, nk, n-1)$. Then $|X_J^c| = \sum_{i=0}^{k-1} C(s, nk-i, n-1)$. By arguing as in the computation of $|X_J|$ above, the run time is $O(dnk)$.

Finally when all these quantities are known we can compute $|L_n|$. As a reminder, $L_{n,s}$ and $L_{n,s,J}$ were defined as subsets of $L_n \cap O_+$ whereas L_n is not restricted to O_+ .

First, $|L_{n,s,J}| = |X_J| + |X_J^c|$ which we can compute in time $O(nk^2) + O(k) = O(nk^2)$ by the computations of $|X_J|$ and $|X_J^c|$ shown above.

Second, $|L_{n,s}| = \binom{d}{s}|L_{n,s,J}|$ where the $\binom{d}{s}$ term chooses s support indices out of d possible. So if we have already computed $|L_{n,s,J}|$, computing this term takes time $O(1)$.

Third, $|L_n| = \sum_{s=0}^{d} 2^s |L_{n,s}| = \sum_{s=0}^{d} 2^s \binom{d}{s} |L_{n,s,J}|$ where the 2^s term accounts for all possible sign combinations on the s support indices. We can compute this sum in time $O(dnk^2)$ since each $|L_{n,s}|$ term requires $O(nk^2)$ time.

■

Claim 24 *We can compute* $C([0, d], [0, nk], n-1)$ *in* $O(dnk)$.

Proof To compute the relevant C quantities, we start with the base cases $C(1, t, n-1) = 1$ for all $1 \le t \le n-1$ and $C(x, x, n-1) = 1$ for all $x \ge 0$. Then we have the recursion

$$C(d, t, n-1) = \sum_{i=1}^{\min(t-(d-1), n-1)} C(d-1, t-i, n-1)$$

because we can pick an integer $i \in \{1, ..., n-1\}$ for the first number in the summation, and then the rest of the summation must total $t - i$. Note that the summation does not extend to $i > t$ because $d-1$ integers in $\{1, \ldots, n-1\}$ cannot sum to less than $d-1$ and does not extend to $i > n-1$ because we cannot choose integers $> n-1$.

Actually, we can optimize this recursion further. Notice that if $t - (d-1) \le n-1$ then the recursive relation reduces to

$$C(d, t, n-1) = \sum_{i=1}^{t-(d-1)} C(d-1, t-i, n-1)$$

and it follows that $C(d,t,n-1) = C(d,t-1,n-1) + C(d-1,t-1,n-1)$. If instead $t-(d-1) > n-1$ the recursive relation reduces to

$$C(d,t,n-1) = \sum_{i=1}^{n-1} C(d-1,t-i,n-1)$$

and it follows that $C(d,t,n-1) = C(d,t-1,n-1) - C(d-1,t-1-(n-1),n-1) + C(d-1,t-1,n-1) = C(d,t-1,n-1) - C(d-1,t-n,n-1) + C(d-1,t-1,n-1)$.

Since we do $O(1)$ work to compute each cell of the dynamic program, the total time to compute the required C table is $O(dnk)$. ■

Claim 26 *We can uniformly sample a point from L_n in time $O(dnk^2)$.*

Proof Begin by computing $|L_{n,s}|$ for all $s \in \{0,...,d\}$ and $|L_n|$ in time $O(dnk^2)$. Select some $s \in \{0,1,...,d\}$ based on the weights $\{2^s|L_{n,s}|\}_{s=0}^d$ in time $O(d)$. Select some $J \subset [d]$ with $|J| = s$ in time $O(d)$. Then flip a coin weighted by $(|X_J|, |X_J^c|)$ to select one of X_J or X_J^c, and note that these weights will be computed during the computation of the $|L_{n,s}|$ terms.

If we pick X_J, we select one of $\{X_{J,1},...,X_{J,k}\}$ proportional to their sizes which will also be computed during the computation of the $|L_{n,s}|$ terms. Suppose we pick $X_{J,i}$. Then we need to form weights for each of the possible branches $\{C(s-i,0,n-1),...,C(s-i,nk-ni,n-1)\}$. Suppose we pick the branch corresponding to the sum $t \in \{0,...,nk-ni\}$. We can step through the recursion to sample a sequence uniformly for $C(s-i,t,n-1)$ to get a vector $v \in \{1,...,n-1\}^{s-i}$: if we're at step $C(a,b,c)$, we pick $j \in \{0,1,...,\min(b,c)\}$ with weights proportional to $C(a-1,b-j,c)$ and then recur into the chosen subbranch. Stepping through this recursion takes run time upper bounded by the time to compute the C table which is $O(dnk)$. Then we sample a random permutation π of J in $O(d)$ time. Initialize a vector $p \in \mathbb{Z}^d$ to be the all zeros vector. Now modify p by setting each of the coordinates in the first i indices of π to n, and embedding v at the remaining coordinates of J in left to right order. Return p.

If we pick X_J^c, we pick one of the sums in $\{nk-(k-1),...,nk\}$ with weights proportional to $\{C(s,nk-(k-1),n-1),...,C(s,nk,n-1)]\}$. We can step through the recursion to sample a sequence $v \in \{1,...,n-1\}^s$ uniformly from the chosen $C(s,nk-i,n-1)$, as before. The run time of this step is similarly $O(dnk)$. Initialize a vector $p \in \mathbb{Z}^d$ to be the all zeros vector. Now modify p by embedding v at the coordinates of J in left to right order. Return p. ■

The following lemma gives an upper-bound on the expected hitting time for ripple mechanism, given that a certain condition holds. It covers all three problems in this paper.

Lemma 27 *In the ripple mechanism, if $\mathcal{Z}(|L_n|)[z]$ can be expressed as a summation of terms of bounded power of z and $1-z^{-1}$*

$$\mathcal{Z}(|L_n|)[z] = \sum_{s=0}^{d} \sum_{w=0}^{d+1} B_{s,w} z^{-w} (1-z^{-1})^{-s}$$

where $B_{s,w}$ are non-negative coefficients, then the expected hitting time T of the reservoir sampling is bounded as $\mathbb{E}[T] = O(\frac{d}{1-e^{-\varepsilon}})$.

Proof Recall that $w_n = \frac{|L_n|e^{-n\varepsilon}}{N}$ is the probability of reservoir sampling stopping at level set L_n. By definition of hitting time,

$$\begin{aligned}
\mathbb{E}[T] &= \sum_{n=0}^{\infty}(n+1)w_n \\
&= \sum_{n=0}^{\infty} w_n + \sum_{n=0}^{\infty} nw_n \\
&= 1 + \frac{1}{N}\mathcal{Z}(n|L_n|)[z]\Big|_{z=e^{\varepsilon}} \\
&= 1 - \frac{z}{N}\frac{\partial \mathcal{Z}(|L_n|)[z]}{\partial z}\Big|_{z=e^{\varepsilon}} \qquad (3) \\
&= 1 - \frac{z}{N}\sum_{s=0}^{d}\sum_{w=0}^{d+1} B_{s,w}(-wz^{-w-1}(1-z^{-1})^{-s} + (-s)z^{-w-2}(1-z^{-1})^{-s-1})\Big|_{z=e^{\varepsilon}} \\
&= 1 + \frac{z}{N}\sum_{s=0}^{d}\sum_{w=0}^{d+1} B_{s,w}z^{-w-1}(1-z^{-1})^{-s-1}(sz^{-1} + w(1-z^{-1}))\Big|_{z=e^{\varepsilon}} \\
&= 1 + (sz^{-1}(1-z^{-1})^{-1} + w)\frac{1}{N}\sum_{s=0}^{d}\sum_{w=0}^{d+1} B_{s,w}z^{-w}(1-z^{-1})^{-s}\Big|_{z=e^{\varepsilon}} \\
&\leq 1 + (dz^{-1}(1-z^{-1})^{-1} + d + 1)\frac{1}{N}\sum_{s=0}^{d}\sum_{w=0}^{d+1} B_{s,w}z^{-w}(1-z^{-1})^{-s}\Big|_{z=e^{\varepsilon}} \qquad (4) \\
&= 2 + d(1 + \frac{e^{-\varepsilon}}{1-e^{-\varepsilon}}) \qquad (5)
\end{aligned}$$

The equality at 3 holds by the property that $\mathcal{Z}(nx_n)[z] = -z\frac{\partial \mathcal{Z}(x_n)[z]}{\partial z}$ (see introduction of Forouzan (2016)). Note that the property listed is for bilateral $\mathcal{Z}$ transforms but we can just set the negative indexed coefficients to 0 and it reduces to the unilateral setting.

The inequality at 4 holds since z^{-1} and $(1-z^{-1})$ are positive for $z = e^{\varepsilon}$ where $\varepsilon > 0$ which is true by assumption. The last step 5 holds because $N = \mathcal{Z}(|L_n|)[e^{\varepsilon}]$. ■

Corollary 28 *Let T be the hitting time random variable for reservoir sampling for Sum. Then $\mathbb{E}[T] = O(\frac{d}{1-e^{-\varepsilon}})$.*

Proof Combing Lemma 18 and Lemma 63 gives

$$\mathcal{Z}(|L_n|)[z] = \sum_{s=0}^{d} 2^s(1-z^{-1})^{-s}\binom{d}{s}\sum_{i=s-k+1}^{s}\sum_{w=0}^{s} A_{s,i,w}z^{-w}$$

which satisfies the hypothesis of Lemma 27 since the terms $2^s, (1-z^{-1})^{-s}, A_{s,i,w}, \binom{d}{s}$ are all non-negative for $z = e^{\varepsilon}$ for $\varepsilon > 0$ which is true by assumption. ■

Algorithm 1 Sum Sampler

Input: Dimension d, ℓ_0 bound k
Compute the normalizing constant N using Corollary 65
Initialize layer index $n = 0$
while True **do**
  Compute the $C([0, d], [0, nk], n-1)$ table using Claim 24
  for $s = 0, \ldots, d$ **do**
    for $i = 1, \ldots, k$ **do**
      Compute $|X_{J,i}|$ using Claim 25
    Compute $|X_J|, |X_J^c|$ using Claim 25
    Compute $|L_{n,s}|$ using Claim 25
  Compute $|L_n|$ using Claim 25
  Flip reservoir sampling coin with heads probability $\frac{w_n}{1-\sum_{j=0}^{n-1} w_n}$
  if heads **then**
    break
  else
    $n \leftarrow n + 1$
Sample $s \propto \{2^s |L_{n,s}|\}_{s=0}^{d}$
Sample $J \subset [d]$ with $|J| = s$
Sample $X \in \{X_J, X_J^c\} \propto \{|X_J|, |X_J^c|\}$
if $X = X_J$ **then**
  Sample $X_{J,i} \propto \{|X_{J,i}|\}_{i=1}^{k}$ using Claim 25
  Sample $t \propto \{C(s-i, t, n-1)\}_{t=0}^{nk-ni}$
  Sample $v \in \{1, ..., n-1\}^{s-i}$ based on t using Claim 26
  Sample random permutation π of J
  Initialize $p \in \mathbb{Z}^d$ to be the all 0s vector
  Set the first i indices of π in p to be n and embed v at the remaining indices of J in left to right order
if $X = X_J^c$ **then**
  Sample $t \propto \{C(s, t, n-1)\}_{t=nk-(k-1)}^{nk}$
  Sample $p \in \{1, ..., n-1\}^s$ based on t using Claim 26
Return p

Corollary 29 *When $\varepsilon = O(1)$, the expected running time of $\mathcal{R}_{\text{SUM}}$ is $O(d^3k^2)$.*

Proof Recall from Claim 25 that the time to compute $|L_n|$ and sample a point from L_n is $O(dnk^2)$. Since $\mathbb{E}[T] = O(\frac{d}{1-e^{-\varepsilon}})$, then $\mathbb{E}[n] = \mathbb{E}[T] = O(d)$ when $\varepsilon = O(1)$. It follows that the expected running time is $O(d) * O(d^2k^2) = O(d^3k^2)$. ■

11. Proofs From Section 5

This section provides the full details for $\mathcal{R}_{\text{COUNT}}$. Sampler pseudocode appears in Algorithm 2.

Lemma 32 *Suppose $J \in J_{p,m}$. Let P be the positive support indices of J, let M be the negative support indices of J. Then we have the internal direct sum decomposition $K_{n,J} = \sqcup_{a=0}^{n}(L_{a,p,P} \oplus -L_{n-a,m,M})$ where $L_{a,p,P}, L_{n-a,m,M}$ are defined in Definition 17 and $-L_{n-a,m,M}$ denotes the set $\{-v : v \in L_{n-a,m,M}\}$.*

Proof First, we show that $K_{n,J} \subset \sqcup_{a=0}^{n}(L_{a,p,P} \oplus -L_{n-a,m,M})$. Let $v \in K_{n,J}$. Write $v = \sum_{i=1}^{n} v_i$ where $v_i \in K_1$. Let V_p be the set of vectors v_i that are in the positive orthant and let V_m be the set of vectors v_i that are in the negative orthant. Then $v = \sum_{u \in V_p} u + \sum_{w \in V_m} w$.

