Approximate Counting of k-Paths in $O^*(2^k)$ Time

Tomohiro Koana
Graduate School of Information Science and Technology,
The University of Tokyo, Japan

Abstract

We resolve the conjecture of Koutis and Williams that k-paths can be approximately counted in $O^*(2^k)$ time. For a simple directed graph with n vertices and m arcs, our randomized algorithm returns a $(1 \pm \varepsilon)$-approximation with failure probability at most δ in $O^*(2^k \varepsilon^{-2} \log(1/\delta))$ time.

1 Introduction

A directed k-path is an ordered tuple $(v_1, \ldots, v_k)$ of distinct vertices such that (v_i, v_{i+1}) is an arc for every $1 \leq i < k$. The k-path problem asks whether a given graph contains such a path. It is a benchmark problem in parameterized algorithms [9, 18]: after the early $k!\,\mathrm{poly}(n)$-time algorithm of Monien and the color-coding method of Alon, Yuster, and Zwick [6, 21], algebraic fingerprints led to randomized algorithms with running times $O^*(2^{3k/2})$ [14] and $O^*(2^k)$ for directed graphs [26]. Here and throughout, O^* suppresses factors polynomial in the parameter and the input size, but not the displayed dependence on ε and δ. In contrast, exactly counting k-paths is #W[1]-hard for both directed and undirected graphs [13]. This gap raises a natural intermediate question.

Can directed k-paths be approximately counted in $O^*(2^k)$ time?

The existence of such an algorithm was conjectured by Koutis and Williams [15]. Color-coding gives a randomized $(1 \pm \varepsilon)$-approximation in $O((2e)^k m \varepsilon^{-2})$ time for constant success probability [3]. Extensor-coding subsequently gave a faster algorithm with a running time of $O^*(4^k)$ [9]. Using representative weight functions, Lokshtanov, Saurabh, and Zehavi improved the running time to $O((2.619^k + n^{o(1)})\varepsilon^{-2}(n+m))$ [18]. We resolve the conjecture of Koutis and Williams:

Theorem 1. *Let $k \geq 1$ and $0 < \varepsilon, \delta < 1$. Given a simple directed graph G, there is a randomized algorithm that returns a $(1 \pm \varepsilon)$-approximation to the number of directed k-paths in G with success probability at least $1 - \delta$ and runs in $O^*(2^k \varepsilon^{-2} \log(1/\delta))$ time.*

Our technique. Our algorithm combines exterior algebra with repeated Alon–Matias–Szegedy (AMS) sketches [5]. We describe the construction for directed k-path. Let G be the input digraph and set $\ell = 2k$. Independently assign each $v \in V(G)$ a vector $\sigma_v \in \{-1, 1\}^k$ with independent uniform coordinates. We work in the exterior algebra $\Lambda(\mathbb{Z}^k)$, whose 2^k basis elements are indexed by subsets of $\{1, \ldots, k\}$. Its multiplication, the wedge product, is multilinear and alternating on vectors. Thus the product $\sigma_{v_1} \wedge \cdots \wedge \sigma_{v_k}$ vanishes if two of the v_i coincide. Otherwise, its coefficient in the one-dimensional degree-k component is $\det[\sigma_{v_1} \ \cdots \ \sigma_{v_k}]$. These labels remove walks with repeated vertices, as in extensor-coding [9].

Form a directed acyclic graph H with k layers. For every $i \in \{1, \ldots, k\}$ and $v \in V(G)$, layer i contains ℓ copies (i, v, a), indexed by colors $a \in \{1, \ldots, \ell\}$. For every arc (u, v) of G and every

$i \in \{2, \dots, k\}$, add an arc from $(i-1, u, b)$ to (i, v, a) for every pair of colors b, a. A path through all k layers of H thus specifies a k-vertex walk in G and a color for each position.

Independently of the vectors σ_v, for each layer $i \in \{2, \dots, k\}$, vertex $v \in V(G)$, and pair of colors $b, a \in \{1, \dots, \ell\}$, independently sample a uniform sign $r_i(v, b, a) \in \{-1, 1\}$. Give every arc from $(i-1, u, b)$ to (i, v, a) the weight $r_i(v, b, a)$. Thus, for fixed i, v, b, a, all incoming arcs share one sign, regardless of the predecessor u. Signs indexed by different tuples (i, v, b, a) are independent, in particular across layers. Independently sample a uniform sign $r_1(v, a) \in \{-1, 1\}$ for every first-layer copy $(1, v, a)$. The weight of a path through all layers is the product of its initial sign and its $k-1$ arc weights.

To explain the role of AMS sketches, consider the tensor Ψ indexed by tuples $(v_1, \dots, v_k) \in V(G)^k$. Its entry is $\det[\sigma_{v_1} \ \cdots \ \sigma_{v_k}]$ if the tuple is a walk in G, and zero otherwise. The squared Frobenius norm $\|\Psi\|_{\mathrm{F}}^2$ is the sum of the squares of its entries. Random-determinant moment bounds show that $\|\Psi\|_{\mathrm{F}}^2/k!$ is an unbiased estimator of the path count with polynomially bounded relative variance. We compress this tensor using AMS sketches, which replace each column of a flattened tensor by ℓ independent random signed sums. The sum of the squares of all output entries, divided by ℓ, is an unbiased estimate of the squared Frobenius norm of the input, since cross terms have expectation zero. The signs $r_1(v, a)$ compress the first vertex coordinate into a color. For each $i \geq 2$, the signs $r_i(v, b, a)$ compress the previous color b together with the next vertex v into a new color a, using the same signs for every choice of the remaining vertex coordinates. After k steps, only the final color remains. This sequential compression is an instance of recursive tensor sketching, as developed by Ahle and Knudsen [2] and in the combined work of Ahle et al. [1].

