Editors' Suggestion

Second moment of Hafnians in Gaussian boson sampling

Adam Ehrenberg [1,2] Joseph T. Iosue [1,2] Abhinav Deshpande [3] Dominik Hangleiter [1,4] and Alexey V. Gorshkov [1,2]

[1] *Joint Center for Quantum Information and Computer Science, NIST/University of Maryland College Park, Maryland 20742, USA*

[2] *Joint Quantum Institute, NIST/University of Maryland College Park, Maryland 20742, USA*

[3] *IBM Quantum, Almaden Research Center, San Jose, California 95120, USA*

[4] *Simons Institute for the Theory of Computing, University of California at Berkeley, Berkeley, California, 94720, USA*

(Received 4 April 2024; revised 11 November 2024; accepted 26 February 2025; published 8 April 2025)

Gaussian boson sampling is a popular method for experimental demonstrations of quantum advantage, but many subtleties remain in fully understanding its theoretical underpinnings. An important component in the theoretical arguments for approximate average-case hardness of sampling is anticoncentration, which is a second-moment property of the output probabilities. In Gaussian boson sampling these are given by hafnians of generalized circular orthogonal ensemble matrices. In a companion work by Ehrenberg *et al.* [Phys. Rev. Lett. **134**, 140601 (2025)], we develop a graph-theoretic method to study these moments and use it to identify a transition in anticoncentration. In this work, we find a recursive expression for the second moment using these graph-theoretic techniques. While we have not been able to solve this recursion by hand, we are able to solve it numerically exactly, which we do up to Fock sector $2n = 80$. We further derive analytical results about the second moment. These results allow us to pinpoint the transition in anticoncentration and furthermore yield the expected linear cross-entropy benchmarking score for an ideal (error-free) device.

DOI: 10.1103/PhysRevA.111.042412

I. INTRODUCTION

One of the major goals of quantum computer science is to find examples of certain tasks on which quantum devices can outperform classical computers. While the ultimate goal is to develop quantum computers that can run, say, Shor's algorithm [1], the qubit numbers, gate fidelities, and error correction needed to accomplish such a task fault-tolerantly are well beyond the current state of the art. Therefore, there is interest in finding near-term examples of quantum advantage.

One area of focus that has strong theoretical evidence for an exponential speedup over the best possible classical algorithms comprises the so-called sampling problems. Aaronson and Arkhipov introduced one such promising framework called boson sampling [2]. The boson sampling task is to produce a sample (that is, a valid output Fock state) according to the outcome distribution generated by measuring indistinguishable photons that have been subjected to a random linear optical network of beam-splitters and phase shifters. In boson sampling, the input states consist of single photons on many input modes. However, because single-photon sources have imperfect efficiency, these states are difficult to produce experimentally, requiring an exponential amount of postselection [3]. Therefore, generalizing this framework to other inputs that are more reliably produced has been an important topic of study.

Gaussian boson sampling represents one such popular generalization. There, the input states are quadratic, meaning they are generated from the vacuum by some combination of displacement and squeezing (assuming pure input states that have no thermal contribution) [4]. Typically, the displacements are ignored because they do not contribute to entanglement between the modes. Hence, the input states are simply squeezed vacuum states, which are much easier to prepare in a laboratory than many parallel single-photon states [3]. Much theoretical work has been done to generalize the original statements from Ref. [2] about the computational complexity of sampling in the Fock basis to this Gaussian setting [5–11]. In due course, many labs have performed experiments claiming to show quantum advantage using Gaussian boson sampling [12–15].

Broadly speaking, the hardness of sampling schemes in general, and therefore of both Fock state and Gaussian boson sampling, is based on certain statistical properties of the output probability distributions. Fock state boson sampling and Gaussian boson sampling have output probabilities defined by permanents and hafnians, respectively, which are combinatorial functions mapping matrices over a field to an element of that field. If one treats the input matrix as a weighted adjacency matrix, then the permanent and the hafnian count the number of perfect matchings in the bipartite and generalized weighted graph, respectively, defined by this adjacency matrix [16]. These functions are, in general, difficult to compute. The permanent is #P-hard to compute exactly [17], and this hardness extends to the hafnian because one can encode the permanent of a matrix as the hafnian of a matrix that is twice as big. Even further, Ref. [2] extended this exact hardness to a proof that it is GapP-hard to approximate the modulus squared of the permanent up to inverse polynomial multiplicative error (which similarly extends to the hafnian). However, showing that it is hard to compute or approximate specific output probabilities is not, in and of itself, enough to demonstrate hardness of actually producing a sample from the Fock or Gaussian boson sampling distributions; many theoretical tools are needed to show that a difficulty in computing probabilities further implies a difficulty in sampling.

One such crucial tool is called anticoncentration. Anticoncentration is a property of the output distribution that says, roughly, that the outputs are not too clustered on individual probabilities, hence making it more difficult to adequately mimic this distribution in a sampling procedure, and it is commonly used as evidence for approximate average-case hardness of sampling [3]. Anticoncentration is usually proven by analyzing the moments of the outcome probability distribution. In a companion piece to this work, Ref. [18], we study anticoncentration in the photon-collision-free limit (where the outcome states are very likely to have at most a single photon in each mode). We develop a graph-theoretic technique to find a closed form for the first moment. Using this result and a few simple analytical results about the second moment (most saliently, that it admits a polynomial expansion in the number of initially squeezed modes and the leading-order coefficient of this expansion), that work shows that there is actually a transition in whether or not anticoncentration holds based on how many of the initial modes are squeezed; when few are squeezed, there is a lack of anticoncentration, but, in the opposite limit, a weak version of anticoncentration holds.

However, the second moment itself deserves a more thorough treatment beyond the few analytic results needed to prove this transition in anticoncentration. For example, linear cross-entropy benchmarking (LXEB) is a tool that has been used to characterize the performance of sampling experiments, most notably in the random circuit sampling experiment of Ref. [19]. It can be shown that the LXEB score that an error-free sampler would achieve when averaged over all possible random networks is precisely given by the second moment of the output probabilities normalized by the square of the first moment. Therefore, a better understanding of the second moment is crucial to achieving a better understanding this popular benchmarking scheme.

To that end, we develop a classically efficient recursion relation that allows us to exactly calculate the second moment up to any desired Fock sector n. This, along with proofs of the analytical results necessary to derive the transition in anticoncentration, is the main technical contribution of this work. The recursion relation follows from the graph-theoretic approach we introduce in Ref. [18], which we generalize and expand upon here. This approach reduces the algebraic evaluation of the hafnian to simply counting the number of connected components of a certain class of graphs. We then carefully study how higher-order graphs reduce to lower-order ones under certain operations, and the effect that this has on the number of connected components, in order to recursively solve for the second moment. Not only does this allow us to make statements about the average LXEB score for an error-free sampler, but it also allows us to pin down more precisely *where* the aforementioned transition in anticoncentration occurs. If k is the number of initially squeezed modes, we provide strong evidence that this transition occurs at $k = \Theta(n^2)$.

The rest of the paper proceeds as follows: In Sec. II, we provide some background information, set up the system and problem of interest, and briefly summarize our main results. In Sec. III, we discuss the graph-theoretic framework for our calculations. Specifically, in Sec. III A, we review results about the first moment from Ref. [18]; in Sec. III B, we discuss how to generalize this framework to the second moment. This latter section sets up the discussion of the recursion in Sec. IV (though most of the technical details are deferred to the Appendix). Section V discusses analytical results and scaling properties of the second moment. Finally, in Sec. VI, we combine these analytical results with a more detailed numerical investigation to give evidence for the exact location of the transition in anticoncentration we derive in Ref. [18].

II. THE OUTPUT DISTRIBUTION OF GAUSSIAN BOSON SAMPLING

In this section, we provide some necessary background information on Gaussian boson sampling and set up our system of interest. We also motivate the study of the moments of the output probabilities. Finally, we provide a brief summary of our main results.

A. Gaussian boson sampling

We consider a paradigmatic Gaussian boson sampling system on m modes [7,8]. These modes pass through a random sequence of beam splitters and phase shifters that effect a linear optical (i.e., photon-number-conserving Gaussian) unitary $U \in \mathrm{U}(m)$ and are then measured in the Fock basis (this non-Gaussian operation is necessary for classical hardness of sampling [10]). We consider the typical case where the initial state on the first k modes consists of single-mode squeezed states of equal squeezing parameter r, and the remaining $m-k$ modes are initialized to the vacuum state.

Reference [7] calculates the outcome probability of the Fock measurement of such a system. Given a unitary U, the probability of obtaining an outcome $\mathbf{n} = (n_1, n_2, \ldots, n_m) \in \mathbb{N}_0^m$ with total photon count $2n = \sum_{i=1}^m n_i$ is given by

$$P_U(\mathbf{n}) = \frac{\tanh^{2n} r}{\cosh^k r} \left|\mathrm{Haf}\left(U_{1_k,\mathbf{n}}^\top U_{1_k,\mathbf{n}}\right)\right|^2. \tag{1}$$

$U_{1_k,\mathbf{n}}$ is the $k \times 2n$ submatrix of U corresponding to its first k rows and its columns determined by the nonzero elements of $\mathbf{n}$ (appropriately repeated n_i times). Haf refers to the hafnian, which, for a $2n \times 2n$ symmetric matrix A, is

$$\mathrm{Haf}(A) = \frac{1}{n!2^n} \sum_{\sigma \in S_{2n}} \prod_{j=1}^{n} A_{\sigma(2j-1),\sigma(2j)}, \tag{2}$$

with S_{2n} the permutation group on $2n$ elements. We specify that the dimensions of A are even because the hafnian of an odd matrix vanishes; it also vanishes if the input matrix is not symmetric. In our setting, this aligns with the physical fact that single-mode squeezed vacuum states are supported only on even Fock states. The hafnian generalizes the permanent (whose computational complexity controls the hardness of Fock state boson sampling) because one can prove that [7]

$$\mathrm{Per}(A) = \mathrm{Haf}\left[\begin{pmatrix} 0 & A \\ A^\top & 0 \end{pmatrix}\right]. \tag{3}$$

Hence, computing the hafnian is at least as hard as computing the permanent.

We work in the regime where the measured output states are, with high probability, photon-collision-free, which means

that the output vector $\mathbf{n}$ has $n_i \in \{0, 1\}$. That is, $U_{1_k,\mathbf{n}}$ has no repeated columns. It suffices for $\mathbb{E}[2n] = k \sinh^2 r = o(\sqrt{m})$ for photon-collision-freeness to hold with high probability [20]. When $n = o(\sqrt{m})$, Ref. [11] provides strong numerical and theoretical evidence that the distribution of submatrices $U_{1_k,\mathbf{n}}$ is well-captured by a generalization of the circular orthogonal ensemble (COE):

Conjecture 1 (hiding [11]). For any k such that $1 \leqslant k \leqslant m$ and $2n = o(\sqrt{m})$, the distribution of the symmetric product $U_{1_k,\mathbf{n}}^\top U_{1_k,\mathbf{n}}$ of submatrices of a Haar-random $U \in \mathrm{U}(m)$ closely approximates in total variation distance the distribution of the symmetric product $X^\top X$ of a complex Gaussian matrix $X \sim \mathcal{N}(0, 1/m)_c^{k\times 2n}$ with mean zero and variance $1/m$.

We note that, in Ref. [11], this conjecture is only formulated for the case $n \leqslant k \leqslant m$. However, here we allow k to reach 1. The reasoning is that the evidence for Conjecture 1 in the regime $k = n$ is based on a proof from Ref. [2] that $n \times n$ submatrices of Haar-random unitaries are approximately Gaussian. Clearly the proof still holds in the case $k < n$ (if $n \times n$ submatrices are approximately Gaussian, then so are smaller submatrices), meaning we can safely extend the conjecture to all $k \leqslant m$.

Roughly speaking, the intuition behind the conjecture and the original proof of the $k = n$ regime in Ref. [2] is that, if one looks at a small enough submatrix of a unitary, this submatrix no longer "notices" the unitary constraints (i.e., the complex orthonormality of rows and columns). Multiplying this small submatrix by its transpose washes out the remaining correlations between elements of the unitary. Hence, the product of the submatrices is approximately the same as a product of i.i.d. Gaussian matrices. Observe also that working in the photon-collision-free regime, $n \in o(\sqrt{m})$, is crucial for this argument to hold; an output state with more than one photon in a given mode leads to a repeated column or row in the respective submatrix, which destroys the independence of these elements. In what follows, we work under the assumption that Conjecture 1 holds. We are therefore interested in the statistical properties of $X^\top X$ when the elements of X are i.i.d. Gaussian.

B. Moments of the Gaussian boson sampling distribution and their significance

To understand the statistical properties of the outcome probabilities of Gaussian boson sampling, we must study not just the distribution over individual matrix elements of $X^\top X$, but how they interact with one another through the hafnian. Under Conjecture 1, the outcome probabilities of Gaussian boson sampling given in Eq. (1) are well-approximated by

$$P_X(\mathbf{n}) = \frac{\tanh^{2n} r}{\cosh^k r} |\mathrm{Haf}(X^\top X)|^2, \tag{4}$$

where $X \sim \mathcal{N}(0, 1/m)_c^{k\times 2n}$ is a random Gaussian matrix. Note that, while the left-hand side (LHS) of Eq. (4) contains a specific photon output $\mathbf{n}$ as an argument, the right-hand side (RHS) is $\mathbf{n}$-independent; this is precisely how Conjecture 1 captures the hiding property in Gaussian boson sampling.

We are interested in moments of the output probability distributions. To capture this interest, we define the moments

$$M_t(k, n) \coloneqq \mathbb{E}_{X\sim\mathcal{G}^{k\times 2n}}[|\mathrm{Haf}(X^\top X)|^{2t}], \tag{5}$$

which are equivalent to the moments of $P_X(\mathbf{n})$ up to prefactors [$\mathcal{G}^{k\times 2n}$ is shorthand for $\mathcal{N}(0, 1)_c^{k\times 2n}$; we consider unit variance in the definition of M_t for computational simplicity, as rescaling X by $1/\sqrt{m}$ leads to an overall prefactor that can be dealt with independently, similarly to the factor $\tanh^{2n} r/\cosh^k r$ that we ignore in M_t]. Specifically, we are most interested in the first and second moments, $t = 1$ and $t = 2$, respectively. We motivate this interest in two ways: the study of anticoncentration and linear cross entropy benchmarking.

We first discuss a useful framework for anticoncentration. The key definition is p_2, the inverse normalized average outcome-collision probability. This is a rather long name, so it is worth slowly describing what each piece means. The outcome-collision probability refers to the probability that two independent trials of a Gaussian boson sampling experiment with the same unitary transformation matrix produce the same outcome, which is just the second moment of the output distribution. Note that this is a different notion of collision than the photon-collision probability described previously (the type of collision in question should be clear from context, but we also try to specify one of photon- and outcome-collision). The average is taken over all possible unitary transformation matrices (weighted equally, meaning the average is over the so-called Haar measure). We normalize this quantity by dividing by the square of the first moment. This ensures that the uniform distribution returns a value of one, but a narrow distribution returns a large value (a distribution peaked on a single value would return the size of the outcome sample space). To finally define p_2 itself, we invert this quantity because this inverted value is what shows up in the current state-of-the-art hardness arguments (though the raw quantity before inverting shows up in the study of linear cross-entropy benchmarking, as we see below). Note that this means that the uniform distribution has $p_2 = 1$, but a highly peaked distribution now returns a very small value of p_2.

Under the hiding conjecture (Conjecture 1), p_2 is approximately given by the ratio of the square of the first moment to the second moment:

$$p_2(\mathrm{U}(m)) = \frac{\mathbb{E}_{U\in\mathrm{U}(m)}[P_U(\mathbf{n})]^2}{\mathbb{E}_{U\in\mathrm{U}(m)}[P_U(\mathbf{n})^2]} \approx \frac{M_1(k,n)^2}{M_2(k,n)} =: m_2(k,n). \tag{6}$$

We refer to $m_2(k, n)$ as the inverse normalized second moment. We may use p_2 to define three different classes of anticoncentration:

(A) We say that P_U *anticoncentrates* if $p_2 = \Omega(1)$.

(WA) We say that P_U *anticoncentrates weakly* if $p_2 = \Omega(1/n^a)$ for some $a = O(1)$.

(NA) We say that P_U *does not anticoncentrate* if $p_2 = O(1/n^a)$ for any constant $a > 0$.

In Appendix A, we explain more thoroughly where p_2 arises in the argument for approximate average-case hardness of Gaussian boson sampling (one can also consult Ref. [3], Sec. IV D 2 for details) and further contextualize our specific definitions in complexity-theoretic terms.

We note also that, of course, it is important how precise the approximation in Eq. (6) is. That is, fully formalizing the complexity-theoretic implications of our work depends on exactly how close in total variation distance the exact and approximate output distributions are. In particular, if the distribution $U_{1_k,\mathbf{n}}^{\top}U_{1_k,\mathbf{n}}$ is not close enough in total variation distance to the distribution $X^{\top}X$, then it is not possible to transfer statements about the normalized second moment m_2 to statements about anticoncentration via p_2. We address this subtlety also in Appendix A, but, in short, we can formalize and sharpen Conjecture 1 such that statements made about the approximate distribution via m_2 imply anticoncentration of the exact distribution via p_2 as well.

Beyond understanding anticoncentration, calculations of $M_1(k,n)$ and $M_2(k,n)$ also allow one to study linear cross-entropy benchmarking in Gaussian boson sampling. Recall that linear cross-entropy benchmarking is a method by which one can compare the outputs of a potentially noisy Gaussian boson sampling experiment with the output of a perfect, error-free experiment. Cross-entropy benchmarking was introduced in the context of random circuit sampling in Refs. [21,22] and later linearized in Ref. [19]. We review this linearized form now, translating from the random circuit sampling language to that of bosonic sampling.

Let $\{\mathbf{n}\}$ be the possible output photon strings sampled in some Gaussian boson sampling experiment that are produced with respective experimental probabilities $\tilde{P}_U(\mathbf{n})$. Let $P_U(\mathbf{n})$ be the ideal probabilities for these outputs; that is, these are the probabilities for an output $\mathbf{n}$ given by Eq. (1). The linear cross-entropy score F_{XEB} for such an experiment is

$$F_{\mathrm{XEB}} = |\Omega_{2n}| \sum_{\mathbf{n}\in\Omega_{2n}} P_U(\mathbf{n})\tilde{P}_U(\mathbf{n}) - 1, \tag{7}$$

where Ω_{2n} is the photon-collision-free sample space with $2n$ output photons in m modes such that $|\Omega_{2n}| = \binom{m}{2n}$. This is the dominant space of outputs assuming that we postselect on outcomes with $2n$ photons and that hiding holds (see Appendix B). If the noisy outputs are correct, i.e., the experiment is error-free, then $\tilde{P}(\mathbf{n}_i) = P(\mathbf{n})$. The ideal cross-entropy score, then, is

$$F_{\mathrm{XEB}}^{\mathrm{ideal}} = |\Omega_{2n}| \sum_{\mathbf{n}\in\Omega_{2n}} P_U(\mathbf{n})^2 - 1. \tag{8}$$

The expected value of the ideal cross-entropy over all possible unitaries is, therefore,

$$\mathbb{E}_{U\in U(m)}\left[F_{\mathrm{XEB}}^{\mathrm{ideal}}\right] = |\Omega_{2n}| \sum_{\mathbf{n}\in\Omega_{2n}} \mathbb{E}_{U\in U(m)}[P_U(\mathbf{n})^2] - 1. \tag{9}$$

Assuming that one operates in the hiding regime, then two facts are true: first, $|\Omega_{2n}| \sim M_1(k,n)$; second, $\mathbb{E}_{U\in U(m)}[P_U(\mathbf{n})^2]$ is independent of $\mathbf{n}$ (see Appendix B for more details). Therefore,

$$\mathbb{E}_{U\in U(m)}\left[F_{\mathrm{XEB}}^{\mathrm{ideal}}\right] \approx \frac{M_2(k,n)}{M_1^2(k,n)} - 1 = m_2(k,n)^{-1} - 1. \tag{10}$$

Thus, anticoncentration and the expected ideal linear cross-entropy benchmarking score both depend on m_2, thereby warranting a more fine-grained study of the second moment beyond asymptotics.

C. Summary of results

We now come to a brief summary of our main results. In Ref. [18], we develop a graph-theoretic formalism that allows us to derive a closed form for the first moment $M_1(k,n)$, which is a key result proving the transition in anticoncentration. We review this framework in Sec. III A. Then, in Sec. III B (with some details deferred to Appendix C), we show how to expand this graph-theoretic framework to study the second moment both numerically and analytically.

We derive an efficiently evaluable recursion relation that allows us to numerically exactly calculate all coefficients of the polynomial expansion of the second moment. We perform this up to Fock sector $2n = 80$. In the photon-collision-free regime, where $n \in o(\sqrt{m})$, this corresponds to approximately 6400 modes, which is well beyond the current state-of-the-art experiments. Therefore, the technique that we develop in this work yields results that can help characterize the output distribution of any near-term Gaussian boson sampling experiment. The recursion is developed in Sec. IV, with details about its efficiency and construction deferred to Appendices E and D, respectively.

We then discuss some simple analytic results about the scaling of the second moment in Sec. IV C. We follow this with substantial numerical investigation of the results of the recursion up to $2n = 80$ in Sec. VI. In particular, we are able to give strong evidence that the transition in anticoncentration occurs at $k = \Theta(n^2)$. We accomplish this with numerical plots of $m_2(k,n)$, the quantity that controls anticoncentration, when k scales polynomially with n. We also provide a brief analytic argument that this transition occurs somewhere between $k = \Omega(n)$ and $k = O(n^2)$.

This result, along with the fact that we operate in the conjectured hiding regime where $2n = o(\sqrt{m})$ and $k \leqslant m$, implies concrete advice for experimental demonstrations of quantum advantage via Gaussian boson sampling. Namely, one should squeeze all m modes with squeezing parameter $\sinh^2 r = o(m^{-1/2})$.

III. GRAPH-THEORETICAL ANALYSIS OF GAUSSIAN BOSON SAMPLING MOMENTS

In this section, we lay out the graph-theoretic framework for analyzing the moments of Gaussian boson sampling output probabilities. This is a review of the same framework we develop in Ref. [18]. We first briefly recall the derivation of the closed form of the first moment $M_1(k,n)$, and we follow this with a discussion of how an extension of this framework also allows us to analyze the second moment $M_2(k,n)$.

A. First moment

In this section, we discuss the first moment of the output probabilities, which is, up to some multiplicative factors, $\mathbb{E}_{X\sim\mathcal{G}^{k\times 2n}}[|\mathrm{Haf}(X^{\top}X)|^2]$. We calculate and analyze this moment in Ref. [18], but we review the key elements of that discussion because they are a useful point of reference for the calculation of the second moment.

Using the definition of the hafnian in Eq. (2) and properties of the expectation value of complex Gaussians—$\mathbb{E}_{X\sim\mathcal{G}^{k\times 2n}}[X_{ij}X_{uv}^{*}] = \delta_{iu}\delta_{jv}$ and $\mathbb{E}_{X\sim\mathcal{G}^{k\times 2n}}[X_{ij}X_{uv}] = 0 =$

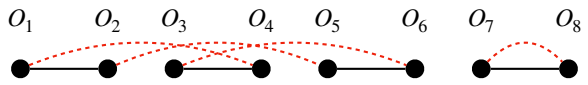


FIG. 1. Graph $G \in \mathbb{G}_n^1$. One of $2^n n!$ permutations that induces this graph is $\tau = \left(\begin{smallmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 3 & 5 & 2 & 4 & 6 & 8 & 7 \end{smallmatrix}\right)$. This graph has two connected components, therefore contributing k^2 to the first moment.

$\mathbb{E}_{X\sim\mathcal{G}^{k\times 2n}}[X^*_{ij}X^*_{uv}]$—we reduce the first moment to a sum over products of Kronecker δs:

$$M_1(k,n) = \frac{(2n)!}{(2^n n!)^2} \sum_{\tau\in S_{2n}} \sum_{\{o_i\}_{i=1}^n}^{k} \prod_{j=1}^{n} \delta_{o_{\lceil \frac{\tau(2j-1)}{2}\rceil} o_{\lceil \frac{\tau(2j)}{2}\rceil}}. \tag{11}$$

We ascribe a graph-theoretic interpretation to this equation; see Fig. 1 for an example. Each permutation τ instantiates a graph G_τ on $2n$ vertices labeled O_1 to O_{2n} with edges defined by two perfect matchings: one fixed black set of edges, and one set of red edges determined by τ. More specifically, each index o_j in the sum splits into two vertices O_ℓ and $O_{\ell'}$ such that $\lceil\tau(\ell)/2\rceil = j = \lceil\tau(\ell')/2\rceil$ (that is, $o_{\lceil\tau(\ell)/2\rceil}$ maps to a vertex O_ℓ). One perfect matching consists of black edges between O_{2j-1} and O_{2j} for all $j \in [n] := \{1, 2, \ldots, n\}$; these edges enforce that $o_{\lceil\tau(2j-1)/2\rceil}$ and $o_{\lceil\tau(2j)/2\rceil}$ are linked by a Kronecker δ. The other perfect matching has red edges between O_ℓ and $O_{\ell'}$ if $\lceil\tau(\ell)/2\rceil = \lceil\tau(\ell')/2\rceil$; these edges ensure that there is an edge between the ℓ, ℓ' mapped to the same value under τ and the ceiling function, meaning the vertices arose from the same lower-case-o index.

This definition of G_τ ensures that the number of connected components of G_τ, $C(G_\tau)$, is equivalent to the number of unconstrained indices in the interior sum in Eq. (11), and, hence, the number of factors of k that τ contributes overall. Therefore,

$$M_1(k,n) = \frac{(2n)!}{(2^n n!)^2} \sum_{\tau\in S_{2n}} k^{C(G_\tau)} \tag{12}$$

We simplify this expression using a degeneracy whereby $2^n n!$ different τ all induce the same final graph; the factor of $n!$ corresponds to choosing which tuple $(2j-1, 2j)$ corresponds to which index $\lceil\tau(\ell)/2\rceil = \lceil\tau(\ell')/2\rceil$, and the factor of 2^n comes from ordering within each tuple. Therefore, we study only these final sets of graphs, which we label $\mathbb{G}_n^1$ (1 refers to the first moment, and n indexes the order). We study the connected components of graphs in $\mathbb{G}_n^1$ by writing down a recursion relation in n and k that, when solved, yields the first theorem of Ref. [18]:

Theorem 1 (Ref. [18]). The sum over graphs in $\mathbb{G}_n^1$ satisfies

$$\sum_{G\in\mathbb{G}_n^1} k^{C(G)} = k(k+2)\cdots(k+2n-2), \tag{13}$$

and hence $M_1(k,n) = (2n-1)!!(k+2n-2)!!/(k-2)!!$.

To summarize: Eq. (11) gives an expression for the first moment of the outcomes of Gaussian boson sampling probabilities in terms of sums of products of Kronecker δs. We then reinterpret this as counting the number of connected components of a certain type of graph with two perfect matchings. We solve this counting problem by developing and evaluating a recursion relation. We use the same overall technique to calculate the second moment, as we explain in the next section.

B. Second moment

We now move on to analyzing the second moment of the output probabilities. Using similar techniques as described for the first moment (namely, expanding the definition of the hafnian and using the aforementioned properties of expectations of Gaussians) we derive in Appendix C an expression for the second moment that is equivalent to Eq. (11):

$$\begin{aligned} M_2(k,n) &:= \mathop{\mathbb{E}}_{X\sim\mathcal{G}^{k\times 2n}} [|\mathrm{Haf}(X^\top X)|^4] \\ &= \left(\frac{1}{2^n n!}\right)^4 (2n)! \sum_{\tau,\alpha,\beta\in S_{2n}} \sum_{\{\ell_i,o_i,p_i\}_{i=1}^n}^{k} \Bigg[\prod_{j=1}^{n} \Big(\delta_{o_{\lceil \frac{\tau(2j-1)}{2}\rceil} o_{\lceil \frac{\tau(2j)}{2}\rceil}} \delta_{p_{\lceil \frac{\alpha(2j-1)}{2}\rceil} q_{\lceil \frac{\beta(2j-1)}{2}\rceil}} \delta_{p_{\lceil \frac{\alpha(2j)}{2}\rceil} q_{\lceil \frac{\beta(2j)}{2}\rceil}} \\ &\quad + \delta_{o_{\lceil \frac{\tau(2j-1)}{2}\rceil} q_{\lceil \frac{\beta(2j)}{2}\rceil}} \delta_{p_{\lceil \frac{\alpha(2j-1)}{2}\rceil} q_{\lceil \frac{\beta(2j-1)}{2}\rceil}} \delta_{p_{\lceil \frac{\alpha(2j)}{2}\rceil} o_{\lceil \frac{\tau(2j)}{2}\rceil}} + \delta_{q_{\lceil \frac{\beta(2j-1)}{2}\rceil} o_{\lceil \frac{\tau(2j)}{2}\rceil}} \delta_{p_{\lceil \frac{\alpha(2j-1)}{2}\rceil} o_{\lceil \frac{\tau(2j-1)}{2}\rceil}} \delta_{p_{\lceil \frac{\alpha(2j)}{2}\rceil} q_{\lceil \frac{\beta(2j)}{2}\rceil}} \\ &\quad + \delta_{q_{\lceil \frac{\beta(2j-1)}{2}\rceil} q_{\lceil \frac{\beta(2j)}{2}\rceil}} \delta_{p_{\lceil \frac{\alpha(2j-1)}{2}\rceil} o_{\lceil \frac{\tau(2j-1)}{2}\rceil}} \delta_{p_{\lceil \frac{\alpha(2j)}{2}\rceil} o_{\lceil \frac{\tau(2j)}{2}\rceil}} \Big) \Bigg]. \end{aligned} \tag{14}$$

The main differences between Eqs. (14) and (11) are threefold: (1) We sum over three permutations (instead of a single one) labeled τ, α, β. (2) There are now $3n$ indices to sum over, $\{o_i, q_i, p_i\}_{i=1}^n$, instead of just the n given by $\{o_i\}_{i=1}^n$. (3) Each factor is a sum of four possible terms instead of just one. However, this expression still possesses a natural graph-theoretic interpretation, as we now review. See Fig. 2 for an example graph as a guide to the following discussion

Each index in $\{o_i, q_i, p_i\}_{i=1}^n$ is again split into two graph vertices $\{O_i, Q_i, P_i\}_{i=1}^{2n}$ that are placed into $2n$ columns and three rows labeled o, p, and q, respectively. As for the first moment, we define two perfect matchings on these vertices given by black and red edges. The black edges are between vertices whose labels are linked under the Kronecker δs, and the red edges connect graph vertices that came from the same original summation index.