We show that we can assume that each $u \in V_p$ has support a subset of P. Suppose not. Then there is some $u \in V_p$ with a coordinate 1 in $[d] - P$. Suppose this is at coordinate c. We know that the c coordinate of v is ≤ 0 since P is the positive support indices of J. That means that there must be some $w \in V_m$ with a coordinate -1 at c. We can replace u by $u - e_c$ and replace w by $w + e_c$ without changing the value of the summation $v = \sum_{i=1}^{n} v_i$. By iterating this process, we can assume that each $u \in V_p$ has support a subset of P and similarly we can assume that each $w \in V_m$ has support a subset of M.

Let $a = |V_p|$ then $n - a = |V_m|$. So far, we know that $\sum_{u \in V_p} u \in L^{\leq}_{a,p,P}$. If it were true that $\sum_{u \in V_p} u \in L_{a',p,P}$ for some $a' < a$ then we could rewrite v as a sum of $a' + n - a < n$ elements of K_1, a contradiction. So it follows that $\sum_{u \in V_p} u \in L_{a,p,P}$, and similarly $\sum_{w \in V_m} w \in -L_{n-a,m,M}$. Then $K_{n,J} \subset \sqcup_{a=0}^{n}(L_{a,p,P} \oplus -L_{n-a,m,M})$ as desired.

Second, we show that $\sqcup_{a=0}^{n}(L_{a,p,P} \oplus -L_{n-a,m,M}) \subset K_{n,J}$. Suppose we have a vector of the form $v = x_p - x_m$ where $x_p \in L_{a,p,P}, x_m \in L_{n-a,m,M}$. Let $v = \sum_{i=1}^{j} v_i$ where $v_i \in K_1$ be an expression of v where j is as small as possible. Write $v = \sum_{u \in V_p} u + \sum_{w \in V_m} w$ where V_p are the set of vectors v_i that are in the positive orthant and V_m are the set of vectors v_i that are in the negative orthant. By using the same replacement procedure as in the second paragraph of this proof, we may assume that each $u \in V_p$ has support a subset of P and each $w \in V_m$ has support a subset of M. If $j < n$ then one of $|V_p| < a$ or $|V_m| < n - a$ must hold. If $|V_p| < a$ holds, then $x_p = \sum_{u \in V_p} u$ expresses v_p as a sum of fewer than a points of L_1, a contradiction to the definition of $L_{a,p,P}$. We get a similar contradiction if $|V_m| < n - a$ holds. It follows that $j = n$. Since $x_p \in L_{a,p,P}, x_m \in L_{n-a,m,M}$, then v has sign vector J so $v \in K_{n,J}$ as desired. ■

Corollary 33 *Suppose $J \in J_{p,m}$. Let P be the positive support indices of J, let M be the negative support indices of J. Then $|K_{n,J}| = \sum_{a=0}^{n} |L_{a,p,P}||L_{n-a,m,M}|$, $|K_{n,p,m}| = \binom{d}{p}\binom{d-p}{m}|K_{n,J}|$, and $|K_n| = \sum_{p=0}^{d}\sum_{m=0}^{d-p}|K_{n,p,m}|$.*

Proof The first equality $|K_{n,J}| = \sum_{a=0}^{n} |L_{a,p,P}||L_{n-a,m,M}|$ follows immediately from the direct sum decomposition $K_{n,J} = \sqcup_{a=0}^{n}(L_{a,p,P} \oplus -L_{n-a,m,M})$ of Lemma 32.

For the second equality, note that $|K_{n,p,m}| = |J_{p,m}||K_{n,J}|$ since $|K_{n,J_1}| = |K_{n,J_2}|$ for all $J_1, J_2 \in J_{p,m}$. Moreover, $|J_{p,m}| = \binom{d}{p}\binom{d-p}{m}$ since we can construct an element of $J_{p,m}$ by first choosing p out of d coordinates to be positive and then choosing m out of the remaining $d - p$ coordinates to be negative.

The third equality is immediate from the disjoint union $K_n = \sqcup_{p=0}^{d} \sqcup_{m=0}^{d-p} K_{n,p,m}$. ■

Lemma 34 *The normalizing constant $N = \sum_{n \geq 0} |K_n| e^{-n\varepsilon} = \sum_{p=0}^{d}\sum_{m=0}^{d-p}\sum_{n \geq 0} |L_{n,p,m}| e^{-n\varepsilon}$, and N can be computed in time $O(d^3)$.*

Proof Recall from Lemma 63 that for any $s \in \{0, ..., d\}$ and any $X \subset [d]$ of size $|X| = s$ we have

$$\mathcal{Z}(|L^{\leq}_{n,s,X}|)[z^{-1}] = (1-z)^{-s-1} \sum_{i=s-k+1}^{s} \sum_{w=0}^{s} A_{s,i,w} z^w$$

Now, let's express $\mathcal{Z}(|L_{n,s,X}|)[z^{-1}]$ in terms of $\mathcal{Z}(|L^{\leq}_{n,s,X}|)[z^{-1}]$. We have $\mathcal{Z}(|L_{n,s,X}|)[z^{-1}] = (1-z)\mathcal{Z}(|L^{\leq}_{n,s,X}|)[z^{-1}]$ since by Lemma 18 $\mathcal{Z}(|L_{n,s}|)[z^{-1}] = (1-z)\mathcal{Z}(|L^{\leq}_{n,s}|)[z^{-1}]$ and $|L_{n,s}| = \binom{d}{s}|L_{n,s,X}|$

Combining the above gives

$$\mathcal{Z}(|L_{n,s,X}|)[z^{-1}] = (1-z)^{-s} \sum_{i=s-k+1}^{s} \sum_{w=0}^{s} A_{s,i,w} z^w$$

Fix $J \in J_{p,m}$ with positive support indices P and negative support indices M. By Corollary 33, $|K_{n,p,m}| = \binom{d}{p}\binom{d-p}{m} \sum_{a=0}^{n} |L_{a,p,P}||L_{n-a,m,M}|$, so we have

$$\begin{aligned}
\mathcal{Z}(|K_n|)[z^{-1}] &= \sum_{p=0}^{d} \sum_{m=0}^{d-p} \sum_{n \geq 0} |K_{n,p,m}| z^n \\
&= \sum_{p=0}^{d} \sum_{m=0}^{d-p} \binom{d}{p} \binom{d-p}{m} \sum_{n \geq 0} \sum_{a=0}^{n} |L_{a,p,P}||L_{n-a,m,M}| z^n \\
&= \sum_{p=0}^{d} \sum_{m=0}^{d-p} \binom{d}{p} \binom{d-p}{m} \mathcal{Z}(|L_{n,p,P}| * |L_{n,m,M}|)[z^{-1}] \\
&= \sum_{p=0}^{d} \sum_{m=0}^{d-p} \binom{d}{p} \binom{d-p}{m} \mathcal{Z}(|L_{n,p,P}|)[z^{-1}] \mathcal{Z}(|L_{n,m,M}|)[z^{-1}] \\
&= \sum_{p=0}^{d} \sum_{m=0}^{d-p} \binom{d}{p} \binom{d-p}{m} \left[(1-z)^{-p} \sum_{i=p-k+1}^{p} \sum_{w=0}^{p} A_{p,i,w} z^w \right] \left[(1-z)^{-m} \sum_{j=m-k+1}^{m} \sum_{x=0}^{m} A_{m,j,x} z^x \right]
\end{aligned}$$

where $*$ denotes convolution. See page 3 of Dahleh (2003) for the $\mathcal{Z}$-transform of the convolution of two sequences which is given by $\mathcal{Z}(x_n * y_n) = \mathcal{Z}(x_n)\mathcal{Z}(y_n)$. Note that the property listed is for bilateral $\mathcal{Z}$ transforms but we can just set the negative indexed coefficients to 0 and it reduces to the unilateral setting.

We can compute all of the A coefficients in $O(d^3)$ as in Lemma 63. Let $x(p)$ denote the value of the first square bracketed term for a fixed p, and let $y(m)$ denote the value of the second square bracketed term for a fixed m. Computing $x(p)$ costs $O(kd)$ to iterate over the two summations so the cost of computing $\{x(p)\}_{p=0}^{d}$ is $O(kd^2)$ and similarly the cost of computing $\{y(m)\}_{m=0}^{d}$ is $O(kd^2)$. We can compute all the binomial coefficients in time $O(d^2)$. After pre-computing $\{x(p)\}_{p=0}^{d}$, $\{y(m)\}_{m=0}^{d}$, and the binomial coefficients, the cost of computing the outer double summation is $O(d^2)$. The total runtime is therefore $O(d^3) + O(2kd^2) + O(d^2) = O(d^3)$. ■

Lemma 35 *We can compute both* $\{|K_{n,p,m}|\}_{p+m=0}^{d}$ *and* $\{|L_{a,p,[p]}|\}_{p=0,a=0}^{d,n}$ *in time* $O(dn^2k^2) + O(d^2n)$.

Proof Recall by Corollary 33 that $|K_{n,p,m}| = \binom{d}{p}\binom{d-p}{m}\sum_{a=0}^{n}|L_{a,p,[p]}||L_{n-a,m,[m]^{-1}}|$ where $[m]^{-1}$ denotes the last m indices of $[d]$. For a fixed $a \in \{0,...,n\}$, we compute $C([0,d],[0,ak],a-1)$ in time $O(dak)$ by Claim 24. This allows us to compute the collection $\{|L_{a,p,[p]}|\}_{p=0}^{d}$ in additional time $O(dak^2)$ by the proof of Claim 25. It follows that we can compute the collection $\{|L_{a,p,[p]}|\}_{p=0,a=0}^{d,n}$ in time $O(dn^2k^2)$, and note that the collection $\{|L_{n-a,m,[m]^{-1}}|\}_{m=0,a=0}^{d,n}$ is equal to the collection $\{|L_{a,p,[p]}|\}_{p=0,a=0}^{d,n}$ up to ordering. We can compute the collection of binomial coefficients up to the dth row of pascal's triangle in time $O(d^2)$. Using all of the above computed quantities, the summation for the expression $|K_{n,p,m}| = \binom{d}{p}\binom{d-p}{m}\sum_{a=0}^{n}|L_{a,p,[p]}||L_{n-a,m,[m]^{-1}}|$ takes time $O(n)$. It follows that we can compute $\{|K_{n,p,m}|\}_{0\le p+m\le d}$ in time $O(dn^2k^2) + O(d^2) + O(d^2n) = O(dn^2k^2 + O(d^2n)$. ■

Lemma 36 *After pre-computing* $\{|K_{n,p,m}|\}_{p+m=0}^{d}$, *we can compute* $|K_n|$ *in* $O(d^2)$.

Proof Using the collection $\{|K_{n,p,m}|\}_{0\le p+m\le d}$ we can compute $|K_n|$ using the equation $|K_n| = \sum_{p=0}^{d}\sum_{m=0}^{d-p}|K_{n,p,m}|$ from Corollary 33 in $O(d^2)$ additional time. ■

Lemma 37 *After pre-computing* $\{|K_{n,p,m}|\}_{0\le p+m\le d}$ *and* $\{|L_{a,p,[p]}|\}_{p=0,a=0}^{d,n}$, *we can sample a point uniformly from* K_n *in time* $O(dnk)$.

Proof First, select signature (p,m) according to the weights $\{|K_{n,p,m}|\}_{0\le p+m\le d}$ in time $O(d^2)$.

Second, select $J \in J_{p,m}$ by taking a random permutation of $[d]$ using Fisher-Yates and setting the first p indices to P and the next m indices to M. These operations take time $O(d)$.

Third, recall that $K_{n,J} = \sqcup_{a=0}^{n}(L_{a,p,P} \oplus -L_{n-a,m,M})$. Select index a according to the weights $\{|L_{a,p,P}||L_{n-a,m,M})|\}_{a=0}^{n}$ which we can compute in $O(n)$ using the pre-computed collection $\{|L_{a,p,[p]}|\}_{p=0,a=0}^{d,n}$.

Fourth, sample $v_p \in L_{a,p,P}$ and $v_m \in L_{n-a,m,M}$ which takes time $O(dak) + O(d(n-a)k) = O(dnk)$ by the second paragraph of Claim 26. Initialize v to be the zero vector in $\mathbb{R}^d$. Embed v_p at the coordinates of P in v and embed $-v_m$ at the coordinates of M in v. Return v as a uniform sample of K_n. ■

Lemma 66 *Let* T *be the hitting time random variable for reservoir sampling. Then* $\mathbb{E}[T] = O(\frac{d}{1-e^{-\varepsilon}})$.