Let $S_k(a)$ be the resulting coordinate for color a. Equivalently, it is the sum, over all paths through the layers ending in color a, of the path weight times the determinant of its original vertex labels. One trial returns $\sum_{a=1}^{\ell} S_k(a)^2/(\ell^k k!)$. After normalization, each AMS compression preserves the expected squared norm and increases the fourth moment by a factor of at most $1 + 2/\ell$. Since $\ell = 2k$, the accumulated factor is at most $(1 + 1/k)^k \leq e$, so the final estimator retains polynomially bounded relative variance. To compute it, dynamic programming on H stores at each copy the sum of weighted wedge products of paths reaching that copy, using σ_v at every copy of v. Summing the final-layer values separately for each color gives the coefficients $S_k(a)$. Each stored exterior-algebra element has at most 2^k coordinates, giving $O^*(2^k)$ time per trial.

The k-path algorithm is a consequence of a more general theorem for counting multilinear monomials in polynomials computed by homogeneous 0–1 right-skew circuits. Associate a directed walk $(v_1, \dots, v_k)$ with the noncommutative monomial $x_{v_1} \cdots x_{v_k}$. The walk is a path exactly when this monomial is multilinear, meaning that no variable occurs twice. Noncommutativity preserves the vertex order, so different walks do not collapse into the same monomial.

We call a noncommutative polynomial a 0–1 *polynomial* if all its coefficients belong to $\{0, 1\}$. For such a polynomial P, let $N(P)$ be the number of its multilinear monomials. A homogeneous 0–1 right-skew circuit is built from constants 0 and 1, addition, and product gates that multiply a previously computed polynomial from the right by one variable. Every gate is required to compute a 0–1 polynomial. See Section 2 for the formal model.

Theorem 2. *Let P be a noncommutative 0–1 polynomial of degree $k \geq 1$ computed by a homogeneous 0–1 right-skew circuit C. For $0 < \varepsilon, \delta < 1$, there is a randomized algorithm that, given C, returns $\widehat{N}$ such that $\Pr[|\widehat{N} - N(P)| \leq \varepsilon N(P)] \geq 1 - \delta$ and runs in $O^*(2^k \varepsilon^{-2} \log(1/\delta))$ time.*

Related work. Arvind and Raman gave early randomized parameterized counting approximation algorithms for patterns of bounded treewidth, including paths [7]. Alon and Gutner later used balanced families of perfect hash functions to derandomize color-coding and obtain deterministic multiplicative approximations for counting paths [4].

Using lifted labels in the exterior algebra on $\mathbb{Z}^{2k}$, Brand, Dell, and Husfeldt gave a randomized $O^*(4^k\varepsilon^{-2})$-time approximation for directed k-path [9]. Lokshtanov, Björklund, Saurabh, and Zehavi subsequently obtained polynomial-space algorithms for directed and undirected k-path counting. Their randomized running time was $O^*(4^{k+O(\log k(\log k+\log(1/\varepsilon)))})$, whereas their deterministic running time was $O^*(4^{k+O(\sqrt{k}(\log^2 k+\log^2(1/\varepsilon)))})$ [17].

Lokshtanov, Saurabh, and Zehavi [18] used representative weight functions to obtain the $O^*(2.619^k\varepsilon^{-2})$-time approximation for k-path. More generally, for a homogeneous degree-k commutative polynomial represented by a monotone arithmetic circuit C, they gave a randomized $(1\pm\varepsilon)$-approximation to the sum of the coefficients of its multilinear monomials in $O^*(3.841^k\varepsilon^{-6})$ time. Pratt gave a randomized constant-success $(1\pm\varepsilon)$-approximation for the corresponding coefficient sum of a homogeneous commutative polynomial with nonnegative real coefficients, given by black-box evaluation, in $O^*(4.075^k\varepsilon^{-2}\log(1/\varepsilon))$ time and polynomial space [24].

Recursive tensor sketching controls the accumulation of error by compressing after each tensor product. Ahle and Knudsen [2] gave a sequential construction, and the combined work of Ahle et al. [1] developed recursive subspace embeddings with sketch dimension polynomial in the tensor order. Ma and Solomonik [19] studied Gaussian embeddings for more general tensor networks, including sequential tensor-train embeddings. Sharma, Khan, and Pratap [25] applied recursive sketching to estimating inner products of higher-order tensors. Our algorithm combines recursive sketching with the determinant-weighted tensor from extensor-coding and evaluates the resulting sketch over a right-skew circuit using exterior algebra, without forming the tensor explicitly. This combination reduces the $O^*(4^k)$ cost of computing the uncompressed squared norm to $O^*(2^k)$ per trial.

For the decision problem, group-algebra fingerprints annihilated nonmultilinear terms and yielded randomized multilinear detection algorithms [14, 15, 26]. Brand and Pratt developed a symbolic-differentiation framework for skew circuits based on determinants [10]. Determinantal sieving detected multilinear monomials whose supports formed bases of a linear matroid. Over general fields, it evaluated a circuit over an exterior algebra [12]. Dynamic representative sets gave a deterministic $O^*(2^{k+O(\sqrt{k}\log^2 k)})$-time algorithm for weighted directed k-path in the word-RAM model, with every edge weight stored in one word [22]. For undirected k-path, narrow sieves gave a polynomial-space randomized $O^*(1.66^k)$-time decision algorithm with constant one-sided error [8].

Organization. Section 2 formalizes the polynomial, tensor, circuit, and exterior-algebra notation used later. Section 3.1 defines the estimator and proves its approximation guarantee. Section 3.2 gives its circuit implementation and proves Theorem 2. Section 4 derives Theorems 1 and 5.

2 Preliminaries

Polynomials and tensors. Fix $n \geq 1$. For a positive integer r, let $[r] = \{1, \ldots, r\}$. We work in the noncommutative polynomial ring $\mathbb{Z}\langle x_1, \ldots, x_n\rangle$. For every positive integer i, each degree-i monomial is uniquely indexed by a tuple $\tau = (\tau_1, \ldots, \tau_i) \in [n]^i$, and we write $x_\tau = x_{\tau_1} \cdots x_{\tau_i}$. The monomial x_τ is *multilinear* if the entries of τ are pairwise distinct.

