Appendix: High Dimensional Discrete Integration over the Hypergrid

Raj Kumar Maity, Arya Mazumdar, Soumyabrata Pal

A Omitted proofs

Proof of Theorem 1. From Lemma 2, we have,

$$\Pr\left[\bigcap_{i=1}^{n} M_i \in [\beta_{\min(i+1,n')}, \beta_{\max(i-1,0)}]\right] \geq 1 - 2n\exp(-\gamma\ell) = 1 - \delta.$$

for $\ell = \frac{1}{\gamma}\ln\frac{2n}{\delta}$ and by definition $M_0 = \beta_0$. The algorithm outputs $M_0 + (t-1)\sum_{i=0}^{n'-1} M_{i+1}t^i$ which lies in the range $[L', U']$ with probability at least $1-\delta$ where $L' = \beta_0 + (t-1)\sum_{i=0}^{n'-1}\beta_{\min\{i+2,n'\}}t^i$ and $U' = \beta_0 + (t-1)\sum_{i=0}^{n'-1}\beta_i t^i$.

Now notice that, as $\beta_0 \geq \beta_1$, we have

$$\begin{aligned} U' &= \beta_0 + (t-1)\sum_{i=0}^{n'-1}\beta_i t^i \\ &= \beta_0 + (t-1)(\beta_0 + \beta_1 t) + (t-1)\sum_{i=2}^{n'-1}\beta_i t^i \\ &\leq t^2\beta_0 + t^2.(t-1).\sum_{i=2}^{n'-1}\beta_i t^{i-2} \\ &\leq t^2(\beta_0 + (t-1)\sum_{i=0}^{n'-1}\beta_{\min\{i+2,n'\}}t^i) = t^2 L'. \end{aligned}$$

The only thing that remains to be proved is that $L' \leq S_w(\Omega) \leq U'$. However that is true, by just following an argument similar to (5). Indeed,

$$\sum_{i=0}^{n'-1}\beta_{i+1}(t^{i+1} - t^i) \leq S_\Omega(w) \leq \sum_{i=0}^{n'-1}\beta_i(t^{i+1} - t^i)$$

which implies $L' \leq S_\Omega(w) \leq U'$.

Therefore Algorithm 1 provides a t^2-approximation to $S_\Omega(w)$. The total number of calls to the MAX-oracle is $n'\ell + 1 = O(n\log(n/\delta))$. □

Proof of Lemma 1. Let A_i denote the ith row of i and b_i denote the ith entry of b. Then $\mathbb{1}[A\sigma + b \prec \alpha_r \cdot \mathbf{1}] = \bigwedge_{i=1}^{m}\mathbb{1}[A_i\sigma + b_i \prec \alpha_r]$.

For all configurations $\sigma \in \Omega, \forall i$, we must have

$$\Pr(A_i\sigma + b_i \prec \alpha_r) = \sum_{j=0}^{r-1}\Pr(A_i\sigma + b_i = \alpha_j) = \frac{r}{q}.$$

As A_i, b_i are independent $1 \leq i \leq m$, we must have that $\Pr(A\sigma + b \prec \alpha_r \cdot \mathbf{1}) = (\frac{r}{q})^m$. Now for any two distinct

configurations $\sigma_1, \sigma_2 \in \mathbb{F}_q^n$,

$$\begin{aligned}
&\Pr(A_i\sigma_1 + b_i \prec \alpha_r \wedge A_i\sigma_2 + b_i \prec \alpha_r)\\
&= \sum_{k=0}^{r-1}\sum_{j=0}^{r-1} \Pr(A_i\sigma_1 + b_i = \alpha_k \wedge A_i\sigma_2 + b_i = \alpha_j)\\
&= \sum_{k=0}^{r-1}\sum_{j=0}^{r-1} \Pr(A_i\sigma_1 + b_i = \alpha_k)\\
&\qquad \cdot \Pr(A_i\sigma_2 + b_i = \alpha_j | A_i\sigma_1 + b_i = \alpha_k)\\
&= \sum_{k=0}^{r-1}\sum_{j=0}^{r-1} \Pr(A_i\sigma_1 + b_i = \alpha_k) \Pr(A_i(\sigma_2 - \sigma_1) = \alpha_j - \alpha_k)\\
&= r^2(1/q)(1/q) = (r/q)^2.
\end{aligned}$$

As all the rows are independent, $\Pr(A\sigma_1 + b \prec \alpha_r \cdot \mathbb{1} \wedge A\sigma_2 + b \prec \alpha_r \cdot \mathbb{1}) = (\frac{r}{q})^{2m}$. □