Proof Recall by Lemma 34

$$\mathcal{Z}(|K_n|)[z] = \sum_{p=0}^{d}\sum_{m=0}^{d-p}\binom{d}{p}\binom{d-p}{m}\left[(1-z^{-1})^{-p}\sum_{i=p-k+1}^{p}\sum_{w=0}^{p}A_{p,i,w}z^{-w}\right]\left[(1-z^{-1})^{-m}\sum_{j=m-k+1}^{m}\sum_{x=0}^{m}A_{m,j,x}z^{-x}\right]$$

Note that all the binomial coefficients and $A_{p,i,w}, A_{m,j,x}$ coefficients are non-negative as shown in Lemma 63. It follows that the entire expression is a sum of terms of the form $B(1-z^{-1})^{-(p+m)}z^{-(w+x)}$ where $B \ge 0$. Moreoever, $0 \le p+m \le d$ and $0 \le w+x \le d$. So $\mathcal{Z}(|K_n|)[z]$ satisfies the hypothesis of Lemma 27. ■

Algorithm 2 Count Sampler

Input: Dimension d, ℓ_0 bound k, layer index upper bound U
Compute the normalizing constant N using Lemma 34
Initialize layer index $n = 0$
while True **do**
  Compute $\{|K_{n,p,m}|\}_{0 \le p+m \le d}$ using Lemma 35
  Compute $|K_n|$ using Corollary 33
  Flip reservoir sampling coin with heads probability $\frac{w_n}{1-\sum_{j=0}^{n-1} w_n}$
  if heads **then**
    break
  else
    $n \leftarrow n+1$
Sample signature (p, m) using Lemma 37
Sample $J \in J_{p,m}$ using Lemma 37
Sample $a \propto \{|L_{a,p,P}||L_{n-a,m,M})|\}_{a=0}^{n}$
Sample $v_p \in L_{a,p,P}$ and $v_m \in L_{n-a,m,M}$
Initialize $v \in \mathbb{Z}^d$ to be the all 0s vector
Embed v_p at the coordinates P in v and $-v_m$ at the coordinates M in v
Return v

Corollary 38 *The expected running time of $\mathcal{R}_{\text{COUNT}}$ is $O(d^4k^2)$ when $\varepsilon = O(1)$.*

Proof Recall from Lemma 35, Lemma 36 and Lemma 37 that the time to compute $|K_n|$ and sample a point from K_n is $O(dn^2k^2) + O(d^2n)$. Since $\mathbb{E}[T] = O(\frac{d}{1-e^{-\varepsilon}})$, then $\mathbb{E}[n] = \mathbb{E}[T] = O(d)$ when $\varepsilon = O(1)$. It follows that the expected running time is $O(d) * O(d^3k^2) = O(d^4k^2)$. ■

12. Proofs From Section 6

This section provides the full details for $\mathcal{R}'_{\text{VOTE}}$. Sampler pseudocode appears in Algorithm 3.

Lemma 68 highlights some elementary facts about Minkowski sums and zonotopes that will be useful for decomposing $\mathcal{L}_n^{\le}$.

Definition 67 *For $a \times n$ matrix N and $a \times m$ matrix M, let $N \oplus M$ denote the horizontal concatenation of N and M.*

Lemma 68

1. *If $S \subset \mathbb{R}^d$ is a convex set and $x, y \ge 0$, then $xS \boxplus yS = (x+y)S$.*
2. *If $x, y \in \mathbb{R}$ and X and Y are matrices with the same number of rows, then $xZ(X) \boxplus yZ(Y) = Z(xX \oplus yY)$.*
3. *If $S, T \subset \mathbb{R}^d$ are sets and $v \in \mathbb{R}^d$ then $S \boxplus (T+v) = (S \boxplus T) + v = (S+v) \boxplus T$.*

The first result relates the permutohedron Π_d to the zonotope defined by A.

Lemma 69 $Z(A) = \Pi_d - v$.

Proof By Proposition 2.3 in Postnikov (2009), Π_d is the zonotope generated by the $\binom{d}{2}$ line segments connecting all pairs of standard bases vectors. Then $[e_i - e_j, 0] + e_j = [e_i, e_j]$ and

$$\begin{aligned} Z(A) &= Z(\{[e_i - e_j, 0] : 1 \leq i < j \leq d\}) \\ &= Z(\{[e_i, e_j] - e_j : 1 \leq i < j \leq d\}) \\ &= Z(\{[e_i, e_j] : 1 \leq i < j \leq d\}) - e_2 - 2e_3 - 3e_4 - ... - (d-1)e_d \\ &= Z(\{[e_i, e_j] : 1 \leq i < j \leq d\}) - v \\ &= \Pi_d - v. \end{aligned}$$

where we have used property 3 of Lemma 68 for the third equality. ■

We can now decompose $\mathcal{L}_n^{\leq}$ using similar zonotopes.

Lemma 70 $\mathcal{L}_n^{\leq} = \sqcup_{i=0}^{n} Z(iA \oplus -(n-i)A) + (-n+2i)v$.

Proof We show that $\mathcal{L}_n^{\leq} = \sqcup_{i=0}^{n}(i\Pi_d \boxplus -(n-i)\Pi_d)$ where $\sqcup$ denotes disjoint union. Recall by property 1 of Lemma 68, that for a convex set S, we have $aS \boxplus bS = (a+b)S$ where $a, b \geq 0$. Then $\mathcal{L}_n^{\leq} = \cup_{b \in B}(\boxplus_{i=1}^{n} b_i \Pi_d) = \sqcup_{i=0}^{n}(i\Pi_d \boxplus -(n-i)\Pi_d)$ where B is the set of vectors in $\{-1, 1\}^n$ which are some -1's (possibly 0) followed by all 1s. Moreover, this is a disjoint union because the elements of $i\Pi_d \boxplus -(n-i)\Pi_d$ have sum of coordinates equal to $(-n+2i)\binom{d}{2}$.

By Lemma 69, we have $Z(A) = \Pi_d - v$, so $i\Pi_d = i(Z(A) + v)$ and $-(n-i)\Pi_d = -(n-i)(Z(A)+v)$. Then by property 2 of Lemma 68, $i\Pi_d \boxplus -(n-i)\Pi_d = Z(iA \oplus -(n-i)A) + (-n+2i)v$, and combining with $\mathcal{L}_n^{\leq} = \sqcup_{i=0}^{n}(i\Pi_d \boxplus -(n-i)\Pi_d)$ yields the result. ■

We use the following result from Ardila et al. (2020), which is a reformulation of Theorem 54 from Shephard (1974) to further decompose the zonotopes into parallelotopes.

Definition 71 *Given matrix M, define $ind(M)$ to be the family of linearly independent subsets of $col(M)$. For a given linearly independent set of vectors S, define the* half-open parallelotope $\square_S = \boxplus_{v \in S}(0, v]$.

Lemma 72 (Proposition 2.2 Ardila et al. (2020)) *The zonotope $Z(M)$ has the disjoint union decomposition $\sqcup_{S \in ind(M)} \square_S$.*

Each of these parallelotopes in the decomposition of $\mathcal{L}_n^{\leq}$ corresponds to an independent set of column vectors of ind$(iA \oplus -(n-i)A)$.

Definition 73 *Let n be a layer index. For $i \in \{0, 1, ..., n\}$, let $C_i = col(iA) \cup col(-(n-i)A)$. Let $C = \cup_{i=0}^{n} C_i$. Given a set of linearly independent column vectors $X \subset C$, define* edge set $e(X)$ *to be the set of column vectors in A equal to some vector in X up to scaling. Note that we can partition the linearly independent sets of C into equivalence classes based on their edge sets.*

For each linearly independent set $Y \in ind(A)$ (including the empty set), let class(Y) denote the edge set equivalence class of Y. For $i \in \{0, 1, ..., n\}$, define Y_i to be the family of independent sets of columns in C_i that are in class(Y). For any subset $S \subset Y$, define $Y_{i,S} \in Y_i$ to be the independent set of columns defined as $Y_{i,S} = (\cup_{w \in S} iw) \cup (\cup_{w \in Y - S} -(n-i)w)$.

Lemma 74 *Let n be a layer index. Then we can decompose $\mathcal{L}_n^{\leq}$ into half-open paralellotopes*

$$\mathcal{L}_n^{\leq} = \sqcup_{i=0}^{n} \sqcup_{Y_{i,S} \in ind(iA \oplus -(n-i)A)} \square'_{Y_{i,S}}$$

where $\square'_{Y_{i,S}} = \square_{Y_{i,S}} + (-n+2i)v = \boxplus_{v \in S}(0, iv]) \boxplus (\boxplus_{v \in Y-S}(0, -(n-i)v]) + (-n+2i)v$.

Note that $i = 0$ forces $S = \emptyset$ since $col(0A)$ is the 0 matrix which has no linearly independent subsets. Similarly, $i = n$ forces $S = Y$. These are the only constraints on the ranges of the variables in the (Y, i, S) tuples.

Proof For each $Y \in \text{ind}(A)$ and each of the $2^{|Y|}$ ways to scale each column of Y by i or $-(n-i)$, if we let $S \subset Y$ denote the set of columns scaled by i, then by Definition 6.11, $Y_{i,S} \in \text{ind}(iA \oplus -(n-i)A)$. Moreover, each element of $\text{ind}(iA \oplus -(n-i)A)$ can be constructed in this way. For each $Y_{i,S}$,

$$\square Y_{i,S} = \boxplus_{v \in Y_{i,S}}(0, v] = (\boxplus_{v \in S}(0, iv]) \boxplus (\boxplus_{v \in Y-S}(0, -(n-i)v]),$$

and by Lemma 72

$$Z(iA \oplus -(n-i)A) + (-n+2i)v = \sqcup_{Y_{i,S} \in \text{ind}(iA \oplus -(n-i)A)} \square_{Y_{i,S}} + (-n+2i)v$$

so applying Lemma 70 yields

$$\begin{aligned} \mathcal{L}_n^{\leq} &= \sqcup_{i=0}^{n} Z(iA \oplus -(n-i)A) + (-n+2i)v \\ &= \sqcup_{i=0}^{n} [\sqcup_{Y_{i,S} \in \text{ind}(iA \oplus -(n-i)A)} \square_{Y_{i,S}} + (-n+2i)v] \\ &= \sqcup_{i=0}^{n} \sqcup_{Y_{i,S} \in \text{ind}(iA \oplus -(n-i)A)} \square'_{Y_{i,S}} \end{aligned}$$

where $\square'_{Y_{i,S}} = \square_{Y_{i,S}} + (-n+2i)v$. ■

The following corollary is immediate from the definition of $P_n^{\leq}$.

Corollary 75 *Let $L(\square'_{Y_{i,S}})$ denote the lattice points of $\square'_{Y_{i,S}}$. Then*

$$P_n^{\leq} = \sqcup_{i=0}^{n} \sqcup_{Y_{i,S} \in ind(iA \oplus -(n-i)A)} L(\square'_{Y_{i,S}})$$

Now, we can show that $P_n^{\leq}$ is the set of lattice points that can be reached by summing n lattice points in $\Pi_d \cup -\Pi_d$, i.e., n points from P_1.

Remark 76 *The rest of this section uses the alias $\square_{Y_{i,S},n} := \square_{Y_{i,S}}$ that emphasizes that $0 \leq i \leq n$. Similarly, $\square'_{Y_{i,S},n} := \square'_{Y_{i,S}}$.*

By Corollary 75, we can index the lattice points of $P_n^{\leq}$ by characterizing the lattice points in each of the $\square'_{Y_{i,S},n}$ parallelotopes.

Lemma 77 *If $1 \leq i \leq n-1$, the lattice points of $\square'_{Y_{i,S},n}$ are indexed by lattice vectors $(c_{v_1}, ..., c_{v_{|Y|}})$ where $Y = \{v_1, ..., v_{|Y|}\}$, $c_{v_k} \in \{1, 2, ..., i\}$ if $v_k \in S$ and $c_{v_k} \in \{-1, -2, ..., -(n-i)\}$ if $v_k \in Y - S$. If $i = 0$, then $c_{v_k} \in \{-1, -2, \ldots, -n\}$ for all $k \in \{1, ..., |Y|\}$. If $i = n$, then $c_{v_k} \in \{1, ..., n\}$ for all $k \in \{1, ..., |Y|\}$. If $Y = \emptyset$, then the only possible lattice vector is the empty vector. In general, the lattice point of $(c_{v_1}, ..., c_{v_{|Y|}})$ is $(-n+2i)v + \sum_{v_k \in Y} c_{v_k} v_k$.*

Proof Suppose $1 \leq i \leq n-1$. Each lattice point $(-n+2i)v + \sum_{v_k \in Y} c_{v_k} v_k$ is distinct by the independence of Y. Furthermore, each $\sum_{v_k \in Y} c_k v_k$ lies in $\square_{Y_{i,S}} = \boxplus_{v \in Y_{i,S}}(0, v]$ because each $c_k v_k$ lies on $(0, iv_k]$ if $v_k \in S$ or on $(0, -(n-i)v_k]$ if $v_k \in Y - S$, so $(-n+2i)v + \sum_{v_k \in Y} c_{v_k} v_k$ lies in $\square'_{Y_{i,S},n}$. There are $i^{|S|}(n-i)^{|Y|-|S|}$ such lattice points, and by Claim 81 $h(Y_{i,S}) = i^{|S|}(n-i)^{|Y|-|S|}$, so these are all of the lattice points of $\square'_{Y_{i,S},n}$.