A homogeneous degree-i polynomial has the form $P = \sum_{\tau\in[n]^i} c_\tau x_\tau$. It is a 0–1 *polynomial* if $c_\tau \in \{0, 1\}$ for every $\tau \in [n]^i$. For such a polynomial, define $N(P) = \sum_{\tau\in[n]^i : x_\tau \text{ multilinear}} c_\tau$. The polynomial P can equivalently be represented by its coefficient tensor $(c_\tau)_{\tau\in[n]^i} \in \mathbb{Z}^{[n]^i}$. More generally, a real tensor $A \in \mathbb{R}^{I_1\times\cdots\times I_r}$ is an array indexed by finite sets $I_1, \ldots, I_r$. Its Frobenius norm is

$$\|A\|_{\mathrm{F}} = \left(\sum_{(a_1,\ldots,a_r)\in I_1\times\cdots\times I_r} A(a_1, \ldots, a_r)^2 \right)^{1/2}.$$

To *flatten* a tensor with specified row coordinates means to view the same entries as a matrix whose row index is the Cartesian product of these coordinates and whose column index is the Cartesian product of the remaining coordinates. Flattening only reindexes the entries and therefore preserves the Frobenius norm.

Circuits. A *homogeneous* 0–1 *right-skew circuit* is a directed acyclic graph with a designated output gate. Each gate is a constant input gate labeled 0 or 1, a binary addition gate $g = h_1 + h_2$, or a product gate $g = hx_j$ with $j \in [n]$, where h, h_1, h_2 are predecessor gates of g. These operations recursively define the noncommutative polynomial computed at each gate, and we require every such polynomial to be a 0–1 polynomial. Every gate also has a syntactic degree. Constants have degree zero, the two inputs and the output of an addition gate have the same degree, and hx_j has degree one more than h.

Exterior algebra. We adapt the concrete presentation of exterior algebra from Brand, Dell, and Husfeldt [9, Section 2] to the coefficient ring $\mathbb{Z}$. Let $e_1, \dots, e_k$ be the standard basis of $\mathbb{Z}^k$. For every $I \subseteq [k]$, let e_I be a formal basis element, where $e_\varnothing = 1$ and $e_{\{j\}} = e_j$. For $0 \le r \le k$, the rth *exterior power* $\Lambda^r(\mathbb{Z}^k)$ is the free $\mathbb{Z}$-module with basis $\{e_I : I \subseteq [k],\ |I| = r\}$. The *exterior algebra* is $\Lambda(\mathbb{Z}^k) = \bigoplus_{r=0}^{k} \Lambda^r(\mathbb{Z}^k)$. We identify $\mathbb{Z}^k$ with $\Lambda^1(\mathbb{Z}^k)$ through the basis $e_1, \dots, e_k$.

The *wedge product* is the $\mathbb{Z}$-bilinear multiplication determined on basis elements by

$$e_I \wedge e_J = \begin{cases} 0, & I \cap J \neq \varnothing, \\ (-1)^{|\{(a,b) \in I \times J : a > b\}|} e_{I \cup J}, & I \cap J = \varnothing. \end{cases}$$

It is associative, respects the grading $\Lambda^r(\mathbb{Z}^k) \wedge \Lambda^s(\mathbb{Z}^k) \subseteq \Lambda^{r+s}(\mathbb{Z}^k)$, and is alternating on vectors. In particular, $y \wedge y = 0$ for every $y \in \mathbb{Z}^k$. If $I = \{i_1 < \cdots < i_r\}$, then $e_I = e_{i_1} \wedge \cdots \wedge e_{i_r}$. Thus $\Lambda^r(\mathbb{Z}^k)$ has $\binom{k}{r}$ coordinates and the full exterior algebra has 2^k coordinates. Finally, for $y_1, \dots, y_k \in \mathbb{Z}^k$, expanding the wedge product gives

$$y_1 \wedge \cdots \wedge y_k = \det[y_1 \ \cdots \ y_k] e_{[k]}.$$

3 The algorithm

In this section, we prove Theorem 2.

Theorem 2. *Let P be a noncommutative 0–1 polynomial of degree $k \ge 1$ computed by a homogeneous 0–1 right-skew circuit C. For $0 < \varepsilon, \delta < 1$, there is a randomized algorithm that, given C, returns $\widehat{N}$ such that $\Pr[|\widehat{N} - N(P)| \le \varepsilon N(P)] \ge 1 - \delta$ and runs in $O^*(2^k \varepsilon^{-2} \log(1/\delta))$ time.*

Throughout this section, fix a polynomial P and a homogeneous 0–1 right-skew circuit C computing it as in Theorem 2, with variables $x_1, \dots, x_n$. For every $\tau \in [n]^k$, let c_τ be the coefficient of x_τ in P. Let $N = N(P)$, set $\ell = 2k$, and fix $0 < \varepsilon, \delta < 1$. In this section, O^* suppresses factors polynomial in k and in the encoding size of C, but not the displayed dependence on ε and δ.

Our algorithm returns 0 whenever $N = 0$, so we assume $N > 0$ below.

We call a random variable Y an *efficient estimator* of N if $\mathbb{E}(Y) = N > 0$ and $\mathbb{E}(Y^2)/(\mathbb{E}(Y))^2$ is bounded by $k^{O(1)}$. The standard median-of-means bound shows that

$$O\left(\frac{\mathbb{E}(Y^2)}{(\mathbb{E}(Y))^2} \varepsilon^{-2} \log(1/\delta) \right)$$

independent samples of Y suffice to compute $\widehat{N}$ such that $\Pr[|\widehat{N} - N| > \varepsilon N] \le \delta$. See, for example, Mitzenmacher and Upfal [20, Sections 3.3–3.4]. It therefore suffices to construct an efficient estimator of N and compute each sample in $O^*(2^k)$ time.