Proof of Lemma 2. Let $t = q/r$. Consider the set of $\lfloor t^j \rfloor$ heaviest configurations

$$\Omega_j = \{\sigma_1, \ldots \sigma_{\lfloor t^j \rfloor}\}.$$

Let $S_j(h_i) = |\{\sigma \in \Omega_j : A^i\sigma + b^i \prec \alpha_r \cdot \mathbf{1}\}|$. Recall h_i is sampled uniformly at random from $\mathcal{H}_{i,n}$. Using Lemma 1,

$$\mathbb{E}S_j(h_i) = \mathbb{E}\sum_{\sigma\in\Omega_j} \mathbb{1}[h_i(\sigma) \prec \alpha_r \cdot \mathbf{1}] = \frac{\lfloor t^j \rfloor}{t^i}.$$

For each configuration σ let us denote the random variable $\bar{Z_\sigma^i} = \mathbb{1}[\{h_i(\sigma) \prec \alpha_r \cdot \mathbf{1}\}] - \frac{1}{t^i}$. By our design $\mathbb{E}\bar{Z_\sigma^i} = 0$. Note that, $S_j(h_i) - \mathbb{E}S_j(h_i) = \sum_{\sigma\in\Omega_j} \bar{Z_\sigma^i}$. Also, from Lemma 1, the random variables $\bar{Z_\sigma^i}$s are pairwise independent. Therefore,

$$\operatorname{var} S_j(h_i) = \operatorname{var}(\sum_{\sigma\in\Omega_j} \bar{Z_\sigma^i}) = \sum_{\sigma\in\Omega_j} \mathbb{E}\bar{Z_\sigma^i}^2 = \frac{\lfloor t^j \rfloor}{t^i}(1 - \frac{1}{t^i}).$$

Now, for any $1 \le k \le \ell$,

$$\begin{aligned}
&\Pr(w_i^{(k)} \ge \beta_j) = \Pr(w_i^{(k)} \ge w(\sigma_{\lfloor t^j \rfloor}))\\
&= \Pr(S_j(h_i) \ge 1) = 1 - \Pr(S_j(h_i) \le 0).
\end{aligned}$$

Let $j = i + 1$. Then, using Chebyshev inequality,

$$\begin{aligned}
&\Pr(S_j(h_i) \le 0) = \Pr\left(S_j(h_i) - \mathbb{E}S_j(h_i) \le -\frac{\lfloor t^j \rfloor}{t^i}\right)\\
&\le \frac{\operatorname{var} S_j(h_i)}{(\frac{\lfloor t^j \rfloor}{t^i})^2} \le \frac{t^i(1 - 1/t^i)}{\lfloor t^j \rfloor} < \frac{t^i - 1}{t^{i+1} - 1} \le \frac{1}{t} = \frac{r}{q}.
\end{aligned}$$

Therefore,

$$\Pr(w_i^{(k)} \ge \beta_{i+1}) \ge 1 - \frac{r}{q}.$$

Also, $\Pr(w_i^{(k)} \le \beta_{i-1}) = \Pr(w_i^{(k)} \le w(\sigma_{\lfloor t^{i-1} \rfloor})) \ge \Pr(S_{i-1}(h_i) = 0)$. Notice that the last inequality is satisfied because $S_{i-1}(h_i) = 0$ implies $w_i^{(k)} \le w(\sigma_{\lfloor t^{i-1} \rfloor})$. Now, continuing the chain of inequalities, using Markov inequality,

$$\begin{aligned}
&\Pr(w_i^{(k)} \le \beta_{i-1}) \ge 1 - \Pr(S_{i-1}(h_i) \ge 1) \ge 1 - \mathbb{E}S_{i-1}(h_i)\\
&= 1 - \frac{\lfloor t^{i-1} \rfloor}{t^i} \ge 1 - \frac{1}{t} = 1 - \frac{r}{q}.
\end{aligned}$$

Recall that, $M_i = \text{Median}(w_i^{(1)}, w_i^{(2)}, \ldots, w_i^{(\ell)})$. Define, $X_i^{(k)}$ to be the indicator random variable of the event $\{w_i^{(k)} \leq \beta_{i+1}\}$. Therefore, $\Pr(M_i \leq \beta_{i+1}) = \Pr(\sum_{k=1}^{\ell} X_i^{(k)} \geq \ell/2)$. On the other hand, note that, $\Pr(X_i^{(k)} = 1) \leq r/q$. We know from Chernoff bound that, if X is a sum of iid $\{0,1\}$ random variables then $\Pr(X \geq \mathbb{E}X(1+\delta)) \leq \exp(-\mathbb{E}X\delta^2/3)$. Therefore,

$$\Pr(M_i \leq \beta_{i+1}) \leq \exp\left(-\frac{\ell q}{3r}\left(\frac{1}{2} - \frac{r}{q}\right)^2\right).$$

Similarly,

$$\Pr(M_i \geq \beta_{i-1}) \leq \exp\left(-\frac{\ell q}{3r}\left(\frac{1}{2} - \frac{r}{q}\right)^2\right).$$