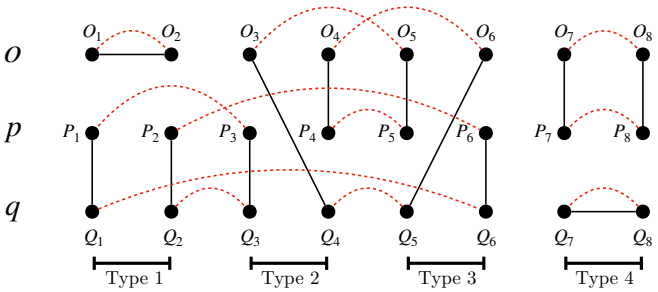


FIG. 2. Example graph on $n = 4$ used in the calculation of the second moment. Each of the four possible sets of black edges are shown. An example of three permutations that would induce this graph is $\tau = \left(\begin{smallmatrix}1&2&3&4&5&6&7&8\\1&2&4&5&3&6&8&7\end{smallmatrix}\right)$, $\alpha = \left(\begin{smallmatrix}1&2&3&4&5&6&7&8\\8&6&7&3&4&5&1&2\end{smallmatrix}\right)$, and $\beta = \left(\begin{smallmatrix}1&2&3&4&5&6&7&8\\8&5&6&2&1&7&3&4\end{smallmatrix}\right)$. This graph has five connected components, so it contributes k^5 to the second moment.

More specifically, consider fixing a set of three permutations τ, α, β. There is a red edge between O_ℓ and $O_{\ell'}$ if $\lceil \tau(\ell)/2 \rceil = \lceil \tau(\ell')/2 \rceil$. An analogous statement holds for P and Q vertices, although one uses permutations α and β, respectively, instead of τ. Note that this implies that red edges are always contained within a single row. Now, the black edges are slightly more complicated. There is only a single Kronecker δ term in each factor in the product Eq. (11), meaning there is only a single set of black edges for the graphs in $\mathbb{G}_n^1$. However, because the second moment as expressed in Eq. (14) contains factors with four Kronecker δ terms, each value of $j \in [n]$ can lead to one of four different patterns of black edges on columns $2j-1$ and $2j$. We refer to these patterns of black edges on a single pair of adjacent columns as type-1, type-2, type-3, and type-4; see Fig. 2 for an example graph that has one of each type. The Kronecker δ terms and their corresponding black edges, listed in order from type-1 to type-4, are given by

$$\delta_{o_{\lceil \frac{\tau(2j-1)}{2} \rceil} o_{\lceil \frac{\tau(2j)}{2} \rceil}} \delta_{p_{\lceil \frac{\alpha(2j-1)}{2} \rceil} q_{\lceil \frac{\beta(2j-1)}{2} \rceil}} \delta_{p_{\lceil \frac{\alpha(2j)}{2} \rceil} q_{\lceil \frac{\beta(2j)}{2} \rceil}} \to \{(O_{2j-1}, O_{2j}), (P_{2j-1}, Q_{2j-1}), (P_{2j}, Q_{2j})\}, \tag{15}$$

$$\delta_{o_{\lceil \frac{\tau(2j-1)}{2} \rceil} q_{\lceil \frac{\beta(2j)}{2} \rceil}} \delta_{p_{\lceil \frac{\alpha(2j-1)}{2} \rceil} q_{\lceil \frac{\beta(2j-1)}{2} \rceil}} \delta_{p_{\lceil \frac{\alpha(2j)}{2} \rceil} o_{\lceil \frac{\tau(2j)}{2} \rceil}} \to \{(O_{2j-1}, Q_{2j}), (P_{2j-1}, Q_{2j-1}), (O_{2j}, P_{2j})\}, \tag{16}$$

$$\delta_{q_{\lceil \frac{\beta(2j-1)}{2} \rceil} o_{\lceil \frac{\tau(2j)}{2} \rceil}} \delta_{p_{\lceil \frac{\alpha(2j-1)}{2} \rceil} o_{\lceil \frac{\tau(2j-1)}{2} \rceil}} \delta_{p_{\lceil \frac{\alpha(2j)}{2} \rceil} q_{\lceil \frac{\beta(2j)}{2} \rceil}} \to \{(O_{2j}, Q_{2j-1}), (P_{2j-1}, O_{2j-1}), (P_{2j}, Q_{2j})\}, \tag{17}$$

$$\delta_{q_{\lceil \frac{\beta(2j-1)}{2} \rceil} q_{\lceil \frac{\beta(2j)}{2} \rceil}} \delta_{p_{\lceil \frac{\alpha(2j-1)}{2} \rceil} o_{\lceil \frac{\tau(2j-1)}{2} \rceil}} \delta_{p_{\lceil \frac{\alpha(2j)}{2} \rceil} o_{\lceil \frac{\tau(2j)}{2} \rceil}} \to \{(O_{2j-1}, P_{2j-1}), (O_{2j}, P_{2j}), (Q_{2j-1}, Q_{2j})\} \tag{18}$$

Because there are four patterns of black edges per pair of adjacent columns, and n such pairs, there are 4^n possible arrangements of black edges on the entire graph. We label these arrangements by an integer $z \in [4^n]$, and we label a graph as $G_{\tau,\alpha,\beta}(z)$.

Analogously to the first moment, we can rewrite the sum over products of Kronecker δs in Eq. (14) as a sum over these graphs, where $G_{\tau,\alpha,\beta}(z)$ contributes a factor of k raised to its number of connected components. Therefore, Eq. (14) becomes

$$M_2(k,n) = \frac{(2n)!}{(2^n n!)^4} \sum_{\substack{\tau,\alpha,\beta \in S_{2n} \\ z \in [4^n]}} k^{C(G_{\tau,\alpha,\beta}(z))} \tag{19}$$

There is again a degeneracy where many permutations all lead to the same set of red edges in a given row, and, hence, the same graph. Specifically, this degeneracy is again $2^n n!$, but for each copy of S_{2n}. We can therefore again ignore the permutations and look only at the underlying graphs. For any given z, we define $\mathbb{G}_n^2(z)$ to be the graphs on $6n$ vertices with two perfect matchings: the zth set of black edges and red edges that pair vertices in the same row. We then define $\mathbb{G}_n^2 = \cup_{z \in [4^n]} \mathbb{G}_n^2(z)$. Thus, accounting for the described degeneracy and these definitions, we get

$$M_2(k,n) = (2n-1)!! \sum_{G \in \mathbb{G}_n^2} k^{C(G)} \tag{20}$$

This implies the following theorem:

Theorem 2 The second moment $M_2(k,n)$ is a degree-$2n$ polynomial in k and can be written as $M_2(k,n) = (2n-1)!! \sum_{i=1}^{2n} c_i k^i$, where c_i is the number of graphs $G \in \mathbb{G}_n^2$ that have i connected components.

Proof. Once Eq. (20) is derived, the theorem follows after deriving the correct limits of summation. Trivially, the fewest possible number of connected components is one. To see that the largest possible number of connected components is $2n$, we consider the four patterns of black edges that are illustrated in Fig. 2 and how many connected components can possibly occur in graphs with those different patterns. See also Fig. 3 for a visual explanation of the following argument

First note that, because all vertices are paired via black edges, every connected component has an even number of vertices. Therefore, the two smallest sizes of connected components are 2 and 4 vertices. To get a connected component of size 2, one must connect a pair of vertices with both a

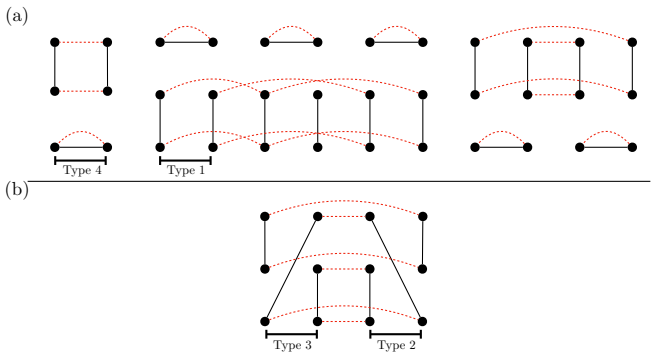


FIG. 3. (a) Example graph in $\mathbb{G}_6^2$ showing how to achieve an average of two connected components per set of six vertices using only type-1 and type-4 sets of edges. All vertices connected by horizontal black (solid) edges are also connected by red (dashed) edges. All type-1 vertical edges are paired off, as are type-4 vertical edges. Note that this graph would correspond to $z = 1 + 3 \times 4^5 + 0 \times 4^4 + 0 \times 4^3 + 0 \times 4^2 + 3 \times 4^1 + 3 \times 4^0 = 3088$. (b) Example showing how using type-2 and type-3 black edges lead to, at most, three connected components per two sets of six vertices.

black and a red edge. Red edges are constrained to lie in a single row, meaning only type-1 and type-4 patterns of black edges, which contain a pair of vertices connected by a black edge in the same row, can yield a connect component of size 2. Pairing off the remaining vertical black edges yields connected components of size 4, the next smallest size.

Therefore, the maximum number of connected components arises from taking only type-1 and type-4 edges. This requires connecting each horizontal black edge by red edge (creating a connected component with two vertices) and then pairing off the vertical edges coming from the same type. This allows for the maximal two connected components per set of six vertices, meaning $2n$ total connected components. ■

Our goal, then, is to determine these coefficients c_i. It is possible to directly compute some individual coefficients, which we discuss in later sections, but these direct techniques do not easily generalize to a way of computing all coefficients. Therefore, we take a more indirect approach, which is to derive a recursion relation that is similar in spirit to the one we use to compute the first moment.

IV. RECURSION FOR THE SECOND MOMENT

In this section, we discuss the development and mechanics of the recursion in more depth, with some details deferred to the Appendices. First, in Sec. IV A, we generalize the graphs that we have considered up to this point. While these generalized graphs do not directly connect to any Gaussian boson sampling calculation (that we know of), they provide a useful intermediate framework for eventually evaluating Eq. (20). Then, in Sec. IV B, we derive the recursion, making strong use of these generalized graphs. Finally, in Sec. IV C, we provide some useful details behind the numerical evaluation of the recursion.

A. Generalized graphs

It is useful to generalize the graphs that we have considered to this point. In Eq. (20), we write the second moment M_2 as a sum over graphs $G \in \mathbb{G}_n^2$, where the red edges of G may not cross between different rows (see Fig. 2 for a reminder). We can, however, define natural expanded sets of graphs that allow for *all* possible perfect matchings of red edges across the $6n$ vertices (with the black edges remaining the same). That is, we can define graphs where red edges may cross between rows, but we still demand that each vertex possesses exactly one incident red edge. While weighted sums over the connected components of these generalized graphs do not directly correspond to any relevant statistical calculation in Gaussian boson sampling, they provide a very useful intermediate calculation that allows for the recursive evaluation of M_2.

More mathematically, we define sets of graphs $\mathbb{G}_n^2(a_{12}, a_{13}, a_{23}, z)$ on $6n$ vertices $\{O_i, P_i, Q_i\}_{i=1}^{2n}$, where z again indexes the 4^n possible sets of black edges defined by Eq. (15)–(18), and a_{12}, a_{13}, a_{23} represent the number of red edges that span the first and second, first and third, and second and third rows, respectively, of the graph. Of course, $\mathbb{G}_n^2(0, 0, 0, z) = \mathbb{G}_n^2(z)$. Two constraints on a_{12}, a_{13}, a_{23} are apparent immediately: (1) $a_{12} + a_{13}$, $a_{12} + a_{23}$, and $a_{13} + a_{23}$ (that is, the number of edges coming out of the first, second, and third row, respectively) must be even; (2) $a_{12} + a_{13}$, $a_{12} + a_{23}$, $a_{13} + a_{23}$ must all be less than or equal to $2n$ (there cannot be more than $2n$ edges coming out of a row with only $2n$ vertices given that there is exactly one red edge incident on every vertex). We also observe that, while we do not explicitly keep track of these edges in the arguments of $\mathbb{G}_n^2(a_{12}, a_{13}, a_{23}, z)$, we can also define a_{11}, a_{22}, a_{33} as the number of "proper" red edges that map between vertices in the first, second, and third rows, respectively (we refer to these red edges as "proper" because they correspond to allowed edges in $\mathbb{G}_n^2$). These edges have a simple relationship to a_{12}, a_{13}, a_{23} that can be derived by counting how many vertices in a given row are left after subtracting those that are used in edges that cross between rows:

$$a_{11} = \frac{2n - a_{12} - a_{13}}{2}, \tag{21}$$

$$a_{22} = \frac{2n - a_{12} - a_{23}}{2}, \tag{22}$$

$$a_{33} = \frac{2n - a_{13} - a_{23}}{2} \tag{23}$$

Because we have the constraints that $a_{12} + a_{13}$, $a_{12} + a_{23}$, $a_{13} + a_{23}$ must all be even, a_{11}, a_{22}, a_{33} are all integral. Also, the fact that $a_{12} + a_{13}$, $a_{12} + a_{23}$, $a_{13} + a_{23}$ must all be less than or equal to $2n$ ensures that a_{11}, a_{22}, a_{33} are all nonnegative as well.

Similarly to how we define $\mathbb{G}_n^2$, we write $\mathbb{G}_n^2(a_{12}, a_{13}, a_{23}) := \cup_{z\in[4^n]}\mathbb{G}_n^2(a_{12}, a_{13}, a_{23}, z)$, which means that, as one might expect, $\mathbb{G}_n^2(0, 0, 0) = \mathbb{G}_n^2$. Furthermore, we also analogously define the weighted sum over connected components of graphs in each of these sets:

$$g(n, a_{12}, a_{13}, a_{23}) := \sum_{G\in\mathbb{G}_n^2(a_{12},a_{13},a_{23})} k^{C(G)} \tag{24}$$

M_2 is then proportional to $g(n, 0, 0, 0)$:

$$M_2(k, n) = (2n - 1)!!g(n, 0, 0, 0) \tag{25}$$

Note also that $g(n, a_{12}, a_{13}, a_{23})$ still admits a polynomial expansion in k, generalizing Theorem 2, where the coefficient in front of k^i represents the number of graphs $G \in \mathbb{G}_n^2(a_{12}, a_{13}, a_{23})$ with i connected components. However, the highest-order term in this expansion need not be $2n$ anymore—generically it can reach $3n$, but no higher (there are $6n$ vertices, and each connected component contains at least two vertices).

A quick counting argument provides the total number of graphs in $\mathbb{G}(a_{12}, a_{13}, a_{23})$, which we write as $|\mathbb{G}(a_{12}, a_{13}, a_{23})|$. There are $6n$ total vertices, and $2n$ in each row. Given a vector $\mathbf{a} = (a_{12}, a_{13}, a_{23})$, we need to choose a_{12} vertices in row 1 and row 2 to link to one another, a_{13} vertices in rows one and three (with no overlap between the vertices chosen in the first row corresponding to a_{12} vs a_{13}), and a_{23} vertices in rows two and three (again, no overlap with previously chosen vertices is allowed). Once these vertices are chosen, it also remains to choose how to connect them. Finally, one must pair off the remaining vertices in each row, then multiply by 4^n to account for the black edges. The

result is

$$\left|\mathbb{G}_n^2(a_{12},a_{13},a_{23})\right| = \binom{2n}{a_{12}}\binom{2n-a_{12}}{a_{13}}\binom{2n}{a_{12}}\binom{2n-a_{12}}{a_{23}}\binom{2n}{a_{13}}\binom{2n-a_{13}}{a_{23}} a_{12}!a_{13}!a_{23}! \times (2n-a_{12}-a_{13}-1)!!(2n-a_{12}-a_{23}-1)!!(2n-a_{13}-a_{23}-1)!!4^n. \tag{26}$$

Equation (26) is useful because

$$\left|\mathbb{G}_n^2(a_{12},a_{13},a_{23})\right| = g(n,a_{12},a_{13},a_{23})|_{k=1} \tag{27}$$

Thus, the sum of the coefficients of the polynomial expansion of $g(n,a_{12},a_{13},a_{23})$ must return the RHS of Eq. (26). This provides a useful way to help check whether numerically derived polynomial expansions of $g(n,a_{12},a_{13},a_{23})$ are correct.

B. Constructing the recursion

With these generalized graphs defined, we are now ready to discuss how the recursive evaluation of the second moment operates. At a high level, the recursion works by determining $g(n,a_{12},a_{13},a_{23})$ as a function of previously computed $g(n-1,b_{12},b_{13},b_{23})$, where it is possible to compute the base cases at $n=1$ entirely by hand. Mathematically, we seek coefficients $c(a_{12},a_{13},a_{23},b_{12},b_{13},b_{23})$ such that

$$g(n,a_{12},a_{13},a_{23}) = \sum_{b_{12},b_{13},b_{23}} c(a_{12},a_{13},a_{23},b_{12},b_{13},b_{23}) \times g(n-1,b_{12},b_{13},b_{23}), \tag{28}$$

where the sum over b_{12}, b_{13}, b_{23} is taken over valid combinations, i.e., those respecting the pairwise parity and sum constraints listed in the paragraph above Eq. (21) [note that, just because b_{12}, b_{13}, b_{23} are valid does not mean that, for a given triplet a_{12}, a_{13}, a_{23}, the coefficient $c(a_{12},a_{13},a_{23},b_{12},b_{13},b_{23})$ will be nonzero]. To find these coefficients, we write how many connected components there are in a graph in $\mathbb{G}_n^2(a_{12},a_{13},a_{23})$ based on how many exist in some properly chosen graph at one lower order. This is done through a "contractive" procedure, which we soon discuss in more detail, that reduces a graph at order n to one at order $n-1$ while keeping careful track of any connected components that may be eliminated in the process. By properly aggregating all of the graphs with the same number of red edges that cross between rows, we can work in terms of the sums $g(n,a_{12},a_{13},a_{23})$ rather than with individual graphs. Overall, this process is morally quite similar to the derivation of the formula for the first moment presented in the companion work Ref. [18], but it requires substantially more machinery.

To find the $c(a_{12},a_{13},a_{23},b_{12},b_{13},b_{23})$ in Eq. (28), and, hence, $g(n,a_{12},a_{13},a_{23})$, we find it necessary to first partition $\mathbb{G}_n^2(a_{12},a_{13},a_{23})$ into different subsets of graphs based on the behavior of the red edges that are incident upon the vertices in the first two columns, $\{O_1,O_2,P_1,P_2,Q_1,Q_2\}$. We refer to this set of vertices as $\mathbb{C}_{1,2}$ (occasionally, we abuse this notation and also let $\mathbb{C}_{1,2}$ refer to the edges incident on these vertices—whether $\mathbb{C}_{1,2}$ refers to just vertices or vertices and edges should be clear from context). There are 17 ways, 24 if one disambiguates symmetric cases, that red edges can connect the vertices in $\mathbb{C}_{1,2}$ to those in the other $2n-2$ columns. We illustrate these cases and describe how to interpret them in Fig. 4 With this partition in mind, we rewrite the weighted sums as

$$g(n,a_{12},a_{13},a_{23}) = \sum_{i\in\text{cases}} g(n,a_{12},a_{13},a_{23})_{\text{case}(i)}, \tag{29}$$

where

$$g(n,a_{12},a_{13},a_{23})_{\text{case}(i)} = \sum_{\substack{G\in\mathbb{G}_n^2(a_{12},a_{13},a_{23}) \\ \mathbb{C}_{1,2}(G)\equiv\text{case}(i)}} k^{C(G)}, \tag{30}$$

and $\mathbb{C}_{1,2}(G)\equiv\text{case}(i)$ means that the red edges incident on the first two columns of G match case (i) in Fig. 4. Continuing along this line of reasoning, we write

$$g(n,a_{12},a_{13},a_{23})_{\text{case}(i)} = \sum_{b_{12},b_{13},b_{23}} c(a_{12},a_{13},a_{23},b_{12},b_{13},b_{23})_{\text{case}(i)} \times g(n-1,b_{12},b_{13},b_{23}), \tag{31}$$

which implicitly defines the case-wise recursive coefficients $c(a_{12},a_{13},a_{23},b_{12},b_{13},b_{23})_{\text{case}(i)}$.

We determine these case-wise recursive coefficients through the contractive procedure to which we previously alluded. It is useful to refer to Fig. 5 throughout the following discussion, which provides a useful example using a graph falling under Case (13s) of Fig. 4. Again, we are interested in the connected components of the graphs in $\mathbb{G}_n^2(a_{12},a_{13},a_{23})$, and a key realization is that the number of connected components in a graph G in this set does not change if one "collapses" vertices that are connected via an edge into a single larger vertex while maintaining the other edges that are incident upon these vertices. If one performs this collapsing operation on all of the vertices in $\mathbb{C}_{1,2}$, which we often refer to as "integrating out" $\mathbb{C}_{1,2}$, one converts a graph with $2n$ columns into one with $2n-2$ columns; that is, one converts a graph at order n to one at order $n-1$. Through this integration, we can write the number of connected components of the original graph as the sum of the remaining connected components in the collapsed graph plus the number of connected components originally contained entirely within $\mathbb{C}_{1,2}$. Our recursion, then, works by considering the effect of this contraction on all graphs at order n, grouped together by the cases listed in Fig. 4

It is worth explicitly highlighting that this approach based on integrating out $\mathbb{C}_{1,2}$ explains why we introduce the generalized sets of graphs $\mathbb{G}_n^2(a_{12},a_{13},a_{23})$; without these new graphs, this integration-based approach would not produce a recursion that closes, as integrating out $\mathbb{C}_{1,2}$ in some $G\in\mathbb{G}_n^2=G_n^2(0,0,0)$ generically induces a lower-order graph with red edges that cross between rows. As an example,

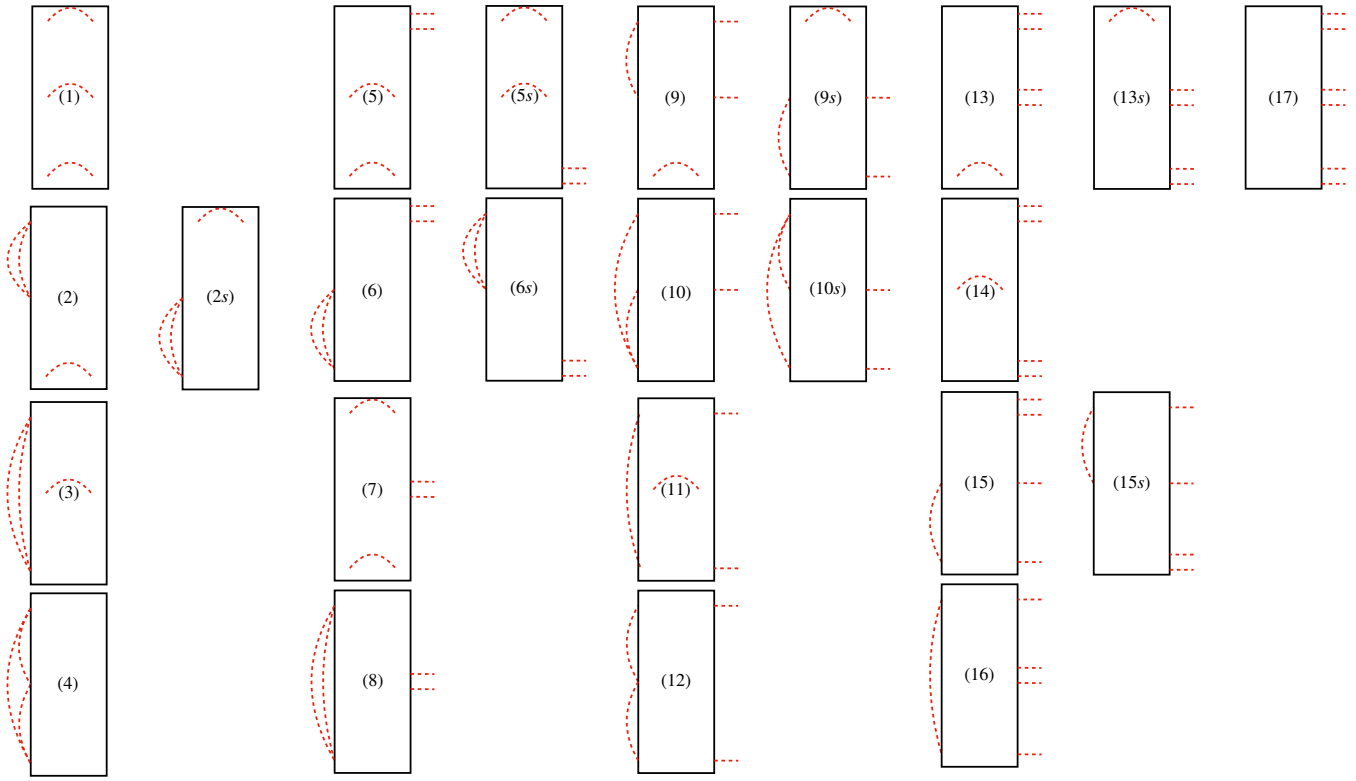


FIG. 4. List of 17 cases (up to symmetry) for how the first two columns in a graph of order n can connect into the rest of the graph. Each block represents $\mathbb{C}_{1,2}$ and a possible configuration of the red edges incident upon those vertices. Red edges connecting the two vertices in a single row are, of course, fixed. Red edges that go between different rows in $\mathbb{C}_{1,2}$ are depicted on the left of the box. These edges are not entirely fixed, as there is more than one way for an edge to connect vertices in two different rows. Red edges that connect $\mathbb{C}_{1,2}$ to the rest of the graph are depicted as protruding from the same row on the right side of the block. We do not draw the four possible sets of black edges within the block, but understanding their effect is crucial to the actual mechanics of the recursion.

the first part of Fig. 5 shows a graph in $\mathbb{G}_4^2$, where $\mathbb{C}_{1,2}$ is integrated out, as denoted by the hashing, and the second figure depicts the consequence of this integration. Consider the path P_3—P_1—Q_1—Q_6 that passes through $\mathbb{C}_{1,2}$. Collapsing the vertices P_1 and Q_1 into P_3 and Q_3, respectively, does not change the number of connected components, but it induces an edge P_3—Q_6 that is not allowed in graphs in $\mathbb{G}_n^2$. Therefore, the newly induced graph is not an element of $\mathbb{G}_3^2$, but instead an element of $\mathbb{G}_3^2(0,0,2)$. Hence, if we only considered graphs in $\mathbb{G}_n^2$, this integration procedure would not allow us to derive the recursion we desire.

To proceed, we work through the effect of the contraction on each case in Fig. 4 to determine the case-wise coefficients. In particular, there are three contributions to these coefficients, which we refer to as loop, vectorial, and combinatorial.

First is the *loop* contribution, which is the easiest to determine. This is, simply, how many internal connected components there are entirely within $\mathbb{C}_{1,2}$. This determines, roughly, how many factors of k there are in the case-wise recursive coefficients. The loop factor from the contraction in Fig. 5 is k due to the connected component O_1—O_2 contained entirely in $\mathbb{C}_{1,2}$.

Second is a *vectorial* contribution that, essentially, tells us how the vectors $\mathbf{a} = (a_{12}, a_{13}, a_{23})$ and $\mathbf{b} = (b_{12}, b_{13}, b_{23})$ are related to one another after the contraction. That is, what kinds of red edges are eliminated by integrating out $\mathbb{C}_{1,2}$, and what red edges are thereby created between the ends of the protruding edges (where a protruding edge is one that connects a vertex in $\mathbb{C}_{1,2}$ to one outside that set). This contribution depends on both the internal red edges determined by the specific case as well as which rows the protruding edges are incident upon in the rest of the graph. Generically, then, determining this contribution will require yet more casework whereby we must consider all possible combinations of where those protruding edges could land in the rest of the graph. In the specific example depicted in Fig. 5, we see that $\mathbf{a} = (0,0,0)$ but $\mathbf{b} = (0,0,2)$ because the contractive procedure generates two edges that span the second and third rows. We often refer to the vectorial contribution in terms of a vector $\mathbf{\Delta} = (\Delta_{12}, \Delta_{13}, \Delta_{23})$, where $\Delta_{ij} := b_{ij} - a_{ij}$. Here, then, $\mathbf{\Delta} = (0,0,2)$. It bears repeating one final time that the existence of this vectorial contribution explains why we must generalize the graphs we consider.

Finally, there is the most complicated contribution, which is *combinatorial* in nature. Essentially, contraction is not an injective procedure—contracting many graphs at a higher order might lead to the same graph at lower order. Consider once more Fig. 5. We see that the contraction procedure induces edges P_3—Q_6 and P_6—Q_3. These come from the paths P_3—P_1—Q_1—Q_6 and P_6—P_2—Q_2—Q_3, respectively. However, imagine that the original graph instead contained the paths P_3—P_2—Q_2—Q_6 and P_6—P_1—Q_1—Q_3. The red

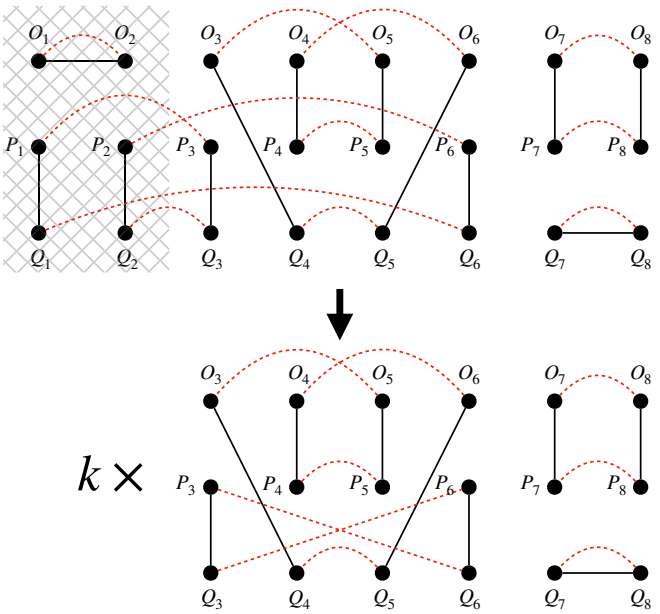


FIG. 5. Explanation of how the contraction procedure at the heart of the recursive analysis works, as well as why we must generalize the kinds of graphs that we consider in order to develop the recursion. We illustrate this procedure on an example graph from Case (13s) in Fig. 4. The crosshatch pattern on the first two columns ($\mathbb{C}_{1,2}$) of the top graph, collapses all vertices within $\mathbb{C}_{1,2}$ while keeping track of any connected components contained exclusively in $\mathbb{C}_{1,2}$. In this example, this "integrating out" induces a multiplicative factor of k due to the connected component O_1—O_2 contained entirely within $\mathbb{C}_{1,2}$, and it also induces red edges P_3—Q_6 and P_6—Q_3. Such red edges are not allowed for graphs in $\mathbb{G}_n^2$, meaning the final graph is in $\mathbb{G}_n^3(0,0,2)$, hence why we consider the more general graphs defined in Sec. IV A.

edges incident on $\mathbb{C}_{1,2}$ in this original graph still match case (13s) in Fig. 4, and contracting these paths by integrating out $\mathbb{C}_{1,2}$ still induces the same final edges P_3—Q_6 and P_6—Q_3, respectively, meaning the final lower-order graph is the same. Thus, because our recursion operates at the level of these sets of graphs partitioned by cases, we must carefully account for this lack of injectivity by including some combinatorial factors (typically based on the number of edges of each type b_{12}, b_{13}, b_{23}). This is, typically, the most difficult part of the casework.