Suppose $i = 0$. Then $S = \emptyset$ since, by Definition 73, $Y_{0,S} = (\cup_{w \in S} 0w) \cup (\cup_{w \in Y-S} - nw)$ must be an independent set and therefore cannot contain 0. By the same logic as above, the second claim follows. The case $i = n$ is similar. ■

The next result uses this expression for the lattice points of each $\square'_{Y_{i,S},n}$ to show that each point is a sum of lattice points from Π_d and $-\Pi_d$.

Lemma 42 *$P_n^{\leq}$ is the set of lattice points that can be reached by summing n lattice points in $\Pi_d \cup -\Pi_d$.*

Proof

By Lemma 77, each lattice point p of $\square'_{Y_{i,S},n}$ can be written as

$$(-n+2i)v + \sum_{v_j \in S} c_{v_j} v_j + \sum_{v_k \in Y-S} c_{v_k} v_k \tag{6}$$

where $c_{v_j} \in \{1, ..., i\}$ and $c_{v_k} \in \{-1, ..., -(n-i)\}$. We rewrite the second term using indicators as

$$\sum_{v_j \in S} c_{v_j} v_j = \sum_{v_j \in S} \sum_{m=1}^{i} \mathbb{1}_{c_{v_j} \geq m} v_j = \sum_{m=1}^{i} \sum_{v_j \in S, c_{v_j} \geq m} v_j = \sum_{m=1}^{i} z_m$$

where we shorthand $z_m = \sum_{v_j \in S, c_{v_j} \geq m} v_j$. A similar rewriting of the third term in Equation (6) produces

$$\sum_{v_k \in Y-S} c_{v_k} v_k = \sum_{v_k \in Y-S} \sum_{m=1}^{n-i} \mathbb{1}_{c_{v_k} \leq -m} v_k = \sum_{m=1}^{n-i} \sum_{v_k \in Y-S, -c_{v_k} \geq m} -v_k = \sum_{m=1}^{n-i} -z'_m$$

where we shorthand $z'_m = \sum_{v_j \in Y-S, -c_{v_k} \geq m} -v_k$. Finally, using $(-n+2i)v = iv - (n-i)v$, Equation (6) becomes

$$iv - (n-i)v + \sum_{m=1}^{i} z_m + \sum_{m=1}^{n-i} -z'_m = \sum_{m=1}^{i} (z_m + v) - \sum_{m=1}^{n-i} (z'_m + v). \tag{7}$$

Each z_m is a sum over distinct columns in A and is therefore a lattice point in $Z(A)$. Each z'_m is also a lattice point in $Z(A)$ for the same reason. With Lemma 69, we see that the last term of Equation (7) is a sum of i lattice points in Π_d and $n-i$ lattice points in $-\Pi_d$, as desired. ■

Our goal is to eventually sample a point uniformly from $P_n \cap \square'_{Y_{i,S},n}$ as a subroutine for the mechanism. In order to understand which subset of lattice points of $\square'_{Y_{i,S},n}$ belong to P_n, we need to understand the relationship between $P_n^{\leq}$ and the sets $P_i^{\leq}$ for $i < n$. The next result completely characterizes this relationship.

Lemma 43 *$P^{\leq}_{n-2} \subset P^{\leq}_{n}$ for all $n \geq 2$. Moreover, $P^{\leq}_{n-1} \cap P^{\leq}_{n} = \emptyset$ for all $n \geq 1$. Consequently, we have the two nested towers $P^{\leq}_{1} \subset P^{\leq}_{3} \subset ...$ and $P^{\leq}_{0} \subset P^{\leq}_{2} \subset ...$ and $P_n = P^{\leq}_{n} - P^{\leq}_{n-2}$.*

Proof By Lemma 70

$$\mathcal{L}^{\leq}_{n-2} = \sqcup_{i=0}^{n-2} Z(iA \oplus -(n-2-i)A) + (-(n-2)+2i)v := \sqcup_{i=0}^{n-2} Z_{n-2,i}$$
$$\mathcal{L}^{\leq}_{n} = \sqcup_{i=0}^{n} Z(iA \oplus -(n-i)A) + (-n+2i)v := \sqcup_{i=0}^{n} Z_{n,i}$$

where $:=$ denotes that $Z_{n-2,i}$ aliases $Z(iA \oplus -(n-2-i)A) + (-(n-2)+2i)v$, and similar aliasing for for $Z_{n,i}$. We show that $Z_{n-2,i-1} \subset Z_{n,i}$ for all $1 \leq i \leq n-1$. We have

$$Z_{n-2,i-1} = Z((i-1)A \oplus -(n-1-i)A) + (-n+2i)v,$$
$$Z_{n,i} = Z(iA \oplus -(n-i)A) + (-n+2i)v.$$

By Lemma 77 for a given parallelotope of $Y_{i-1,S}$, the lattice points of $\square'_{Y_{i-1,S},n-2}$ can be indexed by a lattice vector $(c_{v_1}, ..., c_{v_{|Y|}})$ where $c_{v_k} \in \{1, 2, ..., i-1\}$ if $v_k \in S$ and $c_{v_k} \in \{-1, -2, ..., -(n-1-i)\}$ if $v_k \in Y - S$. Similarly, given the parallelotope of $Y_{i,S}$, the lattice points of of $\square'_{Y_{i,S},n}$ can be indexed by a lattice vector $(c_{v_1}, ..., c_{v_{|Y|}})$ where $c_{v_k} \in \{1, 2, ..., i\}$ if $v_k \in S$ and $c_{v_k} \in \{-1, -2, ..., -(n-i)\}$ if $v_k \in Y - S$. Clearly, $\square'_{Y_{i-1,S},n-2} \subset \square'_{Y_{i,S},n}$ since the c_{v_k} domains in the former are a subset of those in the latter.

As shown in the proof of Lemma 74 the $\square'_{Y_{i-1,S},n-2}$ parallelotopes (for $1 \leq i \leq n-1$)) partition $Z_{n-2,i-1}$, and the $\square'_{Y_{i,S},n}$ parallelotopes (for $0 \leq i \leq n$) partition $Z_{n,i}$, so $Z_{n-2,i-1} \subset Z_{n,i}$. Thus $\mathcal{L}^{\leq}_{n-2} \subset \mathcal{L}^{\leq}_{n}$, and in particular $P^{\leq}_{n-2} \subset P^{\leq}_{n}$.

As already observed in the proof of Lemma 70, each point in $Z_{n,i}$ has sum of coordinates equal to $(-n+2i)\|v\|_1$. For every pair of integers i and j we have $-n+2i \neq (-(n-1)+2j)$, so $(\sqcup_{i=0}^{n} Z_{n,i}) \cap (\sqcup_{j=0}^{n-1} Z_{n-1,j}) = \emptyset$, and in turn $P^{\leq}_{n-1} \cap P^{\leq}_{n} = \emptyset$.

Then $P^{\leq}_{1} \subset P^{\leq}_{3} \subset ...$ and $P^{\leq}_{2} \subset P^{\leq}_{4} \subset ...$, and the fact that $P_n = P^{\leq}_{n} - P^{\leq}_{n-2}$ follows from this. ■

By the previous result, we have $P_n = P^{\leq}_{n} - P^{\leq}_{n-2}$. We can now explicitly specify the lattice points of $P_n \cap \square'_{Y_{i,S},n}$. Roughly speaking, we do this by using the fact that the lattice points of $\square'_{Y_{i,S},n}$ that contribute to P_n are exactly the lattice points of $\square'_{Y_{i,S},n} - \square'_{Y_{i-1,S},n-2}$.

Corollary 78 *If $Y \neq \emptyset$ and $1 \leq i \leq n-1$, the lattice points of $P_n \cap \square'_{Y_{i,S},n}$ are indexed by lattice vector $(c_{v_1}, ..., c_{v_{|Y|}})$ where $c_{v_k} \in \{1, 2, ..., i\}$ if $v_k \in S$ and $c_{v_k} \in \{-1, -2, ..., -(n-i)\}$ if $v_k \in Y - S$ and at least one of $c_{v_k} = -(n-i)$ or $c_{v_k} = i$ occurs.*

If $Y \neq \emptyset$ and $i \in \{0, n\}$ the lattice points of $P_n \cap \square'_{Y_{i,S},n}$ are equal to the lattice points of $\square'_{Y_{i,S},n}$ enumerated by Lemma 77.

In particular, we have the follow special cases. If $Y = S = \emptyset$, we have $P_n \cap \square'_{Y_{i,S},n} = \{-nv, nv\}$ for $i \in \{0, n\}$ and $P_n \cap \square'_{Y_{i,S},n} = \emptyset$ for $1 \leq i \leq n-1$.

Proof Suppose $i \in \{0, n\}$. The second paragraph of Lemma 43 shows that $Z_{n-2,i-1} \subset Z_{n,i}$ on the range $1 \leq i \leq n-1$. Then $Z_{n,0} \cup Z_{n,n} \subset (\sqcup_{i=0}^{n} Z_{n,i} - \sqcup_{i=1}^{n-1} Z_{n-2,i-1})$. It follows that if $i \in \{0, n\}$ then the lattice points of $P_n \cap \square'_{Y_{i,S},n}$ are equal to the lattice points of $\square'_{Y_{i,S},n}$ enumerated by Lemma 77.

Suppose $1 \leq i \leq n-1$. Then by the second paragraph of the proof of Lemma 43, we see that the lattice points of $\square'_{Y_{i,S},n}$ that contribute to $P_n = P_n^{\leq} - P_{n-2}^{\leq}$ are exactly the lattice points of $\square'_{Y_{i,S},n} - \square'_{Y_{i-1,S},n-2}$, which are lattice vectors for which at least one of $c_{v_k} = -(n-i)$ occurs or $c_{v_k} = i$ occurs.

Suppose $Y = S = \emptyset$ and $1 \leq i \leq n-1$. Since the lattice point of the empty lattice vector $()$ is $(-n+2i)v + \sum_{v_k \in Y} c_{v_k} v_k = (-n+2i)v$ by Lemma 77, then the lattice points of $\square'_{Y_{i,S},n}$ and $\square'_{Y_{i-1,S},n-2}$ are the singleton set $\{(-n+2i)v\}$ so there are no lattice points in $\square'_{Y_{i,S},n} - \square'_{Y_{i-1,S},n-2}$.

Suppose $Y = S = \emptyset$ and $i \in \{0, n\}$. then the only lattice point of $\square'_{Y_{i,S},n} - \square'_{Y_{i-1,S},n-2}$ is the singleton set $\{(-n+2i)v\}$ by the same logic as in the first paragraph of this proof. ■

We wish to sample a point uniformly from P_n. To do so, we decompose P_n up into smaller parts that are easy to sample with the appropriate weights. The following result about $|P_n^{\leq}|$ starts this analysis, though it requires some setup.

We use the following result from Stanley (1983), noting briefly that the original result uses a lemma without proof. For completeness, we prove this result as Lemma 79, though it is not directly relevant to this paper.

Lemma 79 *Let L be a rank-d sublattice of $\mathbb{Z}^d$, and let M be an integer $d \times m$ matrix whose columns generate L. Then, the subgroup index of L is the GCD of the volumes of all the lattice d-simplices of M.*

Proof Let G be a group. For a subgroup H of G, define the *subgroup index* $[G : H]$ to be the number of cosets of H. For two groups G_1, G_2 define $G_1 \oplus G_2$ to be the direct sum of the two groups.

A lattice L in $\mathbb{Z}^d$ is an additive subgroup of $\mathbb{Z}^d$. Suppose M is a $d \times m$ matrix of rank d whose columns generate a lattice L. A *lattice d-simplex* of M is any $d \times d$ submatrix of M whose columns generate L.

Let Y be a $d \times d$ matrix whose columns form a basis for L. We show that $[\mathbb{Z}^d : L] = |\det(Y)|$. Since y is an integer matrix, there exist P, Q be invertible integer $d \times d$ matrices such that PYQ is a diagonal matrix in smith normal form with positive diagonal entries $a_1|a_2|...|a_d$ where $a_i|a_{i+1}$ means that a_i divides a_{i+1}. Let $Y' = YQ$. Note that P^{-1} is a matrix whose columns form some other basis for $\mathbb{Z}^d$ and Y' is a matrix whose columns form some other basis for L. Then $P^{-1}(PYQ) = Y'$.

Note that L is automatically normal because $\mathbb{Z}^d$ is abelian, so all subgroups of it are normal. So the quotient group $\mathbb{Z}^d/L$ is well defined. Since $P^{-1}(PYQ) = Y'$, then $\text{col}(Y') = \{a_1 p_1, ..., a_d p_d\}$ where $\text{col}(P^{-1}) = \{p_1, ..., p_d\}$. Since $\{a_1 p_1, ..., a_d p_d\}$ are a basis for L and $\{p_1, ..., p_d\}$ is a basis for $\mathbb{Z}^d$, it follows that $\mathbb{Z}^d/L \cong \mathbb{Z}_{a_1} \oplus ... \oplus \mathbb{Z}_{a_d}$. Moreover, $\det(PYQ) = a_1...a_d$ since PYQ is diagonal. Since $|\det(P)| = |\det(Q)| = 1$, then $|\det(Y)| = |\det(PYQ)| = a_1...a_d = |\mathbb{Z}^d/L| = [\mathbb{Z}^d : L]$.