This proves the lemma. □

Proof of Lemma 3. Notice that $S_{A,R,b}$ is a union of distinct cosets and therefore,

$$S_{A,R,\mathbf{0}} \equiv \bigcup_{y \in \{\alpha_0, \alpha_1, \ldots, \alpha_{r-1}\}^m} S_{A,\mathbf{0}}(y),$$

where $S_{A,\mathbf{0}}(y) \equiv \{Ax + Ry \mid x \in \mathbb{F}_q^{n-m}\}$ is defined as a particular coset with a fixed $y \in \{\alpha_0, \alpha_1, \ldots, \alpha_{r-1}\}^m$. Hence $|S_{A,R,\mathbf{0}}| = q^{n-m}r^m$ and since $S_{A,R,b}$ is simply a random affine shift of $S_{A,R,\mathbf{0}}$, $|S_{A,R,b}| = q^{n-m}r^m$ as well. Now for a vector $\sigma \in \mathbb{F}_q^n$, we must have

$$\Pr(Z_\sigma = 1) = \sum_{y \in S_{A,R,b}} \Pr(\sigma = y) = \frac{|S_{A,R,b}|}{q^n} = \left(\frac{r}{q}\right)^m.$$

Next, for two configurations $\sigma_1, \sigma_2 \in \mathbb{F}_q^n$, we have that

$$\begin{aligned}
&\Pr(Z_{\sigma_1} = 1 \wedge Z_{\sigma_2} = 1) \\
&= \sum_{y_1, y_2} \Pr(\sigma_1 \in S_{A,b}(y_1) \wedge \sigma_2 \in S_{A,b}(y_2)) \\
&= \sum_{y_1, y_2} \Pr(\sigma_1 \in S_{A,b}(y_1) \mid \sigma_2 \in S_{A,b}(y_2)) \Pr(\sigma_2 \in S_{A,b}(y_2)) \\
&= \sum_{y_1, y_2} \Pr(\sigma_1 - \sigma_2 \in S_{A,\mathbf{0}}(y_1 - y_2)) \Pr(\sigma_2 \in S_{A,b}(y_2)).
\end{aligned}$$

Therefore we just need to evaluate the probability of the event $\Pr(\tau \in S_{A,\mathbf{0}}(z))$ for $\tau = \sigma_1 - \sigma_2 \neq 0$ and $z = y_1 - y_2$. Now, if $z = \mathbf{0}$, $\Pr(\tau \in S_{A,\mathbf{0}}(\mathbf{0}))$ is equal to $\frac{q^{n-m}-1}{q^n-1}$ since A is a randomly chosen full rank matrix, i.e., $|\{Ax : x \in \mathbb{F}_q^{n-m}\}| \setminus \{0\} = q^{n-m} - 1$. Now, since $\{Ax + Rz : x \in \mathbb{F}_q^{n-m}\}, z \neq 0$, is a uniformly random coset of $\{Ax : x \in \mathbb{F}_q^{n-m}\}$, we have,

$$\begin{aligned}
\Pr(\tau \in S_{A,\mathbf{0}}(z) \mid z \neq 0, \tau \in \{Ax : x \in \mathbb{F}_q^{n-m}\}) &= 0 \\
\text{and } \Pr(\tau \in S_{A,\mathbf{0}}(z) \mid z \neq 0, \tau \notin \{Ax : x \in \mathbb{F}_q^{n-m}\}) &= \frac{1}{q^m - 1}.
\end{aligned}$$

Hence,

$$\Pr(\tau \in S_{A,\mathbf{0}}(z) \mid z \neq 0) = \left(1 - \frac{q^{n-m}-1}{q^n - 1}\right)\frac{1}{q^m - 1} = \frac{q^{n-m}}{q^n - 1}.$$

Therefore, we have that

$$\begin{aligned}
&\Pr(Z_{\sigma_1} = 1 \wedge Z_{\sigma_2} = 1) \\
&= \sum_{y_1, y_2 : y_1 = y_2} \frac{q^{n-m} - 1}{q^n - 1} + \sum_{y_1, y_2 : y_1 \neq y_2} \frac{q^{n-m}}{q^n - 1} \\
&= r^m \left(\frac{q^{n-m} - 1}{q^n - 1} \right) + \left(r^{2m} - r^m \right) \left(\frac{q^{n-m}}{q^n - 1} \right) \\
&= \frac{r^m (r^m q^{n-m} - 1)}{q^n - 1} \leq \left(\frac{r}{q} \right)^{2m} = \Pr(Z_{\sigma_1} = 1)^2
\end{aligned}$$

and hence we have the statement of the lemma. □

Proof of Lemma 4. We have,

$$\Pr(Z_\sigma = 1) = \sum_{y \in T_{A,b,m}} \Pr(\sigma = y) = \frac{|T_{A,b,m}|}{q^n} = \frac{1}{q^n} \left\lfloor \frac{r^m}{q^{m-n}} \right\rfloor,$$