To summarize, there are three different contributions to $c(\mathbf{a},\mathbf{b})_{\text{case}(i)} \coloneqq c(a_{12},a_{13},a_{23},b_{12},b_{13},b_{23})_{\text{case}(i)}$:

(1) *Loop.* This corresponds to the number of connected components in $\mathbb{C}_{1,2}$. This is the easiest contribution to determine;

(2) *Vectorial.* This corresponds to the relationship between $\mathbf{a}$ and $\mathbf{b}$. When integrating out $\mathbb{C}_{1,2}$, one loses contributions from internal edges that are lost by collapsing the vertices, but one gains edges of the types that are induced between the vertices that have a protruding edge incident upon them. While somewhat simple in spirit, it often requires significant casework;

(3) *Combinatorial.* This corresponds to the combinatorial factors that are associated with how many different higher-order graphs contract to the same lower-order graph. This depends both on the number of protruding edges and how the red and black edges interact via the vertices in $\mathbb{C}_{1,2}$. It is often the most complicated term.

In Appendix D, we provide significant details on how to compute these different components for each of the cases in Fig. 4. These calculations allow us to compute the case-wise recursive coefficients, and hence the recursion itself.

C. Numerical details

Once the theoretical principles behind the recursion in Eq. (28) are developed, we simply account for the contributions from each case and evaluate the recursion numerically and exactly. We accomplish this using the Julia programming language [23] and find $g(n,0,0,0)$ from $n=1$ to $n=40$ (which, recall, means up to photon sector 80).

We now briefly describe our implementation of the exact numerical recursion; the code is available on GitHub [24]. As a consequence of Eq. (26), the polynomial coefficients in $g(n,a_{12},a_{13},a_{23})$ grow at most factorially, so the number of bits needed to store the integers grows polynomially. Therefore, to ensure exact accuracy of all of the integer calculations, we use Julia's `BigInt` type, which allows us to achieve arbitrary-precision arithmetic [23]. Next, in order to avoid performing slow symbolic arithmetic operations, we represent polynomials in k as `BigInt` arrays, where the ith element of the array corresponds to the coefficient in front of the k^i term in the polynomial. Multiplication and addition of polynomials in k is then done at the array level. We begin with $n=1$ and store the base case values of $g(1,a_{12},a_{13},a_{23})$ given in Appendix D 1. To compute the value of $g(n,a_{12},a_{13},a_{23})$, we iterate through the 17 cases described in Appendix D and compute the various loop, vectorial, and combinatorial that show up in the sum in Eq. (28). We then recursively compute the values of $g(n-1,b_{12},b_{13},b_{23})$. The algorithm utilizes memorization every time any value of $g(n,a_{12},a_{13},a_{23})$ is computed so that the recursion rarely needs to go particularly deep. In the end, in order to compute up to $g(40,0,0,0)$, we compute $g(n,a_{12},a_{13},a_{23})$ for around 50 000 combinations of arguments, resulting in almost 200 megabytes of (uncompressed) data.

The evaluation of the recursion is classically efficient. In short, the number of allowed $\mathbf{a}$ (i.e., those that satisfy the necessary pairwise parity and sum constraints) is polynomially bounded, the size of the coefficients cannot be more than factorially large (meaning they can be stored with polynomial space), and the array-based multiplication and addition is classically tractable. More details are presented in Appendix E.

V. ANALYSIS OF THE SECOND MOMENT

In this section, we analyze the results derived from the numerically exact evaluation of the recursion described in the previous section. We begin by comparing our results to a numerical approximation based on calculating the moments through sampling random Gaussian matrices. We then derive a few analytic results about $g(n,0,0,0)$, including its value at $k=1$ and the value of its leading-order coefficient. We can use these results to derive upper and lower bounds for the second moment, which we then compare with the numerically

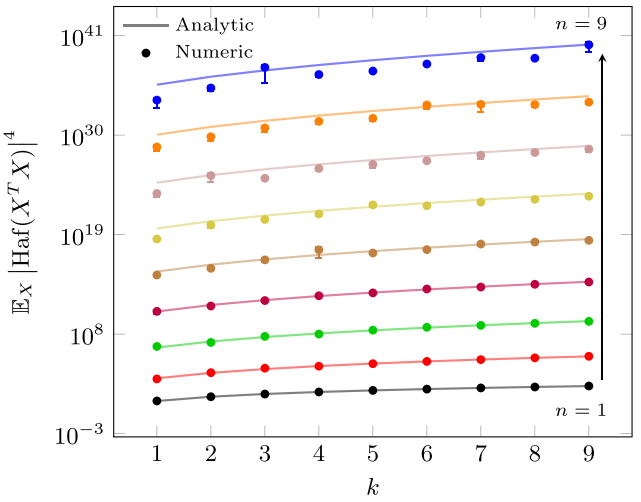


FIG. 6. Numerical test of recursion. The x axis represents k, and the y axis represents $\mathbb{E}_{X\sim\mathcal{G}^{k\times 2n}}[|\mathrm{Haf}(X^\top X)|^4]$. Solid lines, from $n = 1$ through $n = 9$ are the theoretical predictions derived from the recursion relation (see Ref. [24] for the code). Dots and bars represent the expected value and standard error, respectively, estimated by sampling 10^5 random Gaussian matrices and computing the second moment using the code provided by Ref. [25]. Note that, for many points, the size of the error bar is smaller than its associated dot. Furthermore, there is an asymmetry in the error bars due to the log nature of the plot. We see excellent alignment between theory and numerics for $n = 1$ through $n = 5$. For larger n, the agreement is still good, but we seem to undersample the true value in many cases. We suspect that this is because the distribution of the second moment has a long tail, meaning we do not suspect that the given error bars are indicative of the true difference between the sampled and numerically exact data. We believe that were we able to either take sufficiently more samples we would see stronger agreement between the sampled and true means, but this option is too computationally demanding given the size of the matrices involved and the exponential complexity of classically computing the hafnian [26].

exact data to understand how well they capture the scaling of the second moment.

First for various n and k, we numerically sample 10^5 random $X \in \mathcal{G}^{k\times 2n}$, compute $|\mathrm{Haf}[X^\top X]|^4$ using the code provided by Ref. [25], and average the results. This gives a numerical approximation to $(2n-1)!!g(n,0,0,0)$. We perform this calculation for $n, k \in \{1, 2, \ldots, 9\}$. The result is shown in Fig. 6, and we see good agreement between the approximate numerical calculations (data points and error bars) and the theoretical values predicted by the recursion (solid lines)

We next derive a few simple analytic results about the values of the coefficients of the polynomial expansion as well as the overall scaling of the second moment. The former are crucial to demonstrating the transition in anticoncentration, which is the central result of Ref. [18], whereas the latter provide useful intuition behind the behavior of the second moment. We begin with a lemma.

Lemma 1. We have that

(i) $M_2(1, n) = [(2n-1)!!]^4 4^n$.

(ii) $c_{2n} = (2n)!!$.

a. Proof, part (i). Examine Eq. (14). Because $k = 1$, $o_i = p_i = q_i = 1$ for all i. Thus, regardless of the permutation, all Kronecker δs are always satisfied. This means that, independent of the permutation, each factor is always four such that the product becomes 4^n. The sum over the three copies of S_{2n} then simply yields a factor of $(2n)!^3$. The result then follows. It also follows from combining Eqs. (25)–(27).

Proof, part (ii). We argue in the proof of Theorem 2 that the leading-order term in the polynomial expansion of the second moment is k^{2n}, and it comes from graphs that consist of only type-1 and type-4 black edges. Each type-1 and type-4 set of edges contains a horizontal black edge, and the two vertices linked by that black edge also must be linked by a red edge to create a 2-vertex connected component. Additionally, the vertical edges of the type-1 sets need to be paired off via red edges; similarly, the vertical edges of the type-4 sets need to be paired off. This ensures that each other connected component has exactly four vertices, maximizing the number of possible connected components.

Figure 7 visualizes how to now reduce the remaining calculation to the value of the first moment when $k = 2$. If we imagine collapsing each pair of adjacent vertical edges (i.e., those coming from the same group of 6 vertices) onto a pair of vertices connected by a black edge, we reproduce the atomic graph from the proof of the first moment. Here, by atomic graph, we mean the vertices and the fixed black edges which are shared by all graphs; the red edges are not yet included. Explicitly, there are $2n$ vertices, and vertices O_{2i-1}, O_{2i} are connected with a black edge. The black edges here act to identify that the original uncollapsed vertical edges were of the same type. Drawing red edges in the simplified graph on $2n$ vertices corresponds to pairing off vertical edges in the original graph on $6n$ vertices with red edges. Note that this also implies that red edges connect vertical edges of the same type. Therefore, a connected component in the simplified graph could correspond to two pre-images in the original graph: either all type-1 vertical edges, or all type-4 vertical edges. Then, by summing over all graphs and weighting each connected component by 2, we are effectively evaluating the sum in Eq. (13) at $k = 2$; i.e.:

$$\sum_{G\in\mathbb{G}_n^1} 2^{C(G)} = \left.\frac{(k+2n-2)!!}{(k-2)!!}\right|_{k=2} = (2n)!! \tag{32}$$

■

As mentioned, the results of this Lemma 1 are sufficient to derive the transition in anticoncentration in Ref. [18]. However, we can also use these results to gain confidence that our recursive calculation of the $g(n, 0, 0, 0)$ is correct by comparing our numerical calculation to these results and seeing that they match. Even further, we find that the recursively computed values of all $g(n, a_{12}, a_{13}, a_{23})$ match Eq. (26) when evaluated at $k = 1$.

Using Lemma 1, we can also derive upper and lower bounds on the second moment:

Lemma 1. Lemma 1 implies

$$M_2(k, n) \leqslant (2n-1)!!^4 4^n k^{2n}, \tag{33}$$

$$M_2(k, n) \geqslant (2n)! k^{2n}, \tag{34}$$

$$M_2(k, n) \geqslant (2n-1)!!^4 4^n \tag{35}$$

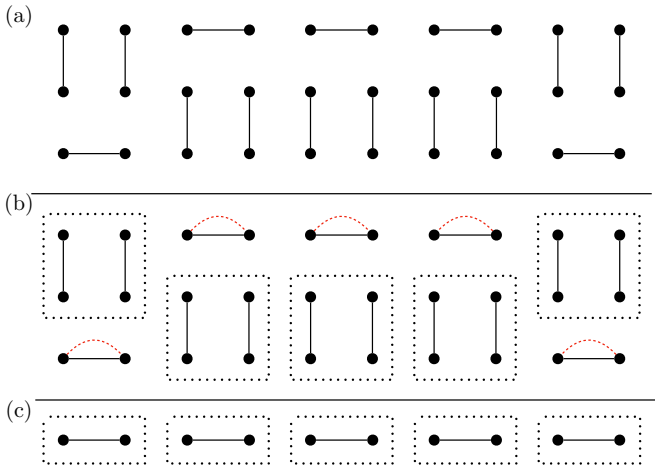


FIG. 7. Visualization of how the calculation of the coefficient of the leading-order term in the second moment can be reduced to the $k = 2$ case of the first moment. Recall that black edges are solid and red edges are dashed. (a) As proven in Theorem 2, graphs that maximize the number of connected components contain only type-1 and type-4 black edges. (b) To maximize the number of connected components, the horizontal black edges must form their own connected component with two vertices, meaning their vertices must be connected by a red edge. Furthermore, each vertical black edge must be paired off with exactly one other vertical black edge of the same type, forming a connected component with four vertices. We draw dotted boxes around the two black vertical edges to show that they come from the same type. (c) If we collapse each vertical edge onto a single vertex and then connect that vertex to the vertex stemming from its adjacent edge in the original graph (i.e., the other vertical edge from the same group of six vertices), then we reduce to the atomic graph (i.e., the graph with the fixed black edges, but without red edges) from the proof of the first moment. Red edges on this collapsed graph then correspond to pairing off vertical edges in the original graph with red edges. Because paired edges in the original graph can only exist between edges of the same type, each connected component in the simplified graph could have come from either type-1 or type-4 vertical edges. This is equivalent to evaluating $\sum_{G\in\mathbb{G}_n^1} k^{C(G)}$ after setting k, the base for the connected components, to two.

Proof. We first prove the upper bound. The leading term in $g(n, 0, 0, 0)$ is of the form k^{2n}, and the total number of graphs with no red edges crossing between rows is $(2n-1)!!^3 4^n$. Thus, the upper bound comes from saying that all graphs have $2n$ connected components.

We next prove the lower bounds. The first lower bound comes from considering only the leading-order term in the polynomial expansion, which is given in Lemma 1(ii). Because each term in the expansion is non-negative, this is a valid lower bound. The second lower bound comes from observing that $g(n, 0, 0, 0)$ is monotonically increasing with k, as there are no negative coefficients in the polynomial expansion. Therefore, we can also take a lower bound which is simply the value at $k = 1$, which we know counts the total number of possible graphs and follows from Lemma 1(i). ■

Stirling's approximation tells us when each lower bound is most useful:

$$(2n)!k^{2n} \sim (nk)^{2n}\left(\frac{4}{e^2}\right)^n, \tag{36}$$

$$(2n-1)!!^4 4^n \sim n^{4n}\left(\frac{64}{e^4}\right)^n \tag{37}$$

For $k \in o(n)$, Eq. (37) is larger, and when $k \in \omega(n)$, Eq. (36) is instead larger. When $k \in \Theta(n)$, then both lower bounds have a leading dependence of n^{4n}, so which is better depends on the constant of proportionality.

Armed with our analytical results and the exact numerical data from the recursion, we can now investigate how the second moment scales with k and n. In Fig. 8(a), we plot the logarithm of the upper and lower bounds, as well as the numerically exactly computed values for $(2n-1)!!g(n, 0, 0, 0)$, for our largest available n, which is $n = 40$. We set $k = n^a$ with $a \in [0, 4]$. We see that, except for when $k = n^0$ and the upper bound is exactly correct (as is the lower bound based on the number of graphs), the lower bound is a much better approximation. In fact, as expected, the lower bound based on the leading order appears to become a very good approximation as k gets larger

We should also point out that the logarithmic scaling of the y axis of Fig. 8(a) means that small differences between the exact values and the corresponding lower bound actually represent large multiplicative differences between the true values. For this reason, in Fig. 8(b), we also plot on a log scale the relative error of the exact data vs the composite lower bound defined by max{Eq. (34), Eq. (35),}. This helps show how the exact data trends toward Eq. (34) as k grows.

Relatedly, we can actually show analytically that Eq. (34) cannot fully capture the scaling of the second moment when $k = O(n^2)$. In Appendix F, we discuss how to compute individual coefficients in the polynomial expansion of the second moment. There, we give a new proof that $c_{2n} = (2n)!!$, and we also prove that $c_{2n-1} = (2n)!!(3n-2)n$. Together, these two results mean

$$\frac{c_{2n}k^{2n}}{c_{2n-1}k^{2n-1}} = \frac{k}{(3n-2)n} \sim \frac{k}{n^2} \tag{38}$$

Therefore, in order for the leading term $c_{2n}k^{2n}$ to asymptotically dominate $c_{2n-1}k^{2n-1}$, we require $k = \omega(n^2)$. *A fortiori*, for the leading term to dominate all other terms, and, therefore, for the leading-order lower bound to be a good approximation for the second moment, k must be $\omega(n^2)$.

In summary, then, the lower bounds in Eqs. (34) and (35) typically track the true value of the second moment much better than the upper bound in Eq. (33). When $k = \omega(n^2)$, the first lower bound, Eq. (34), which is based on the leading-order term, appears to be a very good approximation to the second moment.

VI. LOCATING THE TRANSITION IN ANTICONCENTRATION

We now move on to some of the concrete consequences of our work. The main result of Ref. [18] is identifying a transition in anticoncentration in Gaussian boson sampling as a function of k, the number of initially squeezed modes.

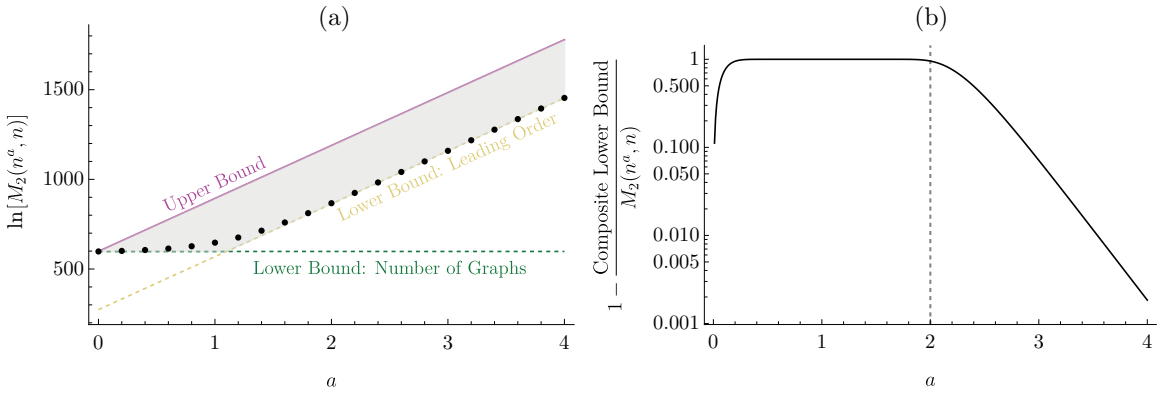


FIG. 8. Plots showing scaling of the second moment compared with upper and lower bounds. For both plots, physically, k should be an integer, but we here simply use the polynomial expansion of the second moment as a function of arbitrary real k. (a) Scaling of logarithm of the second moment and its upper and lower bounds for $n = 40$ and $k = n^a$ with $a \in [0, 4]$. The green horizontal dashed line and the yellow slanted dashed line represent the lower bounds based on the number of graphs [Eq. (35)] and the leading-order term [Eq. (34)], respectively. The maroon solid line represents the upper bound Eq. (33). The bound region is, therefore, highlighted in gray. Numerically exact data are given for $n = 40$ by the black dots [24]. Notice that the black dots representing the exact data stay within the gray region and, for most values of a, closely track the lower bound. Note also that, per Eqs. (36) and (37), the intersection between the two lower bounds occurs around $a = 1$ (it is slightly greater than $a = 1$ for finite n, but it trends toward one as n gets large). (b) Relative error of the composite lower bound compared with the true value, plotted on a logarithmic scale. We see that, around $a = 2$, there is a rapid decrease (that appears to be exponentially fast) in the relative error. This strongly suggests that $a = 2$ indicates where the lower bound starts to become a good approximation.

This result follows entirely from analytic results. Specifically, we show through direct computation that, when $k = 1$, the output probabilities do not anticoncentrate, and we use the leading-order term to show that these output probabilities weakly anticoncentrate in the limit that $k \to \infty$. Hence, we show the existence of a transition, but we do not isolate its exact location. We do conjecture that it occurs at $a = 2$, where k scales with n as $k = \Theta(n^a)$, based on an allusion to scattershot boson sampling [5], which is another generalization of Fock state boson sampling; there the initial state is composed of two-mode squeezed states where one half of each state is measured and postselected on measurements with at most one photon. In short, one can roughly draw a connection between the presence of hiding in scattershot boson sampling and the number of initially squeezed modes.

Note that, *a priori*, it is possible that polynomial scaling is not sufficient to show the transition in anticoncentration, and it is possible that k would need to scale, say, exponentially in n to see it. However, one of the main contributions of this work is to show convincingly that this polynomial scaling is sufficient and that the location of the transition is indeed at $k = \Theta(n^2)$. We accomplish this through numerical arguments based on the exact data generated through the recursion for the second moment and a few more analytic results. We formalize this with the following conjecture:

Conjecture 2 (anticoncentration in Gaussian boson sampling). Let $2n = o(\sqrt{m})$ such that one operates in the (conjectured) hiding regime. Then Gaussian boson sampling does not anticoncentrate for $k = O(n^2)$, but it weakly anticoncentrates with inverse normalized second moment, $m_2(k, n) := M_1^2(k, n)/M_2(k, n)$, scaling as $1/\sqrt{\pi n}$ for $k = \omega(n^2)$.

Our evidence for Conjecture 2 is twofold and based on results regarding the anticoncentration of the approximate distribution (see Appendix A 2 for details on how to convert these statements to those about anticoncentration of the exact distribution):

(1) We provide a sequence of numerical plots of $\ln([m_2(k, n)\sqrt{\pi n}]^{-1})$ and its symmetric difference with respect to n for various polynomial scalings of k with n. The numerical plots of the function itself show an exponential scaling when $k = O(n^2)$, but that the function becomes approximately constant when $k = \omega(n^2)$. Similarly, the plots of the symmetric difference are positive in the $k = O(n^2)$ regime but approximately vanish when $k = \omega(n^2)$.

(2) We show that, assuming the lower bound for $M_2(k, n)$ is a good approximation, weak anticoncentration holds for $k = \omega(n^2)$. We also show that there is a lack of anticoncentration when $k = o(n)$.

We begin with the numerical evidence. In Fig. 9, we set $k = n^a$ and plot $\ln([m_2(k, n)\sqrt{\pi n}]^{-1})$ for various values of a. We choose this quantity because, in the asymptotic limit of large k, $[m_2(k, n)\sqrt{\pi n}]^{-1} \sim 1$, but when $k = 1$, it is exponentially big [18]. Therefore, we hope to use Fig. 9 to understand how this quantity interpolates between the exponential and polynomial behavior of $m_2(k, n)^{-1}$. In Fig. 9(a), we plot $\ln([m_2(k, n)\sqrt{\pi n}]^{-1})$ for $a = 0.5$ to $a = 4.0$ with spacing 0.5. We see that for $a \leqslant 2$, this quantity seems to linearly increase with n, meaning that $m_2(k, n)^{-1}$ is exponentially large in n. However, for $a > 2$, it trends to a small constant. Because $m_2(k, n) \sim 1/\sqrt{\pi n}$ is derived in the limit of asymptotically large k using the leading-order lower bound for the second moment in Eq. (34), this suggests that the use of this lower bound is a good approximation to the second moment when $a > 2$; this aligns well with Fig. 8. Thus, we see that, when $a > 2$, the normalized second moment trends to its asymptotic-in-k value of $\sqrt{\pi n}$. In Fig. 9(b), we enlarge the suspected transition point and plot the same quantity when $a \in \{1.95, 1.99, 2.00, 2.01, 2.05, 2.10, 2.15, 2.20\}$. We see similar behavior in this plot; namely, at approximately

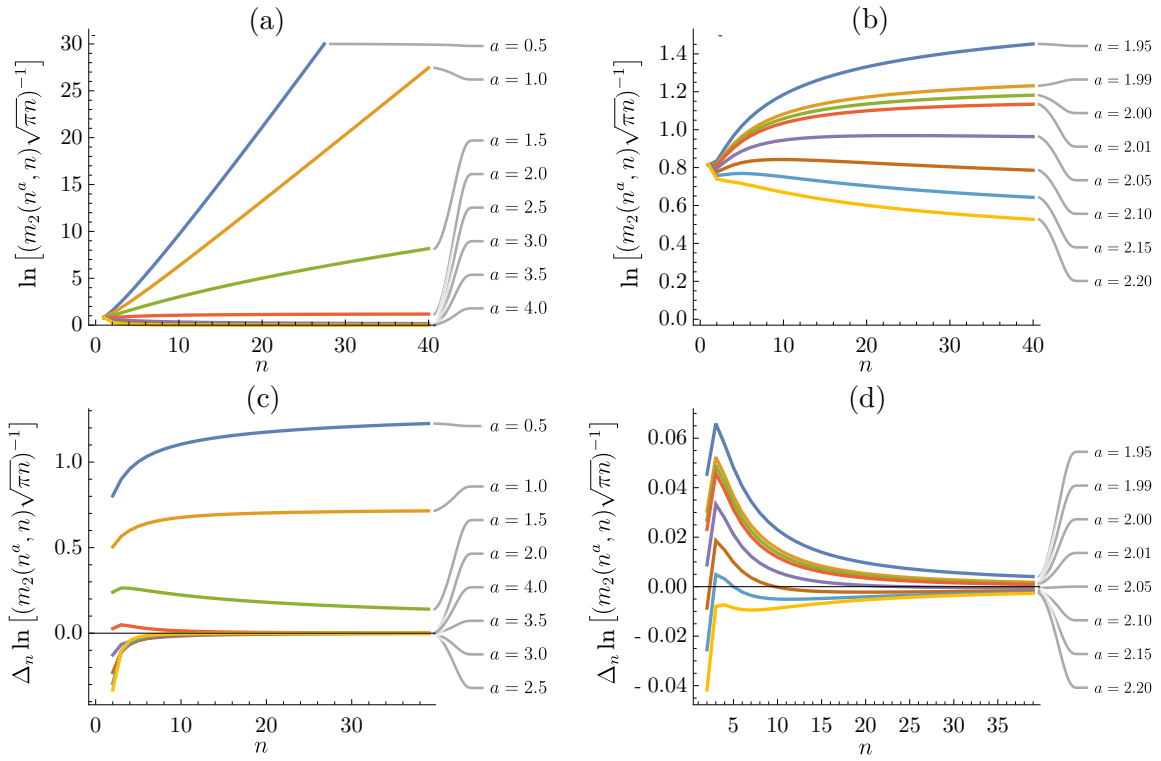


FIG. 9. Plots of $\ln([m_2(k,n)\sqrt{\pi n}]^{-1})$ and its symmetric difference, denoted Δ_n, as a function of n for $k = n^a$. Recall that $m_2(k,n) := M_1(k,n)^2/M_2(k,n)$ and, for asymptotically large k, $m_2(k,n) \sim 1/\sqrt{\pi n}$ [18]. (a) $a \in [0.5, 4.0]$, equally spaced by 0.5. (b) $a \in \{1.95, 1.99, 2.00, 2.01, 2.05, 2.10, 2.15, 2.20\}$ to show the regime around $a = 2$ more clearly. (c) The symmetric difference of $\ln([m_2(k,n)\sqrt{\pi n}]^{-1})$ with respect to n, again with $a \in [0.5, 4.0]$. (d) Zooming in on the symmetric difference when a is around 2, with the same values as plot (b). Note that each of the curves in plots (a) and (b) are composed of numerically exact data at 40 points ($n \in \{1, \ldots, 40\}$) that are smoothed for visualization. The same holds for plots (c) and (d), except there are only 38 points ($n = 1$ and $n = 40$ are excluded because we compute the symmetric difference). Finally, while k physically must be an integer, we do not enforce that for these plots; we instead just using the polynomial expansion of the moments to extend k to arbitrary real numbers.

$a = 2$, the curves transition from growing in n to decreasing toward zero. To clarify this point even further, we also plot the symmetric difference of the above quantity as a function of n (excluding the minimum and maximum values of n). Here, the symmetric difference of a function $f(n)$, which we refer to as $\Delta_n f(n)$, is defined as $[f(n+1) - f(n-1)]/2$. Figures 9(c) and 9(d) use the same values of a as Figs. 9(a) and 9(b), respectively. We see that, up to some finite-size effects, when $a > 2$ this symmetric difference trends to zero, but it remains positive for $a \leqslant 2$

We next plot in Fig. 10 the symmetric difference $\Delta_n \ln([m_2(k,n)\sqrt{\pi n}]^{-1})$ with respect to n at $n = 39$ (so that $n + 1 = 40$ is the largest Fock sector for which we ran our numerical recursion) as a function of a. We see the symmetric difference vanish near $a = 2$, as would be expected if the transition occurs at $k = \Theta(n^2)$. The inset of Fig. 10 clarifies this by plotting the logarithm of this symmetric difference such that its vanishing instead becomes a divergence

For our second, more analytic argument, we show that if the lower bound is a good approximation to the second moment, then weak anticoncentration holds for $k = \omega(n^2)$ and there is a lack of anticoncentration when $k = o(n)$.

First, consider the case $a < 1$. Note that $k = n^a$ is negligible to n (asymptotically in n). Therefore, up to subleading order,

$$\frac{(k+2n-2)!!}{(k-2)!!} \sim (2n)!! \tag{39}$$

Using Eq. (35), which is a valid lower bound, we get

$$\frac{M_2(k,n)}{M_1(k,n)^2} \gtrsim \frac{(2n-1)!!^4 4^n}{[(2n-1)!!(2n)!!]^2} \tag{40}$$

$$= 4^n \frac{(2n-1)!!^2}{(2n)!!^2} \tag{41}$$

$$\sim \frac{4^n}{\pi n}, \tag{42}$$

which is exponentially big, demonstrating a lack of anticoncentration (accounting for the subleading contribution of k does not change the conclusion). Here, we have used Stirling's approximation and

$$\frac{(2n)!!}{(2n-1)!!} \sim \frac{\sqrt{2\pi n}(2n/e)^n}{\sqrt{2}(2n/e)^n} = \sqrt{\pi n} \tag{43}$$

We now examine the case where $k = n^a$ with $a > 2$. We use that, according to Fig. 8, the lower bound $M_2(k,n) \geqslant (2n)!k^{2n}$ is actually an extremely good approximation to the

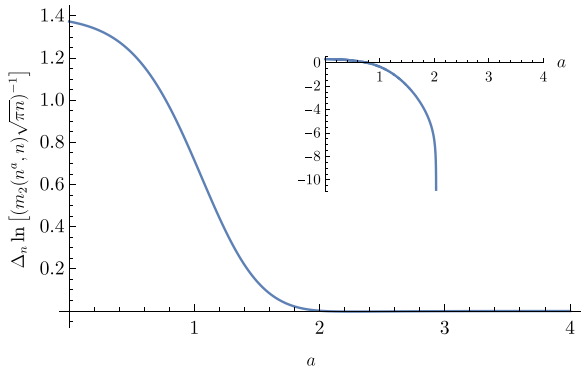


FIG. 10. Symmetric difference $\Delta_n \ln([m_2(k,n)\sqrt{\pi n}]^{-1})$ evaluated at $n = 39$. Here, $k = n^a$, and a represents the x axis. Again, physically, k must be an integer, but for this plot we are simply using the polynomial expansions of the moments where k can be an arbitrary real number. This symmetric difference vanishes very close to $a = 2$, suggesting that, when $k = \Omega(n^2)$, the quantity $m_2(k,n)\sqrt{\pi n}$ is a constant, meaning the normalized second moment appears to scale as $\sqrt{\pi n}$. The inset simply plots the ln of the y axis in the main plot (still with a along the x axis) in order to visualize more clearly the transition. The divergence occurs somewhere around $a = 2.03$, but we suspect this difference is due solely to finite-size effects. Beyond this divergence, the symmetric difference is negative, meaning the logarithm is complex and, thus, not plotted.

second moment. Here, k now dominates n, so

$$\frac{(k+2n-2)!!}{(k-2)!!} \sim \sqrt{k^{2n}} = n^{an} \tag{44}$$

Correspondingly, the normalized second moment scales as

$$\frac{M_2(k,n)}{M_1(k,n)^2} \sim \frac{(2n-1)!!(2n)!!k^{2n}}{(2n-1)!!^2 k^{2n}} \tag{45}$$

$$= \frac{(2n)!!}{(2n-1)!!} \tag{46}$$

$$\sim \sqrt{\pi n} \tag{47}$$

Therefore, when $k = \omega(n^2)$, weak anticoncentration holds (again, the inclusion of any subleading terms does not change the conclusion). Note that this argument is similar to the argument used to demonstrate the existence of the transition in the first place, but it uses the fact that the second moment is already well approximated by the leading-order lower bound at $k = \omega(n^2)$ instead of just in the asymptotic limit of large k. Unfortunately, our current results are insufficient to more formally handle the regime $a \in [1, 2]$ regime.