3.1 Constructing an efficient estimator

Let $\sigma_{j,r}$, $j \in [n]$ and $r \in [k]$, be mutually independent random variables, each uniform on $\{-1, 1\}$, and define $\sigma_j = (\sigma_{j,1}, \ldots, \sigma_{j,k})^{\mathsf{T}} \in \mathbb{Z}^k$. For every $\tau = (\tau_1, \ldots, \tau_k) \in [n]^k$, let $D_\tau = \det[\sigma_{\tau_1} \cdots \sigma_{\tau_k}]$. Define $\Psi \in \mathbb{R}^{[n]^k}$ by $\Psi(\tau) = c_\tau D_\tau$. Recall that its Frobenius norm is $\|\Psi\|_{\mathrm{F}} = (\sum_{\tau \in [n]^k} \Psi(\tau)^2)^{1/2}$. Since $c_\tau \in \{0, 1\}$ for every $\tau \in [n]^k$, we have

$$\|\Psi\|_{\mathrm{F}}^2 = \sum_{\tau \in [n]^k} c_\tau^2 D_\tau^2 = \sum_{\tau \in [n]^k} c_\tau D_\tau^2.$$

If x_τ is not multilinear, then the matrix defining D_τ has two equal columns, so $D_\tau = 0$.

We first show that the normalized squared Frobenius norm $\|\Psi\|_{\mathrm{F}}^2/k!$ is an efficient estimator of N. The algorithm of Brand, Dell, and Husfeldt uses this estimator [9, Section 3.6]. We include the proof in our setting for completeness.

Lemma 3. *$\mathbb{E}(\|\Psi\|_{\mathrm{F}}^2) = k!N$ and $\mathbb{E}(\|\Psi\|_{\mathrm{F}}^4) \le (k!)^2 k^3 N^2$.*

Proof. Fix $\tau \in [n]^k$ such that x_τ is multilinear. Since $\tau_1, \ldots, \tau_k$ are distinct, D_τ has the same distribution as $\det(B)$, where B is a $k \times k$ *random sign matrix* with independent entries uniform in $\{-1, 1\}$. The determinant-moment formulas of Nyquist, Rice, and Riordan [23] imply

$$\mathbb{E}(\det(B)^2) = k! \qquad \text{and} \qquad \mathbb{E}(\det(B)^4) \le (k!)^2 k^3.$$

Let $\mathcal{M}$ be the set of $\tau \in [n]^k$ such that $c_\tau = 1$ and x_τ is multilinear. Then $|\mathcal{M}| = N$. Since $D_\tau = 0$ when x_τ is not multilinear,

$$\|\Psi\|_{\mathrm{F}}^2 = \sum_{\tau \in \mathcal{M}} D_\tau^2.$$

Taking expectations gives $\mathbb{E}(\|\Psi\|_{\mathrm{F}}^2) = k!N$. Moreover, Cauchy–Schwarz gives

$$\|\Psi\|_{\mathrm{F}}^4 = \left(\sum_{\tau \in \mathcal{M}} D_\tau^2 \right)^2 \le \left(\sum_{\tau \in \mathcal{M}} 1 \right) \left(\sum_{\tau \in \mathcal{M}} D_\tau^4 \right) = N \sum_{\tau \in \mathcal{M}} D_\tau^4.$$

Taking expectations and using $\mathbb{E}(D_\tau^4) \le (k!)^2 k^3$ for every $\tau \in \mathcal{M}$ gives $\mathbb{E}(\|\Psi\|_{\mathrm{F}}^4) \le (k!)^2 k^3 N^2$. □

The extensor-coding construction of Brand, Dell, and Husfeldt computes $\|\Psi\|_{\mathrm{F}}^2$ in $O^*(4^k)$ arithmetic operations [9]. It uses *lifts* that map each $\sigma_j \in \mathbb{Z}^k \subseteq \Lambda(\mathbb{Z}^k)$ to an element of $\Lambda^2(\mathbb{Z}^{2k})$ and evaluates C over $\Lambda(\mathbb{Z}^{2k})$. We instead compress Ψ using random sign matrices.

Compression by random sign matrices. Our compression applies the Alon–Matias–Szegedy (AMS) sketch [5] to the columns of a flattened tensor. An AMS sketch stores random signed sums of vector coordinates. Applying ℓ such sums to each column and dividing the sum of all squared outputs by ℓ gives an unbiased estimate of the squared Frobenius norm of the matrix. At each step, we use the same random sign matrix for all columns of the flattened tensor.

Independently of the vectors $\sigma_1, \ldots, \sigma_n$, let $R_1 \in \{-1, 1\}^{[\ell] \times [n]}$ and $R_i \in \{-1, 1\}^{[\ell] \times ([\ell] \times [n])}$ for every $2 \le i \le k$ be random matrices whose entries are mutually independent and uniform. Flatten Ψ with its first coordinate as the row coordinate, multiply it from the left by R_1, and let the resulting tensor be $S_1 \in \mathbb{R}^{[\ell] \times [n]^{k-1}}$. Here and below, a factor $[n]^0$ is omitted. For each $i = 2, \ldots, k$, flatten $S_{i-1} \in \mathbb{R}^{[\ell] \times [n]^{k-i+1}}$ with its first two coordinates as the row coordinates, multiply it from the left by R_i, and denote the result by $S_i \in \mathbb{R}^{[\ell] \times [n]^{k-i}}$. Thus $S_k \in \mathbb{R}^{[\ell]} \cong \mathbb{R}^\ell$. For every finite-dimensional real vector y, write $\|y\|_2^2$ for the sum of the squares of its coordinates. The final vector is given, for every $a_k \in [\ell]$, by

$$S_k(a_k) = \sum_{\tau \in [n]^k} \Psi(\tau) \sum_{a_1, \ldots, a_{k-1} \in [\ell]} R_1[a_1, \tau_1] \left(\prod_{t=2}^{k} R_t[a_t, (a_{t-1}, \tau_t)] \right). \tag{3.1}$$

We now verify that S_k yields an efficient estimator of N. The key point is that, after normalization, successive compression by random sign matrices preserves the expected squared Frobenius norm and increases its fourth moment by only a constant factor.