which proves the first claim. Next, for two distinct configurations $\sigma_1, \sigma_2 \in \mathbb{F}_q^n$, we have that

$$\begin{aligned}
&\Pr(Z_{\sigma_1} = 1 \wedge Z_{\sigma_2} = 1) \\
&= \sum_{y_1, y_2 \in T_{A,b,m}} \Pr(\sigma_1 = y_1 \wedge \sigma_2 = y_2) \\
&= \sum_{y_1, y_2 \in \mathcal{S}_m} \Pr(\sigma_1 = A y_1 + b \wedge \sigma_2 = A y_2 + b) \\
&= \sum_{y_2 \in \mathcal{S}_m} \Pr(\sigma_2 = A y_2 + b) \\
&\quad \times \sum_{y_1 \in \mathcal{S}_m} \Pr(\sigma_1 = A y_1 + b \mid \sigma_2 = A y_2 + b) \\
&= \sum_{y_2 \in \mathcal{S}_m} \Pr(\sigma_2 = A y_2 + b) \sum_{y_1 \in \mathcal{S}_m} \Pr(\sigma_1 - \sigma_2 = A(y_1 - y_2))
\end{aligned}$$

Since $\sigma_1 \neq \sigma_2$, we must have that $\Pr(\sigma_1 - \sigma_2 = A(y_1 - y_2) \mid y_1 = y_2) = 0$. For $y_1 \neq y_2$, every configuration $\sigma \in \mathbb{F}_q^n, \sigma \neq 0$ is equally probable to be $A(y_1 - y_2)$ since A is uniformly and randomly sampled full rank matrix. Hence,

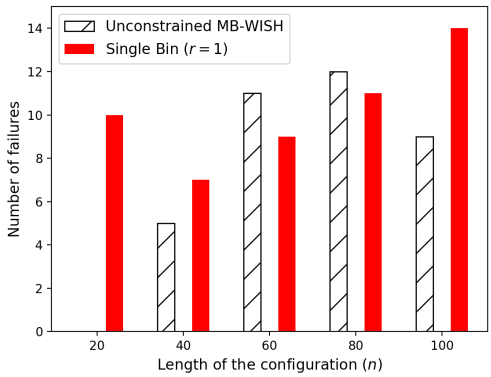
$$\begin{aligned}
&\Pr(Z_{\sigma_1} = 1 \wedge Z_{\sigma_2} = 1) \\
&= \frac{1}{q^n (q^n - 1)} \left\lfloor \frac{r^m}{q^{m-n}} \right\rfloor \left(\left\lfloor \frac{r^m}{q^{m-n}} \right\rfloor - 1 \right) \leq (\Pr(Z_\sigma = 1))^2.
\end{aligned}$$

□

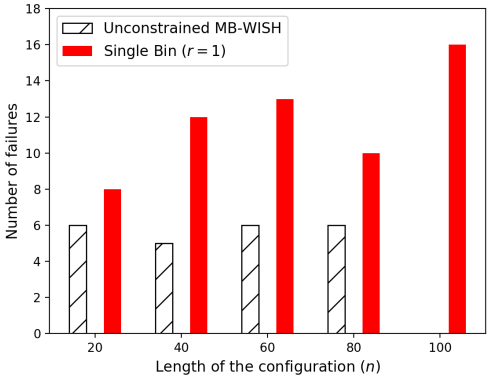
B Experimental results on computing Total Variation distance

We show one more instance of discrete integration where `MB-WISH` is useful. The purpose of this experiment is to show the effectiveness of `MB-WISH` by choosing counting problems where good theoretical results are available.

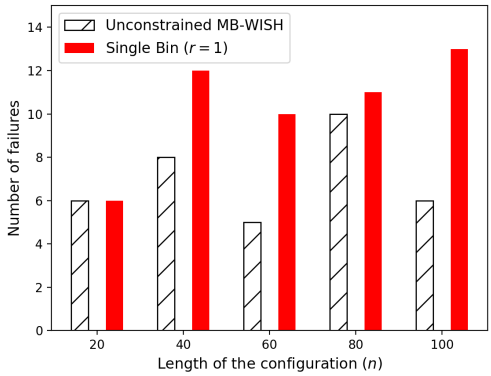
For this, we use `Unconstrained MB-WISH` to compute *total variation distances* between two high dimensional (up to dimension 100) probability distributions, generated via an iid model. Although, it is computationally hard to compute the total variation distance between two distributions, for the special case of product distributions, we can derive theoretical expressions that are known to bound the total variation distance from above and below.