To recap, we have shown the following results: First, we have provided numerics in Figs. 9 and 10 that suggest that $\sqrt{\pi n}$ is a good approximation to the normalized second moment when $k = \omega(n^2)$. This is the value of the normalized second moment that is calculated when one uses the lower bound in Eq. (34) that is based on the leading-order term. Similarly, these plots numerically indicate that when $k = O(n^2)$, the normalized second moment grows exponentially in n, meaning there is a lack of anticoncentration. Next, we have shown that, if the leading order is a good approximation to the second moment, which, according to Fig. 8 occurs when $k = \omega(n^2)$, then the normalized second moment scales as $\sqrt{\pi n}$, meaning weak anticoncentration holds in that regime. We have also shown that for $k = O(n)$, there is a lack of anticoncentration. Altogether, the totality of the evidence presented here strongly suggests the veracity of Conjecture 2 and that the transition between lack of anticoncentration and weak anticoncentration in the approximate output distribution occurs at $k = \Theta(n^2)$.

VII. CONCLUSION

In this work, we have studied the output distribution of the prototypical setup for Gaussian boson sampling in the hiding regime. Our main theoretical contribution is the development of a recursion relation that allows one to compute numerically exactly in polynomial time the second moment of these output probabilities for any photon Fock sector. We additionally detail separate ways to calculate individual coefficients of the polynomial expansion of the second moment. Together, these results provide strong evidence for our conjecture that the transition in anticoncentration, whose existence is proven in Ref. [18], occurs at $k = \Theta(n^2)$.

Ideally we would have been able to derive a closed-form expression for the polynomial description of the second moment akin to Theorem 1, as this might have allowed us to formally prove this conjecture, but we leave this important question to future work. It would also be nice to develop a better, more intuitive understanding for why this transition occurs. It appears to be related to the transition between photon-collisional and photon-collision-free outputs in scattershot boson sampling, but the connection is not perfect, and further investigation seems worthwhile.

Related to all of these points, the precise nature of the crossover at $k = \Theta(n^2)$ is an interesting realm of future study. Specifically, we conjecture that weak anticoncentration holds for $k = \omega(n^2)$ and there is a lack of anticoncentration when $k = O(n^2)$, which of course places the transition at $k = \Theta(n^2)$. But precisely how the normalized moment behaves as we tune a through $a = 2$ deserves special attention.

Our results may open the door for answering other questions of interest. As we show in Sec. II B, our calculation of the normalized second moment immediately provides the expected linear cross-entropy benchmarking score for an error-free Gaussian boson sampling device. This means that all of the above statements about anticoncentration in terms of the normalized second moment (including, specifically, the transition in scaling as a function k) also hold for linear cross-entropy benchmarking. Furthermore, now that this ideal score is derived, it is possible that this information could be helpful in evaluating experimental implementations of Gaussian boson sampling. For example, any difference between the calculated cross-entropy score and the expected score for an error-free sampler could shed some light on the type or degree of error in the system. Relatedly, it would be interesting to see whether our techniques can be expanded to calculating moments in imperfect settings, such as when photons are partially distinguishable [27], or when the measurement detectors only distinguish between the presence or absence of photons [28]. Should any of these extensions be possible, it then suggests that one might be able to calculate the expected cross-entropy scores assuming specific error models. This might allow for

even more fine-grained analysis of the types of error present in a given experiment using calculations related to cross-entropy. Finally, our results may make it possible to evaluate how well certain classical algorithms may spoof cross-entropy benchmarking in Gaussian boson sampling [29]. Further exploration along all of these axes is worthwhile.

Finally, the graph-theoretic approach that we have developed in this paper is surprisingly flexible, and it deserves continued treatment. In Appendix G, we present another way to use the graphs in $\mathbb{G}_2^n$ in order to develop a recursion that can solve for the second moment. In short, this other approach observes that there are really only five types of black edges in our graphs: ones that stay in row 1, ones that stay in row 3, and ones that go between rows 1 and 2, rows 1 and 3, and rows 2 and 3. Because we are interested only in the number of connected components, and because we sum over *all* perfect matchings defined by red edges in each row, we are free to drag the black edges around and order them in new, convenient ways. Therefore, looking at these graphs from the perspective of the total number of each type of black edge allows us to conceive of a different kind of recursion for the second moment. While we only sketch the idea behind this alternative recursion, we believe that it may be a promising new way of looking at the problem. In particular, this new approach allows us to find an admittedly somewhat complicated expression for c_1 (which reproduces our expression for c_1 found via the original recursion up to $n = 40$). However, this new approach should not be viewed as a strict alternative to what we have derived in this paper, but a complementary approach that might yield new insights. We leave exploring it to future work.

Note added. Recently, Ref. [30] was posted to the arXiv, where the authors also study second moments of Gaussian boson sampling.

ACKNOWLEDGMENTS

We thank Changhun Oh, Bill Fefferman, Marcel Hinsche, Max Alekseyev, and Benjamin Banavige for helpful discussions. We thank Jacob Bringewatt for providing feedback on the Appendix discussing the classical complexity of evaluating the recursion. This material is based upon work supported by the U.S. Department of Energy, Office of Science, Accelerated Research in Quantum Computing, Fundamental Algorithmic Research toward Quantum Utility (FAR-Qu). Additional support is acknowledged from DARPA SAVaNT ADVENT, AFOSR MURI, DoE ASCR Quantum Testbed Pathfinder program (Awards No. DE-SC0019040 and No. DE-SC0024220), NSF QLCI (Award No. OMA-2120757), NSF STAQ program, AFOSR, and NQVL:QSTD:Pilot:FTL. Support is also acknowledged from the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers, Quantum Systems Accelerator. J.T.I. thanks the Joint Quantum Institute at the University of Maryland for support through a JQI fellowship. D.H. acknowledges funding from the US Department of Defense through a QuICS Hartree fellowship and from the Simons Insitute for the Theory of Computing, supported by DOE QSA. Specific product citations are for the purpose of clarification only, and are not an endorsement by the authors or NIST.

DATA AVAILABILITY

The Julia code used in this work is available on GitHub [24].

APPENDICES

In the Appendices, we provide details and derivations that supplement the discussion in the main text

(1) Appendix A: We motivate our definition of anticoncentration with respect to the typical arguments for hardness of Gaussian boson sampling, and we also formalize Conjecture 1 to show how anticoncentration of the approximate distribution implies anticoncentration of the exact distribution;

(2) Appendix B: We discuss the relationship between the hiding property, Conjecture 1, and the evaluation of the first moment.

(3) Appendix C: We derive Eqs. (14) and (20) of the main text, which are the starting points of the graph-theoretic discussion of the second moment.

(4) Appendix D: We provide the graph-theoretic details for how to derive the recursion for the second moment.

(5) Appendix E: We show that evaluating the recursion for the second moment is efficient (i.e., the time and space required scale polynomially) in the Fock sector n.

(6) Appendix F: We discuss how to compute individual coefficients of the polynomial expansion of the second moment. Specifically, we give a combinatorial method to calculate the leading and first subleading terms in the polynomial expansion of the second moment.

(7) Appendix G: We discuss an alternative method for developing a recursion to solve for the second moment. We apply this alternative picture to find an expression for the constant term in the polynomial expansion of the second moment.

APPENDIX A: DETAILS ON ANTICONCENTRATION

In this Appendix, we discuss some of the details behind our definition of anticoncentration and how it relates to the standard notion of anticoncentration often used in the literature. We also discuss how these different definitions interact when it comes to showing anticoncentration holds for the exact distribution of the output probabilities of Gaussian boson sampling (i.e., the distribution defined in terms of unitary submatrices) given anticoncentration of the approximate distribution (i.e., the distribution defined in terms of random Gaussian matrices).

1. Anticoncentration in hardness arguments

We first discuss in somewhat more detail the relevance of anticoncentration to the argument for hardness of sampling from the output distribution of Gaussian boson sampling (GBS). This argument makes use of an approximate counting algorithm due to of an approximate counting algorithm due Stockmeyer [31]. Roughly, we assume that there is an efficient sampling algorithm for GBS that, given a linear-optical unitary U, samples from a distribution Q_U that is ϵ-close in total-variation distance to the ideal GBS distribution P_U [recall

Eq. (1) of the main tex] where $\epsilon > 0$ is a constant:

$$\mathsf{tvd}(P_U, Q_U) \coloneqq \frac{1}{2}\sum_{\mathbf{n}} |P_U(\mathbf{n}) - Q_U(\mathbf{n})| \leqslant \epsilon \tag{A1}$$

Given the so-called hiding property (see Sec. IV C 4 of Ref. [3] for details), we can use this sampling algorithm (supposing it exists) as input to Stockmeyer's algorithm. Stockmeyer's algorithm then approximates the probability $P_U(\mathbf{n})$ up to an error given by

$$\varepsilon = \frac{1}{\mathrm{poly}(n)} P_U(\mathbf{n}) + \frac{2\epsilon}{|\Omega|\delta}\left(1 + \frac{1}{\mathrm{poly}(n)}\right), \tag{A2}$$

with probability $1-\delta$ over $\mathbf{n} \in \Omega$, where Ω is the sample space on which P_U is defined. Note that ε is a combination of additive and relative errors with respect to the total-variation distance error ϵ. If it is sufficiently hard (#P-hard, to be precise) to approximate the outcome probabilities $P_U(\mathbf{n})$ up to the error (A2), on the instances on which our approximation scheme achieves this error, this rules out the approximate sampling algorithm up to very reasonable complexity-theoretic conjectures (one of which is the noncollapse of the polynomial hierarchy, a generalization of the famous $\mathsf{P} \neq \mathsf{NP}$ conjecture). The required property is thus what we call "approximate average-case hardness," that is, the statement that any algorithm which is able to compute $P_U(\mathbf{n})$ with probability $1-\delta$ over the instances up to the error in Eq. (A2) is able to solve any #P-hard problem [of the same difficulty as approximating the outcome probabilities $P_U(\mathbf{n})$ up to the error in Eq. (A2)].

While we know average-case hardness of approximating the outcome probabilities up to error $2^{-\Omega(n\ln n)}$ [11,32], it is only conjectured for the relevant approximation error given by either $c_1 P_U(\mathbf{n})$ or $c_2/|\Omega|$ for constants $c_1, c_2 > 0$. Anticoncentration serves as evidence for the truth of the conjecture, the idea being the following: suppose that most of the outcome probabilities are very close to zero, i.e., $\ll \epsilon/2^{-n}$, meaning only a vanishing fraction of them are relevant. Then a high approximation error on the relevant probabilities is tolerable, because we only need to distinguish between relevant and irrelevant outcomes, and a sufficiently good approximation to the irrelevant ones is zero. This is a significantly easier task than if the distribution is highly spread out and a large fraction of the probabilities is "relevant" in the sense that all of the relevant probabilities are of the same order of magnitude as those of the uniform distribution.

In the standard argument, this intuition is formalized as the statement

$$\Pr_{U\in\mathrm{U}(m)}\left[P_U(\mathbf{n}) \geqslant \frac{\alpha}{|\Omega|}\right] \geqslant \gamma(\alpha), \tag{A3}$$

for some constants $\alpha, \gamma(\alpha) > 0$. In this formulation, we have made crucial use of the hiding property, which asserts that the distribution over circuits is invariant under a procedure by which we "hide" a particular outcome $\mathbf{n}$ in the probability of obtaining a different outcome $\mathbf{n}'$ of a random circuit. This allows us to restrict our attention to the distribution over circuits of a fixed outcome $\mathbf{n}$.

The anticoncentration property (A3) implies that the mixed additive and relative error (A2) is dominated by the first term on a $\gamma(\alpha)(1-\delta)$ fraction of the instances because, with probability $\gamma(\alpha)$, we can upper bound the second term by $P_U(\mathbf{n})$. But, if a large fraction of the probabilities is larger than uniform, then none of them can be much larger than uniform and, hence, the approximation error needs to be exponentially small. Thus, we expect that, in the presence of anticoncentration, approximating the outcome probabilities up to the error (A2) is much harder than without anticoncentration, lending credibility to the approximate average-case hardness conjecture.

In our definition of anticoncentration, we consider the (normalized) average outcome-collision probability

$$P_2(\mathrm{U}(m)) \coloneqq |\Omega| \sum_{\mathbf{n}\in\Omega} \mathbb{E}_{U\in\mathrm{U}(m)}[P_U(\mathbf{n})^2] \tag{A4}$$

$$\overset{\text{hiding}}{=} |\Omega|^2 \mathbb{E}_{U\in\mathrm{U}(m)}[P_U(\mathbf{n})^2] \tag{A5}$$

The outcome-collision probability is the probability that, were one to sample the distribution twice (using the same transformation unitary U), one would receive the same outcome both times. For very flat distributions it is small, while it is large for heavily peaked distributions; with the given normalization, the outcome-collision probability of the uniform distribution is given by 1, but the normalized outcome-collision probability of a fully peaked distribution with a single unit probability is given by $|\Omega|$.

The average outcome-collision probability is thus another measure of the anticoncentration of the outcome probabilities in the ensemble of linear-optical unitaries. It is a more coarse-grained measure, though, because it is only an average quantity. Indeed, a (constantly) small average outcome-collision probability implies anticoncentration in the sense of (A3) via the Paley-Zygmund inequality, as

$$\Pr_{U\in\mathrm{U}(m)}\left[P_U(\mathbf{n}) \geqslant \frac{\alpha}{|\Omega|}\right] \geqslant (1-\alpha)^2 \frac{1}{P_2(\mathrm{U}(m))} \tag{A6}$$

The relevant quantity of interest to anticoncentration is thus the inverse normalized average outcome-collision probability $p_2(\mathrm{U}(m)) = 1/P_2(\mathrm{U}(m))$. Assuming that hiding holds, the first moment $\mathbb{E}_U[P_U(\mathbf{n})]$ must evaluate to the inverse size of the sample space (see Appendix B for a thorough discussion of this argument), so we can rewrite p_2 for GBS as

$$p_2(\mathrm{U}(m)) = \frac{\mathbb{E}_{U\in\mathrm{U}(m)}[P_U(\mathbf{n})]^2}{\mathbb{E}_{U\in\mathrm{U}(m)}[P_U(\mathbf{n})^2]} \approx \frac{M_1(k,n)^2}{M_2(k,n)} = m_2(k,n) \tag{A7}$$

In the main text, we define various degrees of anticoncentration in terms of the inverse normalized average outcome-collision probability p_2, which we recall here

(A) We say that $P_U, U \in \mathrm{U}(m)$ *anticoncentrates* if $p_2 = \Omega(1)$.

(WA) We say that P_U *anticoncentrates weakly* if $p_2 = \Omega(1/n^a)$ for some $a = O(1)$.

(NA) We say that P_U *does not anticoncentrate* if $p_2 = O(1/n^a)$ for any constant $a > 0$.

Here, we motivate those definitions in more detail.

(a) Anticoncentration. Clearly (A) implies anticoncentration in the sense of Eq. (A3), hence the definition.

(b) Lack of anticoncentration (NA). Ignoring the average over unitaries, the collision probability p_2 of a fixed

U upper-bounds the support of the outcome distribution of U by $p_2|\Omega|$. Let us assume for simplicity that p_2 is actually exponentially small. An exponentially small value of p_2 implies that the average fractional support of the outcome distributions P_U is exponentially small, implying that at least a constant fraction (over U) of the distributions P_U has exponentially small support and, conversely, exponentially larger than uniform probabilities on that support. At least for those distributions, this implies an exponentially larger error tolerance compared with $1/|\Omega|$. Such an exponentially larger error tolerance makes the approximate average-case hardness conjecture significantly stronger, presumably even untenable.

While it is possible that for a constant fraction of the U the outcomes are highly concentrated while, for another constant fraction, the probabilities are highly spread out, yielding a superpolynomially small p_2 as well as the anticoncentration property in the sense of (A3) (see Sec. V C of Ref. [33] for an example), this seems like an extremely unlikely state of affairs for typical instances, since it is highly fine-tuned and not likely to occur in practical scenarios. Indeed, the hiding property implies that it should not matter whether we talk about the distribution over unitaries or over outcomes, which means that the situation described above is a generic feature, rendering (A3) false in the case p_2 is exponentially small.

(c) Weak anticoncentration (WA). Our results show that weak anticoncentration holds in the regime where sufficiently many of the initial modes are squeezed. But why do we think of a polynomially decaying p_2 as *weak* anticoncentration rather than a lack of anticoncentration?

We argue that this is a meaningful regime in the sense that there is a stronger—but not inconceivable—approximate average-case hardness conjecture associated with the weak anticoncentration regime. To see this, observe that weak anticoncentration implies anticoncentration in the sense of Eq. (A3) with $\gamma(\alpha) = \Omega(1/\mathrm{poly}(n))$, which means that an inverse polynomial fraction of the outcome probabilities are larger than uniform. Technically, using Stockmeyer's algorithm we can thus achieve a multiplicative error for an inverse polynomial fraction of the outcome probabilities. To rule out an efficient classical sampler, we thus need to conjecture approximate average-case hardness with constant relative errors for any inverse polynomial fraction of the instances. Equivalently, we can formulate a similar conjecture for a polynomially large relative or subexponentially large additive error on a constant fraction. While clearly much stronger than the requirement of anticoncentration, this is qualitatively different from the lack of anticoncentration scenario (NA), where the difference is superpolynomial.

2. Anticoncentration of the exact distribution

We also need to show that our definition of anticoncentration allows us to translate between anticoncentration of the approximate distribution based on the hafnians of random Gaussian matrices, which we refer to as $P_X(\mathbf{n})$, and anticoncentration of the true distribution, $P_U(\mathbf{n})$. For a given output $\mathbf{n}$, let $\mathcal{D}_U$ be the distribution of the symmetric product $U^{\top}_{1_k,\mathbf{n}}U_{1_k,\mathbf{n}}$ with $U \in \mathrm{U}(m)$. Let $\mathcal{D}_X$ be the distribution of the symmetric product $X^{\top}X$ with $X \sim \mathcal{N}(0, 1/m)^{k\times 2n}_c$. In Conjecture 1, we conjecture that $\mathcal{D}_U$ and $\mathcal{D}_X$ become close in total variation distance when $n = o(\sqrt{m})$. However, precisely how close these two distributions are is crucial to whether anticoncentration translates between the two output probability distributions. In what follows, we refer to anticoncentration in the sense of Eq. (A3) as "standard" anticoncentration, and our definition of anticoncentration as "moment-based."

Ideally, we would be able to prove that statements about moment-based anticoncentration of $P_X(\mathbf{n})$ imply equivalent statements about moment-based anticoncentration of $P_U(\mathbf{n})$. However, under worst-case assumptions, we can only show that moment-based anticoncentration of $P_X(\mathbf{n})$ implies standard anticoncentration of $P_U(\mathbf{n})$. To understand this, let us fix some notation. Let $\mathbb{1}[\cdot]$ be an indicator function which is 1 if the argument is true and 0 if it is false. Let $\mathrm{d}\mu$ be the Lebesgue measure on $\mathbb{C}^{2kn}$ (as we consider $k \times 2n$ complex matrices) and $p_U(A)$, $p_X(A)$ be the respective probabilities of generating A from $\mathcal{D}_U$ and $\mathcal{D}_X$.

Now, let the total-variation distance between $\mathcal{D}_U$ and $\mathcal{D}_X$ be less than δ. Then

$$\Pr_{U\in\mathrm{U}(m)}[P_U(\mathbf{n}) \geqslant \epsilon] = \int p_U(A)\mathbb{1}[P_A(\mathbf{n}) \geqslant \epsilon]\,\mathrm{d}\mu(A) \tag{A8}$$

$$= \int [p_U(A) - p_X(A) + p_X(A)]\mathbb{1}[P_A(\mathbf{n}) \geqslant \epsilon]\,\mathrm{d}\mu(A) \tag{A9}$$

$$= \int [p_U(A) - p_X(A)]\mathbb{1}[P_A(\mathbf{n}) \geqslant \epsilon]\,\mathrm{d}\mu(A) + \int p_X(A)\mathbb{1}[P_A(\mathbf{n}) \geqslant \epsilon]\,\mathrm{d}\mu(A) \tag{A10}$$

$$\geqslant -2\delta + \Pr_{X\in\mathcal{G}^{k\times 2n}}[P_X(\mathbf{n}) \geqslant \epsilon] \tag{A11}$$

In this calculation, we have used the Radon-Nikodym theorem [34] to express the probability measures that define $\mathcal{D}_U$ and $\mathcal{D}_X$ as $p_U(A)\mathrm{d}\mu$ and $p_X(A)\mathrm{d}\mu$, respectively. Therefore

$$\Pr_{U\in\mathrm{U}(m)}\left[P_U(\mathbf{n}) \geqslant \frac{\alpha}{|\Omega_{2n}|}\right] \geqslant \Pr_{X\in\mathcal{G}^{k\times 2n}}\left[P_X(\mathbf{n}) \geqslant \frac{\alpha}{|\Omega_{2n}|}\right] - 2\delta \geqslant (1-\alpha)^2\frac{1}{m_2(k,n)} - 2\delta \tag{A12}$$

The final step follows from the Paley-Zygmund inequality for the approximate distribution. This proves that we can translate statements on anticoncentration as long as 2δ is smaller than $m_2(k,n)^{-1}$, which, as we show in the main text, means $\delta = o(n^{-1/2})$.

With this in mind, we can make the following more precise version of Conjecture 1 such that, if it holds, moment-based weak anticoncentration of the approximate distribution implies standard weak anticoncentration of the exact distribution:

Conjecture 3 (formal). Let $\mathcal{D}_U$ be the distribution of the symmetric product $U^\top_{1_k,\mathbf{n}}U_{1_k,\mathbf{n}}$ with U unitary and $\mathbf{n}$ some photon-collision-free outcome of a Gaussian boson sampling experiment. Let $\mathcal{D}_X$ be the distribution of the symmetric product $X^\top X$ with $X \sim \mathcal{N}(0, 1/m)_c^{k\times 2n}$. Then, for any k such that $1 \leqslant k \leqslant m$, and for any $\delta > 0$ such that $m \geqslant n^2/\delta$,

$$\mathsf{tvd}(\mathcal{D}_U, \mathcal{D}_X) = O(\delta) \tag{A13}$$

Specifically, if $\delta = o(n^{-1/2})$, then $m \geqslant n^{5/2}$.

The motivation behind the choice of $m \geqslant n^2/\delta$ is based on the equivalent conjecture for Fock boson sampling in Ref. [2]. There, the authors are able to prove the equivalent result for $m \geqslant n^{5+\epsilon}/\delta$ (for arbitrarily small, constant ϵ), but they suspect that the result can be pushed further to $m \geqslant n^2/\delta$. We note that this choice makes our formal conjecture slightly stronger than the equivalent formal conjecture in Ref. [11].

As we have shown, in order to translate our results on moment-based weak anticoncentration from the approximate to the true distribution in the worst case, we require $\delta = o(n^{-1/2})$. Therefore, in order to translate statements about anticoncentration, the formal version of our conjecture requires $m \geqslant n^{5/2}$.

However, it is worth noting that we do not believe that this worst-case scenario truly reflects the way in which $P_X(\mathbf{n})$ approaches $P_U(\mathbf{n})$, i.e., where all of the error is concentrated on a single probability. In general, the intuition is that, if hiding holds, then it is more likely that the errors are more evenly distributed among all of the exponentially many output probabilities. Using this intuition, each individual probability only receives an error of approximately $\delta/|\Omega_{2n}|$ (where we are specifying to the photon-collision free sample space Ω_{2n}). If this is true, then we can show that moment-based weak anticoncentration of $P_X(\mathbf{n})$ does actually imply the same for $P_U(\mathbf{n})$. Specifically, say that $P_U(\mathbf{n}) \approx P_X(\mathbf{n}) \pm \delta/|\Omega_{2n}| \approx P_X(\mathbf{n}) \pm \delta\mathbb{E}[P_X(\mathbf{n})]$ (as per Appendix B). Then

$$\frac{\mathbb{E}[P_U(\mathbf{n})^2]}{(\mathbb{E}[P_U(\mathbf{n})])^2} \approx \frac{\mathbb{E}[(P_X(\mathbf{n}) \pm \delta\mathbb{E}[P_X(\mathbf{n})])^2]}{(\mathbb{E}[P_X(\mathbf{n}) \pm \delta\mathbb{E}[P_X(\mathbf{n})]])^2} \tag{A14}$$

$$= \frac{\mathbb{E}\left[P_X(\mathbf{n})^2\right] \pm 2\delta\mathbb{E}[P_X(\mathbf{n})]^2 + \delta^2\mathbb{E}[P_X(\mathbf{n})]^2}{(1\pm\delta)^2\mathbb{E}[P_X(\mathbf{n})]^2} \tag{A15}$$

$$\approx \frac{1}{(1\pm\delta)^2}\frac{\mathbb{E}[P_X(\mathbf{n})^2]}{(\mathbb{E}[P_X(\mathbf{n})])^2} + \frac{\pm 2\delta + \delta^2}{(1\pm\delta)^2} \tag{A16}$$

$$= \frac{1}{(1\pm\delta)^2}\frac{\mathbb{E}[P_X(\mathbf{n})^2]}{(\mathbb{E}[P_X(\mathbf{n})])^2} + 1 - \frac{1}{(1\pm\delta)^2} \tag{A17}$$

$$\leqslant \frac{1}{(1-\delta)^2}\frac{\mathbb{E}[P_X(\mathbf{n})^2]}{(\mathbb{E}[P_X(\mathbf{n})])^2} + 1 \tag{A18}$$

In our case, where the normalized second moment of $P_X(\mathbf{n})$ scales at least polynomially in n, and δ scales inverse polynomially in n, weak anticoncentration or lack of anticoncentration of $P_X(\mathbf{n})$ in terms of the normalized second moment adequately translates to $P_U(\mathbf{n})$ as well. Note that, in this case, we are assuming that δ, which is the total variation distance between the distributions of matrices, extends to a bound on the total variation distance between the probabilities themselves. This intuitively arises from the fact that any map from the distribution of the matrices to probabilities must be bounded, meaning we can translate the total variation distance from one to the other (however, formalizing this would require dealing with some subtleties induced by the fact that the hafnian of a product of Gaussians is not technically bounded, but any large hafnians only arise with extremely small probabilities).

APPENDIX B: APPROXIMATE HIDING AND ASYMPTOTICS OF THE FIRST MOMENT

In this Appendix, we discuss more thoroughly the connection between hiding, the relevant sample space, and the first moment of squared hafnians of generalized circular orthogonal ensemble (COE) matrices.

In the main text (and elucidated upon in Appendix A), we introduce the normalized average outcome-collision probability as a measure of anticoncentration. Fixing the output state to have $2n$ photons, we write this as $|\Omega_{2n}|\mathbb{E}_{U\in U(m)}[\sum_{\mathbf{n}\in\Omega_{2n}} P_U(\mathbf{n})^2]$, where Ω_{2n} is the space of photon-collision-free outcomes with $2n$ photons in m modes, and its size, which we write as $|\Omega_{2n}|$, is simply $\binom{m}{2n}$. We here work specifically with the photon-collision-free sample space because, in order for hiding to hold, photon collisions have to be negligible (a non-negligible likelihood of repeated columns in $U^\top_{1_k,\mathbf{n}}U_{1_k,\mathbf{n}}$ would prevent this distribution from being well approximated by $X^\top X$ with X Gaussian). And, indeed, when $n = o(\sqrt{m})$, it is easy to see that the size of the full sample space of $2n$ photons in m modes,

$$\binom{m+2n-1}{2n},$$

approaches $|\Omega_{2n}| = \binom{m}{2n}$ when $n \gg 1$. In particular,

$$\frac{m^{2n}}{(2n)!} \leqslant \frac{(m+2n-1)!}{(m-1)!(2n)!} \leqslant \frac{(m+2n-1)^{2n}}{(2n)!} \tag{B1}$$

and

$$\frac{\binom{m}{2n}}{\binom{m+2n-1}{2n}} \sim \frac{\frac{m^{2n}}{(2n)!}}{\frac{(m+2n-1)^{2n}}{(2n)!}} = \frac{1}{\left(1+\frac{2n-1}{m}\right)^{2n}} \xrightarrow{n\gg 1} 1, \tag{B2}$$

where, in the last step, we are using that $(2n-1)/m = o(1/n)$, which means that the limit of the expression is simply unity [were $(2n-1)/m = \Theta(1/n)$, this limit would approach a constant depending on the constant of proportionality, and it would vanish if $(2n-1)/m = \omega(1/n)$]. Thus, Ω_{2n} is the dominant contribution to the full sample space. We also note that the above calculation is merely a simple subcase of the general bosonic birthday paradox presented in Ref. [20].

We proceed to then replace $|\Omega_{2n}|$ with the expected value of the outcome probabilities, $\mathbb{E}_U[P_U(\mathbf{n})]$, that is, the first moment over input unitaries of a specific outcome. This holds assuming that the hiding property in Conjecture 1 holds. Roughly,

hiding ensures that we do not preference any individual outcome, meaning we can replace the expected value over all probabilities with that over unitaries for a single probability. By linearity of expectation (and the fact that probabilities sum to unity), this expectation over unitaries should simply be the inverse of the size of the sample space of photon-collision-free outcomes. Finally, Conjecture 1 also gives us an approximate equality between $\mathbb{E}_U[P_U(\mathbf{n})]$ and M_1, the first moment of the squared hafnian of generalized COE matrices (properly rescaled to contain the correct prefactors).