Lemma 4. *$\|S_k\|_2^2/(\ell^k k!)$ is an efficient estimator of N.*

Proof. The claim below follows from the variance bound of Alon, Matias, and Szegedy [5, Section 2.2], whose proof we include for completeness, and the Cauchy–Schwarz inequality.

Claim. *Let d and t be positive integers. Let X be a random matrix in $\mathbb{R}^{d\times t}$, and let $R \in \{-1,1\}^{\ell\times d}$ be a random sign matrix independent of X. Then*

$$\mathbb{E}(\|RX\|_{\mathrm{F}}^2) = \ell\,\mathbb{E}(\|X\|_{\mathrm{F}}^2) \quad \textit{and} \quad \mathbb{E}(\|RX\|_{\mathrm{F}}^4) \le \ell^2(1+2/\ell)\mathbb{E}(\|X\|_{\mathrm{F}}^4).$$

Proof. First suppose that $X \in \mathbb{R}^{d\times t}$ is fixed. For every $b \in [t]$, let $y_b \in \mathbb{R}^d$ be its bth column and define $U_b = \|Ry_b\|_2^2$.

Fix $z = (z_1,\dots,z_d) \in \mathbb{R}^d$, let $r_1,\dots,r_d$ be the entries of one row of R, and define $Q = (\sum_{u=1}^d r_u z_u)^2$. Since $r_u^2 = 1$ for every u,

$$Q = \sum_{u=1}^d z_u^2 + 2\sum_{1\le u<v\le d} z_u z_v r_u r_v.$$

The signs are independent and have mean zero, so $\mathbb{E}(Q) = \sum_{u=1}^d z_u^2 = \|z\|_2^2$. When we square $Q - \mathbb{E}(Q)$, every cross term indexed by two distinct pairs has expectation zero: at least one sign occurs exactly once and is independent of the others. The terms indexed by the same pair have $r_u^2 r_v^2 = 1$. Hence

$$\mathrm{Var}(Q) = 4\sum_{1\le u<v\le d} z_u^2 z_v^2 \le 2\left(\sum_{u=1}^d z_u^2\right)^2 = 2\|z\|_2^4.$$

For each fixed $b \in [t]$, the variable U_b is the sum of ℓ independent copies of Q with $z = y_b$, one for each row of R. Linearity of expectation and additivity of variance give

$$\mathbb{E}(U_b) = \ell\|y_b\|_2^2 \quad \text{and} \quad \mathrm{Var}(U_b) \le 2\ell\|y_b\|_2^4.$$

Therefore,

$$\mathbb{E}(U_b^2) = \mathrm{Var}(U_b) + (\mathbb{E}(U_b))^2 \le (2\ell+\ell^2)\|y_b\|_2^4 = \ell^2(1+2/\ell)\|y_b\|_2^4.$$

Since $\|RX\|_{\mathrm{F}}^2 = \sum_{b=1}^t U_b$, linearity of expectation gives $\mathbb{E}(\|RX\|_{\mathrm{F}}^2) = \ell\|X\|_{\mathrm{F}}^2$. Cauchy–Schwarz gives

$$\begin{aligned}\mathbb{E}(\|RX\|_{\mathrm{F}}^4) = \sum_{b,c=1}^t \mathbb{E}(U_b U_c) &\le \left(\sum_{b=1}^t \sqrt{\mathbb{E}(U_b^2)}\right)^2 \\ &\le \ell^2(1+2/\ell)\left(\sum_{b=1}^t \|y_b\|_2^2\right)^2 = \ell^2(1+2/\ell)\|X\|_{\mathrm{F}}^4.\end{aligned}$$

This argument allows the U_b to share the randomness in R. For random X independent of R, conditioning on X and taking expectations proves both bounds. □

We apply the claim at each flattening. Let $S_0 = \Psi$. For each $i \in [k]$, the matrix multiplied by R_i is a flattening of S_{i-1}. It depends only on $\sigma_1,\dots,\sigma_n, R_1,\dots,R_{i-1}$ and is therefore independent of R_i. Since flattening preserves the Frobenius norm, applying the preceding bounds inductively gives, for every $i \in [k]$,

$$\begin{aligned}&\mathbb{E}(\|S_i\|_{\mathrm{F}}^2) = \ell\,\mathbb{E}(\|S_{i-1}\|_{\mathrm{F}}^2),\\ &\mathbb{E}(\|S_i\|_{\mathrm{F}}^4) \le \ell^2(1+2/\ell)\mathbb{E}(\|S_{i-1}\|_{\mathrm{F}}^4).\end{aligned}$$

By iterating from $S_0 = \Psi$, using $\|S_k\|_2 = \|S_k\|_{\mathrm{F}}$, and applying Lemma 3, we obtain

$$\mathbb{E}(\|S_k\|_2^2/(\ell^k k!)) = \mathbb{E}(\|\Psi\|_{\mathrm{F}}^2)/k! = N,$$
$$\mathbb{E}((\|S_k\|_2^2/(\ell^k k!))^2) \le \left(1 + \frac{2}{\ell}\right)^k k^3 N^2 = \left(1 + \frac{1}{k}\right)^k k^3 N^2.$$

Since $(1 + 1/k)^k \le e$, the lemma follows. □

By Lemma 4 and the median-of-means bound above, $O^*(\varepsilon^{-2}\log(1/\delta))$ independent trials suffice to return $\widehat{N}$ such that $\Pr[|\widehat{N} - N| > \varepsilon N] \le \delta$. Thus it remains only to compute each sample within the claimed running time.