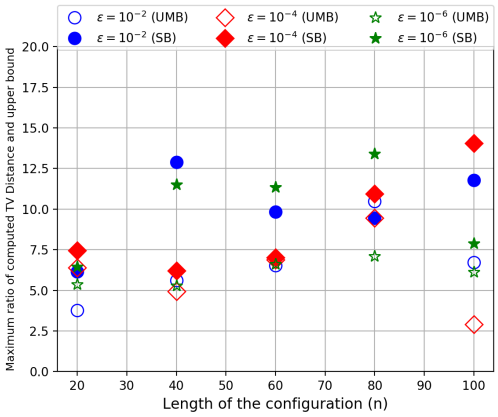
(a) Number of times, among 100 trials, the computed total variation distance is above 4× the theoretical upper bound for $\epsilon = 10^{-2}$



(b) Number of times, among 100 trials, the computed total variation distance is above 4× the theoretical upper bound for $\epsilon = 10^{-4}$



(c) Number of times, among 100 trials, the computed total variation distance is above 4× the theoretical upper bound for $\epsilon = 10^{-6}$



(d) The maximum ratio of the computed total variation distance and the upper bound

Figure 3: Behavior of computed total variation distance by Algorithm 2 using $r = 2$ (Unconstrained MB-WISH or UMB) and $r = 1$ (Single Bin or SB) with respect to the theoretical upper bound.

The total variation (TV) distance between any two discrete distributions P and Q with common sample space $\mathcal{P}$ is defined to be

$$\|P - Q\|_{TV} = \sup_{A \subseteq \mathcal{P}} |P(A) - Q(A)| = \frac{1}{2} \sum_{\sigma \in \mathcal{P}} |P(\sigma) - Q(\sigma)|.$$

Simply consider finding TV distance between joint distributions of n random variables that can take value in $\{0, 1, \ldots, q-1\}$. In that case, we seek to find,

$$\frac{1}{2} \sum_{\sigma \in \{0,1,\ldots,q-1\}^n} |P^n(\sigma) - Q^n(\sigma)|,$$

which is in the exact form of Eq. (1). Therefore we can use `MB-WISH` algorithm to estimate the total variation distance. The following are well-known upper and lower bounds on TV distance based on Hellinger distance, $h(P,Q)^2 \equiv \sum_{\sigma \in \mathcal{P}} (\sqrt{P(\sigma)} - \sqrt{Q(\sigma)})^2$ [6]

$$\frac{1}{2} h(P,Q)^2 \leq \|P - Q\|_{TV} \leq h(P,Q) \sqrt{1 - h(P,Q)^2/4}.$$

Furthermore,

$$h(P^n, Q^n)^2 = 2 - 2\prod_{i=1}^{n}(1 - \frac{1}{2}h(P_i, Q_i)^2).$$

For 'near-uniform' distributions, it is known that the upper bound is a good approximation [20].

For the experiments, we choose two distributions defined over q points in the following manner: We choose a vector $\mathbf{v} \in [0,1]^q$ randomly and normalize the vector (so that the sum of the elements is 1) in order to have the first distribution $P \equiv [p_1, p_2, \ldots, p_q]$. The second distribution Q is then chosen to be

$$Q \equiv [p_1, p_2 + \epsilon, p_3 - \epsilon, \ldots, p_q - \epsilon].$$

where ϵ is a small number chosen in order to make the two distributions very close to each other. Here the distribution P^n and Q^n are supported on $\{0, 1, 2, \ldots, q-1\}^n$ where n can be any natural number. Now, we choose $q = 5$ and for three different values of $\epsilon = 10^{-2}, 10^{-4}$ and 10^{-6} and five different values of $n = 20, 40, 60, 80$ and 100, we repeat the experiment described above 100 times for each setting and use `Unconstrained MB-WISH` to compute the total variation distance.

We perform our experiments in a time constrained manner (10 minute for each calls to MAX-oracle). We have shown in Figure 3 histograms of the number of times the computed total variation distance is above four times the upper bound for $\epsilon = 10^{-2}, 10^{-4}$ and 10^{-6} respectively in Figures 3a, 3b and 3c respectively. We chose a factor of four because the theoretical approximation factor guaranteed by `Unconstrained MB-WISH` is ~ 4. We observed that the total variation distance is always above the upper bound with Hellinger distance but on the other hand, in very few trials the computed value is above four times the upper bound. Finally, we have also shown the maximum ratio of the computed total variation distance and the upper bound for each value of ϵ and n in Figure 3d.

We have compared the results obtained by `Unconstrained MB-WISH` with the corresponding results obtained by its single bin counterpart (Ermon et al.'s method, choosing $r = 1$) in Figure 3. Even though $q = 5$ is not large, the improvement in performance by using `Unconstrained MB-WISH` is clear. In Figures 3a, 3b, 3c, it can be observed that in the case of single bin, the number of failures (solid red) is almost always larger than the corresponding setting with multiple bins. Moreover, in Figure 3d, the maximum ratio in the setting of single bin (solid) is always much higher than the the setting of multiple bins (hollow).