Let g be the GCD of all $d \times d$ minors of M. We show that $g = \det(Y)$. First, we show that $|\det(Y)|$ divides g. Let M' be a full-rank $d \times d$ submatrix of M. Then $M' = YT$ for some $d \times d$ integer matrix T since the columns of M' are in L, i.e. each of these columns can be expressed as a linear combination of columns of Y. Then $\det(M') = \det(Y)\det(T)$, and since $\det(T)$ is an integer then $|\det(Y)|$ divides $|\det(M')|$, and since M' was arbitrary, $|\det(Y)|$ divides g.

Second, we show that g divides $|\det(Y)|$. There exists an $m \times d$ integer matrix R such that $Y = MR$ since the columns of M span L. By Cauchy-Binet, $\det(Y) = \sum_{s \in S} \det(M_s)\det(R_s)$ where S is the collection of size-d subsets of $\{1, ..., n\}$, M_s is the submatrix formed by taking the columns of M corresponding to s, and R_s is the submatrix formed by taking the rows of R

corresponding to s. Since g divides each $|\det(M_s)|$, and $|\det(R_s)|$ is an integer, then g divides $|\det(Y)|$. It follows that $g = |\det(Y)| = [\mathbb{Z}^d : L]$. ■

As in the Sum ripple mechanism, we use the Ehrhart polynomial to count lattice points in different sets.

Lemma 80 (Theorem 2.2 Stanley (1983)) *Let M be a $d \times m$ matrix. Then the Ehrhart polynomial (Definition 19) $E(Z(M), q) = \sum_X h(X) q^{|X|}$ where X ranges over all independent subsets of the columns of M and $h(X)$ is the GCD of all minors of size $|X|$ of the $d \times |X|$ matrix whose columns are X.*

This brings us to Lemma 45, which connects $P_n^{\leq}$ and Q_k.

Lemma 45 $|P_n^{\leq}| = (n+1) \sum_{k=0}^{d-1} |Q_k| n^k$ *where* $0^0 = 1$ *for* $n = 0$.

Proof Recall that $P_n^{\leq}$ is the set of lattice points in $\mathcal{L}_n^{\leq}$. Lemma 70 decomposes $\mathcal{L}_n^{\leq}$ into a disjoint union of translated zonotopes, so $P_n^{\leq}$ is the disjoint union of their sets of lattice points, and $|P_n^{\leq}| = \sum_{i=0}^{n} E(Z(iA \oplus -(n-i)A) + (n-2i)v, 1)$ Since translation by a lattice point does not change the number of lattice points covered by $Z(iA \oplus -(n-i)A)$, we get $|P_n^{\leq}| = \sum_{i=0}^{n} E(Z(iA \oplus -(n-i)A), 1)$.

Consider $Z(iA \oplus -(n-i)A)$. Take any independent set $X \subset \text{col}(iA) \cup \text{col}(-(n-i)A)$. Let A_X be the $d \times |X|$ submatrix of $iA \oplus -(n-i)A$ whose columns are the elements of X. Our goal is to determine the GCD of all minors of size $|X|$ of A_X so that we can apply Lemma 80.

Suppose $A_{X,R}$ is the $|X| \times |X|$ minor formed by taking the indices in $R \subset \{1, ..., d\}$ where $|R| = |X|$. Let $G(A_{X,R})$ be the undirected graph on d vertices $\{1, 2..., d\}$ such that (i, j) is an edge if and only if there is a vector in $\text{col}(A_{X,R})$ with two non-zero entries exactly at row indices $i, j \in R$ where these row indices are labeled with respect to their row index in the ambient matrix A. Then $G(A_{X,R})$ induces a matroid whose elements are the edges of $G(A_{X,R})$ and whose independent sets are the forests of $G(A_{X,R})$. Throughout the rest of the proof, we abuse notation and use $A_{X,R}$ to mean this matroid. By doing so, we can pass between talking about cycles of the matroid while also talking about the matrix properties of $A_{X,R}$ without cluttering notation.

For each $Y \in \text{ind}(A)$ and each of the $2^{|Y|}$ ways to scale each column of Y by i or $-(n-i)$, if we let $S \subset Y$ denote the set of columns scaled by i, then by Definition 6.11, $Y_{i,S} \in \text{ind}(iA \oplus -(n-i)A)$. Moreover, each element of $\text{ind}(iA \oplus -(n-i)A)$ can be constructed in this way. Then by Lemma 80,

$$\begin{aligned} |P_n^{\leq}| = \sum_{i=0}^{n} E(Z(iA \oplus -(n-i)A), 1) &= \sum_{i=0}^{n} \sum_{X \subset \text{ind}(iA \oplus -(n-i)A)} h(X) \\ &= \sum_{Y \in \text{ind}(A)} \sum_{i=0}^{n} \sum_{y \in Y_i} h(y) \\ &= \sum_{Y \in \text{ind}(A)} \sum_{i=0}^{n} \sum_{S \subset Y} h(Y_{i,S}). \end{aligned}$$

Our goal now is to simplify the expression $h(Y_{i,S})$.

Claim 81 *For any independent set $X \subset col(iA) \cup col(-(n-i)A)$, we have $h(X) = i^j(n-i)^{|X|-j}$ where j is the number of columns of X in $col(iA)$. In particular, $h(Y_{i,S}) = i^{|S|}(n-i)^{|Y|-|S|}$ for all $Y \in ind(A), 0 \leq i \leq n, S \subset Y$.*

Proof If $A_{X,R}$ has a cycle or an all 0s row, then it is a dependent set so $\det(A_{X,R}) = 0$ and this minor does not contribute to calculating the GCD since every integer divides 0.

If $A_{X,R}$ is acyclic and has no all 0s row, we show that $|\det(A_{X,R})| = i^j(n-i)^{|X|-j}$ where j is the number of columns of X in col(iA) and $|X| - j$ is the number of columns of X in col$(-(n-i)A)$. Let $A'_{X,R}$ be formed from $A_{X,R}$ by scaling the columns of X in col(iA) by i^{-1} and scaling the columns of X in col$(-(n-i)A)$ by $(n-i)^{-1}$. We show that $\det(A'_{X,R}) = \pm 1$ by induction on the number of vertices. Since $A'_{X,R}$ is a forest, it must have some leaf, i.e. there is a row index r with only one non-zero entry coordinate, equal to ± 1, at column c and the rest of the coordinates are 0. We will compute the determinant using minors and cofactors along r. We have $\det(A'_{X,R}) = (-1)^{r+c}(\pm 1)\det(A'_{X,R}[r,c])$ where $A'_{X,R}[r,c]$ is the minor of $A'_{X,R}$ formed by removing row r and column c. This corresponds to deleting the vertex r from the matroid, so the resulting graph is still acyclic, but on one fewer vertex. By induction, $\det(A'_{X,R}[r,c]) = \pm 1$. It remains to prove the base case where $d = 2$. In this case, there is only one column in A, namely $(1,-1)^T$, so $X = \{(1,-1)^T\}$ and clearly each 1×1 minor of X has determinant ± 1.

Then $\det(A_{X,R}) = i^j(n-i)^{|X|-j}\det(A'_{X,R})$ since the determinant is an alternating form in its columns, meaning that it is multilinear in its arguments. This is true for all minors of A_X, so $h(X) = i^j(n-i)^{|X|-j}$. ■

Using Claim 81 gives

$$\sum_{Y \in \text{ind}(A)} \sum_{i=0}^{n} \sum_{S \subset Y} h(Y_{i,S}) = \sum_{Y \in \text{ind}(A)} \sum_{i=0}^{n} \sum_{j=0}^{|Y|} \binom{|Y|}{j} i^j (n-i)^{|Y|-j} \tag{8}$$

$$= \sum_{Y \in \text{ind}(A)} \sum_{i=0}^{n} (i + (n-i))^{|Y|}$$

$$= (n+1) \sum_{Y \in \text{ind}(A)} n^{|Y|} \tag{9}$$

$$= (n+1) \sum_{k=0}^{d-1} |Q_k| n^k \tag{10}$$

where the last equality uses the interpretation of each set $Y \in \text{ind}(A)$ as a forest with an edge for each column. ■

.

We can apply it to compute several further cardinalities needed for sampling.

Definition 82 *Define $R_{n,k}$ to be the set of lattice points that lie in any set $P_n \cap \square'_{Y_{i,S},n}$ where the columns of Y correspond to a member of Q_k.*

We will show that we have the decomposition $|P_n| = \sum_{k=0}^{d-1} |R_{n,k}|$. Using this, we will be able to sample a set Q_k where $0 \leq k \leq d-1$ with weights proportional to $|R_{n,k}|$, and this will be the first step in sampling one of the sets $P_n \cap \square'_{Y_{i,S},n}$ with the appropriate weight.

Lemma 83 (Modification of Lemma 46) $|P_0| = |P_0^{\leq}| = 1$, $|P_1| = |P_1^{\leq}| = 2\sum_{k=0}^{d-1}|Q_k|$, *and for* $n \geq 2$, $|P_n| = \sum_{k=0}^{d-1}|Q_k|((n+1)n^k - (n-1)(n-2)^k)$. *In particular,* $|P_n| = \sum_{k=0}^{d-1}|R_{n,k}|$ *where* $|R_{n,k}| = |Q_k|((n+1)n^k - (n-1)(n-2)^k)$.

Proof We have $P_n = P_n^{\leq}$ for $n \in \{0,1\}$ since $P_0^{\leq} \cap P_1^{\leq} = \emptyset$ by Lemma 43. Then by Lemma 45, $|P_0| = |P_0^{\leq}| = 1$, $|P_1| = |P_1^{\leq}| = 2\sum_{k=0}^{d-1}|Q_k|$, and for $n \geq 2$,

$$\begin{aligned}|P_n| &= |P_n^{\leq}| - |P_{n-2}^{\leq}| \\ &= (n+1)\sum_{k=0}^{d-1}|Q_k|n^k - (n-1)\sum_{k=0}^{d-1}|Q_k|(n-2)^k \\ &= \sum_{k=0}^{d-1}|Q_k|((n+1)n^k - (n-1)(n-2)^k).\end{aligned}$$

Equation (9) and Equation (10) in the proof of Lemma 45 established that

$$|P_n^{\leq}| = (n+1)\sum_{Y \in \text{ind}(A)} n^{|Y|} = (n+1)\sum_{k=0}^{d-1}|Q_k|n^k$$

so each term $(n+1)|Q_k|n^k$ counts the contribution to $|P_n^{\leq}|$ of lattice points indexed by any $Y \in \text{ind}(A)$ where, using the graph interpretation of a matrix from Lemma 45, $\text{col}(Y) \in Q_k$. Similarly, in $|P_{n-2}^{\leq}| = (n-1)\sum_{k=0}^{d-1}|Q_k|(n-2)^k$, each $(n-1)(n-2)^k|Q_k|$ counts the contribution to $|P_{n-2}^{\leq}|$ of the lattice points indexed by any $Y \in \text{ind}(A)$ where $\text{col}(Y) \in Q_k$. In particular, $|Q_k|((n+1)n^k - (n-1)(n-2)^k)$ counts the contribution of lattice points that lie in a set $P_n \cap \square'_{Y_{i,S},n}$ where $\text{col}(Y) \in Q_k$ since $\square'_{Y_{i-1,S},n-2} \subset \square'_{Y_{i,S},n}$ for $1 \leq i \leq n-1$. It follows that $|R_{n,k}| = |Q_k|((n+1)n^k - (n-1)(n-2)^k)$. ■

We can now compute the normalizing constant, as shown in Lemma 47.

Lemma 84 (Modification of Lemma 47)

$$\begin{aligned}\mathcal{Z}(|P_n^{\leq}|)[z^{-1}] = 1 &+ \frac{z}{(1-z)}\left(1 + \frac{1}{1-z}\right) \\ &+ \sum_{k=1}^{d-1}|Q_k|(1-z)^{-k-2}\left[\sum_{j=0}^{k}\mathcal{A}(k+1,j)z^{j+1} + (1-z)\sum_{j=0}^{k-1}\mathcal{A}(k,j)z^{j+1}\right]\end{aligned}$$

where $\mathcal{A}(k,j)$ *is the Eulerian number defined as the number of permutations on* $\{1,...,k\}$ *with* j *descents.*

Moreover, $\mathcal{Z}(|P_n|)[z^{-1}] = (1-z^2)\mathcal{Z}(|P_n^{\leq}|)[z^{-1}]$.