We therefore now show that our calculation of the first moment in the hiding regime is consistent with the above discussion in the sense that $\mathbb{E}_U[P_U(\mathbf{n}|\sum_i \mathbf{n}_i = 2n)] = \mathbb{E}_U[P_U(\mathbf{n})]/P(2n)$ is asymptotically equal to $|\Omega_{2n}|^{-1} = \binom{m}{2n}^{-1}$ assuming Conjecture 1. Here, $P(2n)$ is the probability that our output is in the $2n$-photon sector (i.e., the probability that Ω_{2n} is the proper sample space to consider in the first place).

Recall our input state has the first k of m modes prepared in the single-mode squeezed vacuum state with identical squeezing parameter r, and the remaining $m-k$ modes are prepared in the vacuum state. The probability of an outcome $\mathbf{n}$ is given by Eq. (1):

$$P_U(\mathbf{n}) = \frac{\tanh^{2n} r}{\cosh^k r}\left|\mathrm{Haf}\left(U^\top_{1_k,\mathbf{n}} U_{1_k,\mathbf{n}}\right)\right|^2, \tag{B3}$$

where $U_{1_k,\mathbf{n}}$ is the submatrix of U given by the first k rows and the columns dictated by where $\mathbf{n}$ is nonzero. Define $\tilde{U}_{1_k,\mathbf{n}} := mU_{1_k,\mathbf{n}}$. Using multiplicativity of the Hafnian, one finds

$$P_U(\mathbf{n}) = \frac{\tanh^{2n} r}{\cosh^k r}\frac{1}{m^{2n}}\left|\mathrm{Haf}\left(\tilde{U}_{1_k,\mathbf{n}} \tilde{U}^\top_{1_k,\mathbf{n}}\right)\right|^2 \tag{B4}$$

Assuming Conjecture 1, then approximately $U^\top_{1_k,\mathbf{n}} U_{1_k,\mathbf{n}} \approx X^\top X$ (where this is an approximation of distributions), where $X \sim \mathcal{N}(0, 1/m)^{k\times 2n}_c$, which means $\tilde{U}^\top_{1_k,\mathbf{n}} \tilde{U}_{1_k,\mathbf{n}} \approx X^\top X$, but now $X \sim \mathcal{N}(0,1)^{k\times 2n}_c$. Then, by Conjecture 1 and Theorem 2, we find

$$\begin{aligned}
&\mathop{\mathbb{E}}_{U\in \mathrm{U}(m)}\left[\left|\mathrm{Haf}\left(\tilde{U}^\top_{1_k,\mathbf{n}} \tilde{U}_{1_k,\mathbf{n}}\right)\right|^2\right] \\
&\quad\approx \mathop{\mathbb{E}}_{X\in\mathcal{G}^{k\times 2n}}[|\mathrm{Haf}(X^\top X)|^2] \\
&\quad= \frac{(2n)!}{2^n n!}\frac{(k+2n-2)!!}{(k-2)!!},
\end{aligned} \tag{B5}$$

where the first part of the equation is not an equality precisely because the hiding in Conjecture 1 is not exact. This implies that

$$\mathbb{E}_U[P_U(\mathbf{n})] \approx \frac{\tanh^{2n} r}{\cosh^k r}\frac{1}{m^{2n}}\frac{(2n)!}{2^n n!}\frac{(k+2n-2)!!}{(k-2)!!} \tag{B6}$$

Now, a single-mode squeezed vacuum state with squeezing parameter r and phase ϕ has Fock-state expansion given by

$$|\mathrm{SMSV}\rangle = \frac{1}{\sqrt{\cosh r}}\sum_{\ell=0}^{\infty}(-e^{i\phi}\tanh r)^\ell \frac{\sqrt{(2\ell)!}}{2^\ell \ell!}|2\ell\rangle \tag{B7}$$

Therefore, the probability of measuring 2ℓ photons is

$$|\langle 2\ell|\mathrm{SMSV}\rangle|^2 = \frac{\tanh^{2\ell r}}{\cosh r}\frac{(2\ell)!}{(2^\ell \ell!)^2} \tag{B8}$$

Given k independent single-mode squeezed vacuum states, the probability of finding $2n$ total photons is the k-fold convolution of the Fock-basis probability distribution of one single-mode squeezed vacuum state:

$$\begin{aligned}
P(2n) &= \sum_{2\ell_1+\cdots+2\ell_k=2n}\prod_{i=1}^{k}\frac{\tanh^{2\ell_i r}}{\cosh r}\frac{(2\ell_i)!}{(2^{\ell_i}\ell_i!)^2} \\
&= \frac{\tanh^{2n} r}{\cosh^k r}\frac{1}{2^{2n}}\sum_{2\ell_1+\cdots+2\ell_k=2n}\prod_{i=1}^{k}\binom{2\ell_1}{\ell_1}
\end{aligned} \tag{B9}$$

This probability distribution is unchanged if the k independent single-mode squeezed vacuum states are acted upon by a linear-optical unitary before measurement (such a unitary does not change the photon number, only the location of the photons). The combinatorial identity at the core of this k-fold convolution has been calculated before in Refs. [35,36]. Specifically,

$$\sum_{2\ell_1+\cdots+2\ell_k=2n}\prod_{i=1}^{k}\binom{2\ell_1}{\ell_1} = 4^n\binom{n-1+k/2}{n}, \tag{B10}$$

where we note that Eq. (B10) holds even in the case where k is odd using a generalization of the binomial coefficients in terms of the Γ function.

The overall probability of finding $2n$ photons from k independent single-mode squeezed vacuum states, even after the application of a linear optical unitary, is therefore

$$\begin{aligned}
P(2n) &= \frac{\tanh^{2n} r}{\cosh^k r}\frac{1}{2^{2n}}4^n\binom{n-1+k/2}{n} \\
&= \frac{\tanh^{2n} r}{\cosh^k r}\binom{n-1+k/2}{n}
\end{aligned} \tag{B11}$$

We note that this expression, but not the full derivation, is also provided in Ref. [8]. A bit of algebraic manipulation reveals

$$\begin{aligned}
P(2n) &= \frac{\tanh^{2n} r}{\cosh^k r}\binom{n-1+k/2}{n} \\
&= \frac{\tanh^{2n} r}{\cosh^k r}\frac{(2n-1)!!(k+2n-2)!!}{(2n)!(k-2)!!} \\
&= \frac{1}{(2n)!}\frac{\tanh^{2n} r}{\cosh^k r}\mathop{\mathbb{E}}_{X\sim\mathcal{G}^{k\times 2n}}[|\mathrm{Haf}(X^\top X)|^2]
\end{aligned} \tag{B12}$$

According to Eq. (B6), then

$$P(2n) \approx \frac{m^{2n}}{(2n)!}\mathbb{E}_U[P_U(\mathbf{n})], \tag{B13}$$

which finally implies

$$\frac{\mathbb{E}_U[P_U(\mathbf{n})]}{P(2n)} \approx \frac{(2n)!}{m^{2n}} \approx \binom{m}{2n}^{-1} = |\Omega_{2n}|^{-1}, \tag{B14}$$

where the first approximation is due to the fact that hiding is not exact, and the second approximation holds in the photon collision-free regime.

APPENDIX C: ALGEBRAIC DETAILS OF THE SECOND MOMENT—DERIVATION OF EQS. (13) AND (19)

In this Appendix, we generalize the calculation of the first moment of the output probabilities given in the Appendix of Ref. [18] to the second moment. The structure of the derivation is very similar, but the details are more nuanced due to the increased number of copies of X. We begin with some algebraic manipulations:

$$\begin{aligned}\mathop{\mathbb{E}}_{X\sim\mathcal{G}^{k\times 2n}}[|\mathrm{Haf}(X^\top X)|^4] &= \left(\frac{1}{2^n n!}\right)^4 \sum_{\sigma,\tau,\alpha,\beta\in S_{2n}} \mathop{\mathbb{E}}_{X\sim\mathcal{G}^{k\times 2n}} \left[\prod_{j=1}^{n}\left(\sum_{\ell_j=1}^{k} X_{\ell_j\sigma(2j-1)}X_{\ell_j\sigma(2j)}\right)\left(\sum_{o_j=1}^{k} X^*_{o_j\tau(2j-1)}X^*_{o_j\tau(2j)}\right)\right.\\ &\quad\left.\times\left(\sum_{p_j=1}^{k} X_{p_j\alpha(2j-1)}X_{p_j\alpha(2j)}\right)\left(\sum_{q_j=1}^{k} X^*_{q_j\beta(2j-1)}X^*_{q_j\beta(2j)}\right)\right] \qquad \text{(C1)}\\ &= \frac{1}{(2^n n!)^4}\sum_{\sigma,\tau,\alpha,\beta\in S_{2n}}\sum_{\{\ell_i,o_i,p_i,q_i\}_{i=1}^{n}=1}^{k} \mathop{\mathbb{E}}_{X\sim\mathcal{G}^{k\times 2n}}\\ &\quad\times\left[\prod_{j=1}^{n} X_{\ell_j\sigma(2j-1)}X_{\ell_j\sigma(2j)}X^*_{o_j\tau(2j-1)}X^*_{o_j\tau(2j)}X_{p_j\alpha(2j-1)}X_{p_j\alpha(2j)}X^*_{q_j\beta(2j-1)}X^*_{q_j\beta(2j)}\right]. \qquad \text{(C2)}\end{aligned}$$

This first equation simply comes from the definition of the hafnian, and the second from exchanging product and sum and using the linearity of expectation. As in the proof of the first moment, we must properly match the indices of the Gaussian elements; because our Gaussian distribution is complex with mean zero, in order for the expectation value not to vanish, the indices i, j must show up an equal number of times in a conjugated and nonconjugated copy of X. To proceed, first recall that permutations are bijective. Therefore, for all j and any given permutation η, there is a unique value y_j such that $\sigma(2j-1)=\eta(2y_j-1)$ or $\sigma(2j-1)=\eta(2y_j)$. Similarly, there is a unique value y'_j such that $\sigma(2j)=\eta(2y'_j-1)$ or $\sigma(2j)=\eta(2y'_j)$. Using this bijectivity and the independence of matrix elements allows us to separate the single expectation value on the $8n$ matrix elements in Eq. (C2) into a product of $2n$ expectation values of 4 elements:

$$\prod_{j=1}^{n}\mathop{\mathbb{E}}_{X\sim\mathcal{G}^{k\times 2n}}\left[X_{\ell_j\sigma(2j-1)}X_{p_{k_j}\sigma(2j-1)}X^*_{o_{i_j}\sigma(2j-1)}X^*_{q_{m_j}\sigma(2j-1)}\right]\mathop{\mathbb{E}}_{X\sim\mathcal{G}^{k\times 2n}}\left[X_{\ell_j\sigma(2j)}X_{p_{k'_j}\sigma(2j)}X^*_{o_{i'_j}\sigma(2j)}X^*_{q_{m'_j}\sigma(2j)}\right] \qquad \text{(C3)}$$

To explain more thoroughly: we have defined i_j, k_j, m_j to be the indices that map to $\sigma(2j-1)$ under τ,α,β, respectively, in the sense that either $\eta(2y_j-1)=\sigma(2j-1)$ or $\eta(2y_j)=\sigma(2j-1)$ for $\eta\in\{\tau,\alpha,\beta\}$ and $y\in\{i,k,m\}$, respectively. Because two matrix elements are necessarily independent if they do not match on the second index, we can separate all elements with $\sigma(2j-1)$ as the second element into a single expectation value, hence the first term. To get the second term, we repeat this argument where i'_j, k'_j, m'_j are the indices that map to $\sigma(2j)$ under τ,α,β, respectively, in the sense that either $\eta(2y_j-1)=\sigma(2j)$ or $\eta(2y_j)=\sigma(2j)$ for $\eta\in\{\tau,\alpha,\beta\}$ and $y\in\{i,k,m\}$, respectively.

Now consider the first expectation value. For it to be nonvanishing, we must appropriately match the first indices of the matrix elements. We have three options: either all four indices can match, or the indices can be paired off in one of two ways. In the former case, the expectation value yields two given that the elements are complex Gaussian with mean zero and variance one. By the same logic, the latter two cases yield an expectation value of one. In summary:

$$\ell_j = p_{k_j} = o_{i_j} = q_{m_j} \Rightarrow \mathbb{E}\to 2, \qquad \text{(C4)}$$

$$\left(\ell_j \neq p_{k_j}\right)\wedge\left(\ell_j = o_{i_j}\right)\wedge\left(p_{k_j} = q_{m_j}\right)\Rightarrow \mathbb{E}\to 1, \qquad \text{(C5)}$$

$$\left(\ell_j \neq p_{k_j}\right)\wedge\left(\ell_j = q_{m_j}\right)\wedge\left(p_{k_j} = o_{i_j}\right)\Rightarrow \mathbb{E}\to 1 \qquad \text{(C6)}$$

One might naively think that there should be another contribution from matching indices as

$$\left(\ell_j = p_{k_j}\right)\wedge\left(\ell_j \neq q_{m_j}\right)\wedge\left(q_{m_j} = o_{i_j}\right) \qquad \text{(C7)}$$

However, the expectation value in this case actually vanishes; again, we are working with *complex* Gaussian random variables, meaning the indices need to be matched such that there are an equal number of conjugated and nonconjugated indices.

We can write this in one simple expression using Kronecker δs as

$$2\delta_{\ell_j p_{k_j} o_{i_j} q_{m_j}} + \delta_{\ell_j o_{i_j}}\delta_{p_{k_j} q_{m_j}}\left(1-\delta_{\ell_j p_{k_j}}\right) + \delta_{\ell_j q_{m_j}}\delta_{p_{k_j} o_{i_j}}\left(1-\delta_{\ell_j p_{k_j}}\right) = \delta_{\ell_j o_{i_j}}\delta_{p_{k_j} q_{m_j}} + \delta_{\ell_j q_{m_j}}\delta_{p_{k_j} o_{i_j}} \qquad \text{(C8)}$$

That is,

$$\mathop{\mathbb{E}}_{X\sim\mathcal{G}^{k\times 2n}}\left[X_{\ell_j\sigma(2j-1)}X_{p_{k_j}\sigma(2j-1)}X^*_{o_{i_j}\sigma(2j-1)}X^*_{q_{m_j}\sigma(2j-1)}\right] = \delta_{\ell_j o_{i_j}}\delta_{p_{k_j} q_{m_j}} + \delta_{\ell_j q_{m_j}}\delta_{p_{k_j} o_{i_j}}, \qquad \text{(C9)}$$

which is essentially an application of Isserlis' or Wick's theorem. It is straightforward to derive that we may rewrite each of o_{i_j}, p_{k_j}, q_{m_j} in terms of j (a similar example of this in the context of the first moment is worked out explicitly in the Appendix of Ref. [18]), giving

$$\delta_{\ell_j o_{i_j}}\delta_{p_{k_j}q_{m_j}} + \delta_{\ell_j q_{m_j}}\delta_{p_{k_j}o_{i_j}} = \delta_{\ell_j o_{\lceil \frac{\tau^{-1}(\sigma(2j-1))}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha^{-1}(\sigma(2j-1))}{2}\rceil}q_{\lceil \frac{\beta^{-1}(\sigma(2j-1))}{2}\rceil}} + \delta_{\ell_j q_{\lceil \frac{\beta^{-1}(\sigma(2j-1))}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha^{-1}(\sigma(2j-1))}{2}\rceil}o_{\lceil \frac{\tau^{-1}(\sigma(2j-1))}{2}\rceil}}. \tag{C10}$$

Thus,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{X\sim\mathcal{G}^{k\times 2n}}\left[X_{\ell_j\sigma(2j-1)}X_{p_{k_j}\sigma(2j-1)}X^*_{o_{i_j}\sigma(2j-1)}X^*_{q_{m_j}\sigma(2j-1)}\right]\mathop{\mathbb{E}}_{X\sim\mathcal{G}^{k\times 2n}}\left[X_{\ell_j\sigma(2j)}X_{p_{k'_j}\sigma(2j)}X^*_{o_{i'_j}\sigma(2j)}X^*_{q_{m'_j}\sigma(2j)}\right]\\
&\quad=\left(\delta_{\ell_j o_{\lceil \frac{\tau^{-1}(\sigma(2j-1))}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha^{-1}(\sigma(2j-1))}{2}\rceil}q_{\lceil \frac{\beta^{-1}(\sigma(2j-1))}{2}\rceil}} + \delta_{\ell_j q_{\lceil \frac{\beta^{-1}(\sigma(2j-1))}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha^{-1}(\sigma(2j-1))}{2}\rceil}o_{\lceil \frac{\tau^{-1}(\sigma(2j-1))}{2}\rceil}}\right)\\
&\quad\times\left(\delta_{\ell_j o_{\lceil \frac{\tau^{-1}(\sigma(2j))}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha^{-1}(\sigma(2j))}{2}\rceil}q_{\lceil \frac{\beta^{-1}(\sigma(2j))}{2}\rceil}} + \delta_{\ell_j q_{\lceil \frac{\beta^{-1}(\sigma(2j))}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha^{-1}(\sigma(2j))}{2}\rceil}o_{\lceil \frac{\tau^{-1}(\sigma(2j))}{2}\rceil}}\right).
\end{aligned} \tag{C11}$$

Therefore,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{X\sim\mathcal{G}^{k\times 2n}}[|\mathrm{Haf}(X^\top X)|^4]\\
&=\left(\frac{1}{2^n n!}\right)^4\sum_{\sigma,\tau,\alpha,\beta\in S_{2n}}\sum_{\{\ell_i,o_i,p_i,q_i\}_{i=1}^n=1}^{k}\left[\prod_{j=1}^{n}\left(\delta_{\ell_j o_{\lceil \frac{\tau^{-1}(\sigma(2j-1))}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha^{-1}(\sigma(2j-1))}{2}\rceil}q_{\lceil \frac{\beta^{-1}(\sigma(2j-1))}{2}\rceil}} + \delta_{\ell_j q_{\lceil \frac{\beta^{-1}(\sigma(2j-1))}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha^{-1}(\sigma(2j-1))}{2}\rceil}o_{\lceil \frac{\tau^{-1}(\sigma(2j-1))}{2}\rceil}}\right)\right.\\
&\quad\left.\times\left(\delta_{\ell_j o_{\lceil \frac{\tau^{-1}(\sigma(2j))}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha^{-1}(\sigma(2j))}{2}\rceil}q_{\lceil \frac{\beta^{-1}(\sigma(2j))}{2}\rceil}} + \delta_{\ell_j q_{\lceil \frac{\beta^{-1}(\sigma(2j))}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha^{-1}(\sigma(2j))}{2}\rceil}o_{\lceil \frac{\tau^{-1}(\sigma(2j))}{2}\rceil}}\right)\right].
\end{aligned} \tag{C12}$$

We can reparametrize our sums over the permutations by performing a change of variables $(\eta^{-1}\circ\sigma)\to\eta$ for $\eta\in\{\tau,\alpha,\beta\}$. This yields

$$\begin{aligned}
\mathop{\mathbb{E}}_{X\sim\mathcal{G}^{k\times 2n}}[|\mathrm{Haf}(X^\top X)|^4] &= \left(\frac{1}{2^n n!}\right)^4(2n)!\sum_{\tau,\alpha,\beta\in S_{2n}}\sum_{\{\ell_i,o_i,p_i,q_i\}_{i=1}^n=1}^{k}\left[\prod_{j=1}^{n}\left(\delta_{\ell_j o_{\lceil \frac{\tau(2j-1)}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha(2j-1)}{2}\rceil}q_{\lceil \frac{\beta(2j-1)}{2}\rceil}}\right.\right.\\
&\quad\left.\left.+\delta_{\ell_j q_{\lceil \frac{\beta(2j-1)}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha(2j-1)}{2}\rceil}o_{\lceil \frac{\tau(2j-1)}{2}\rceil}}\right)\left(\delta_{\ell_j o_{\lceil \frac{\tau(2j)}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha(2j)}{2}\rceil}q_{\lceil \frac{\beta(2j)}{2}\rceil}} + \delta_{\ell_j q_{\lceil \frac{\beta(2j)}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha(2j)}{2}\rceil}o_{\lceil \frac{\tau(2j)}{2}\rceil}}\right)\right].
\end{aligned} \tag{C13}$$

Expanding the product and summing over ℓ_j yields

$$\begin{aligned}
\mathop{\mathbb{E}}_{X\sim\mathcal{G}^{k\times 2n}}[|\mathrm{Haf}(X^\top X)|^4] &= \left(\frac{1}{2^n n!}\right)^4(2n)!\sum_{\tau,\alpha,\beta\in S_{2n}}\sum_{\{o_i,p_i,q_i\}_{i=1}^n=1}^{k}\left[\prod_{j=1}^{n}\left(\delta_{o_{\lceil \frac{\tau(2j-1)}{2}\rceil}o_{\lceil \frac{\tau(2j)}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha(2j-1)}{2}\rceil}q_{\lceil \frac{\beta(2j-1)}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha(2j)}{2}\rceil}q_{\lceil \frac{\beta(2j)}{2}\rceil}}\right.\right.\\
&\quad+\delta_{o_{\lceil \frac{\tau(2j-1)}{2}\rceil}q_{\lceil \frac{\beta(2j)}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha(2j-1)}{2}\rceil}q_{\lceil \frac{\beta(2j-1)}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha(2j)}{2}\rceil}o_{\lceil \frac{\tau(2j)}{2}\rceil}} + \delta_{q_{\lceil \frac{\beta(2j-1)}{2}\rceil}o_{\lceil \frac{\tau(2j)}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha(2j-1)}{2}\rceil}o_{\lceil \frac{\tau(2j-1)}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha(2j)}{2}\rceil}q_{\lceil \frac{\beta(2j)}{2}\rceil}}\\
&\quad\left.\left.+\delta_{q_{\lceil \frac{\beta(2j-1)}{2}\rceil}q_{\lceil \frac{\beta(2j)}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha(2j-1)}{2}\rceil}o_{\lceil \frac{\tau(2j-1)}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha(2j)}{2}\rceil}o_{\lceil \frac{\tau(2j)}{2}\rceil}}\right)\right].
\end{aligned} \tag{C14}$$

This is Eq. (14) of the main text, which is the starting point of a new graph-theoretic approach.

As discussed in the main text, we use Eq. (C14) to define graphs, examples of which are provided in Figs. 2 and 3(a). Specifically, we let $G_{\tau,\alpha,\beta}(z)$ be a graph on $6n$ vertices, with labels $\{O_i,P_i,Q_i\}_{i=1}^{2n}$, and z an integer from 1 to 4^n. We use the Kronecker δs to define black and red edges. z enumerates the different patterns of black edges, and τ,α,β determine the red edges. Specifically, there is a red edge between O_j and $O_{j'}$ if $\lceil\tau(j)/2\rceil=\lceil\tau(j')/2\rceil$, and similarly for the O and Q vertices using permutations α and β, respectively. However, given a choice of permutations, there are 4^n possible sets of black edges that correspond to the 4^n possible combinations of terms in Eq. (C14). The sets of edges corresponding to each term are listed below:

$$\delta_{o_{\lceil \frac{\tau(2j-1)}{2}\rceil}o_{\lceil \frac{\tau(2j)}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha(2j-1)}{2}\rceil}q_{\lceil \frac{\beta(2j-1)}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha(2j)}{2}\rceil}q_{\lceil \frac{\beta(2j)}{2}\rceil}} \to \{(O_{2j-1},O_{2j}),(P_{2j-1},Q_{2j-1}),(P_{2j},Q_{2j})\}, \tag{C15}$$

$$\delta_{o_{\lceil \frac{\tau(2j-1)}{2}\rceil}q_{\lceil \frac{\beta(2j)}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha(2j-1)}{2}\rceil}q_{\lceil \frac{\beta(2j-1)}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha(2j)}{2}\rceil}o_{\lceil \frac{\tau(2j)}{2}\rceil}} \to \{(O_{2j-1},Q_{2j}),(P_{2j-1},Q_{2j-1}),(O_{2j},P_{2j})\}, \tag{C16}$$

$$\delta_{q_{\lceil \frac{\beta(2j-1)}{2}\rceil}o_{\lceil \frac{\tau(2j)}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha(2j-1)}{2}\rceil}o_{\lceil \frac{\tau(2j-1)}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha(2j)}{2}\rceil}q_{\lceil \frac{\beta(2j)}{2}\rceil}} \to \{(O_{2j},Q_{2j-1}),(P_{2j-1},O_{2j-1}),(P_{2j},Q_{2j})\}, \tag{C17}$$

$$\delta_{q_{\lceil \frac{\beta(2j-1)}{2}\rceil}q_{\lceil \frac{\beta(2j)}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha(2j-1)}{2}\rceil}o_{\lceil \frac{\tau(2j-1)}{2}\rceil}}\delta_{p_{\lceil \frac{\alpha(2j)}{2}\rceil}o_{\lceil \frac{\tau(2j)}{2}\rceil}} \to \{(O_{2j-1},P_{2j-1}),(O_{2j},P_{2j}),(Q_{2j-1},Q_{2j})\} \tag{C18}$$

We refer to these sets of black edges as type-1, type-2, type-3, and type-4, respectively. We take the convention that our graphs have the vertices organized into three rows and $2n$ columns. The first, second, and third rows correspond to type-O, $-P$, and $-Q$ vertices, respectively. The columns are ordered by index i. Using this convention, black edges are constrained to lie within groups of two columns $2i-1$ and $2i$ using one of the four patterns described above. Again, see Figs. 2 and 3(a) for examples [please note that Fig. 3(a) is not fully general, as it only has type-1 and type-4 black edges, but it does show that patterns of black edges can repeat, and it shows how z identifies the patterns of black edges present in the graph].

These graphs are useful in the following way: Evaluating the algebraic expression in Eq. (C14) amounts to computing how many "unconstrained" or "free" indices there are in the sum over $\{o_i, p_i, q_i\}_{i=1}^n$ for every combination of permutations in $S_{2n} \times S_{2n} \times S_{2n}$. Unconstrained or free indices are those that are left over after one accounts for all of the dependencies between indices that are enforced by the Kronecker δs, and each free index yields a factor of k. By construction, the number of such free indices for a given set of permutations exactly maps to the number of connected components $C(G_{\tau,\alpha,\beta}(z))$ of the graph $G_{\tau,\alpha,\beta}(z)$. Thus, this graph contributes $k^{C(G_{\tau,\alpha,\beta}(z))}$ to the sum, and the second moment can be written as

$$M_2(k,n) = \frac{(2n)!}{(2^n n!)^4} \sum_{\tau,\alpha,\beta\in S_{2n}} \sum_{z\in[4^n]} k^{C(G_{\tau,\alpha,\beta}(z))}, \tag{C19}$$

which is Eq. (19) of the main text.

Now, there is a degeneracy where many permutations induce the same final graph. For any fixed set of black edges (i.e., for any fixed z), the graph possesses one of $(2n-1)!!^3$ possible sets of red edges (this is the number of ways of pairing off three sets of $2n$ elements when order does not matter). For each graph G corresponding to some assignment of the red edges, there are $2^n n!$ of each of τ, α, β such that $G_{\tau,\alpha,\beta} = G$, leading to a degeneracy factor of $(2^n n!)^3$.

Therefore, removing the degeneracies induced by different permutations, and defining $\mathbb{G}_n^2(z)$ to be the set of graphs for the zth set of black edges and $\mathbb{G}_n^2 := \bigcup_{z=1}^{4^n} \mathbb{G}_n^2(z)$, we get a final result of

$$M_2(k,n) = (2n-1)!! \sum_{G\in\mathbb{G}_n^2} k^{C(G)} \tag{C20}$$

This is Eq. (20) of the main text.

APPENDIX D: BUILDING THE RECURSION

We now describe precisely how to derive and evaluate the recursion for the second moment. Using the framework developed in this Appendix, we implement the full recursion numerically exactly [24] in both the Julia programming language [23] and *Mathematica* [37]. We show in the next Appendix, Appendix E, that these numerical implementations are efficient in n.

Recall that the recursion is defined by Eq. (28), which we copy here for convenience:

$$g(n,a_{12},a_{13},a_{23}) = \sum_{b_{12},b_{13},b_{23}} c(a_{12},a_{13},a_{23},b_{12},b_{13},b_{23}) \times g(n-1,b_{12},b_{13},b_{23}) \tag{D1}$$

$g(n,a_{12},a_{13},a_{23})$ is a polynomial in k, where the coefficient in front of k^i is the number of graphs of type $\mathbf{a} = (a_{12},a_{13},a_{23})$ that have i connected components, and where a graph of type $\mathbf{a}$ has a_{ij} edges between rows i, j.

We first describe the base case, i.e., $g(1,a_{12},a_{13},a_{23})$ for all valid vectors $\mathbf{a} = (a_{12},a_{13},a_{23})$. We then describe how to handle each of the possible 17 cases that contribute to the recursion that are depicted in Fig. 4, which is copied here as Fig. 11 for convenience

The way that we handle each case is as follows. We consider all graphs of order n such that the leftmost two columns, which, recall, we refer to as $\mathbb{C}_{1,2}$, have red edges that correspond to that case. We then "integrate out" these edges to determine how to write the contribution of that case at order n in terms of the terms at order $n-1$. When we say integrate out, we mean that we collapse any path that goes through $\mathbb{C}_{1,2}$ into a new edge that remains entirely in the graph of order $n-1$ by collapsing together vertices connected by these paths. In doing this, we must account for three main contributions: (1) how many loops are contained solely within $\mathbb{C}_{1,2}$—each of these loops, of course, leads to a factor of k multiplied by the contribution at order $n-1$; (2) what edges are erased when integrating out the case, as well as what edges are created after collapsing the paths into new edges—this tells us what $\mathbf{b}$ at lower order contribute to $\mathbf{a}$ at a higher order; (3) a combinatorial factor accounting for the fact that integrating out $\mathbb{C}_{1,2}$ in multiple graphs at order n could lead to the same graph at order $n-1$, meaning we may need to multiply the contributions at order $n-1$ by something to get the correct final answer. The loop calculation is usually quite simple, but the vectorial and combinatorial calculations require more significant casework.

In the abstract, this is quite complicated, but we explain it more thoroughly through detailed examples as we proceed. We group our analysis of these cases into four categories corresponding to the number of edges, i.e., 0, 2, 4, or 6, that protrude from the cases: (1)–(4), (5)–(12), (13)–(16), and (17), respectively. However, as mentioned, we begin with the base cases, to which we turn now.