3.2 Sampling from the efficient estimator

Use the random vectors $\sigma_1, \ldots, \sigma_n$ and random matrices $R_1, \ldots, R_k$ defined in Section 3.1. Evaluate the ancestors of the output gate over the exterior algebra $\Lambda(\mathbb{Z}^k)$ from Section 2.

For every gate g of degree $i \ge 1$ and every $\tau \in [n]^i$, let $c_g(\tau)$ be the coefficient of x_τ in the noncommutative polynomial computed at g; since C is a 0–1 circuit, $c_g(\tau) \in \{0, 1\}$. The values below store signed sums of these monomial contributions after replacing variables by their exterior-algebra labels. Evaluate these gates in topological order. Degree-zero gates store their values in $\{0, 1\}$. For every gate g of degree $i \ge 1$ and every $a \in [\ell]$, store $F_{g,a} \in \Lambda^i(\mathbb{Z}^k)$. Addition gates are evaluated componentwise. If $g = hx_j$ has degree one, we may assume that h stores 1, since otherwise g computes zero and can be ignored. Define $F_{g,a} = R_1[a, j]\sigma_j$. If $g = hx_j$ has degree $i \ge 2$, define $F_{g,a} = (\sum_{b=1}^{\ell} R_i[a, (b, j)]F_{h,b}) \wedge \sigma_j$.

For every gate g of degree $i \ge 1$ and every $a_i \in [\ell]$, we claim that

$$F_{g,a_i} = \sum_{\tau \in [n]^i} c_g(\tau) \sum_{a_1, \ldots, a_{i-1} \in [\ell]} R_1[a_1, \tau_1] \left(\prod_{t=2}^{i} R_t[a_t, (a_{t-1}, \tau_t)] \right) \cdot \sigma_{\tau_1} \wedge \cdots \wedge \sigma_{\tau_i}. \tag{3.2}$$

For $i = 1$, the inner sum is omitted and the product is interpreted as 1.

We prove (3.2) by induction over the gates. At a degree-one product gate $g = hx_j$, we may assume that h stores 1. The coefficient of x_j is then one, all other coefficients are zero, and $F_{g,a_1} = R_1[a_1, j]\sigma_j$. At an addition gate $g = h_1 + h_2$, both the coefficients and the stored elements are the sums of the corresponding values at h_1 and h_2.

Now let $g = hx_j$ have degree $i \ge 2$. A monomial $x_\tau = x_{\tau_1} \cdots x_{\tau_i}$ has nonzero coefficient at g only if $\tau_i = j$, and in this case $c_g(\tau) = c_h((\tau_1, \ldots, \tau_{i-1}))$. Substituting the expansion of each $F_{h,b}$ into the recurrence makes the sum over b the sum over a_{i-1}, introduces the factor $R_i[a_i, (a_{i-1}, \tau_i)]$, and appends σ_{τ_i} to the wedge product. This proves (3.2).

Let o be the output gate. Since $\Psi(\tau) = c_\tau D_\tau$, $c_o(\tau) = c_\tau$, and $\sigma_{\tau_1} \wedge \cdots \wedge \sigma_{\tau_k} = D_\tau e_{[k]}$, (3.1) and (3.2) give, for every $a_k \in [\ell]$,

$$F_{o,a_k} = \sum_{\tau \in [n]^k} c_\tau D_\tau \sum_{a_1, \ldots, a_{k-1} \in [\ell]} R_1[a_1, \tau_1] \left(\prod_{t=2}^{k} R_t[a_t, (a_{t-1}, \tau_t)] \right) e_{[k]} = S_k(a_k) e_{[k]}.$$

Thus the stored output coefficients determine $\|S_k\|_2^2/(\ell^k k!) = (\ell^k k!)^{-1} \sum_{a=1}^{\ell} S_k(a)^2$.

Running time. Let L be the encoding size of C. After relabeling the variables occurring in C, we may assume that $n \le L$. By Section 2, an element of $\Lambda^i(\mathbb{Z}^k)$ has at most 2^k coefficients. Since $\ell = 2k$, every gate can be evaluated using $2^k \operatorname{poly}(k)$ integer operations. For a degree-i gate, each coefficient in (3.2) is a sum of at most $L^i \ell^{i-1}$ terms, each of which is a coefficient of an i-fold wedge of sign vectors and hence has absolute value at most $i!$. Thus every integer computed in one trial has $O(k(1 + \log L + \log k))$ bits. Squaring and summing the output coefficients

changes this bound only by a constant factor. Therefore every integer operation within one trial takes time polynomial in k and L, and one trial takes $O^*(2^k)$ time. Each estimator value has denominator $\ell^k k!$, and averaging $O(k^3\varepsilon^{-2})$ values adds only $O(1 + \log k + \log(1/\varepsilon))$ bits to its numerator and denominator. The group averages have a common denominator, so their median can be selected using a linear number of comparisons between their numerators. These comparisons and the remaining amplification arithmetic take polynomial time. The amplification in Section 3.1 uses $O^*(\varepsilon^{-2}\log(1/\delta))$ trials, giving a total running time of $O^*(2^k\varepsilon^{-2}\log(1/\delta))$.

4 Applications

For completeness, we give the noncommutative-polynomial construction underlying Theorem 1. We also derive, by the same approach, a result for set packing.

4.1 Directed paths

A directed k-path is an ordered tuple $(v_1, \ldots, v_k)$ of distinct vertices such that (v_i, v_{i+1}) is an arc for every $1 \le i < k$.