Proof By the first equation in section 2 and the first equation in section 3 in Luschny (2010) we have

$$\sum_{n \geq 1} n^k z^n = (1-z)^{-k-1}\sum_{j=0}^{k-1}\mathcal{A}(k,j)z^{j+1} \tag{11}$$

In the following calculation, we need to split off the $k = 0$ term since the above equation does not make sense for $k < 1$.

$$\mathcal{Z}(|P_n^{\leq}|)[z^{-1}] = \sum_{n\geq 0} |P_n^{\leq}| z^n = 1 + \sum_{n\geq 1}(n+1)\sum_{k=0}^{d-1} |Q_k| n^k z^n$$

by Lemma 83. Then we expand the series as

$$\begin{aligned}
\sum_{n\geq 1}(n+1)\sum_{k=0}^{d-1}|Q_k|n^k z^n &= \sum_{n\geq 1}\left[\sum_{k=0}^{d-1}|Q_k|n^{k+1}z^n + \sum_{k=0}^{d-1}|Q_k|n^k z^n\right] \\
&= \sum_{k=0}^{d-1}|Q_k|\sum_{n\geq 1} n^{k+1}z^n + \sum_{k=0}^{d-1}|Q_k|\sum_{n\geq 1} n^k z^n \\
&= \sum_{n\geq 1} n z^n + \sum_{n\geq 1} z^n + \sum_{k=1}^{d-1}|Q_k|\sum_{n\geq 1} n^{k+1}z^n + \sum_{k=1}^{d-1}|Q_k|\sum_{n\geq 1} n^k z^n \quad (12)
\end{aligned}$$

since $|Q_0|$, the number of forests on d vertices with 0 edges, is 1.

By using the formula for geometric series, the derivative of geometric series, and Equation (11), we obtain

$$\begin{aligned}
(12) &= \frac{z}{(1-z)^2} + \frac{z}{(1-z)} + \sum_{k=1}^{d-1}|Q_k|\left[(1-z)^{-k-2}\sum_{j=0}^{k}\mathcal{A}(k+1,j)z^{j+1} + (1-z)^{-k-1}\sum_{j=0}^{k-1}\mathcal{A}(k,j)z^{j+1}\right] \\
&= \frac{z}{(1-z)}\left(1 + \frac{1}{1-z}\right) + \sum_{k=1}^{d-1}|Q_k|(1-z)^{-k-2}\left[\sum_{j=0}^{k}\mathcal{A}(k+1,j)z^{j+1} + (1-z)\sum_{j=0}^{k-1}\mathcal{A}(k,j)z^{j+1}\right]
\end{aligned}$$

Next, we express $\mathcal{Z}(|P_n|)[z^{-1}]$ in terms of $\mathcal{Z}(|P_n^{\leq}|)[z^{-1}]$. Recall from Definition 40 that for $n \in \{0, 1\}$, we have $P_n = P_n^{\leq}$. Then, by the same result,

$$\begin{aligned}
\mathcal{Z}(|P_n|)[z^{-1}] &= \sum_{n\geq 0}|P_n|z^n \\
&= |P_0| + |P_1|z + \sum_{n=2}^{\infty}(|P_n^{\leq} - P_{n-2}^{\leq}|)z^n \quad (13)
\end{aligned}$$

by Lemma 43. Then

$$\begin{aligned}
(13) &= |P_0| + |P_1|z + \sum_{n=2}^{\infty}|P_n^{\leq}|z^n - \sum_{n=2}^{\infty}|P_{n-2}^{\leq}|z^n \\
&= |P_0| + |P_1|z + (\mathcal{Z}(|P_n^{\leq}|)[z^{-1}] - |P_0^{\leq}| - |P_1^{\leq}|z) - z^2\sum_{n=2}^{\infty}|P_{n-2}^{\leq}|z^{n-2} \\
&= (1-z^2)\mathcal{Z}(|P_n^{\leq}|)[z^{-1}].
\end{aligned}$$

■

We now turn to computation of the F table described in Section 6.2.

Lemma 85 (of Integer Sequences (OEIS)) *Let $F_{d,m}$ denote the number of forests on d labeled vertices with exactly m components. Then $F_{d,m} = \sum_{k=1}^{d-m+1} \binom{d-1}{k-1} k^{k-2} F_{d-k,m-1}$ where $F_{0,0} = 1$ and $F_{d,0} = 0$ for $d > 1$.*

Proof The base cases are obvious. In the remaining cases, the vertex labeled 1 is in some tree T_1. Suppose $|T_1| = k$ where $1 \leq k \leq d - m + 1$. We can pick the vertices that belong to T_1 by choosing $k-1$ out of the vertices in $\{2, ..., d\}$ in addition to the vertex 1. There are k^{k-2} trees on the k labeled vertices of T_1. For each of these possible trees, we can extend it to a forest on all d labeled vertices by combining it with a forest on $\{1, ..., d\} - V(T_1)$ with $m-1$ components, and there are $F_{d-k,m-1}$ such forests. ■

The next result describes how to compute F and use it to sample a forest. The forest sampling will be relevant to a sampling subroutine to come.

Lemma 86 (Modification of Lemma 48) *We can compute the $F_{d,m}$ table for all $m \in \{1, ...d\}$ in $O(d^2 \log(d))$. After computing this table, there is an algorithm to uniformly sample a forest on d labeled vertices with m components in time $O(d^2)$.*

Proof First, we compute the $F_{d,m}$ table. An exponential generating function (EGF) for sequence $a_1, a_2, \ldots$ is a function $E(x) = \sum_{k=1}^{\infty} a_k \frac{x^k}{k!}$. Cayley's formula says that the number of trees on d labeled vertices is d^{d-2}, so its EGF is $T(x) = \sum_{k=1}^{\infty} t_k \frac{x^k}{k!}$ where $t_k = k^{k-2}$. Consider $T(x)^m$. Then its coefficient for $\frac{x^d}{d!}$ is made up of every way to pick nonnegative integers $(k_1, ..., k_m)$ that sum to d, multiplied by the number of resulting ways to partition d and the product of the coefficients,

$$\sum_{k_1+...+k_m=d} \binom{d}{k_1, \ldots, k_m} t_{k_1} \cdots t_{k_m}.$$

Each permutation of a tuple $(k_1, ..., k_m)$ in the summation index counts the same set of forests with m trees, so we must divide by $m!$ to correct for this over-count. Thus the coefficients $F_m(x) = \frac{T(x)^m}{m!} = \sum_{k=1}^{\infty} F_{k,m} \frac{x^k}{k!}$ are column m of the desired table. We also have

$$F_m(x) = \frac{1}{m} \left[\frac{1}{(m-1)!} T(x)^{m-1} \right] T(x) = \frac{1}{m} \left[F_{m-1}(x) T(x) \right],$$

so, after directly computing the first column using Cayley's formula in time $O(d^2)$, each column of $F_{d,m}$ is a convolution of the last column's generating function and $T(x)$. Each such convolution can be done in time $O(d \log(d))$ using FFTs, so it takes time $O(d^2 \log(d))$ to compute the whole table.

Once the table is computed, we can sample a number of components m for the forest proportional to $F_{d,m}$. It remains to sample a forest on d labeled vertices with m components. We start by computing the binomial coefficients in the dth row of Pascal's triangle in time $O(d^2)$ and then computing the values k^{k-2} for $1 \leq k \leq d$ in time $O(d^2)$.

Once these are computed, we pick the size k of the tree T_1 containing node 1 according to the weights $\{\binom{d-1}{k-1} k^{k-2} F_{d-k,m-1}\}_{k=1}^{d-m+1}$ specified by Lemma 85, which we construct in time $O(d)$.

Next, sample a random permutation of $\{2, ..., d\}$ in time $O(d)$ (by, e.g., Fisher-Yates shuffle) and pick the first $k-1$ vertices of the permutation to be the remaining nodes of T_1. Next we randomly sample the edges of T_1. A Prüfer sequence is a vector of the form $\{v_1, ..., v_k\}^{k-2}$. We randomly sample a Prüfer sequence in $\{v_1, ..., v_k\}^{k-2}$. There is a bijection between the set of all such sequences and the labeled trees on $\{v_1, \ldots, v_k\}$, and we convert the sampled sequence into a labeled tree T_1 in time $O(d)$ Wang et al. (2009).

We then combine this with a recursively uniformly sampled forest on $\{1, ..., d\} - V(T_1)$ with $m-1$ components. Each recursion reuses the binomial coefficients and values k^{k-2} computed at the beginning of the sampling process, so the overall time is $O(d^2)$. ■

This brings us to the final result necessary for our sampler. It shows that we can compute $|P_n|$ (for the weights w_n in reservoir sampling) and sample P_n fast.

Lemma 49 *Given $F_{d,m}$ for $1 \leq m \leq d$, for any $n \geq 0$, we can compute $|P_n|$ in time $O(d)$ and sample a point uniformly from P_n in time $O(d^2 \log(d))$.*

Proof By Lemma 83, we have $|P_n| = \sum_{k=0}^{d-1} |R_{n,k}|$ where $|R_{n,k}| = |Q_k|((n+1)n^k - (n-1)(n-2)^k)$. As $|Q_k| = F_{d,d-k}$, we can compute $|P_n|$ in $O(d)$. Using $|P_n|$, we can form the $w_n = \frac{|P_n|e^{-\varepsilon n}}{N}$ weights for reservoir sampling.

Suppose we have sampled a layer index n. Our goal is to sample a point uniformly from P_n. Since we decomposed P_n into parts $P_n \cap \square'_{Y_{i,S},n}$ that are indexed by elements of $\text{ind}(A)$, the first object that we sample is an element $Y^* \in \text{ind}(A)$.

Claim 87 *We can sample an element $Y^* \in \text{ind}(A)$ in time $O(d^2)$.*

Proof Lemma 83 established that, for any $k \in \{0, 1, \ldots, d-1\}$, the elements $Y \in \text{ind}(A)$ with $\text{col}(Y) \in Q_k$ in total contribute $|R_{n,k}|$ lattice points to P_n. We therefore sample one of these Q_k proportional to

$$|R_{n,k}| = |Q_k|((n+1)n^k - (n-1)(n-2)^k) = F_{d,d-k}((n+1)n^k - (n-1)(n-2)^k)$$

since any forest on d vertices with k edges has $d-k$ components. We have already computed each $F_{d,d-k}$, so this takes time $O(d)$.

Suppose we choose Q^*. Next, we sample a forest on $\{1, ..., d\}$ from Q^* which we can do in $O(d^2)$ by Lemma 86. We can convert each edge of this forest into a column of $\text{col}(A)$ where the edge (i, j) corresponds to the column $e_i - e_j \in \text{col}(A)$. Let $Y^* \in \text{ind}(A)$ be this set of columns. ■

If $Y^* = \emptyset$ then we can return one of the 2 possible lattice points $\{-nv, nv\}$ uniformly at random according to Corollary 78, and this is the desired sample point from P_n. If instead $Y \neq \emptyset$, our ultimate goal is to sample a lattice point from some $P_n \cap \square'_{Y_{i,S},n}$, so we need to sample an appropriate index i.

Claim 88 *We can sample an index $i^* \in \{0, ..., n\}$ corresponding to one of the elements of $\{Y_i^*\}_{i=0}^n$ in time $O(\log(n))$.*

Proof Recall from Definition 73 that Y_i^* is the family of independent sets of $C_i = \text{col}(iA) \cup \text{col}(-(n-i)A)$ that are in the class of Y^*. We want to sample one of the families of $\{Y_i^*\}_{i=0}^n$ with

the appropriate weighting. For $0 \le i \le n$, let $P^{\le}_{n,Y^*,i}$ (resp. $P_{n,Y^*,i}$) denote the number of lattice points contributed by Y^*_i to $P^{\le}_n$ (resp. P_n).

First, for $1 \le i \le n-1$, the lattice points that contribute to $P_{n,Y^*,i}$ belong to $Z_{n,i} - Z_{n-2,i-1}$, which was defined to be $Z_{n,i} = Z(iA \oplus -(n-i)A) + (-n+2i)v$ in Lemma 43. By Equation (9) in the proof of Lemma 45, $|P^{\le}_{n,Y^*,i}| = n^{|Y^*|}$. The proof of Lemma 43 established that $Z_{n-2,i-1} \subset Z_{n,i}$, so

$$\begin{aligned}|P_{n,Y^*,i}| &= |P^{\le}_{n,Y^*,i} - P^{\le}_{n-2,Y^*,i-1}| \\ &= |P^{\le}_{n,Y^*,i}| - |P^{\le}_{n-2,Y^*,i-1}| \\ &= n^{|Y^*|} - (n-2)^{|Y^*|}.\end{aligned}$$

Second, for $i \in \{0, n\}$, all possible lattice vectors contribute to P_n by Corollary 78, so $|P_{n,Y^*,i}| = n^{|Y^*|}$.