1. Base cases for recursion

Here we calculate the base cases for the recursion; that is, we determine all valid $\mathbf{a}$ when $n=1$, construct all graphs with each $\mathbf{a}$, and count their connected components. Recall that the vector $\mathbf{a}$ must satisfy non-negativity, pairwise sums being even, and pairwise sums being at most $2n$; should any one of these conditions not be met, then $g(n,\mathbf{a}) = g(n,a_{12},a_{13},a_{23}) = 0$. For $n=1$, there are five possible options for $\mathbf{a}$: (0,0,0), (2,0,0), (0,2,0), (0,0,2), (1,1,1). It remains then to construct the graphs and count their connected components. This is tedious, but the diagrams are shown in

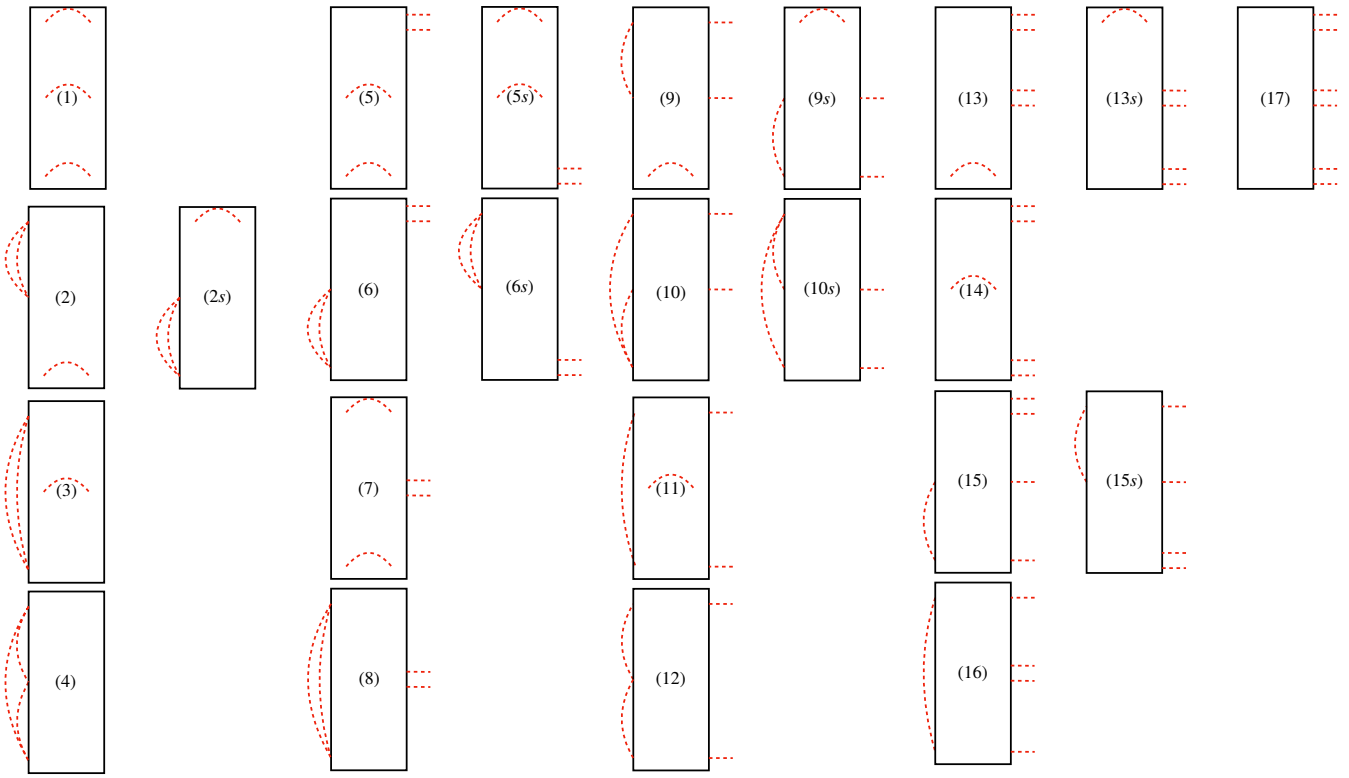


FIG. 11. Copy of Fig. 4. List of 17 cases (up to symmetry) for how the first two columns in a graph of order n can connect into the rest of the graph.

Figs. 12 and 13, and the final results are

$$g(1,0,0,0) = 2k^2 + 2k, \tag{D2}$$

$$g(1,2,0,0) = k^3 + 3k^2 + 4k, \tag{D3}$$

$$g(1,0,0,2) = k^3 + 3k^2 + 4k, \tag{D4}$$

$$g(1,0,2,0) = 2k^2 + 6k, \tag{D5}$$

$$g(1,1,1,1) = 2k^3 + 14k^2 + 16k \tag{D6}$$

This completes the base cases, and we now move on to the recursion.

2. Cases (1)–(4)

We now handle cases (1)–(4). There are no protruding edges, meaning many of the contributions are easy to derive because these cases are "independent" of from the lower-order graph consisting of the final $n-1$ pairs columns. Therefore, when we integrate out $\mathbb{C}_{1,2}$, none of the paths affect the graph at lower order, meaning it is much simpler to calculate their contribution.

In fact, it is simple to see that the evaluation of the loops mimics exactly the calculation of the base cases:

$$\text{Loop } (1) \to 2k^2 + 2k, \tag{D7}$$

$$\text{Loop } (2) \to k^3 + 3k^2 + 4k, \tag{D8}$$

$$\text{Loop } (2s) \to k^3 + 3k^2 + 4k, \tag{D9}$$

$$\text{Loop } (3) \to 2k^2 + 6k, \tag{D10}$$

$$\text{Loop } (4) \to 2k^3 + 14k^2 + 16k \tag{D11}$$

Next, examining the diagrams for each case, one can derive simple relationships between $\mathbf{a}$ and $\mathbf{b}$ that yield a nontrivial contribution in Eq. (D1):

$$\text{Vector } (1) \to (b_{12}, b_{13}, b_{23}) = (a_{12}, a_{13}, a_{23}), \tag{D12}$$

$$\text{Vector } (2) \to (b_{12}, b_{13}, b_{23}) = (a_{12} - 2, a_{13}, a_{23}), \tag{D13}$$

$$\text{Vector } (2s) \to (b_{12}, b_{13}, b_{23}) = (a_{12}, a_{13}, a_{23} - 2), \tag{D14}$$

$$\text{Vector } (3) \to (b_{12}, b_{13}, b_{23}) = (a_{12}, a_{13} - 2, a_{23}), \tag{D15}$$

$$\text{Vector } (4) \to (b_{12}, b_{13}, b_{23}) = (a_{12} - 1, a_{13} - 1, a_{23} - 1) \tag{D16}$$

These can be understood by looking at the diagram for each case and observing what kind of edges are eliminated when collapsing all of the paths that pass through the vertices in $\mathbb{C}_{1,2}$.

Finally, there are no combinatorial contributions because there are no protruding edges that have to be connected to the existing graph. That is, any graph that comes from integrating out one of these cases arises uniquely.

Therefore, we can easily combine everything to get the contributions to the recursion from each of these cases:

$$g(n, a_{12}, a_{13}, a_{23})_{\text{case}(1)} = (2k^2 + 2k)g(n-1, a_{12}, a_{13}, a_{23}), \tag{D17}$$

$$g(n, a_{12}, a_{13}, a_{23})_{\text{case}(2)} = (k^3 + 3k^2 + 4k)g(n-1, a_{12} - 2, a_{13}, a_{23}), \tag{D18}$$

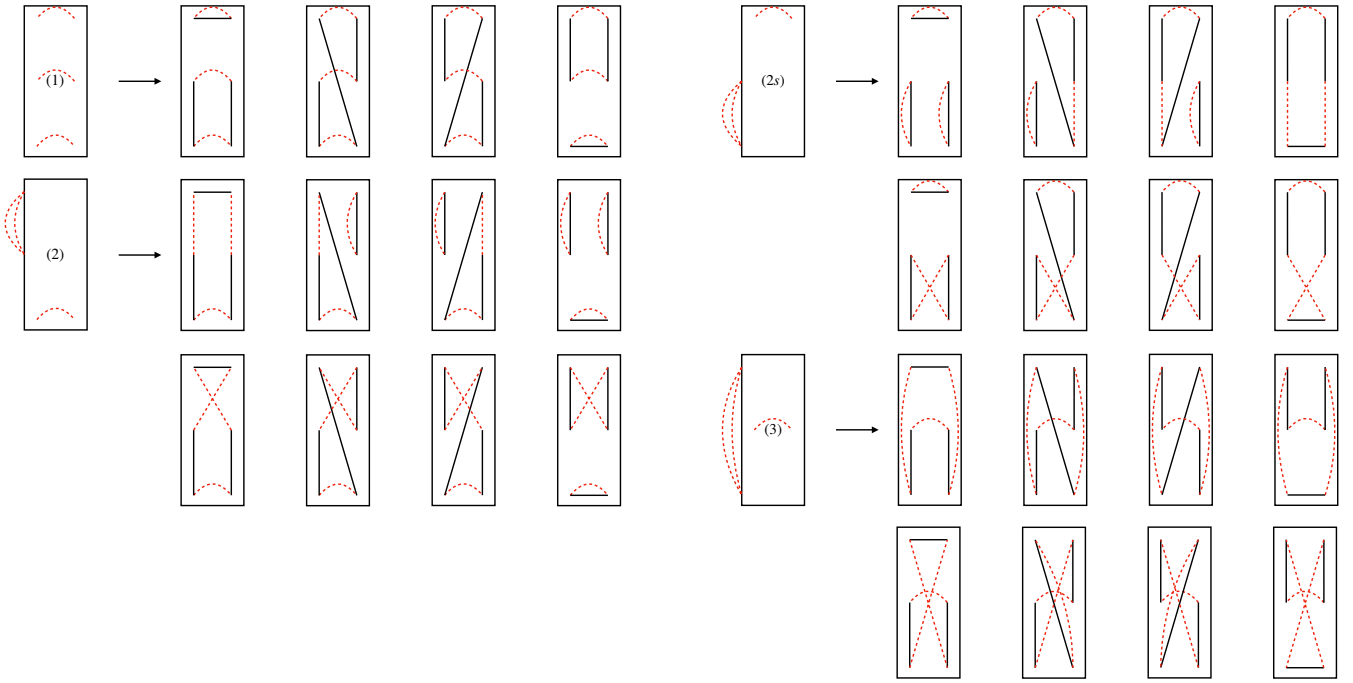


FIG. 12. Base cases corresponding to (1), (2), (2s), and (3). Counting the connected components of the graphs in each case yields contributions of $2k^2+2k$, k^3+3k^2+4k, k^3+3k^2+4k, and $2k^2+6k$, respectively.

$$g(n,a_{12},a_{13},a_{23})_{\text{case}(2s)} = (k^3+3k^2+4k)g(n-1,a_{12},a_{13},a_{23}-2), \quad \text{(D19)}$$

$$g(n,a_{12},a_{13},a_{23})_{\text{case}(3)} = (2k^2+6k)g(n-1,a_{12},a_{13}-2,a_{23}), \quad \text{(D20)}$$

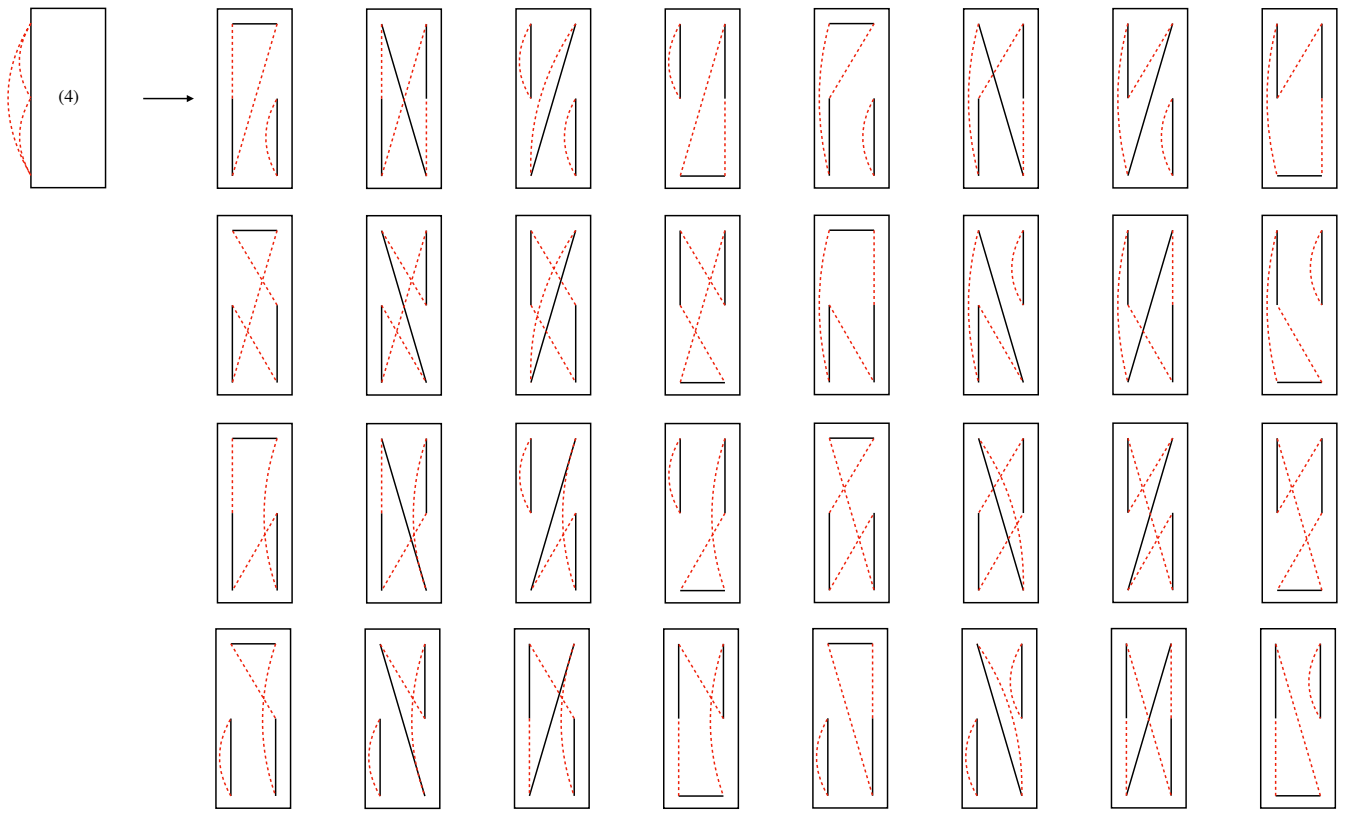


FIG. 13. Base case corresponding to (4). Counting the connected components of the graphs in each case yields $2k^3+14k^2+16k$.

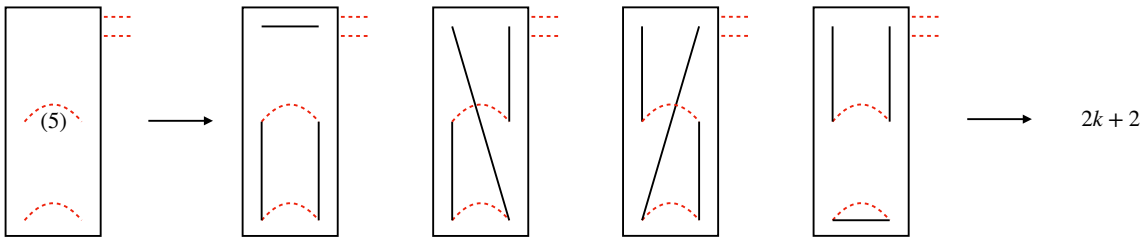


FIG. 14. Loop contribution for case (5).

$$g(n, a_{12}, a_{13}, a_{23})_{\text{case(4)}} = (2k^3 + 14k^2 + 16k)g(n-1, a_{12}-1, a_{13}-1, a_{23}-1) \tag{D21}$$

Recall that the notation $g(n, a_{12}, a_{13}, a_{23})_{\text{case}(i)}$, refers to the contribution to $g(n, a_{12}, a_{13}, a_{23})$ from graphs where the vertices in $\mathbb{C}_{1,2}$ and their corresponding red edges fall into case (i). That is, $g(n, \boldsymbol{a}) = \sum_{i\in\text{cases}} g(n, \boldsymbol{a})_{\text{case}(i)}$, which is just Eq. (29).

3. Cases (5)–(12)

We now tackle cases (5)–(12), which have two edges that protrude and attach to the rest of the graph. Because of these two protruding edges, we have to carefully derive all three of the loop, vectorial, and combinatorial contributions. To understand each of these contributions, we carefully walk through case (5), which contains two edges protruding from the first row, and then argue the behavior of the other cases by analogy.

Vectorial. We start with the vectorial contributions, as understanding them allows us to more easily explain and derive the loop and combinatorial contributions. We take an existing graph of order n where $\mathbb{C}_{1,2}$ and the respective red edges match case (5). We then count how the numbers of edges of each type change after collapsing all of the paths that pass through the vertices in $\mathbb{C}_{1,2}$ into edges that lie within the other $2(n-1)$ columns.

It is crucial to observe the following extremely important fact for *all cases* (5)–(12): the two protruding edges are always part of the same path that goes through $\mathbb{C}_{1,2}$, regardless of which of the four types of black edges are present between the vertices in $\mathbb{C}_{1,2}$. Therefore, when $\mathbb{C}_{1,2}$ is integrated out in graphs that match these cases, the edge that is created in the lower-order graph is simply given by the two rows upon which those protruding edges are incident. That is, if the protruding edges connected to rows i and j, then, after integrating, an edge of type ij is created.

Now, there are, of course, six types of edges that can be created by collapsing a path: 11, 22, 33, 12, 13, and 23. However, it is somewhat convenient to actually describe nine possible edges, 11, 22, 33, 12, 13, 23, 21, 31, and 32. The last three are equivalent to 12, 13, and 23 edges, respectively, but we order the edges in this way to account for the two possible ways that the protruding edges can connect into the graph (that is, *which* edge connects to row i or j, for example). Note that this separation is extraneous for certain cases, i.e., those with two edges protruding from the *same* row, but it is useful when considering cases with edges protruding from different rows.

To determine the vector contribution for a graph of order n with a_{12}, a_{13}, and a_{23} edges, we consider what edges b_{12}, b_{13}, and b_{23} on the graph of order $n-1$ remain after integrating out $\mathbb{C}_{1,2}$. Case (5) has two protruding edges coming from the first row, and then additional red edges of type 22 and 33. These 22 and 33 edges do not change the 12, 13, or 23 edge counts. Therefore, the only changes come from the collapse of the path associated with the two protruding edges from row 1.

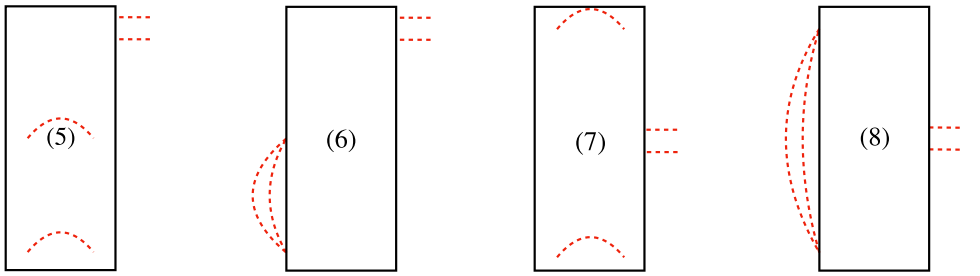
Let us say that these two protruding edges are originally incident on rows 2 and 3. In this example, this means that when integrating out $\mathbb{C}_{1,2}$, we lose one edge of type 12 and one of type 13, but we *create* one of type 23. Therefore, we must have that $b_{12} = a_{12} - 1$, $b_{13} = a_{13} - 1$, and $b_{23} = a_{23} + 1$. Or, if we define $\Delta_{ij} := b_{ij} - a_{ij}$, then $(\Delta_{12}, \Delta_{13}, \Delta_{23}) = (-1, -1, +1)$. We then consider all possible vertices that these two protruding edges could have been connected to in the remainder of the graph, and that defines all possible $g(n-1, b_{12}, b_{13}, b_{23})$ that can contribute to $g(n, a_{12}, a_{13}, a_{23})_{\text{case(5)}}$. This completes our study of the vectorial contribution of case (5).

Combinatorial. We must also consider some combinatorial factors $\mathcal{C}$. The combinatorial factors are really just a shorthand for determining how many times a contribution $g(n-1, b_{12}, b_{13}, b_{23})$ arises when integrating out a given case, here case (5), from all the relevant graphs of order n. This is because different graphs at order n, when appropriately collapsed, can lead to the same graph at order $n-1$. The combinatorial factor is just a way of encoding this information.

Say that we are again considering an example where the original protruding edges attach to vertices in rows 2 and 3. Then an edge of type 23 is created. But if we look from the perspective of the lower-order graph, *any* of the 23 edges could have been the one that was generated—that is, for some graph of order n with case (5) integrated out, a different 23 edge that is present is the one generated. Therefore, when we sum up all the contribution from integrating out case (5) over all relevant graphs of order n, we get a factor of b_{23}. Note also that, as we derived above, $b_{23} = a_{23} + 1$. Also note that, were we looking at protruding edges attached to the same row, we would get an additional factor of 2 due to the ambiguity of which edge attaches to which endpoint.

Loop. Finally, we consider the loop contribution. The calculation for case (5) is a relatively straightforward diagrammatic proof, which is detailed in Fig. 14. In short, we draw all possible diagrams consistent with case (5) and count up the loops that are induced. There are only four cases, as the red edges are essentially fixed and there are four possible sets of black edges. The result is a factor $2k + 2$. That is, there are

TABLE I. Information for vectorial and combinatorial contributions to cases (5)–(8). Observe that there is a symmetry when the endpoints of the protruding edges are ij and ji. Also observe that, when the endpoints are the same, i.e., ii, there is an extra factor of two in the combinatorial term because of the ambiguity between how the protruding edges originally attach.



Protruding endpoints	$\mathcal{C}$	Δ_{12}	Δ_{13}	Δ_{23}	Δ_{12}	Δ_{13}	Δ_{23}	Δ_{12}	Δ_{13}	Δ_{23}	Δ_{12}	Δ_{13}	Δ_{23}
11	$2b_{11}$	0	0	0	0	0	−2	−2	0	0	−2	−2	0
12	b_{12}	0	0	0	0	0	−2	0	0	0	0	−2	0
13	b_{13}	0	0	0	0	0	−2	−1	+1	−1	−1	−1	−1
21	b_{12}	0	0	0	0	0	−2	0	0	0	0	−2	0
22	$2b_{22}$	−2	0	0	−2	0	−2	0	0	0	0	−2	0
23	b_{23}	−1	−1	+1	−1	−1	−1	0	0	0	0	−2	0
31	b_{13}	0	0	0	0	0	−2	−1	+1	−1	−1	−1	−1
32	b_{23}	−1	−1	+1	−1	−1	−1	0	0	0	0	−2	0
33	$2b_{33}$	0	−2	0	0	−2	−2	0	0	−2	0	−2	−2

two sets of black edges that lead to an internal loop, leading to an extra factor of k, and there are two sets of black edges where the protruding edges snake through all vertices in $\mathbb{C}_{1,2}$ such that collapsing them just leads to a graph of order $n-1$ without any extra loop factors

So, putting together three factors, we have that a full contribution from case (5) is

$$\begin{aligned}
&g(n, a_{12}, a_{13}, a_{23})_{\text{case(5)}} \\
&= (2k+2)[(2b_{11}+2b_{12}+2b_{13})g(n-1, a_{12}, a_{13}, a_{23}) \\
&\quad + 2b_{22}g(n-1, a_{12}-2, a_{13}, a_{23}) \\
&\quad + 2b_{23}g(n-1, a_{12}-1, a_{13}-1, a_{23}+1) \\
&\quad + 2b_{33}g(n-1, a_{12}, a_{13}-2, a_{23})] \\
&= (2k+2)[(2(n-1)+a_{12}+a_{13})g(n-1, a_{12}, a_{13}, a_{23}) \\
&\quad + (2(n-1)-(a_{12}-2)-a_{23})g(n-1, a_{12}-2, a_{13}, a_{23}) \\
&\quad + 2(a_{23}+1)g(n-1, a_{12}-1, a_{13}-1, a_{23}+1) \\
&\quad + (2(n-1)-(a_{13}-2)-a_{23})g(n-1, a_{12}, a_{13}-2, a_{23})].
\end{aligned} \tag{D22}$$

This includes the loop, combinatorial, and vectorial factors. We also note that, should any of the combinatorial factors actually be negative, they should be set to zero, as that indicates that the graph that is constructed at lower order when integrating out the given case does not really exist (this is also handled by the vector input to g being negative—that is, one of the edge counts b_{12}, b_{13}, b_{23} is negative). One can get the contribution from case (5s) by simply mapping $1 \leftrightarrow 3$.

We list the combinatorial and vectorial contributions for cases (5)–(8) in Table I and cases (9)–(12) in Table II (the main difference in the latter cases is that there is no longer a symmetry between red edges attaching to vertices ij and ji because, by convention, we attach the top protruding edge to the vertex in row i and the bottom protruding edge to the vertex in row j, which gives us different types of new edges, generically). The first column of these tables gives what kind of edge is created at order $n-1$. The second column tells us the combinatorial factor. The next four multicolumns give the vector information for each of the cases. Note that we do not give the symmetric cases, as they can be obtained by simply mapping $1 \leftrightarrow 3$.

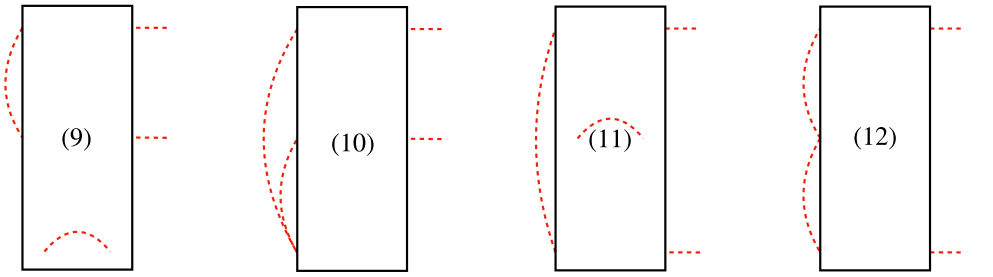
We also provide the loop contributions for cases (5)–(12) in Table III. These are derived in an analogous way to the diagrammatic approach in Fig. 14, but there are many more graphs to consider. Therefore, using all of this information, we can derive an equivalent version of Eq. (D22) for each case up to (12) (including the symmetric ones), accounting for all of their contributions.

4. Cases (13)–(16)

We now move on to more complicated cases that have four protruding edges. The vectorial contribution is more difficult to calculate, as we must account for $3^4 = 81$ possibilities for how the protruding edges attach to the lower-order graph. Furthermore, there is more interaction between the vectorial, combinatorial, and loop terms. This did not occur in the previous sets of cases because the protruding edges were always part of the same path through the black edges attached to the vertices in $\mathbb{C}_{1,2}$. However, one must now keep track of which protruding edges connect to one another through the vertices in $\mathbb{C}_{1,2}$.

For example, we look at the possibilities for case (13), shown in Fig. 15 By convention, we take the top-left vertex to row a, the top-right vertex to row b, the middle-left vertex to row c, and the middle-right vertex to row d, where $a, b, c, d \in \{1, 2, 3\}$. We see that, when the black edges attached to the vertices in $\mathbb{C}_{1,2}$ are type-1, then the red edges that protrude

TABLE II. Information for vectorial and combinatorial contributions to cases (9)–(12). Observe that there is no longer a symmetry between ij and ji, but the ii cases still have an extra factor of 2 in the combinatorial term because the ambiguity between how the protruding edges originally attach still exists.



Protruding endpoints	$\mathcal{C}$	Δ_{12}	Δ_{13}	Δ_{23}	Δ_{12}	Δ_{13}	Δ_{23}	Δ_{12}	Δ_{13}	Δ_{23}	Δ_{12}	Δ_{13}	Δ_{23}
11	$2b_{11}$	-2	0	0	-1	-1	-1	0	-2	0	-1	-1	-1
12	b_{12}	0	0	0	$+1$	-1	-1	$+1$	-1	-1	0	0	-2
13	b_{13}	-1	$+1$	-1	0	0	-2	0	0	0	-1	$+1$	-1
21	b_{12}	-2	0	0	-1	-1	-1	0	-2	0	-1	-1	-1
22	$2b_{22}$	-2	0	0	-1	-1	-1	-1	-1	-1	-2	0	-2
23	b_{23}	-2	-0	0	-1	-1	-1	-1	-1	$+1$	-2	0	0
31	b_{13}	-2	0	0	-1	-1	-1	0	-2	0	-1	-1	-1
32	b_{23}	-1	-1	$+1$	0	-2	0	0	-2	0	-1	-1	-1
33	$2b_{33}$	-1	-1	-1	0	-2	-2	0	-2	0	-1	-1	-1

from the top row are connected to one another, which means that one generates an edge of type ab when collapsing this path. However, if the black edges associated with $\mathbb{C}_{1,2}$ are type-2, then it is instead ac and bd that are connected. In total, one of the possible types of black edges connect edges ab and cd, and three connect ac and bd. In the case where ab and cd are connected, this means that we generate edges of type ab and cd but we lose edges of type $1a$, $1b$, $2c$, $2d$. When ac and bd are connected, we of course gain edges of type ac and bd, but we still lose edges of type $1a$, $1b$, $2c$, $2d$. We use these observations to build up the vectorial contribution of the graph by summing over all 81 possibilities of $a, b, c, d \in \{1, 2, 3\}$. This is tedious to do by hand, but simple numerically.

TABLE III. Loop contributions for each of the cases (5)–(12). Notice that symmetric versions of cases have the same loop contribution; only their vectorial and combinatorial contributions are different.

Case	Loop contribution
(5)	$2k+2$
(5s)	$2k+2$
(6)	k^2+3k+4
(6s)	k^2+3k+4
(7)	$2k+2$
(8)	$2k+6$
(9)	$2k^2+6k+8$
(9s)	$2k^2+6k+8$
(10)	$2k^2+14k+16$
(10s)	$2k^2+14k+16$
(11)	$4k+12$
(12)	$2k^2+14k+16$

We need also account for the loop and combinatorial factors that associate to each of these vectorial contributions. Luckily, we do not need to consider 81 cases parametrized by a, b, c, d, but we must consider each of the subcases defined by the four possible sets of black edges in connecting the vertices in $\mathbb{C}_{1,2}$. Loop-wise, we simply need to count how many loops are induced. Working from the left to right in Fig. 15, we get 0,0,0,1 loops, respectively, leading to factors of 1, 1, 1, k, respectively. The combinatorial factor is given by

$$2^{\delta_{ab}}2^{\delta_{cd}}\Big[(\delta_{ac}\delta_{bd}+\delta_{ad}\delta_{bc}-\delta_{abcd})2\binom{b_{ab}}{2} + [1-(\delta_{ac}\delta_{bd}+\delta_{ad}\delta_{bc}-\delta_{abcd})]b_{ab}b_{cd}\Big] \tag{D23}$$

in the case where edges ab and cd are connected. If instead ac and bd are connected, we replace each instance of ab and cd with ac and bd, respectively. We then again account for all 81 cases and attach each combinatorial factor and loop factor to its associated vectorial term.