C Derandomization: structured hashes

For the analysis of [7, 8] to go through, we needed a family of hash functions that are pairwise independent[1]. A hash family $\mathcal{H} = \{h : \Omega \to \tilde{\Omega}\}$ is called uniform and pairwise independent if the following two criteria are met for a randomly and uniformly chosen h from $\mathcal{H}$: 1) for every $x \in \Omega$, $h(x)$ is uniformly distributed in $\tilde{\Omega}$ and 2) for any two distinct $x, y \in \Omega$ and $u, v \in \tilde{\Omega}$, $\Pr(h(x) = u, h(y) = v) = \Pr(h(x) = u)\Pr(h(y) = v)$. By identifying Ω with $\mathbb{F}_2^n$ (and $\tilde{\Omega}$ with $\mathbb{F}_2^m$) and by using a family of hashes $\{x \mapsto h_{A,b}(x) = Ax + b : A \in \mathbb{F}_2^{m \times n}, b \in \mathbb{F}_2^m\}$ defined in (4), [7] show the family to be pairwise independent and thereby achieve their objective.

The size of the hash family $\mathcal{H}$ determines how many random bits are required for the randomized algorithm to work. By defining the hash family by a random binary matrix, Ermon et al. reduce the number of random bits from potentially $m2^n$ to $mn + m = m(n+1)$ bits (see, p. 3 of [7]). Here, we show that it is possible to construct pairwise independent hash family $\{\mathbb{F}_2^n \to \mathbb{F}_2^m\}$ using only $O(n)$ random bits such that any hash function from the family still has the structure $h(x) = Ax + b$. While memory optimal pairwise independent hash functions are quite standard, we feel for completeness it would be good to show that they can be represented as the above matrix-vector product form. All of the statements of this section can be easily extended to q-ary alphabets.

Construction 1: Let $f(x) \in \mathbb{F}_2[x]$ be an irreducible polynomial of degree n. We construct the finite field $\mathbb{F}_{2^n}$ with the ζ, root of $f(x)$ as a generator of $\mathbb{F}_{2^n}^*$. Now, any $x \in \mathbb{F}_2^n$ can be written as a power of ζ via a natural map $\phi : \mathbb{F}_2^n \to \mathbb{F}_{2^n}$. Indeed, for any element $\zeta^k \in \mathbb{F}_{2^n}^*$ consider the polynomial $\zeta^k \mod f(\zeta)$ of degree $n-1$. The coefficients of this

[1] It is sufficient to have the hash family satisfy some weaker constraints, such as being pairwise negatively correlated.

polynomial from an element of $\mathbb{F}_2^n$. ϕ is just the inverse of this map. Also, assume that the all-zero vector is mapped to 0 under ϕ.

Let $x \in \mathbb{F}_2^n$ be the configuration to be hashed. Suppose the hash function is $h_{v,b}$, indexed by $v \in \mathbb{F}_2^n$ and $b \in \mathbb{F}_2^m$. The hash function is defined as follows: Let $v \in \mathbb{F}_2^n$. Compute $z = \phi^{-1}(\phi(x) \cdot \phi(v) \mod f(\zeta)) \in \mathbb{F}_2^n$. Let $y \in \mathbb{F}_2^m$ be the first m bits of z. Finally, output $y + b$, where $b \in \mathbb{F}_2^m$.

Proposition 1. *The hash function $h_{v,b}$ can be written as an affine transform ($x \mapsto Ax + b$) over $\mathbb{F}_2^n$.*

Proof. It is sufficient to show that z can be obtained as a linear transform of v. Note that the product of $\phi(x)$ and $\phi(v)$ can be written as a convolution between x and $v \equiv (v_1, v_2, \ldots, v_n)$ (as we can view this as product between two polynomials). Let Γ be the $(2n-1) \times n$ matrix,

$$\Gamma = \begin{bmatrix} v_1 & 0 & 0 & \ldots & 0 \\ v_2 & v_1 & 0 & \ldots & 0 \\ v_3 & v_2 & v_1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ v_n & v_{n-1} & v_{n-2} & \ldots & v_1 \\ 0 & v_n & v_{n-1} & \ldots & v_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & v_n \end{bmatrix}.$$

The reduction modulo $f(\zeta)$ can also be written as a linear operation. Just consider the $n \times (2n-1)$ matrix P whose ith column contains the coefficients of the polynomial $\zeta^{i-1} \mod f(\zeta), 1 \le i \le 2n-1$. Note that the first n columns of the matrix is simply the identity matrix. We can write, $z = P\Gamma x$. □

Note that, to chose a random and uniform hash function from $\{h_{v,b}, v \in \mathbb{F}_2^n, b \in \mathbb{F}_2^m\}$, one needs $m + n$ random bits. It follows that the hash family is pairwise independent.