Now, we can sample an index $i^* \in \{0, 1, ..., n\}$ according to weights $\{P_{n,Y^*,i}\}_{i=0}^n$. The weights for $i \in \{0, n\}$ are equal, and the weights for $i \in \{1, ..., n-1\}$ are equal, so we can sample one of these two groups of indices in constant time. If we pick the group $\{0, n\}$ we can sample of the two indices in constant time, and if we pick the group $\{1, ..., n-1\}$ we can sample of the $n-1$ indices in $O(\log(n))$ time by repeated subdivision. ■

It remains to sample the S component of the eventual $P_n \cap \square'_{Y_{i,S},n}$.

Claim 89 *We can sample some subset $S^* \subset Y^*$ corresponding to $Y^*_{i^*,S^*}$ in time $O(d)$.*

Proof Let $Y^*_{i^*,j} = \{Y_{i,S} : S \subset Y^*, |S| = j\}$. For $0 \le j \le |Y^*|$, let M_j be the number of lattice points contributed by $Y^*_{i^*,j}$ to P_n. We will first sample j^* with weight proportional to M_{j^*}. If $i^* = 0$, then by the reasoning at the end of the proof of Lemma 77, $|S| = 0$, so the only option is $j^* = 0$. Similarly, if $i^* = n$, then $j^* = n$.

If $1 \le i^* \le n-1$, we have $Z_{n-2,i^*-1} \subset Z_{n,i^*}$ so we can use Equation (8) to obtain $M_j = \binom{|Y^*|}{j}(i^*)^j(n-i^*)^{|Y^*|-j} - \binom{|Y^*|}{j}(i^*-1)^j(n-1-i^*)^{|Y^*|-j}$ for $0 \le j \le |Y^*|$. In this case, we sample j^* according to the M_j weights, which can be computed in time $O(d)$.

Each element of $Y^*_{i^*,j^*}$ contributes an equal number of lattice points so we can pick S^* by randomly permuting $\{1, ..., |Y^*|\}$ in $O(d)$ and choosing the first j^* indices to be S^*. ■

The last step is sampling a point from the chosen set $P_n \cap \square_{Y^*_{i^*,S^*},n}$.

Claim 90 *We can sample a lattice point from $P_n \cap \square_{Y^*_{i^*,S^*},n}$ in time $O(d^2)$.*

Proof

We split into cases, though all are essentially direct applications of Corollary 78. In each case, the lattice points are indexed by lattice vectors $(c_{v_1}, \ldots, c_{v_{|Y^*|}})$, so we sample from these lattice vectors. The first several cases are distinct but simple.

1. If $i^* = 0$, then the lattice points of $P_n \cap \square_{Y^*_{i^*,S^*},n}$ are the lattice points of $\square_{Y^*_{i^*,S^*},n}$. By Lemma 77, each $c_{v_k} \in \{-1, -2, \ldots, -n\}$.

2. If $i^* = n$, then the lattice points of $P_n \cap \square_{Y^*_{i^*,S^*},n}$ are the lattice points of $\square_{Y^*_{i^*,S^*},n}$. By Lemma 77, each $c_{v_k} \in \{1, 2, \ldots, n\}$.

3. If $1 \leq i^* \leq n-1$, then for each $v_k \in S^*$ we have $c_{v_k} \in \{1, 2, \ldots, i^*\}$, for each $v_k \notin S^*$ we have $c_{v_k} \in \{-1, -2, \ldots, -(n-i^*)\}$, and at least one of $c_{v_k} = i^*$ and $c_{v_k} = -(n-i^*)$ occurs.

   (a) If $i^* = 1$, and $S^* \neq \emptyset$ (i.e., $j^* \geq 1$), then for each $v_k \in S^*$, $c_{v_k} = 1$. Since this satisfies $c_{v_k} = i^*$, we uniformly at random sample the remaining sublattice vector from $\{-1, -2, \ldots, -(n-1)^*\}^{|Y^*|-j^*}$.

   (b) Similarly, if $i^* = n-1$ and $S^* \neq Y^*$ (i.e., $j^* \leq n^* - 1$), then for each $v_k \in Y^* - S^*$, $c_{v_k} = -1$, and we uniformly at random sample the random sublattice vector from $\{1, 2, \ldots, i^*\}^{j^*}$.

In the remaining cases, there is no v_k for which we know $c_{v_k} \in \{i^*, -(n-i^*)\}$. To sample a lattice vector in these cases, we populate a matrix P such that for $0 \leq x \leq j^*$ and $0 \leq y \leq |Y^*| - j^*$, entry $P_{x,y}$ is the number of valid lattice vectors with x coordinates where $c_{v_k} = i^*$ and y coordinates where $c_{v_k} = -(n-i^*)$. Then $P_{0,0} = 0$ and for the remaining x and y we have $P_{x,y} = \binom{j^*}{x}\binom{|Y^*|-j^*}{y}(i^*-1)^{j^*-x}(n-i^*-1)^{|Y^*|-j^*-y}$, as we choose x of the $v_k \in S^*$ to have $c_{v_k} = i^*$ and the rest to lie in $\{1, 2, ..., i^*-1\}$ and choose y of the $v_k \in Y^* - S^*$ to have $c_{v_k} = -(n-i)^*$ and the rest to lie in $\{-1, -2, \ldots, -(n-i^*-1)\}$. Sampling according to the weights in P according to the weights takes time $O(d^2)$. We then sample the remaining sublattice vectors uniformly at random as in 3a) and 3b) above.

This fully specifies $(c_{v_1}, ..., c_{v_{|Y^*|}})$ in all cases. We then return the lattice point $(-n+2i^*)v + \sum_{v_k \in Y^*} c_{v_k} v_k$ in time $O(d)$. ■

■

The penultimate proof required for the Vote ripple mechanism is the hitting time analysis for its reservoir sampling.

Lemma 50 *The hitting time random variable for reservoir sampling, T, satisfies $\mathbb{E}[T] = O(\frac{d}{1-e^{-\varepsilon}})$.*

Proof Recall by Lemma 84

$$\begin{aligned}
\mathcal{Z}(|P_n|)[z^{-1}] &= (1-z^2)\mathcal{Z}(|P_n^{\leq}|)[z^{-1}] \\
&= \frac{z(1-z^2)}{(1-z)}\left(1 + \frac{1}{1-z}\right) + (1-z^2)\sum_{k=1}^{d-1}|Q_k|(1-z)^{-k-2}\left[\sum_{j=0}^{k}\mathcal{A}(k+1,j)z^{j+1} + (1-z)\sum_{j=0}^{k-1}\mathcal{A}(k,j\right. \\
&= z + \frac{z}{1-z} + z^2 + \frac{z^2}{1-z} + (1+z)\sum_{k=1}^{d-1}|Q_k|(1-z)^{-k-1}\left[\sum_{j=0}^{k}\mathcal{A}(k+1,j)z^{j+1} + (1-z)\sum_{j=0}^{k-1}\mathcal{A}(k,j)\right. \\
&= \sum_{s=0}^{d}\sum_{w=0}^{d+1}B_{s,w}z^w(1-z)^{-s}
\end{aligned}$$

Algorithm 3 Vote Sampler

Input: Dimension d
Compute the normalizing constant N using Lemma 84
Compute $F_{d,m}$ table for $1 \leq m \leq d$ using Lemma 86
Initialize layer index $n = 0$
while True **do**
  Compute $|P_n|$ using Lemma 83
  Flip reservoir sampling coin with heads probability $\frac{w_n}{1-\sum_{j=0}^{n-1} w_n}$
  if heads **then**
    break
  else
    $n \leftarrow n + 1$
Sample $Y^* \in \text{ind}(A)$ using the $F_{d,m}$ table and Claim 87
if $Y^* = \emptyset$ **then**
  Sample $p \in \{-nv, nv\}$ uniformly at random
if $Y^* \neq \emptyset$ **then**
  Sample $i^* \in \{0, ..., n\}$ according to Claim 88
  Sample $S^* \subset Y^*$ using Claim 89
  Sample a lattice point $p \in P_n \cap \square_{Y^*_{i^*}, S^*, n}$ using Claim 90
Return p

where the $B_{s,w}$ coefficients group terms with same power of z and $(1-z)$. Moreover, the $B_{s,w}$ terms are non-negative since each of $|Q_k|$ and $\mathcal{A}(k,j)$ terms are non-negative. Then $\mathcal{Z}(|P_n|)[z] = \sum_{s=0}^{d} \sum_{w=0}^{d+1} B_{s,w} z^{-w} (1-z^{-1})^{-s}$ satisfies the hypothesis of Lemma 27. ■

We include the simple proof of Corollary 51 for completeness.

Corollary 51 *When $\varepsilon = O(1)$, the expected running time of $\mathcal{R}_{\text{VOTE}}$ is $O(d^2 \log(d))$.*

Proof Let R denote a random variable for the runtime to sample a point from the Vote ripple mechanism, and let T denote the random variable for the hitting time of reservoir sampling. Each tails coin flip in reservoir sampling requires computing a size $|P_n|$, which takes time $O(d)$ by Lemma 49. The (single) heads flip requires $|P_n|$ and then sampling from P_n, in total time $O(d^2 \log(d))$. Thus $R \sim dT + d^2 \log(d)$, so by Lemma 50 we have $\mathbb{E}[R] = O(d^2 \log(d))$. ■

13. Computing Sum, Count and Vote Norms

This section contains a lot of known conclusions in fields like Integer Programming so no proof is provided for such folklore conclusions.

Theorem 91 *(folklore) For a polytope $P = \{x \in \mathbb{R}^d : Ax \leq b\}$, for a point x, $\|x\|_P = \max_{i:a_i^T \in row(A)} \frac{a_i^T x}{b_i}$.*

13.1. Sum

Lemma 92 *For the Sum problem in d-dimensions with contribution-bounding limit k, given a point x in space $\mathbb{R}^d$,*

$$\|x\|_P = \max\left\{\frac{\|x\|_1}{k}, \|x\|_\infty\right\}$$

The Sum unit ball is $S = \{x : \max\{\frac{\|x\|_1}{k}, \|x\|_\infty\} = 1\}$ so the unit $\|\cdot\|_P$ ball and S are equal, i.e. the norm $\|\cdot\|_P$ is equal to the norm induced by S.

13.2. Count

From the Count Section we know that the polytope $P = CH(K \cup -K)$, where

$$K = \{x \in \mathbb{R}^d : x \geq 0, \max\left(\frac{\|x\|_1}{k}, \|x\|_\infty\right) \leq 1\}$$

is in the first orthant.

Lemma 93 *For any vector $x \in \mathbb{R}^d$, let x_+ be the positive part of x (replace negative entries with 0) and x_- be the magnitude of the negative part of x (replace positive entries with 0, flip negatives to positive); $x = x_+ - x_-$. The norm $\|x\|_P$, where P is the count polytope for contribution bound k, is given by:*

$$\|x\|_P = \max\left(\|x_+\|_\infty, \frac{\|x_+\|_1}{k}\right) + \max\left(\|x_-\|_\infty, \frac{\|x_-\|_1}{k}\right)$$

Proof Let T be the Count unit ball. Let $S = \{x : \|x\|_1 \leq k, \|x\|_\infty \leq 1\}$ be the Sum unit ball. We only have to prove that the set $\{x : \|x\|_P \leq 1\}$ is equivalent to T. By definition $T = CH(S \cup -S)$. Fix some $x \in T$. Write $x = x_+ - x_-$. Then x can be written as a convex combination of vertices of T. Let B_k be the set of binary vectors with at most k ones. The vertices of T are $B_k \cup -B_k - \{0\}$. Write $x = \sum_{v \in B_k} c_v v + \sum_{w \in -B_k} c_w w$. If there is some v and w that both have support at coordinate i, then we can replace v by $v - e_i$ and replace w by $w + e_i$ without changing the convex combination. So we may assume that each $v \in B_k$ in the convex combination has support equal to the support of x_+ and each $w \in -B_k$ in the convex combination has support equal to the support of x_-.

Let $\|\cdot\|_S$ denote the Sum induced norm.

$$\begin{aligned}
\|x\|_P &= \left\|\sum_{v \in B_k} c_v v\right\|_S + \left\|\sum_{w \in -B_k} c_w w\right\|_S \\
&\leq \sum_{v \in B_k} \|c_v v\|_S + \sum_{w \in -B_k} \|c_w w\|_S \\
&= \sum_{v \in B_k} c_v + \sum_{w \in -B_k} c_w \\
&= 1
\end{aligned}$$

where we have used triangle inequality. This shows that T is a subset of the unit ball of $\|\cdot\|_P$.

Conversely, suppose $x \in \mathbb{R}^d$ has $\|x\|_P \leq 1$. Suppose $c_+ = \max\left(\|x_+\|_\infty, \frac{\|x_+\|_1}{k}\right)$ and $c_- = \max\left(\|x_-\|_\infty, \frac{\|x_-\|_1}{k}\right)$. Then $\|x\|_P = c_+ + c_- \leq 1$. Let S_+ be the unit Sum ball on the subspace of coordinates in the support of x_+, and Let S_- be the unit Sum ball on the subspace of coordinates in the support of x_-. By Lemma 92, we know $x_+ \in c_+ S_+$ and $x_- \in c_- S_-$. Moreover, $S_+ \subset S$ and $S_- \subset -S$, so we can write $x = c_+ v_+ + c_- v_-$ for some $v_+ \in S$ and $v_- \in -S$. Let $c_0 = 1 - c_+ - c_-$. Then $x = c_+ v_+ + c_- v_- + c_0(0)$ expresses x as a convex combination of points in T, which shows that the unit ball of $\|\cdot\|_P$ is a subset of T.