To understand Eq. (D23), consider the following, where we assume we are dealing with type-1 black edges so that we are creating edges ab and cd. We get a factor of two when a and b are the same because they correspond to protruding edges coming from the same row, meaning there is a choice of which edge to connect where. The same holds for c and d. If all four edges connect to the same row, i.e., $a=b=c=d$, then one might naively think we need to add an extra factor of six (to get to a total of 4! possible connections), but this is incorrect, as ab and cd are always paired given their connection through case (13) with black edges of type-1. Now, if $a=c$ and $b=d$ or $a=d$ and $b=c$, then the two edges ab and cd are the same type, meaning we are creating two edges of the same type in the graph of order $n-1$. There are therefore $\binom{b_{ab}}{2}$ choices of

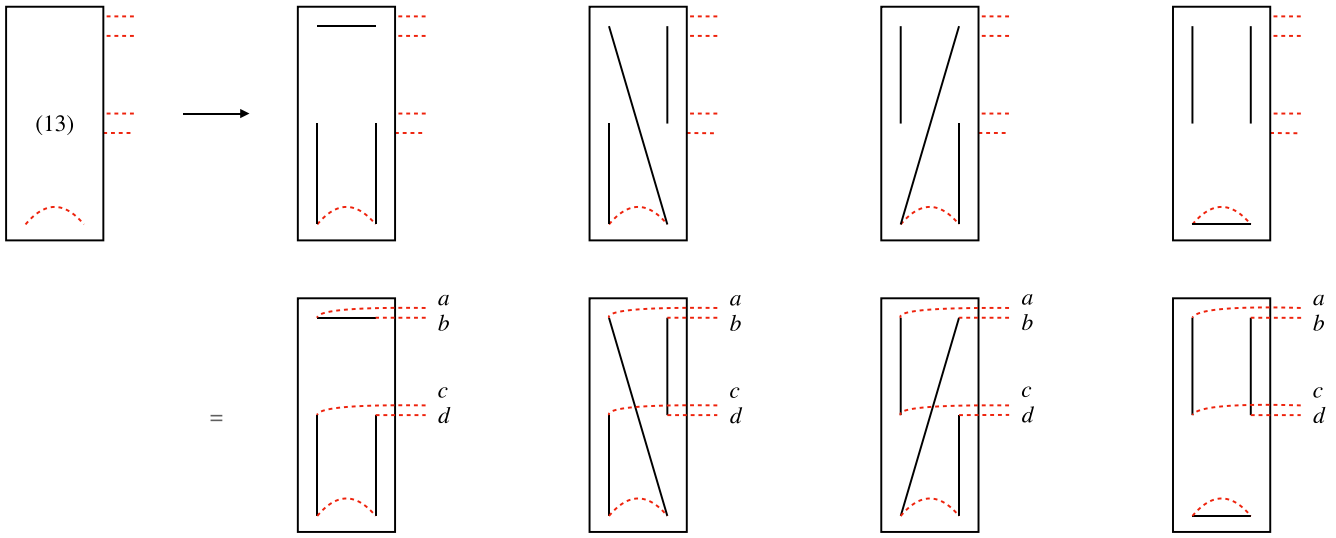


FIG. 15. Evaluation of case (13). By convention, we take the top-left vertex to row a, the top-right vertex to row b, the middle-left vertex to row c, and the middle-right vertex to row d, where $a, b, c, d \in \{1, 2, 3\}$. The types of edges that are created after integrating out the two leftmost columns are determined by the type of the black edges.

which edges these are in the lower-order graph, but we also need an extra factor of two to decide which one the groups of protruding edges each maps to. If ab and cd correspond to different types of edges, then we just get a factor of $b_{ab}b_{cd}$, as we simply need to account for which of these edges are generated through the integration process.

Therefore, we see that cases (13)–(16) raise substantially more complications in their evaluation. In particular, the type of black edges leads to far more interaction between the loop, vectorial, and combinatorial contributions that must be carefully combined in code to achieve the correct recursion. While we have only described case (13) in detail, cases (14)–(16) follow in the exact same manner, although there are more graphs to consider in the cases where two rows have only one protruding edge.

5. Case (17)

Case (17) raises the same issues, although there are only four graphs to consider. However, we have $243 = 3^6$ possible options for how the protruding edges may connect to the graph at lower order (this is true in general, but not all of these are possible when n is small). See Fig. 16 We repeat the convention for cases (13)–(16) by taking the top-left vertex to row a, the top-right vertex to row b, the middle-left vertex to row c, and the middle-right vertex to row d, but we now also take the bottom left to e and the bottom right to f, where $a, b, c, d, e, f \in \{1, 2, 3\}$. Now, for type-1 black edges, we create ab, ce, and df; for type-2, it is af, bd, and ce; for type-3 it is ac, be, and df; and for type-4 it is ac, bd, and ef. We always lose edges of type $1a$, $1b$, $2c$, $2d$, $3e$, $3f$ regardless of the type of the black edges. Furthermore, the loop contribution is always a factor of one because there are no internal loops to case (17).

The combinatorial factor, however, is quite complicated. Assume for now that we are working with type-1 black edges such that ab, ce, and df are linked. The combinatorial factor is

$$(2^{\delta_{ab}})^3 3! \binom{b_{ab}}{3} \times 1[\{a,b\} = \{c,e\} = \{d,f\}] \tag{D24}$$

$$+2^{\delta_{ab}} 2^{\delta_{ce}} \times 2 \binom{b_{ab}}{2} \times 2^{\delta_{df}} b_{df} \times 1[\{a,b\} = \{c,e\} \neq \{d,f\}] \tag{D25}$$

$$+2^{\delta_{ab}} 2^{\delta_{df}} \times 2 \binom{b_{ab}}{2} \times 2^{\delta_{ce}} b_{ce} \times 1[\{a,b\} = \{d,f\} \neq \{c,e\}] \tag{D26}$$

$$+2^{\delta_{ce}} 2^{\delta_{df}} \times 2 \binom{b_{ce}}{2} \times 2^{\delta_{ab}} b_{ab} \times 1[\{a,b\} \neq \{c,e\} = \{d,f\}] \tag{D27}$$

$$+2^{\delta_{ab}} 2^{\delta_{ce}} 2^{\delta_{df}} b_{ab} b_{ce} b_{df} \times 1[\{a,b\} \neq \{c,e\} \neq \{d,f\}] \tag{D28}$$

Here, 1[A] is an indicator function that is 1 if statement A is true and 0 if it is false. For example, $1[\{a,b\} = \{c,e\}, \{d,f\}]$ is 1 if $\{a,b\}$, $\{c,e\}$, and $\{d,f\}$ are all equal as sets (that is, order does not matter). The middle-three lines [Eqs. (D25)–(D27)] are just repetitions of the combinatorial factors for cases (13)–(16), but accounting for which sets of four edges may be sent to the same row. The last line [Eq. (D28)] is simple and accounts for the case where all of the edge types ab, ce, df are different. The first line [Eq. (D24)] requires a bit of explanation. In the case where $a \neq b$, we simply have to choose three edges of type ab where the order matters (they each could have been created by integrating out different graphs at a higher order). In the case where $a = b$, this is still the case, but now we need a factor of two for each edge, as we can flip which vertices are connected where.

Again, it is hard to account for all of these elements by hand, but it is simple numerically. With this final case sorted out, we simply combine contributions of all of the cases $g(n, \mathbf{a})_{\text{case}(i)}$ to find $g(n, \mathbf{a})$.

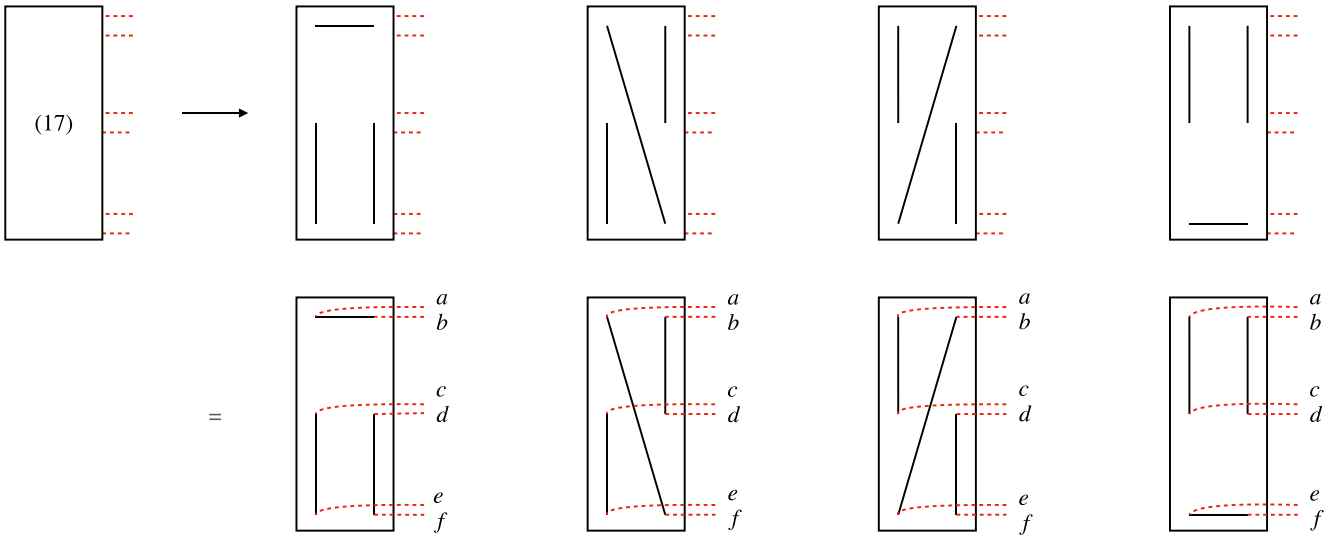


FIG. 16. Evaluation of case (17). We repeat the convention for cases (13)–(16) by taking the top-left vertex to row a, the top-right vertex to row b, the middle-left vertex to row c, and the middle-right vertex to row d, but we now also take the bottom left to e and the bottom right to f, where $a, b, c, d, e, f \in \{1, 2, 3\}$. The types of edges that are created are still determined by the type of the black edges.

APPENDIX E: CLASSICAL COMPLEXITY OF EVALUATING THE RECURSION FOR THE SECOND MOMENT

In this Appendix, we argue that the numerical evaluation of the recursion and, hence, the second moment, is classically efficient (that is, the runtime and space used are at most polynomial) in n, which corresponds to the Fock sector of interest in the output samples.

We recall the setup of the recursion as we describe it in the main text. Specifically, we define

$$g(n, a_{12}, a_{13}, a_{23}) := \sum_{G \in \mathbb{G}_n^2(a_{12}, a_{13}, a_{23})} k^{C(G)} \tag{E1}$$

$\mathbb{G}_n^2(a_{12}, a_{13}, a_{23})$ is the set of second-moment graphs of order n with a_{ij} red edges that cross between rows i and j. $C(G)$ is the number of connected components of G. The second moment is given by $(2n-1)!!g(n, 0, 0, 0)$. We then write down the recursion using these $g(n, a_{12}, a_{13}, a_{23})$ as

$$g(n, a_{12}, a_{13}, a_{23}) = \sum_{b_{12}, b_{13}, b_{23}} c(a_{12}, a_{13}, a_{23}, b_{12}, b_{13}, b_{23}) \times g(n-1, b_{12}, b_{13}, b_{23}) \tag{E2}$$

We list the following constraints on $\mathbf{a}$, which is shorthand for (a_{12}, a_{13}, a_{23}). First, $a_{12}+a_{13}$, $a_{12}+a_{23}$, and $a_{13}+a_{23}$ (the edges that exit the first, second, and third rows, respectively) must be even. Second, $a_{12}+a_{13}$, $a_{12}+a_{23}$, $a_{13}+a_{23}$ must all be less than or equal to $2n$, as there cannot be more than $2n$ edges coming out of a row with only $2n$ vertices given that there is exactly one red edge incident on every vertex. Finally, we also add here that, clearly, a_{12}, a_{13}, a_{23} are non-negative. These constraints imply a finite number of valid vectors $\mathbf{a} = (a_{12}, a_{13}, a_{23})$ for a given order n, and any vector satisfying these constraints corresponds to a valid set of graphs and, therefore, a term $g(n, \mathbf{a})$ in the recursion. We provide an example of all possible $\mathbf{a}$ when $n = 4$ in Table IV.

Clearly, as n grows, the number of possible $\mathbf{a}$ for which one must evaluate $g(n, \mathbf{a})$ also grows. However, we can bound this growth as being polynomial in n using some arguments about partitions. Recall that a partition of a positive integer t of size s is a set (i.e., order does not matter) of s positive integers whose sum is t. A weak partition of t of size s relaxes the positivity constraint of the set such that it contains s non-negative elements (t is still positive).

Let $t := a_{12} + a_{13} + a_{23}$. Then $t \leqslant 3n$, which follows from the fact that

$$\begin{aligned} &2a_{12} + 2a_{13} + 2a_{23} \\ &\quad = (a_{12} + a_{13}) + (a_{12} + a_{23}) + (a_{13} + a_{23}) \leqslant 6n \end{aligned} \tag{E3}$$

The conditions listed above on $\mathbf{a}$ imply that each $\mathbf{a}$ is a weak partition of size 3 of $t \leqslant 3n$ that satisfies two further

TABLE IV. All possible $\mathbf{a}$, up to permutations of the vector elements, for $2n = 8$. Each entry satisfies the constraints that $a_{12}+a_{13}$, $a_{12}+a_{23}$, and $a_{13}+a_{23}$ are even and less than or equal to $2n$, a_{12}, a_{13}, and a_{23} are non-negative, and $a_{12}+a_{13}+a_{23} = t$.

t	$\boldsymbol{a}$
0	(0,0,0)
1	∅
2	(2,0,0)
3	(1,1,1)
4	(2, 2, 0), (4, 0, 0)
5	(3,1,1)
6	(6, 0, 0), (4, 2, 0), (2, 2, 2)
7	(5, 1, 1), (3, 3, 1)
8	(8, 0, 0), (6, 2, 0), (4, 4, 0), (4, 2, 2)
9	(7, 1, 1), (5, 3, 1), (3, 3, 3)
10	(6, 2, 2), (4, 2, 2)
11	(5,3,3)
12	(4,4,4)

constraints: all three elements of the set must have the same parity as t, and no element can be larger than $2n$.

Now, the number of partitions of t of size at most three is $\lfloor (t+3)^2/12 \rceil$ [38] (note that $\lfloor T \rceil$ refers to the closest integer to T). Therefore, the number of partitions of t of size exactly 3, or $p_3(t)$, is bounded by this value, which implies that $\sum_{t=0}^{3n} p_3(t) = O(n^3)$. In turn, the number of $\mathbf{a}$, up to permutations of the elements of $\mathbf{a}$, is bounded by $O(n^3)$ (because they form an even more restricted class of weak permutations). We can overcount for these permutations with a simple constant multiplicative factor of 3! (this overcounts because, when numbers are repeated in the partition, there are fewer distinct permutations). Thus, we have a polynomial bound on the number of terms in our recursion at any Fock sector n (note that we could tighten this bound a bit by accounting more precisely for the parity constraint on the elements $\mathbf{a}$, but, because we are interested only in classical efficiency, this polynomial bound that arises from considering only size-3 partitions is sufficient).

To be sure that the recursion is efficiently computable, however, the actual values of the terms in the recursion must not grow too quickly. In particular, recall that each term $g(n, \mathbf{a})$ has a polynomial expansion in k of order at most $3n$ (this is the largest number of connected components possible when each one must have at least two vertices). The sum of the coefficients of $g(n, \mathbf{a})$ is the same as the number of graphs in $\mathbb{G}_n^2(a_{12}, a_{13}, a_{23})$, which we derive to be

$$\begin{aligned}\left|\mathbb{G}_n^2(a_{12}, a_{13}, a_{23})\right| = &\binom{2n}{a_{12}}\binom{2n-a_{12}}{a_{13}}\binom{2n}{a_{12}}\binom{2n-a_{12}}{a_{23}}\binom{2n}{a_{13}}\binom{2n-a_{13}}{a_{23}} a_{12}!a_{13}!a_{23}! \\ &\times (2n-a_{12}-a_{13}-1)!!(2n-a_{12}-a_{23}-1)!!(2n-a_{13}-a_{23}-1)!!4^n. \end{aligned} \tag{E4}$$

This is, at most, factorially big in n, which means that the number of bits needed to store these numbers, and, hence, $g(n, \mathbf{a})$ is polynomial in n.

Therefore, we have a polynomial bound on the number of terms in the recursion, as well as on the space needed to represent each of these terms. Finally, because the actual recursion consists only of polynomial numbers of multiplication and addition, which can each be accomplished in time polynomial in the size of the inputs, the actual computation is efficient.

APPENDIX F: COMPUTING INDIVIDUAL COEFFICIENTS

In this Appendix, we discuss the various methods by which one can compute individual coefficients in the polynomial expansion of the second moment. Recall that, per Theorem 2, the second moment may be expanded as

$$M_2(k, n) = (2n-1)!! \sum_{i=1}^{2n} c_i k^i \tag{F1}$$

Ideally, one would simply be able to find a closed functional form for the right-hand side of this equation (as was possible for the equivalent definition of the first moment). But, unfortunately, such a result currently eludes us. Therefore, the best we can do is find individual coefficients. We now discuss methods of calculating c_{2n} and c_{2n-1}.

1. Leading-order coefficient c_{2n}

We begin with the leading-order coefficient c_{2n}, which, per Lemma 1(ii), is $(2n)!!$. We prove this in the main text by reducing the calculation to a special case of the first moment, but we can also prove this purely combinatorially. As discussed in the proof of the main text, c_{2n} contains contributions only from graphs that possess solely type-1 and type-4 sets of black edges. Again, in order to create the maximal number of connected components, the horizontal black edges must also be connected by red edges to create a size-2 connected component. The remaining type-1 vertical black edges are paired off, and the type-4 vertical black edges are similarly paired off. So, for a graph of order n, say that there are p sets of type-1 black edges and, therefore, $n-p$ sets of type-4 black edges. There are $\binom{n}{p}$ sets of black edges with this type distribution. There are then $(2p-1)!!$ ways to pair off the $2p$ vertical type-1 black edges, and $(2n-2p-1)!!$ ways to pair off the $2n-2p$ vertical type-4 black edges. Therefore, summing over $p \in \{0, 1, \ldots, n\}$, we get that

$$c_{2n} = \sum_{p=0}^{n} \binom{n}{p} (2p-1)!!(2n-2p-1)!! \tag{F2}$$

We can massage the right-hand side a bit using the fact that $(2x-1)!! = (2x)!/(2x)!! = (2x)!/(2^x x!)$. Expanding out the binomial coefficient and converting all terms to single factorials yields

$$c_{2n} = \frac{n!}{2^n} \sum_{p=0}^{n} \binom{2p}{p}\binom{2n-2p}{n-p} \tag{F3}$$

The summation evaluates to 4^n using the convolution of the Taylor series for $(1-4x)^{-1/2}$ [39]. Therefore,

$$c_{2n} = 2^n n! = (2n)!!, \tag{F4}$$

matching the known result.

2. First subleading coefficient c_{2n-1}

We now generalize the above combinatorial version of the c_{2n} calculation to c_{2n-1}. It is slightly more complicated, as there is a bit of casework to consider, but the general idea is the same. In particular, the key idea is that because $2n$ is the maximal number of connected components, finding a graph with $2n-1$ connected components comes down to counting the ways that one can create a "deficit" of exactly one connected component from the maximal number. There are nine ways to accomplish this.

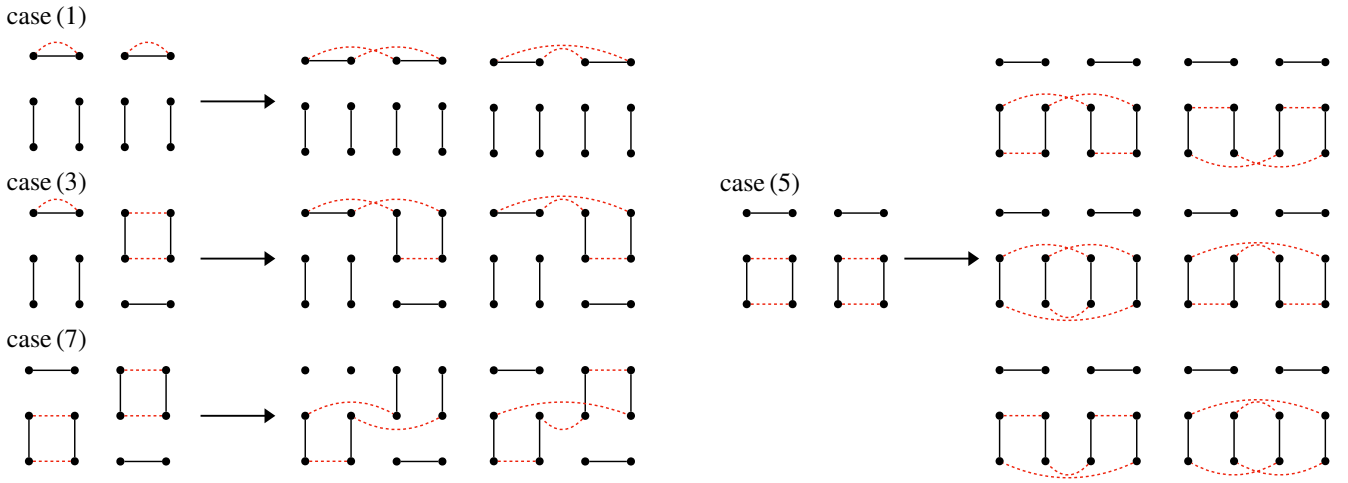


FIG. 17. Possible ways of merging type-1 and type-4 vertices to create a deficit of a single connected component. Here, we only show cases (1), (3), (5), and (7), as (2), (4), and (6) are symmetric with (1), (3), and (5) with type-1 and type-4 edges switched.

First, consider starting with graphs with a maximal number of connected components, meaning they have only type-1 and type-4 black edges. The connected components have either two vertices (red and black edge between two vertices in the same row) or four (two vertical black edges of the same type that are paired off via red edges). We refer to these as type-x2-vertex and 4-vertex connected components, respectively (where x is either 1 or 4). One can convert these graphs with maximal connected components into graphs with a deficit of a single connected component in the following ways, all of which involve merging two connected components into a single one:

(1) merge two type-1 2-vertex connected components
(2) merge two type-4 2-vertex connected components
(3) merge one type-1 2-vertex connected component with one type-4 4-vertex connected component
(4) merge one type-4 2-vertex connected component with one type-1 4-vertex connected component
(5) merge two type-1 4-vertex connected components
(6) merge two type-4 4-vertex connected components
(7) merge one type-1 4-vertex connected component with one type-4 4-vertex connected component

These options are visualized (up to the symmetry of exchanging the roles of type-1 and type-4 edges) in Fig. 17

Next, we must also consider cases with type-2 and type-3 black edges. There are two options here: either the graph can have exactly one set of type-2 or type-3 edges, or it can have exactly two sets (it does not matter whether it is two type-2 sets of edges, two type-3 sets of edges, or one of each). The rest of the sets of black edges must all be of type 1 or type 4. Then, creating a deficit can be done in the following ways:

(8) connect one type-2 or type-3 edge (the edge connecting the top row to the bottom row) to one type-1 vertical edge and one type-4 vertical edge to make a 6-vertex loop
(9) connect two type-2 or type-3 edges (again, the top-to-bottom edges) to form a 4-vertex connected component

These are visualized in Fig. 18 The rest of horizontal black edges must be connected with red edges to form 2-vertex connected components, and the remaining vertical edges must be appropriately paired off in order to ensure $2n-2$ other connected components are formed.

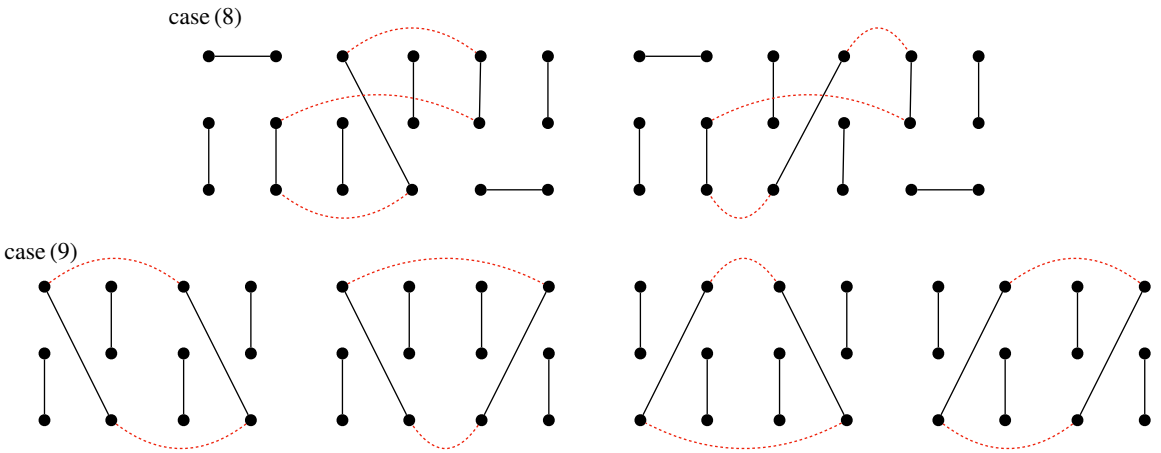


FIG. 18. Possible ways of creating a deficit of a single connected component while using type-2 and/or type-3 edges.

The end result of accounting for all of these cases is a (double) sum that computes c_{2n-1}:

$$
\begin{aligned}
c_{2n-1} = \sum_{p=0}^{n} \binom{n}{p} \times \Bigg[& \underbrace{2\binom{p}{2}(2p-1)!![2(n-p)-1]!!}_{(1)} + \underbrace{2\binom{n-p}{2}(2p-1)!![2(n-p)-1]!!}_{(2)} \\
& + \underbrace{2\binom{p}{1}\binom{2(n-p)}{2}(2p-1)!![2(n-p-1)-1]!!}_{(3)} + \underbrace{2\binom{n-p}{1}\binom{2p}{2}[2(p-1)-1]!![2(n-p)-1]!!}_{(4)} \\
& + \underbrace{6\binom{2p}{4}[2(p-2)-1]!![2(n-p)-1]!!}_{(5)} + \underbrace{6\binom{2(n-p)}{4}(2p-1)!![2(n-p-2)-1]!!}_{(6)} \\
& + \underbrace{2\binom{2p}{2}\binom{2(n-p)}{2}[2(p-1)-1]!![2(n-p-1)-1]!!}_{(7)} \Bigg] \\
& + \sum_{p=0}^{n-1} \underbrace{2\binom{n}{1}\binom{n-1}{p}(2p+1)[2(n-p-1)+1](2p-1)!![2(n-p-1)-1]!!}_{(8)} \\
& + \sum_{p=0}^{n-2} \underbrace{4\binom{n}{2}\binom{n-2}{p}(2p+1)!![2(n-p-2)+1]!!}_{(9)} . \qquad \text{(F5)}
\end{aligned}
$$

The last sum should be taken to be 0 when $n = 1$ and the sum is empty (this is because this case of course requires at least $n = 2$ to have two sets of type-2 or -3 edges). Each of these terms can be derived through a simple combinatorial argument regarding which types of edges are present and how they must be connected. For each case, say that there are p type-1 sets of black edges. This means there are $n - p$, $n - p - 1$, and $n - p - 2$ sets of type-4 black edges for cases (1)-(7), case (8), and case (9), respectively [in the latter two cases, the remaining set(s) of edges are type-2 and/or type-3]. Each case then comes down to deciding how to order the sets of edges, how to choose which edges are connected together, and then pairing off the remaining edges of the same type to build the remaining 2- and 4-vertex connected components. We do not detail how to count every single case, but we discuss two examples, case (1) and case (8). The rest should be straightforward to derive by extending these arguments.

In case (1), we merge two 2-vertex connected components of type 1. First, we have a factor of $\binom{n}{p}$ to account for all ways of having p type-1 sets of edges. We then must select 2 of the p horizontal black edges to merge into a single connected component, hence the factor of $\binom{p}{2}$; see Fig. 17. The additional factor of two comes from the two possible ways of merging these into a single connected component. Finally, the remaining double factorial factors are the number of ways of pairing off the vertical black edges with those of the same type. We then must sum from $p = 0$ to n to account for all possible black edge type distributions.

Case (8) proceeds similarly. First, we have a factor of $\binom{n}{1}$, or n, to choose where the type-2 or type-3 set of edges is. The factor of two out front now actually accounts for whether it is type-2 or type-3. Next, we have $\binom{n-1}{p}$ to account for the placement of the p type-1 sets of edges. There are now $2p + 1$ black edges that span the second and third rows (i.e., they are black edges that arise from type-1 sets of black edges). It is $2p + 1$ because the type-2 or type-3 set of black edges contributes one, and the p type-1 sets contribute $2p$. Analogously, there are also $[2(n - p - 1) + 1]$ black edges spanning the first and second rows. We have to select one of each to connect to the black edge that spans the first and third rows to make a single 6-vertex connected component. The remaining factors are again the number of ways to pair off the remaining vertical black edges with those of the same type (horizontal black edges must form 2-vertex connected components to reach the required number of connected components).

It is possible, but quite tedious, to simplify this double sum by looking at each individual term and then applying a similar technique as in the evaluation of the sum for c_{2n}. That is, for each term in the sum, we use the convolution of various Taylor series and compare the coefficients of x^n. We start with the first term

$$
\begin{aligned}
(1) &\to \sum_{p=0}^{n} \binom{n}{p} 2\binom{p}{2}(2p-1)!![2(n-p)-1]!! \\
&= \frac{n!}{2^n} \sum_{p=0}^{n} p(p-1)\binom{2p}{p}\binom{2n-2p}{n-p} \qquad \text{(F6)}
\end{aligned}
$$

One then has through Taylor expansion that

$$
x^2 \frac{\mathrm{d}^2}{\mathrm{d}x^2} \frac{1}{\sqrt{1-4x}} = \sum_{n=0}^{\infty} \binom{2n}{n} n(n-1) x^n , \qquad \text{(F7)}
$$

which implies that

$$12x^2\frac{1}{(1-4x)^3}=\left(x^2\frac{\mathrm{d}^2}{\mathrm{d}x^2}\frac{1}{\sqrt{1-4x}}\right)\frac{1}{\sqrt{1-4x}}$$
$$=\sum_{n=0}^{\infty}\sum_{p=0}^{n}p(p-1)\binom{2p}{p}\binom{2n-2p}{n-p}x^n \tag{F8}$$

Using the Online Encyclopedia of Integer Sequences (OEIS), we find the threefold convolution of powers of 4 A038845 [40] has formula $(n+2)(n+1)2^{2n-1}$, meaning

$$\sum_{n=0}^{\infty}12(n+2)(n+1)2^{2n-1}x^{n+2}=12x^2\frac{1}{(1-4x)^3}$$
$$=\sum_{n=0}^{\infty}\sum_{p=0}^{n}p(p-1)\binom{2p}{p}\binom{2n-2p}{n-p}x^n \tag{F9}$$

Therefore, comparing powers of x, we get that

$$\sum_{p=0}^{n}p(p-1)\binom{2p}{p}\binom{2n-2p}{n-p}=12n(n-1)2^{2n-5}, \tag{F10}$$

meaning the first term in the sum is (after some algebra)

$$\sum_{p=0}^{n}\binom{n}{p}2\binom{p}{2}(2p-1)!![2(n-p)-1]!!=(2n)!!\frac{3n(n-1)}{8} \tag{F11}$$

Note also by the symmetry between p and $n-p$, the contribution of the second term is the same.