Proposition 2. *The hash family $\{h_{v,b}, v \in \mathbb{F}_2^n, b \in \mathbb{F}_2^m\}$ is uniform and pairwise independent.*

Proof. Suppose v, b are randomly and uniformly chosen. For any $x_1, x_2 \in \mathbb{F}_2^n$ and $y_1, y_2 \in \mathbb{F}_2^m$, first of all

$$\Pr(h_{v,b}(x_1) = y_1) = \frac{1}{2^m},$$

since b is uniform. Now,

$$\begin{aligned} &\Pr(h_{v,b}(x_1) = y_1, h_{v,b}(x_2) = y_2) \\ &= \frac{1}{2^m} \Pr(h_{v,b}(x_2) = y_2 | h_{v,b}(x_1) = y_1) \\ &= \frac{1}{2^m} \Pr(h_{v,b}(x_2) - h_{v,b}(x_1) = y_2 - y_1 | h_{v,b}(x_1) = y_1) \\ &= \frac{1}{2^m} \Pr(h_{v,b}(x_2) - h_{v,b}(x_1) = y_2 - y_1). \end{aligned}$$

Now, since $\Pr((\phi(x_1) - \phi(x_2)) \cdot \phi(v) \mod f(\zeta) = u) = \frac{1}{2^n}$ for any u, we must have $\Pr(h_{v,b}(x_2) - h_{v,b}(x_1) = y_2 - y_1) = \frac{1}{2^m}$. Therefore the claim is proved. □

Moreover the randomness used to construct this hash function is also optimal. It can be shown that, the size of a pairwise independent hash family $\{h : \{0,1\}^n \to \{0,1\}^m\}$ is at least $2^{m+n} - 2^n + 1$ (see, [22]). This implies that $m + n$ random bits were essential for the construction.

Construction 2: Toeplitz matrix. In [9], a Toeplitz matrix was used as the hash function. In a Toeplitz matrix, each descending diagonal from left to right is fixed, i.e., if $A_{i,j}$ is the (i,j)th entry of a Toeplitz matrix, then $A_{i,j} = A_{i-1,j-1}$. So to specify an $m \times n$ Toeplitz matrix one needs to provide only $m + n - 1$ entries (entries of the first row and first

column). Consider the random $m \times n$ Toeplitz matrix A_T where each of the entries of the first row and first column are chosen with equal probability from $\{0, 1\}$, i.e., each entry in the first row and column is a Bernoulli(0.5) random variable. The hash function $h_{A_T,b} : x \mapsto A_T x + b$, is constructed by choosing a uniformly random $b \in \mathbb{F}_2^m$.

Proposition 3. *The hash family $\{h_{A_T,b}\}$ is uniform and pairwise independent [9].*

Proof. First of all, the uniformity of the family is immediate since b is uniformly chosen. For any $x_1, x_2 \in \mathbb{F}_2^n$ and $y_1, y_2 \in \mathbb{F}_2^m$, $\Pr(h_{A_T,b}(x_1) = y_1, h_{A_T,b}(x_2) = y_2) = \frac{1}{2^m} \Pr(h_{A_T,b}(x_2) = y_2 | h_{A_T,b}(x_1) = y_1) = \frac{1}{2^m} \Pr(A_T(x_1 - x_2) = y_1 - y_2)$. It remains to prove that $\Pr(A_T x = y) = \frac{1}{2^m}$ for any fixed x, y. Let the kth coordinate of x is the first to be in the support of x. Now consider the inner product of the jth row of A_T with x. This product will contain the entry $A_T(j, k)$, the (j, k)th entry of A_T. Note that, this entry would not have appeared in any of the inner products of ith row of A_T and x, for $i < j$. Therefore the probability that this inner product is any fixed value is exactly $\frac{1}{2}$ given inner product of all previous rows with x. Therefore, $\Pr(A_T x = y) = \frac{1}{2^m}$. □

Note that, the number of random bits required from this construction is $2m + n - 1$. Toeplitz matrix allow for much faster computation of the hash function (matrix-vector multiplication with Toeplitz matrix takes only $O(n \log n)$ time compared to $\Omega(mn)$ for unstructured matrices).

We remark that *sparse Toeplitz Matrices* also can be used as our hash family, further reducing the randomness. In particular, we could construct a Toeplitz matrix with Bernoulli(p) entries for $p < 0.5$. While the pairwise independence of the hash family is lost, it is still possible to analyze the MB-WISH algorithm with this family of hashes since they form a *strongly universal* family [22]. The number of random bits used in this hash family is $(m + n - 1)h(p) + m$. This construction allows us to have sparse rows in the matrix for small values of p, which can lead to further speed-up.

Both the constructions of this section extend to q-ary alphabet straightforwardly.

D MB-WISH for computing permanent

For computing the *permanent*, the domain of integration is the symmetric group S_n. However S_n can be embedded in $\mathbb{F}_q^n$ for a $q \geq n$. Therefore we can try to use MB-WISH algorithm and same set of hashes on elements of S_n treating them as q-ary vectors, $q \geq n$. We need to be careful though since it is essential that the MAX-oracle returns a permutation and not an arbitrary vector. The modified MAX-oracle for permanents therefore must have some additional constraints. However those being affine constraints, it turns out MAX-oracle is still implementable in optimization softwares.