Theorem 1. *Let $k \ge 1$ and $0 < \varepsilon, \delta < 1$. Given a simple directed graph G, there is a randomized algorithm that returns a $(1 \pm \varepsilon)$-approximation to the number of directed k-paths in G with success probability at least $1 - \delta$ and runs in $O^*(2^k\varepsilon^{-2}\log(1/\delta))$ time.*

Proof. Let the input digraph have vertex set $[n]$ and arc set A, where $|A| = m$. If $n < k$, return 0, so assume $n \ge k$. Define $Z_0 = 0$ and $Z_i = Z_{i-1}x_1$ for $i \in [k]$; thus Z_i is a syntactically homogeneous zero polynomial of degree i. For every $v \in [n]$, define $P_{v,1} = x_v$. For $i = 2, \ldots, k$ and $v \in [n]$, define

$$P_{v,i} = \left(\sum_{u:(u,v)\in A} P_{u,i-1} \right) x_v,$$

where an empty sum is represented by Z_{i-1}. The polynomial $P_{v,i}$ stores the directed walks on i vertices that end at v.

We prove this claim by induction on i. It holds for $i = 1$. For $i \ge 2$, each summand indexed by an arc (u, v) appends v to a walk ending at u. Conversely, deleting the final vertex of a walk ending at v identifies its summand. The monomial records the entire vertex sequence and therefore occurs with coefficient one.

Let $P_k = \sum_{v\in[n]} P_{v,k}$. The last variable identifies the endpoint, so every monomial of P_k still has coefficient one. Its monomials encode all directed walks on k vertices, and a monomial is multilinear exactly when its variables are distinct. Thus the multilinear monomials of P_k are in bijection with the directed k-paths.

Every intermediate polynomial is 0–1: the predecessor sums combine disjoint supports, right multiplication is injective on monomials, and the final sum combines monomials with distinct last variables. Thus these recurrences and the final sum form a polynomial-size homogeneous 0–1 right-skew circuit. Applying Theorem 2 to P_k gives the claimed approximation and running time. □

4.2 Set packing

A k-packing is a k-element subfamily of pairwise disjoint sets. Counting k-matchings is #W[1]-hard even in bipartite graphs [11]. Liu, Wang, and Wang gave fixed-parameter randomized approximation schemes for counting packings in families of sets of fixed size [16]. For families of d-element sets, the representative-weight method gives a randomized $(1 \pm \varepsilon)$-approximation in $O^*(2.619^{dk}\varepsilon^{-2})$ time [18]. The following theorem improves this exponential factor to 2^{dk}.

Theorem 5. *Let $d, k \geq 1$ and $0 < \varepsilon, \delta < 1$. Given a family $\mathcal{E}$ of $m \geq k$ distinct d-element sets, there is a randomized algorithm that returns a $(1 \pm \varepsilon)$-approximation to the number of k-packings in $\mathcal{E}$ with success probability at least $1 - \delta$ and runs in $O^*(2^{dk}\varepsilon^{-2}\log(1/\delta))$ time.*

Proof. Let $\mathcal{E} = \{E_1, \ldots, E_m\}$ be the input family. Index a variable x_u by every ground element u. For each $h \in [m]$, choose an ordering $u_{h,1}, \ldots, u_{h,d}$ of E_h and define $M_h = x_{u_{h,1}} \cdots x_{u_{h,d}}$. We list selected sets in increasing input order because a commutative product would merge distinct subfamilies with the same union.

For every $h \in [m]$ and $0 \leq j \leq \min\{h, k\}$, the polynomial $P_{h,j}$ stores the j-element subfamilies of $\{E_1, \ldots, E_h\}$. Define $P_{0,0} = 1$ and $P_{h,0} = 1$ for $h \in [m]$. For every $h \in [m]$ and $1 \leq j \leq \min\{h, k\}$, define

$$P_{h,j} = \begin{cases} P_{h-1,j} + P_{h-1,j-1}M_h, & j < h, \\ P_{h-1,h-1}M_h, & j = h. \end{cases}$$

Expanding each multiplication by M_h into d consecutive right-product gates implements these recurrences by a polynomial-size homogeneous right-skew circuit with output $P_{m,k}$.

Induction on h, using $P_{h,0} = 1$ as the base state, shows that for every $1 \leq j \leq \min\{h, k\}$, the monomials of $P_{h,j}$ are exactly $M_{i_1} \cdots M_{i_j}$ indexed by the sequences $1 \leq i_1 < \cdots < i_j \leq h$. The two terms of the recurrence correspond to subfamilies that omit or contain E_h, respectively. These cases are disjoint and exhaustive. Each such monomial has a unique decomposition into blocks of length d, and the input sets are distinct, so different subfamilies give different monomials. Each monomial has coefficient one. Each intermediate right-product gate appends a fixed prefix of M_h and therefore also computes a 0–1 polynomial. Hence the circuit above is a homogeneous 0–1 right-skew circuit.

Such a monomial is multilinear exactly when the selected sets are pairwise disjoint. Thus $P_{m,k}$ is a homogeneous 0–1 polynomial of degree dk, and $N(P_{m,k})$ is the number of k-packings in $\mathcal{E}$. Applying Theorem 2 gives the claimed approximation and running time. □

Acknowledgments

The author thanks Jesper Nederlof for pointing out a possible connection to the sketching literature.

This work was supported in part by JST CREST Grant Number JPMJCR24Q2 and JST ERATO Grant Number JPMJER2301.

Declaration of AI use

The original algorithmic ideas underlying this work were proposed by OpenAI's GPT-5.6 Pro. The author subsequently developed and substantially simplified the proofs, carefully checked all mathematical arguments, and takes full responsibility for the mathematical content of the paper.

References

[1] T. D. Ahle, M. Kapralov, J. B. T. Knudsen, R. Pagh, A. Velingker, D. P. Woodruff, and A. Zandieh. Oblivious sketching of high-degree polynomial kernels. In *Proceedings of the 2020 ACM–SIAM Symposium on Discrete Algorithms (SODA 2020)*, pages 141–160. SIAM, 2020. doi:10.1137/1.9781611975994.9.