We can perform similar manipulations for the other terms. In particular,

$$(3)\to\sum_{p=0}^{n}\binom{n}{p}2p\binom{2n-2p}{2}(2p-1)!![2(n-p-1)-1]!!$$
$$=\frac{n!}{2^{n-1}}\sum_{p=0}^{n}(n-p)p\binom{2p}{p}\binom{2n-2p}{n-p} \tag{F12}$$

Instead of taking the Taylor expansion for the second derivative of $(1-4x)^{-1/2}$ and convolving it with that for $(1-4x)^{-1/2}$, we convolve the Taylor series for the first derivative with itself. That is,

$$\frac{4x^2}{(1-4x)^3}=\left(x\frac{\mathrm{d}}{\mathrm{d}x}\frac{1}{\sqrt{1-4x}}\right)^2$$
$$=\sum_{n=0}^{\infty}\sum_{p=0}^{n}p(n-p)\binom{2p}{p}\binom{2n-2p}{n-p}x^n, \tag{F13}$$

which, using the same result as for (1) (just with a difference of a factor of three), yields

$$\sum_{p=0}^{n}p(n-p)\binom{2p}{p}\binom{2n-2p}{n-p}=4n(n-1)2^{2n-5} \tag{F14}$$

This means that the third term yields a contribution of

$$(3)\to\frac{n!}{2^{n-1}}4n(n-1)2^{2n-5}=(2n)!!\frac{2n(n-1)}{8} \tag{F15}$$

Again, by the symmetry between n and $n-p$, the contribution from (4) is the same.

Next,

$$(5)\to\sum_{p=0}^{n}\binom{n}{p}6\binom{2p}{4}[2(p-2)-1]!![2(n-p)-1]!!$$
$$=\frac{n!}{2^n}\sum_{p=0}^{n}p(p-1)\binom{2p}{p}\binom{2n-2p}{n-p}=(2n)!!\frac{3n(n-1)}{8} \tag{F16}$$

because this is the exact same as (1). Again, by symmetry, (6) has the same contribution.

We also have that

$$(7)\to\sum_{p=0}^{n}\binom{n}{p}2\binom{2p}{2}\binom{2(n-p)}{2}$$
$$\times[2(p-1)-1]!![2(n-p-1)-1]!!$$
$$=\frac{n!}{2^{n-1}}\sum_{p=0}^{n}p(n-p)\binom{2p}{p}\binom{2n-2p}{n-p}$$
$$=(2n)!!\frac{2n(n-1)}{8}, \tag{F17}$$

which follows because this term happens to be the same as (3).

We now move on to the final two cases. Again, similar manipulations yield that

$$(8)\to\sum_{p=0}^{n-1}2\binom{n}{1}\binom{n-1}{p}(2p+1)[2(n-p-1)+1]$$
$$\times(2p-1)!![2(n-p-1)-1]!!$$
$$=\frac{n!}{2^{n-1}}\sum_{p=0}^{n}(2p+1)(n-p)\binom{2p}{p}\binom{2n-2p}{n-p}$$
$$=\frac{n!}{2^{n-1}}2\sum_{p=0}^{n}p(n-p)\binom{2p}{p}\binom{2n-2p}{n-p}$$
$$+\frac{n!}{2^{n-1}}\sum_{p=0}^{n}(n-p)\binom{2p}{p}\binom{2n-2p}{n-p}. \tag{F18}$$

We have expanded the upper limit to $p=n$ because the factor of $n-p$ sets this additional contribution to zero. The first term in the last equation is simply twice the contribution of (3), which is $(2n)!!4n(n-1)/8$. The second term requires yet another manipulation of Taylor series. By very similar arguments to the above, we have that

$$x\frac{\mathrm{d}}{\mathrm{d}x}\frac{1}{\sqrt{1-4x}}=\sum_{n=0}^{\infty}\binom{2n}{n}nx^n, \tag{F19}$$

which implies that

$$\frac{2x}{(1-4x)^2}=\left(x\frac{\mathrm{d}}{\mathrm{d}x}\frac{1}{\sqrt{1-4x}}\right)\frac{1}{\sqrt{1-4x}}$$
$$=\sum_{n=0}^{\infty}\sum_{p=0}^{n}p\binom{2p}{p}\binom{2n-2p}{n-p}x^n, \tag{F20}$$

which is the same as the sum we are interested in (up to the symmetry of replacing $n-p$ with p). Using OEIS sequence A002697 [40], that is, the convolution of powers of four, we find that

$$\frac{2x}{(1-4x)^2} = \sum_{n=0}^{\infty} 2(n+1)4^n x^{n+1}, \tag{F21}$$

which means that, comparing powers of x^n,

$$\frac{n!}{2^{n-1}} \sum_{p=0}^{n} (n-p)\binom{2p}{p}\binom{2n-2p}{n-p} = \frac{n!}{2^{n-1}} 2n4^{n-1} = n(2n)!! \tag{F22}$$

Finally, then

$$(8) \to (2n)!!\frac{4n(n-1)}{8} + (2n)!!n = \frac{4n(n+1)}{8} \tag{F23}$$

Last, we get that

$$\begin{aligned}(9) &\to \sum_{p=0}^{n-2} 4\binom{n}{2}\binom{n-2}{p}(2p+1)!![2(n-p-2)+1]!! \\ &= \frac{n!}{2^{n-1}} \sum_{p=0}^{n-2} \binom{2p+2}{p+1}\binom{2n-2p-2}{n-p-1} \\ &\quad \times (p+1)(n-p-1) \\ &= \frac{n!}{2^{n-1}} \sum_{x=0}^{n} \binom{2x}{x}\binom{2n-2x}{n-x}(x)(n-x),\end{aligned} \tag{F24}$$

where we have set $x = p+1$ and then expanded the limits of summation to include $x = 0$ and $x = n$ (because these terms contribute zero). Therefore, this contribution is the same as (3), (4), and (7), which is $(2n)!!2n(n-1)/8$.

In total, we have that

$$\begin{aligned}\frac{c_{2n-1}}{(2n)!!} &= 4\frac{3n(n-1)}{8} + 4\frac{2n(n-1)}{8} + \frac{4n(n-1)}{8} + n \\ &= (3n-2)n,\end{aligned} \tag{F25}$$

meaning

$$c_{2n-1} = (2n)!!(3n-2)n \tag{F26}$$

Numerically evaluating the sums yields the same value up to $n = 40$, and this also matches the value of c_{2n-1} computed via the recursion. We note that $(3n-2)n$ are the so-called octagonal numbers, which are OEIS entry A000567 [40]. However, we are not sure whether there is a deeper connection between these numbers and the graph theoretic problem at the core of this calculation. Additionally, while it is nice that we have been able to find an exact formula for a second coefficient, this calculation does not seem scalable, meaning other methods are likely needed to try to find the full expansion of the second moment.

APPENDIX G: ALTERNATIVE METHOD FOR COMPUTING COEFFICIENTS c_i

In this Appendix, we present an alternative method for computing coefficients c_i in

$$M_2(k,n) = (2n-1)!! \sum_{i=1}^{2n} c_i k^i \tag{G1}$$

Using this method, we obtain a useful expression for c_1. We also outline how this method can be used to set up an alternative recursive code for computing the coefficients c_i for all i. While we have not implemented this code, there is a possibility it is more efficient than the recursive code discussed in the main text. It is also possible that this new method may yield other useful analytical results about c_i, including their asymptotic behavior.

We start by recalling Eq. (20):

$$M_2(k,n) = (2n-1)!! \sum_{G\in\mathbb{G}_n^2} k^{C(G)}, \tag{G2}$$

where the sum goes over all graphs possessing the allowed assignments of black and red edges. The new method relies on the following key simplifying observation: for a given fixed assignment of black edges, the contribution to $M_2(k,n)$ (summed over all allowed red-edge assignments) depends only on $\mathbf{e} = (e_{11}, e_{12}, e_{13}, e_{23}, e_{33})$, where e_{ij} is the number of black edges that connect row i to row j. In particular, the answer does not depend on what columns the black edges are connecting. The proof of this key observation is simple: for a fixed set of black edges, the contribution to $M_2(k,n)$ is summed over all possible red perfect matchings in each of the three rows. This means that we can swap any two vertices in a given row (while pulling the ends of the black edges to the new destinations) without changing the answer. This completes the proof.

Let p_1 be the number of type-1 sets of black edges, p_4 be the number of type-4 sets of black edges, and p be the combined number of type-2 and type-3 sets of black edges (type-2 and type-3 sets are equivalent as far as their contributions to e_{ij}). Then $e_{11} = p_1$, $e_{33} = p_4$, $e_{12} = p + 2p_4$, $e_{23} = p + 2p_1$, and $e_{13} = p$. We then write

$$M_2(k,n) = \sum_{p_1=0}^{n} \sum_{p_4=0}^{n-p_1} \binom{n}{p_1}\binom{n-p_1}{p_4} 2^p g(\mathbf{e}), \tag{G3}$$

where $p = n - p_1 - p_4$. The combinatorial factors come from choosing p_1 sets of type-1 black edges out of n possible locations, then p_4 sets of type-4 black edges from $n-p_1$ possible locations, and finally multiplying by a factor of two for each choice of whether a given contribution to p is type-2 or type-3. Additionally,

$$g(\mathbf{e}) = \sum_{i=1}^{2n} d_i(\mathbf{e}) k^i, \tag{G4}$$

where $d_i(\mathbf{e})$ is the number of ways (using the allowed red-edge assignments) to make i loops given the black edges specified by $\mathbf{e}$.

The coefficients $d_i(\mathbf{e})$ can then be computed with the help of the visualization shown in Fig. 19(a). The three black dots

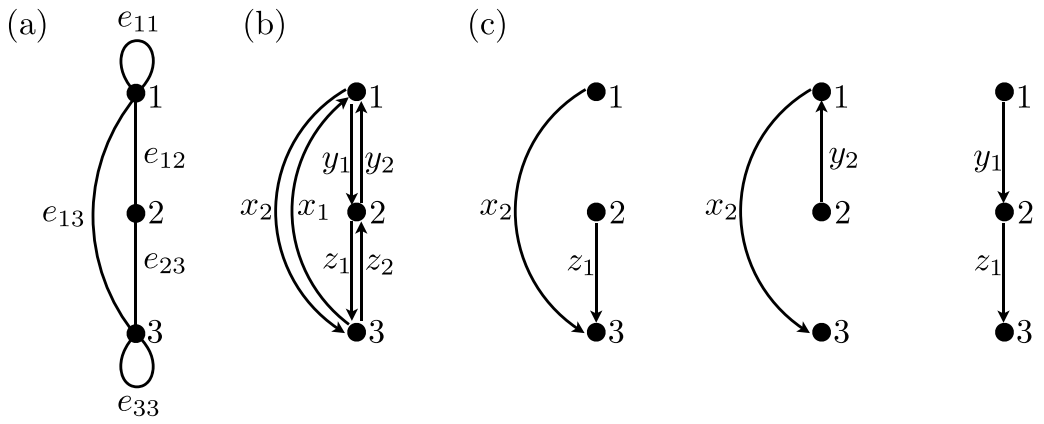


FIG. 19. Graphs useful for understanding the new method for calculating coefficients c_i. (a) Once the types of black edges are assigned, the contribution to $M_2(k,n)$ depends only on the number of black e_{ij} edges connecting row i to row j. (b) To compute c_1, the number of single-loop graphs contributing to M_2, we first set $e_{11} = e_{33} = 0$ (later adding in the effect of nonzero values), fix the winding number w of the loop, and break up 1-3 edges into $x_1 = (e_{13} + w)/2$ clockwise edges and $x_2 = (e_{13} - w)/2$ counterclockwise edges. We similarly break up the 1-2 and 2-3 edges. We then use the BEST theorem [41] to count the number of Eulerian circuits on this directed graph. (c) For $u = 3$, the three types of arborescences contributing to $t_u(G) = x_2z_1 + y_2x_2 + y_1z_1$ for the graph G shown in panel (b).

labeled 1, 2, and 3 represent the three rows. The numbers e_{jk} on the five edges (including the two loops) show how many black edges connect row j to row k. Roughly speaking, the coefficient $d_i(\mathbf{e})$ is the number of ways to connect all the black edges specified by $\mathbf{e}$ into exactly i loops. The red edges are used to connect the black edges to each other and are taken into account automatically, which is one of the key advantages of this approach (slightly more specifically, for any two black edges that share a row, it is possible to connect them with a red edge between the vertices in that shared row). Each way of joining the edges $\mathbf{e}$ into loops also comes with a combinatorial factor that takes into account the fact that all edges are distinguishable and the fact that edges that stay in the same row can each be traversed in one of two directions

The coefficients $g(\mathbf{e})$ can be computed using a recursive procedure. Instead of doing a recursion on n (which is what we do in the main text, with details presented in Appendix D), we perform the recursion on the number of black edges $e_{11} + e_{12} + e_{13} + e_{23} + e_{33}$. As in the main text, we need to define a more general function $g(\mathbf{e}, \sigma, c, s)$ to make the recursion work. σ is a binary variable, so that $\sigma = 1$ means we are in the process of building a loop, while $\sigma = 0$ means that we need to start a new loop. If $\sigma = 1$, we need to also specify $s \in \{1, 2, 3\}$ (standing for start) indicating the row where the current loop started and $c \in \{1, 2, 3\}$ (standing for current) indicating the row where we currently are.

As in the main text, the recursive procedure is efficient, i.e., takes polynomial time in the number of edges. We first directly compute $g(\mathbf{e}, \sigma, c, s)$ for small values of $e_{11} + e_{12} + e_{13} + e_{23} + e_{33}$. Then the recursive step goes as follows: If $\sigma = 0$, we can either (1) close the loop right away by reducing e_{11} or e_{33} by one, keep $\sigma = 0$, and multiply by k, or (2) set $\sigma = 1$, start a new loop at row i, set $s = i$, reduce e_{ij} by one (for some j), and set $c = j$. If $\sigma = 1$, we can either (1) close the loop by reducing e_{cs} by one, set $\sigma = 0$, and multiply by k, or (2) continue building the loop, keep $\sigma = 1$, keep s unchanged, reduce e_{cj} by one (for some j), and change the value of c to j. As we do these calculations, we need to also include appropriate combinatorial factors deciding which black edge to take (e.g., if we pick one of e_{ij} edges, we need to multiply by e_{ij}, and if $i = j$, we need to multiply by another factor of 2).

While we have not coded up this procedure, we believe that it offers another complementary way of understanding and analyzing the second moment.

1. Computing c_1

Again, while we have not coded up the above recursive procedure, we show how to use the new approach to compute c_1 in Eq. (G1), i.e., the number of ways to build a single-loop graph, which we were not able to directly compute using the original method.

To proceed, we first ignore the contributions of the edges e_{11} and e_{33} (effectively pretending that they are equal to zero), but we address how to deal with them later on. We also assign a direction to this single loop, and we later divide the final answer by two because each loop will be counted twice (because there are two possible directions around a loop). While it may seem to make things more difficult to add directionality to a previously undirected graph, it actually allows us to make use of known results.

To proceed, we sort the contributions to c_1 according to the winding number w of the loop around the triangle formed by rows 1, 2, and 3, which can now be well defined because we have added directionality to the edges. Once w is fixed, the total numbers of edges in the triangle of each directionality also become fixed. Specifically, as shown in Fig. 19(b), $x_1 = (e_{13} + w)/2$ is the number of 1-3 edges traversed (i.e., directed) from 3 to 1, $x_2 = (x - w)/2$ is the number of 1-3 edges traversed from 1 to 3, $y_1 = (e_{12} + w)/2$ is the number 1-2 edges traversed from 1 to 2, $y_2 = (e_{12} - w)/2$ is the number of 1-2 edges traversed from 2 to 1, $z_1 = (e_{23} + w)/2$ is the number 2-3 edges traversed from 2 to 3, and $z_2 = (e_{23} - w)/2$ is the number 2-3 edges traversed from 3 to 2. Note that, here, we are treating a positive winding number as going *clockwise* around the graph. There will also be combinatorial factors associated with *which* edges go in which direction, but we

handle that factor later. We are now interested in the number of Eulerian circuits on the resulting directed graph G, i.e., the number of directed closed paths that visit each edge exactly once. The BEST theorem [41] says that the number of such Eulerian circuits is

$$ec(G) = t_u(G) \prod_{v \in V} [\deg(v) - 1]!, \tag{G5}$$

where $V = \{1, 2, 3\}$ is the set of three vertices of our graph, $\deg(v)$ is the indegree of vertex v, and $t_u(G)$ is the number of arborescences of G with root u, i.e., the number of directed tree subgraphs of G such that, for any vertex v, there is exactly one directed path from v to u. If graph the G has an Eulerian circuit, it is known that $t_u(G)$ is independent of the choice of u. Choosing $u = 3$, the three types of trees (arborescences) contributing to $t_u(G)$ for the graph G in Fig. 19(b) are shown in Fig. 19(c). The result is $t_u(G) = x_2 z_1 + y_2 x_2 + y_1 z_1$. The term $x_2 z_1$ [corresponding to the first graph in Fig. 19(c)] counts trees (arborescences) made up of a $1 \to 3$ edge and a $2 \to 3$ edge; the term $y_2 x_2$ [corresponding to the second graph in Fig. 19(c)] counts trees made up of a $2 \to 1$ edge and a $1 \to 3$ edge; and the term $y_1 z_1$ [corresponding to the third graph in Fig. 19(c)] counts trees made up of a $1 \to 2$ edge and a $2 \to 3$ edge. Plugging in the definitions of x_i, y_i, and z_i, we find $t_u(G) = (w^2 + e_{12}e_{13} + e_{13}e_{23} + e_{23}e_{12})/4$. Therefore,

$$\begin{aligned} ec(G) &= \frac{1}{4}(w^2 + e_{12}e_{13} + e_{13}e_{23} + e_{23}e_{12})\left(\frac{e_{12}+e_{13}}{2} - 1\right)!\left(\frac{e_{12}+e_{23}}{2} - 1\right)!\left(\frac{e_{13}+e_{23}}{2} - 1\right)! \\ &= \frac{1}{4}[w^2 + 3n^2 - (p_1 - p_4)^2 - 2n(p_1 + p_4)](n - p_1 - 1)!(n-1)!(n - p_4 - 1)! \end{aligned} \tag{G6}$$

We now include the aforementioned combinatorial factors that account for which edges receive which directionality. When choosing which x_1 of the e_{13} edges to make into $3 \to 1$ edges, we pick up a combinatorial factor of

$$\binom{e_{13}}{x_1} = \binom{n - p_1 - p_4}{(n - p_1 - p_4 + w)/2}$$

Similarly for e_{12} and e_{23}:

$$\binom{e_{12}}{y_1} = \binom{n - p_1 + p_4}{(n - p_1 + p_4 + w)/2} \quad \text{and} \quad \binom{e_{23}}{z_1} = \binom{n + p_1 - p_4}{(n + p_1 - p_4 + w)/2}$$

We can now also account for the fact that e_{11} and e_{33} may actually be nonzero. We keep G defined as before (i.e. using only 1-2, 1-3, and 2-3 edges), but we now dress the loops defined on G (and counted above) with additional 1-1 and 3-3 edges. The number of times our loop visits vertex 1 is given by $\deg(1) = n - p_1$, so we need to sort $e_{11} = p_1$ edges into $n - p_1$ buckets, which gives a factor of

$$\binom{e_{11} + n - p_1 - 1}{e_{11}} = \binom{n-1}{p_1}$$

(by the standard "stars and bars" argument). Similarly, $e_{33} = p_4$ loops give

$$\binom{e_{33} + n - p_4 - 1}{e_{33}} = \binom{n-1}{p_4}$$

Because all $e_{11} = p_1$ edges are distinguishable and can be traversed in two different ways, we also get a factor of $p_1! 2^{p_1}$ (that is, after the bucket counts are decided, we still have to order the edges and assign each a direction). We similarly get a factor of $p_4! 2^{p_4}$. Putting all these elements together, we have

$$\begin{aligned} c_1 &= \sum_{p_1=0}^{n} \sum_{p_4=0}^{n-p_1} \binom{n}{p_1}\binom{n-p_1}{p_4} 2^p d_1(\mathbf{e}) \\ &= \sum_{p_1=0}^{n-1} \sum_{p_4=0}^{n-\max(p_1,1)} \binom{n}{p_1}\binom{n-p_1}{p_4} 2^p \sum_w \frac{1}{8}(w^2 + 3n^2 - (p_1 - p_4)^2 - 2n(p_1 + p_4))(n - p_1 - 1)!(n-1)!(n - p_4 - 1)! \\ &\quad \times \binom{n - p_1 - p_4}{(n - p_1 - p_4 + w)/2}\binom{n - p_1 + p_4}{(n - p_1 + p_4 + w)/2}\binom{n + p_1 - p_4}{(n + p_1 - p_4 + w)/2}\binom{n-1}{p_1}\binom{n-1}{p_4} p_1! 2^{p_1} p_4! 2^{p_4} \\ &= n![(n-1)!]^3 2^{n-3} \sum_{p_1=0}^{n} \sum_{p_4=0}^{n-p_1} \sum_{w=-n+p_1+p_4}^{n-p_1-p_4} \frac{\binom{n-p_1+p_4}{(n-p_1+p_4+w)/2}\binom{n+p_1-p_4}{(n+p_1-p_4+w)/2}(w^2 + 3n^2 - (p_1 - p_4)^2 - 2n(p_1 + p_4))}{p_1! p_4! [(n - p_1 - p_4 - w)/2]! [(n - p_1 - p_4 + w)/2]!} \end{aligned} \tag{G7}$$

In the second line, we have introduced an extra factor of $1/2$ because we counted every loop twice because of the two directions in which each loop can be traversed. In the second line, we also excluded the cases where all black edge sets are of type-1 ($p_1 = n$) and where all black edge sets are of type-4 ($p_4 = n$), as there is no single-loop contribution in this case [allowing for $p_1 = n$ would make $\binom{n-1}{p_1}$ undefined; similarly for $p_4 = n$]. In the last line, to simplify the expression, we allow $p_1 = n$ and $p_4 = n$ because the corresponding contribution is now well-defined and vanishes anyway. In the last line, the sum over w runs in increments of 2 due to a parity constraint (flipping the directionality of a single edge actually changes the winding number by two). While one can evaluate the sum over w in the final expression in Eq. (G7) in terms of hypergeometric functions, we were not able to then evaluate the remaining sums over p_4 and p_1 to obtain a closed-form expression for c_1.

Numerical evaluation of the final expression in Eq. (G7) agrees with the evaluation of c_1 using the recursive method in the main text up to $n = 40$ (which is the largest n we apply the latter method to). The final expression in Eq. (G7) is, however, so simple that it can easily be evaluated for much larger values of n. For example, *Mathematica* [37] on a personal computer evaluates it for $n = 200$ in about 15 seconds. One can also use Eq. (G7) to study in detail the asymptotic dependence of c_1 on n. We also hope that the method introduced in this section can yield other useful analytical results about c_i.

[1] P. W. Shor, in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM* (IEEE, Piscataway, NJ, 1994), pp. 124–134.

[2] S. Aaronson and A. Arkhipov, Theory Comput. **9**, 143 (2013).

[3] D. Hangleiter and J. Eisert, Rev. Mod. Phys. **95**, 035001 (2023).

[4] A. Serafini, *Quantum Continuous Variables* (CRC Press, Boca Raton, FL, 2017).

[5] A. P. Lund, A. Laing, S. Rahimi-Keshari, T. Rudolph, J. L. O'Brien, and T. C. Ralph, Phys. Rev. Lett. **113**, 100502 (2014).

[6] S. Rahimi-Keshari, A. P. Lund, and T. C. Ralph, Phys. Rev. Lett. **114**, 060501 (2015).

[7] C. S. Hamilton, R. Kruse, L. Sansoni, S. Barkhofen, C. Silberhorn, and I. Jex, Phys. Rev. Lett. **119**, 170501 (2017).

[8] R. Kruse, C. S. Hamilton, L. Sansoni, S. Barkhofen, C. Silberhorn, and I. Jex, Phys. Rev. A **100**, 032326 (2019).

[9] D. Grier, D. J. Brod, J. M. Arrazola, M. B. d. A. Alonso, and N. Quesada, Quantum **6**, 863 (2022).

[10] U. Chabaud and M. Walschaers, Phys. Rev. Lett. **130**, 090602 (2023).

[11] A. Deshpande, A. Mehta, T. Vincent, N. Quesada, M. Hinsche, M. Ioannou, L. Madsen, J. Lavoie, H. Qi, J. Eisert, D. Hangleiter, B. Fefferman, and I. Dhand, Sci. Adv. **8**, eabi7894 (2022).

[12] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu, P. Hu, X.-Y. Yang, W.-J. Zhang, H. Li, Y. Li, X. Jiang, L. Gan, G. Yang, L. You, Z. Wang *et al.*, Science **370**, 1460 (2020).

[13] H.-S. Zhong, Y.-H. Deng, J. Qin, H. Wang, M.-C. Chen, L.-C. Peng, Y.-H. Luo, D. Wu, S.-Q. Gong, H. Su, Y. Hu, P. Hu, X.-Y. Yang, W.-J. Zhang, H. Li, Y. Li, X. Jiang, L. Gan, G. Yang, L. You *et al.*, Phys. Rev. Lett. **127**, 180502 (2021).

[14] L. S. Madsen, F. Laudenbach, M. F. Askarani, F. Rortais, T. Vincent, J. F. F. Bulmer, F. M. Miatto, L. Neuhaus, L. G. Helt, M. J. Collins, A. E. Lita, T. Gerrits, S. W. Nam, V. D. Vaidya, M. Menotti, I. Dhand, Z. Vernon, N. Quesada, and J. Lavoie, Nature (London) **606**, 75 (2022).

[15] Y.-H. Deng, Y.-C. Gu, H.-L. Liu, S.-Q. Gong, H. Su, Z.-J. Zhang, H.-Y. Tang, M.-H. Jia, J.-M. Xu, M.-C. Chen, H.-S. Zhong, J. Qin, H. Wang, L.-C. Peng, J. Yan, Y. Hu, J. Huang, H. Li, Y. Li, Y. Chen *et al.*, arXiv:2304.12240.

[16] A. Barvinok, *Combinatorics and Complexity of Partition Functions*, Algorithms and Combinatorics (Springer International Publishing, Cham, 2016), Vol. 30.

[17] L. Valiant, Theor. Comput. Sci. **8**, 189 (1979).

[18] A. Ehrenberg, J. T. Iosue, A. Deshpande, D. Hangleiter, and A. V. Gorshkov, companion paper, Phys. Rev. Lett. **134**, 140601 (2025).

[19] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler *et al.*, Nature (London) **574**, 505 (2019).

[20] A. Arkhipov and G. Kuperberg, Geom. Topol. Monogr. **18**, 1 (2012).

[21] C. Neill, P. Roushan, K. Kechedzhi, S. Boixo, S. V. Isakov, V. Smelyanskiy, A. Megrant, B. Chiaro, A. Dunsworth, K. Arya, R. Barends, B. Burkett, Y. Chen, Z. Chen, A. Fowler, B. Foxen, M. Giustina, R. Graff, E. Jeffrey, T. Huang *et al.*, Science **360**, 195 (2018).

[22] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, R. Babbush, N. Ding, Z. Jiang, M. J. Bremner, J. M. Martinis, and H. Neven, Nat. Phys. **14**, 595 (2018).

[23] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, SIAM Rev. **59**, 65 (2017).

[24] J. T. Iosue and A. Ehrenberg, jtiosue/LXEB GitHub repository (2024), https://github.com/jtiosue/LXEB.

[25] B. Gupt, J. Izaac, and N. Quesada, J. Open Source Softw. **4**, 1705 (2019).

[26] A. Björklund, B. Gupt, and N. Quesada, ACMJ. Exp. Algor. **24**, 1.11:1 (2019).

[27] J. Shi and T. Byrnes, npj Quantum Inf. **8**, 54 (2022).

[28] N. Quesada, J. M. Arrazola, and N. Killoran, Phys. Rev. A **98**, 062322 (2018).

[29] C. Oh, L. Jiang, and B. Fefferman, Phys. Rev. Lett. **131**, 010401 (2023).

[30] J. Martínez-Cifuentes, H. de Guise, and N. Quesada, PRX Quantum **5**, 040312 (2024).

[31] L. Stockmeyer, in *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing* (ACM, New York, NY, 1983), pp. 118–126.

[32] A. Bouland, B. Fefferman, Z. Landau, and Y. Liu, in *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS) Denver, CO* (IEEE, Denver, Piscataway, NJ, 2022), pp. 1308–1317.

[33] A. Deshpande, P. Niroula, O. Shtanko, A. V. Gorshkov, B. Fefferman, and M. J. Gullans, PRX Quantum **3**, 040329 (2022).

[34] G. Folland, *Real Analysis: Modern Techniques and Their Applications* (Wiley, New York, 2007).
[35] N. N. Li and Wenchang Chu, Math. Commun. **24**, 279 (2019).
[36] G. Chang and C. Xu, Am. Math. Mon. **118**, 175 (2011).
[37] W. R. Inc., *Mathematica, Version 14.0* (Champaign, 2024).
[38] G. H. Hardy, Some Famous Problems of the Theory of Numbers and in Particular Waring's Problem. An Inaugural Lecture Delivered before the University of Oxford (2011).
[39] R. Johnson, https://math.stackexchange.com/users/13854/robjohn, Binomial sum gives 4^n, Mathematics Stack Exchange (2016), https://math.stackexchange.com/q/1595627 (version: 2016-01-01).
[40] OEIS Foundation Inc., The On-Line Encyclopedia of Integer Sequences, published electronically at http://oeis.org (2022).
[41] T. van Aardenne-Ehrenfest and N. G. de Bruijn, Simon Stevin : Wis- en Natuurkundig Tijdschrift **28**, 203 (1951).