Since the unit $\|\cdot\|_P$ ball and T coincide, then the norm $\|\cdot\|_P$ is equal to the norm induced by T. ■

13.3. Vote

Theorem 94 *(folklore) The d-dimensional* permutahedron Π_d *is known to have rank $d-1$ and can be described as:*

$$\sum_{i=1}^{d} x_i = \frac{d(d-1)}{2}$$

$$\forall I \subset [d], I \neq \emptyset, \sum_{i \in I} x_i \leq \sum_{k=d-|I|}^{d-1} k$$

In other words the numbers sum up to exactly $0+1+2+\ldots+(d-1)$, and for any index set I is at most $(d-1)+\ldots+(d-|I|)$

Definition 95 *The mathematical definition of a prism using the Minkowski sum is as follows: Let $K \subset \mathbb{R}^n$ be a convex set (the "base" of the prism). Let $L \subset \mathbb{R}^n$ be a line segment (the "extrusion" or "interval"). The prism P is defined as the Minkowski sum of the base and the segment:*

$$P = K \oplus L = \{k + l \mid k \in K, l \in L\}$$

Lemma 96 *The d-dimensional permutahedron Π_d is symmetric around the point $(\frac{n-1}{2}, \ldots, \frac{n-1}{2}) = \frac{n-1}{2}\mathbf{1}$; in other words $\Pi_d = (n-1)\mathbf{1} - \Pi_d$.*

Here $\mathbf{1}$ denotes a d-dimensional vector of all 1's. Hence we have:

Lemma 97 *The polytope $CH(\Pi_d \cup -\Pi_d)$ is a prism and can be written as $\Pi_d - [0,1] \cdot (n-1)\mathbf{1}$.*

Deriving the linear program that characterizes the prism is a standard technique in polyhedral geometry. The method typically projects any point in the prism along the direction of the *translation vector* (the vector along which the cap is extended into the prism, for us this is $\mathbf{1}$) to the cap, and plug the projection into the linear constraints characterizing the cap. The detailed explanation can be found in, e.g., Ziegler (1995). Following this technique we have:

Theorem 98 *The polytope $CH(\Pi_d \cup -\Pi_d)$ can be described as:*

$$\textit{Cap constraint: } -\frac{d(d-1)}{2} \leq \sum_{i=1}^{d} x_i \leq \frac{d(d-1)}{2}$$

$$\textit{Wall constraint: } \forall I \subset [d], I \neq \emptyset, \sum_{i \in I} x_i - \frac{|I|}{d}\left(\sum_{j=1}^{d} x_j\right) \leq \frac{|I|(d-|I|)}{2}$$

Theorem 99 *Let $P = CH(\Pi_d \cup -\Pi_d)$. For any point x, the norm $\|x\|_P$ can be computed within time $O(d \log d)$.*

Proof There are 2^d constraints, but we don't have to compute $\frac{a_i^T x}{b_i}$ for every one of them. The cap constraint can be evaluated rather easily. For the wall constraints, without loss of generality, assume the coordinates are sorted in descending order. For any size $s \in 1, 2, \ldots, d-1$, the term $\frac{|I|}{d}\left(\sum_{j=1}^{d} x_j\right)$ is fixed; to maximize the LHS, we only have to pick $I = [s]$, the index set corresponding to the maximum s coordinates. Therefore for each $s \in [d]$ we only have to compute in $O(1)$ time. Hence the run-time is dominated by the sorting of d coordinates that takes $O(d \log d)$ time. ■

14. Additional Plots From Section 7

15. Proofs From Section 8

Lemma 55 *If P has NMP, take any probability mass function $f(\cdot)$ that is ε-DP for P, and average the mass over every level set L_k: $\forall v \in L_k, f_{avg}(v) = \frac{\sum_{u \in L_k} f(u)}{|L_k|}$, the resulting $f_{avg}(\cdot)$ is still ε-DP.*

Proof The constraint we want is equivalent to

$$e^{-\varepsilon} \sum_{v \in L_k} f(v) \cdot \frac{|L_{k-1}|}{|L_k|} \leq \sum_{u \in L_{k-1}} f(u) \leq e^{\varepsilon} \sum_{v \in L_k} f(v) \cdot \frac{|L_{k-1}|}{|L_k|}. \tag{14}$$

If this holds, the theorem also holds. But what we have is

$$e^{-\varepsilon} f(v) \leq f(u) \leq e^{\varepsilon} f(v) \tag{15}$$

for every *neighboring* $u \in L_{k-1}$) and $v \in L_k$, and u, v are neighbors, i.e., $v - u \in P$. If such a NMP flow exists between $L(k)$ and $L(k+1)$, we show how to use the flows on the edges to construct a linear combination of such inequalities that gives us (14).

Let $w_{u,v}$ be the amount of flow from $u \in L_{k-1}$ to $v \in L_k$ in the graph G_k. Multiply all sides of (15) with $w_{u,v}$, and take the summation over all such inequalities for any neighboring (u, v). The equation we get is:

$$e^{-\varepsilon} \sum_{(u,v) \in (L_{k-1}, L_k)} w_{u,v} f(v) \leq \sum_{(u,v) \in (L_{k-1}, L_k)} w_{u,v} f(u) \leq e^{\varepsilon} \sum_{(u,v) \in (L_{k-1}, L_k)} w_{u,v} f(v).$$

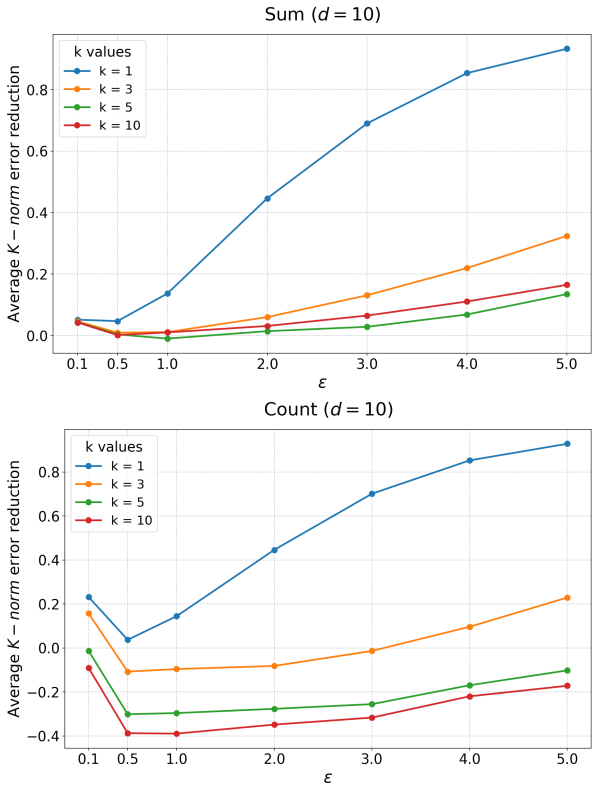

Figure 3: Relative norm error for Sum and Count at $d = 10$.

Slightly rearranging the summation signs yields

$$e^{-\varepsilon} \sum_{v \in L_k} \sum_{u \in L_{k-1}:(u,v) \in G_k} w_{u,v} f(v) \leq \sum_{u \in L_{k-1}} \sum_{v \in L_k:(u,v) \in G_k} w_{u,v} f(u) \leq e^{-\varepsilon} \sum_{v \in L_k} \sum_{u \in L_{k-1}:(u,v) \in G_k} w_{u,v} f(v)$$

In this inequality, in the middle, for every $u \in L_{k-1}$, by definition of the flow $w(\cdot, \cdot)$, the coefficient $\sum_{v \in L_k:(u,v) \in G_k} w_{u,v}$ before $f(u)$ is its total out-going flow, hence this term is equal to 1. Similarly, on the left/right-hand side, for each $v \in L_k$, the term that follows is the total in-coming flow, and is equivalent to $\frac{|L_{k-1}|}{|L_k|}$, multiplied by an $e^{-\varepsilon}$ and e^{ε}, respectively. Hence the resulting inequality is (14). ■

Theorem 56 *For any problem whose convex hull of sensitivity space is a well-behaved polytope P with NMP, the ripple mechanism has minimum expected $\| \cdot \|_P$ error within class $\mathcal{M}_{S(T,\mathcal{X}),\varepsilon}$.*

Proof If P has NMP, take any ε-DP PMF $f(\cdot)$ on support $\mathbb{Z}^d$. By definition, each point in $\mathbb{Z}^d$ has some integer norm value k, and lies in $\partial kP = L_k$. We get the averaged pmf $f_{avg}(\cdot)$ by averaging out $f(\cdot)$ on each L_k. By Lemma 55, this operation maintains ε-DP; the expected error is also the same, since all points on L_k have norm value k. We compare $f_{avg}(\cdot)$ with the ripple mechanism's pmf $f_P(\cdot)$. Both have the same probabilities over every level set L_k; the difference is that for f_P the rate of probability decline from L_{k-1} to L_k is exactly e^{ε}, but for f_{avg} this is not necessarily true.

In f_{avg}, at any integer k, for $u \in L_{k-1}$ and $v \in L_k$, if $\frac{f_{avg}(v)}{f_{avg}(u)} > e^{-\varepsilon}$, we can construct a new pmf f'_{avg} by letting

$$f'_{avg}(x) = \begin{cases} (1+\delta) f_{avg}(x), & x \in L(i), i \leq k \\ (1-\delta') f_{avg}(x), & x \in L(i), i > k \end{cases} \tag{16}$$

where the constants δ, δ' are normalizing constants to make the probability sum up to 1. Obviously this will have smaller expectation than $f_{avg}(\cdot)$, since we have moved mass from outside of L_k to inside of L_k. We can do this for any such k where the decline ratio is less than e^{ε}. ■

Lemma 57 *When the sensitivity polytope P is the ℓ_1 unit ball: $P = \{x \in \mathbb{Z}^d : \|x\|_1 \leq 1\}$, it has NMP.*

Before proving Lemma 57, we need to introduce the definition of *posets* and their properties.

Definition 100 *A* partial order *is a homogeneous binary relation that is reflexive, antisymmetric, and transitive. A partially ordered set (*poset *for short) is an ordered pair $P = (X, \leq)$ consisting of a set X (called the ground set of P) and a partial order $\leq$ on X. When $a < b$ and there exists no c such that $a < c < b$, a is said to be* covered *by b, and b covers a. A* graded poset *is one where there exists a* rank *ρ function, such that:*

- *the rank function is compatible with the ordering, meaning that for all x and y in the order, if $x < y$ then $\rho(x) < \rho(y)$, and*
- *the rank is consistent with the covering relation of the ordering, meaning that for all x and y, if y covers x then $\rho(y) = \rho(x) + 1$.*

Below is the definition of the **Cartesian product** of posets (there are other possible definitions of product between posets, but this one fits our setting).

Definition 101 *For any two existing posets $(P, \leq_P)$ and $(Q, \leq_Q)$, their* Cartesian product poset*, written as $P \times Q$, is defined on the set of all ordered pairs (p, q) where $p \in P$ and $q \in Q$. The ordering is $(p, q) \leq (p', q')$ if and only if $p \leq p'$ and $q \leq q'$. If $p =_P p'$ and $q =_Q q'$, $(p, q) =_{P \times Q} (p', q')$; otherwise if either $p <_P p'$ or $q <_Q q'$ holds, $(p, q) <_{P \times Q} (p', q')$. The Cartesian product of two graded posets is also a graded poset. Its rank is the sum of the ranks of the individual posets: $\rho(p, q) = \rho(p) + \rho(q)$.*

This definition can be found on page 3 of Ehrenborg and Readdy (2015). Note that the equivalence relationship here does not mean that they have the same rank . It means they are the same element.

The following result can be found in several papers, such as Hsieh and Kleitman.

Lemma 102 *The Cartesian product of two NMP posets is also NMP.*

This finally brings us to the proof of Lemma 57.

Proof [Lemma 57] Define a graded poset P on the 1-D integer space $\mathbb{Z}$, where $\rho(x) = |x|$, and $x < y$ if and only if $|y| = |x| + 1$ and $x - y \in \{\pm 1\}$ both hold. Then, in $\mathbb{Z}^d$, construct the d-th product graded poset $P^d = P \times P \times ... \times P$ (d times). It is easy to verify that in P^d, $\rho(x) = \|x\|_1$. Then $L(k)$ is equivalent to all points with rank k in P^d; and for any $x \in L(k)$ and $y \in L(k+1)$, x and y are neighbors if and only if $x < y$ in P^d. Hence by Lemma 102, ℓ_1 shells have NMP, which gives us optimality. ■