Recall the permanent of a matrix as defined in Eq. (2): $\text{Perm}(D) \equiv \sum_{\sigma \in S_n} \prod_{i=1}^n D_{i,\sigma(i)}$. We will show that it is possible to approximate the permanent with a modification of the MB-WISH algorithm and our idea of using multiple bins for optimization in the calls to MAX-oracle. Also, recall from Section 3 that we set $\mathbb{F}_q \equiv \{\alpha_0, \alpha_1, \ldots, \alpha_{q-1}\}$ where there exists a fixed ordering among the elements. We set $q \geq n$ and consider any $\sigma \in S_n$ as an n-length vector over $\mathbb{F}_q$ (that is by identifying $1, 2, \ldots, n$ as $\alpha_0, \alpha_1, \ldots, \alpha_{n-1}$ respectively). Then we define a modified hash family $\mathcal{H}_{m,n} = \{h_{A,b} : A \in \mathbb{F}_q^{m \times n}, b \in \mathbb{F}_q^m\}$ with $h_{A,b} : S_n \to \mathbb{F}_q^m : \sigma \mapsto A\sigma + b$, the operations are over $\mathbb{F}_q$.

However, when calling the MAX-oracle, we need to make sure that we are getting a permutation as the output. Hence the modified MAX-oracle for computing permanent will be:

$$\begin{aligned} &\max_{\sigma \in \mathbb{F}_q^n} w(\sigma) \\ &\text{s.t., } A\sigma + b \prec \alpha_r \cdot \mathbf{1}; \sigma \prec \alpha_{n-1} \cdot \mathbf{1}; \sigma(i) \neq \sigma(j) \forall i \neq j, \end{aligned} \tag{11}$$

where, $w(\sigma) = \prod_{i=1}^n D_{i,\sigma(i)}$. These constraints ensures that the MAX-oracle returns a permutation over n elements. With this change we propose Algorithm 3 to compute permanent of a matrix and call it PERM-WISH. The full algorithm is provided as Algorithm 3.

Algorithm 3 PERM-WISH for and matrix D; $\Omega = S_n$; weight function $w(\sigma) = \prod_{i=1}^{n} D_{i,\sigma(i)}$

Initialize: $\ell \to \lceil \frac{1}{\gamma} \ln \frac{2n}{\delta} \rceil$, $q > n$, $r = \lfloor \frac{q-1}{2} \rfloor$, $n' = \lceil n \log_{q/r} q \rceil$
  $M_0 \equiv \max_{\sigma \in S_n} w(\sigma)$
  for $i \in \{1, 2, \ldots, n'\}$ **do**
    for $k \in \{1, \ldots, \ell\}$ **do**
      Sample hash functions $h_i \equiv h_{A^i, b^i}$ uniformly at random from $\mathcal{H}_{i,n}$ as defined in (7)
      $w_i^{(k)} = \max_{\sigma \in \mathbb{F}_q^n} w(\sigma)$ such that $A^i\sigma + b^i \prec \alpha_r \cdot \mathbf{1}; \sigma \prec \alpha_{n-1} \cdot \mathbf{1}; \sigma(k) \neq \sigma(l) \forall k \neq l$.
    end for
    $M_i = \text{Median}(w_i^{(1)}, w_i^{(2)}, \ldots, w_i^{(\ell)})$
  end for
  Return $M_0 + (\frac{q}{r} - 1) \sum_{i=0}^{n'-1} M_{i+1} \left(\frac{q}{r}\right)^i$

The main result of this section is the following.

Theorem 3. *Let D be any $n \times n$ matrix. Let $q > n$ be a power of prime and $r = \lfloor \frac{q-1}{2} \rfloor$. For any $\delta > 0$, Algorithm 3 makes $\Theta(n^2 \text{poly}(\log \frac{n}{\delta}))$ calls to the MAX-oracle and, with probability at least $1-\delta$ outputs a $(\frac{q}{r})^2 = (4+O(1/n))$-approximation of $\text{Perm}(D)$.*

The proof of Theorem 3 follows the same trajectory as in Theorem 1. The constraints in MAX-oracle ensures that a permutation is always returned. So in the proof of Theorem 1, the $w_i^{(k)}$s can be though of as permutations instead in this setting. It should be noted that, we must take $q > n$ for PERM-WISH to work. That is the reason we get a $(4 + O(1/n))$-approximation for the permanent.

It also has to be noted that, since q is large, the straightforward extension of WISH algorithm would have provided only a $q^2 = n^2$-approximation of the permanent. Therefore the idea of using optimizations with multiple bins are crucial here as it lead to a close to 4-approximation.