[2] T. D. Ahle and J. B. T. Knudsen. Almost optimal tensor sketch. arXiv:1909.01821, 2019. doi:10.48550/arXiv.1909.01821.

[3] N. Alon, P. Dao, I. Hajirasouliha, F. Hormozdiari, and S. C. Sahinalp. Biomolecular network motif counting and discovery by color coding. In *Proceedings of the 16th International Conference on Intelligent Systems for Molecular Biology (ISMB 2008)*, pages 241–249, 2008. doi:10.1093/bioinformatics/btn163.

[4] N. Alon and S. Gutner. Balanced families of perfect hash functions and their applications. *ACM Transactions on Algorithms*, 6(3), Article 54, pages 1–12, 2010. doi:10.1145/1798596.1798607.

[5] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and System Sciences*, 58(1):137–147, 1999. doi:10.1006/jcss.1997.1545.

[6] N. Alon, R. Yuster, and U. Zwick. Color-coding. *Journal of the ACM*, 42(4):844–856, 1995. doi:10.1145/210332.210337.

[7] V. Arvind and V. Raman. Approximation algorithms for some parameterized counting problems. In *Proceedings of the 13th International Symposium on Algorithms and Computation (ISAAC 2002)*, LNCS 2518, pages 453–464. Springer, 2002. doi:10.1007/3-540-36136-7_40.

[8] A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto. Narrow sieves for parameterized paths and packings. *Journal of Computer and System Sciences*, 87:119–139, 2017. doi:10.1016/j.jcss.2017.03.003.

[9] C. Brand, H. Dell, and T. Husfeldt. Extensor-coding. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2018)*, pages 151–164. ACM, 2018. doi:10.1145/3188745.3188902.

[10] C. Brand and K. Pratt. Parameterized applications of symbolic differentiation of (totally) multilinear polynomials. In *Proceedings of the 48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*, LIPIcs 198, pages 38:1–38:19. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ICALP.2021.38.

[11] R. Curticapean and D. Marx. Complexity of counting subgraphs: Only the boundedness of the vertex-cover number counts. In *Proceedings of the 55th IEEE Annual Symposium on Foundations of Computer Science (FOCS 2014)*, pages 130–139. IEEE Computer Society, 2014. doi:10.1109/FOCS.2014.22.

[12] E. Eiben, T. Koana, and M. Wahlström. Determinantal sieving. *TheoretiCS*, 4, Article 21, 2025. doi:10.46298/theoretics.25.21.

[13] J. Flum and M. Grohe. The parameterized complexity of counting problems. *SIAM Journal on Computing*, 33(4):892–922, 2004. doi:10.1137/S0097539703427203.

[14] I. Koutis. Faster algebraic algorithms for path and packing problems. In *Proceedings of the 35th International Colloquium on Automata, Languages, and Programming (ICALP 2008)*, LNCS 5125, pages 575–586. Springer, 2008. doi:10.1007/978-3-540-70575-8_47.

[15] I. Koutis and R. Williams. Algebraic fingerprints for faster algorithms. *Communications of the ACM*, 59(1):98–105, 2016. doi:10.1145/2742544.

[16] Y. Liu, S. Wang, and J. Wang. Parameterized counting matching and packing: A family of hard problems that admit FPTRAS. *Theoretical Computer Science*, 734:83–93, 2018. doi:10.1016/j.tcs.2017.09.022.

[17] D. Lokshtanov, A. Björklund, S. Saurabh, and M. Zehavi. Approximate counting of k-paths: Simpler, deterministic, and in polynomial space. *ACM Transactions on Algorithms*, 17(3), Article 26, pages 1–44, 2021. doi:10.1145/3461477.

[18] D. Lokshtanov, S. Saurabh, and M. Zehavi. Efficient computation of representative weight functions with applications to parameterized counting (extended version). In *Proceedings of the 2021 ACM–SIAM Symposium on Discrete Algorithms (SODA 2021)*, pages 179–198. SIAM, 2021. doi:10.1137/1.9781611976465.13.

[19] L. Ma and E. Solomonik. Cost-efficient Gaussian tensor network embeddings for tensor-structured inputs. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, pages 38980–38993, 2022. doi:10.52202/068431-2825.

[20] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. Second edition. Cambridge University Press, 2017.

[21] B. Monien. How to find long paths efficiently. *Annals of Discrete Mathematics*, 25:239–254, 1985. doi:10.1016/S0304-0208(08)73110-4.

[22] J. Nederlof. Weighted k-path and other problems in almost $O^*(2^k)$ deterministic time via dynamic representative sets. In *Proceedings of the 66th IEEE Annual Symposium on Foundations of Computer Science (FOCS 2025)*, pages 2813–2824. IEEE, 2025. doi:10.1109/FOCS63196.2025.00143.

[23] H. Nyquist, S. O. Rice, and J. Riordan. The distribution of random determinants. *Quarterly of Applied Mathematics*, 12(2):97–104, 1954. JSTOR 43634123.

[24] K. Pratt. Waring rank, parameterized and exact algorithms. In *Proceedings of the 60th IEEE Annual Symposium on Foundations of Computer Science (FOCS 2019)*, pages 806–823. IEEE Computer Society, 2019. doi:10.1109/FOCS.2019.00053.

[25] A. Sharma, M. A. Khan, and R. Pratap. Efficient and accurate tensor compression via recursive sketching. In *Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS 2026)*, Proceedings of Machine Learning Research 300, pages 2062–2070. PMLR, 2026. `https://proceedings.mlr.press/v300/sharma26a.html`.

[26] R. Williams. Finding paths of length k in $O^*(2^k)$ time. *Information Processing Letters*, 109(6):315–318, 2009. doi:10.1016/j.ipl.2008.11.004.