Classical algorithm for quantum SU(2) Schur sampling

Vojtěch Havlíček*
Department of Computer Science, University of Oxford, Wolfson Building, Parks Road, Oxford OX1 3QD, United Kingdom

Sergii Strelchuk
Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Wilberforce Road, Cambridge, CB2 3HU, United Kingdom

Kristan Temme
IBM T.J. Watson Research Center, Yorktown Heights, New York 10598, USA

(Received 13 February 2019; published 27 June 2019)

Many quantum algorithms can be represented in a form of a classical circuit positioned between quantum Fourier transformations. Motivated by the search for new quantum algorithms, we turn to circuits where the latter transformation is replaced by the SU(2) quantum Schur transform—a global transformation which maps the computational basis to a basis defined by angular momenta. We show that the output distributions of these circuits can be approximately classically sampled in polynomial time if they are sufficiently close to being sparse, thus isolating a regime in which they could lead to algorithms with exponential computational advantages. Our paper is primarily motivated by a conjecture that underpinned the hardness of permutational quantum computing, a restricted quantum computational model that has the above circuit structure in one of its computationally interesting regimes. The conjecture stated that approximating transition amplitudes of permutational quantum computing model to inverse-polynomial precision on a classical computer is computationally hard in this case. We disprove the extended version of this conjecture—even in the case when the hardness of approximation originated from a difficulty of finding the large elements in the output probability distributions. Finally, we present some evidence that output of the above permutational quantum computing circuits could be efficiently approximately sampled from on a classical computer.

DOI: 10.1103/PhysRevA.99.062336

I. INTRODUCTION

Characterizing the power of quantum computers is one of the two major challenges in quantum computation, with the other being their scalable implementation. A seminal approach to the former problem is the study of conditions which make quantum algorithms amenable to methods of efficient classical simulation. A number of important quantum algorithms can be cast in a form of a classical circuit positioned between a pair of circuits which implement quantum Fourier transformation. These are, for example, algorithms for the hidden subgroup problem which in particular include the Shor's factoring algorithm [1,2]. While the latter provides strong evidence that quantum computers outperform the classical ones, Schwarz and Van den Nest [3] showed that the respective quantum circuit could be efficiently classically simulated if its output distribution was sufficiently close to being sparse.

In our current paper, we aim to characterize a different class of circuits that instead of the quantum Fourier transform contain the quantum Schur transform (QST) as depicted in Fig. 1. QST is a map from the computational basis to a basis defined by angular momentum [4–6] and it underpins a variety of quantum information processing tasks, including spectrum estimation [7,8], hypothesis testing [9–12], quantum computing using decoherence-free subspaces [13], communication without a shared reference frame [14,15], and quantum color coding [16]. A quantum circuit that efficiently implements this transform was first described in Refs. [4–6] and recently improved by Kirby and Strauch [17,18]. The extent to which circuits using QST could be used to devise new quantum algorithms is, to our knowledge, largely unexplored—possibly with the exception of Refs. [19,20].

QST is a centerpiece in the analysis of permutational quantum computing (PQC) [21]—a restricted quantum computational model based on recoupling of angular momenta [20,22]. It has been conjectured that PQC has supraclassical computational power. One of the conjectures supporting this belief stated that an approximation of its transition amplitudes in the regime where they encode matrix elements of the symmetric group irreducible representation matrices in the Young's orthogonal form [20,23] is hard to compute classically if we require inverse polynomial precision (in the number of input qubits). While in our previous work we presented an efficient classical algorithm for approximating such transition amplitudes [21], an intriguing question remained: Is it also possible to identify all PQC transition amplitudes that can be approximated using classical methods with the inverse polynomial precision? Since the expected output probability of an n-qubit quantum circuit C with an input state $|y\rangle$ is given

*vojtech.havlicek@keble.ox.ac.uk

 ©2019 American Physical Society

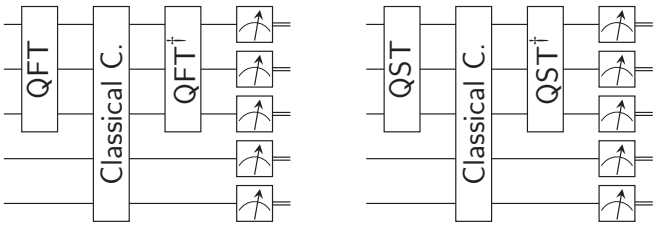


FIG. 1. Schematic diagrams of the quantum circuit used in Shor's factoring algorithm (left) and the circuits we consider here (right). QST denotes the SU(2) quantum Schur transformation. The classical circuits between the transforms can represent, for example a polynomially long sequence of Toffoli gates.

by

$$\mathbb{E}_x(|\langle x|C|y\rangle|^2) = \frac{1}{2^n}\sum_x |\langle x|C|y\rangle|^2 = \frac{1}{2^n},$$

approximating these values with an inverse polynomial precision cannot distinguish the majority of $\langle x|C|y\rangle$ amplitudes from zeros (see Fig. 2). Could we exploit the difficulty that arises from finding large matrix elements encoded in the output of the algorithm and thus demonstrate the (exponential) quantum computational advantage?

We show that this is not the case by describing a classical method that finds all large output probabilities in polynomial time. Our proof technique uses the simulation technique of Schwarz and Van den Nest [3], where the authors studied an analogous problem in the context of the quantum Fourier transform. This approach uses a variant of the Kushilevitz-Mansour algorithm used in classical learning theory [24,25]. We adapt it for distributions arising in the class of circuits using the QST, which include the relevant regime of PQC. We then show how to classically approximately sample their output distributions. The sampling algorithm becomes efficient for output distributions that are sufficiently close to sparse.

Our results additionally imply that sampling from the quantum Schur circuits can only lead to exponential computational advantages if the individual elements of the output

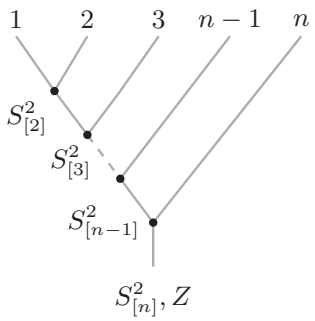


FIG. 3. Sequentially coupled basis on n qubits. The numbers at the leaf nodes label qubits. Every vertex • carries a total spin operator S^2_A that forces qubits in set A to one of its eigenstates. Similar diagrams can be used to label basis states and are shown in Appendix A or Ref. [20].

distribution *cannot* be resolved by polynomial approximation with the quantum device by taking polynomially many samples. A way to circumvent this restriction, similar to the case of circuits that use the quantum Fourier transform, could be to use a technique utilized in the Shor's algorithm that reconstructs group generators by sampling $\log|G|$ group elements for a superpolynomially large $|G|$. There is no meaningful counterpart to this approach for the QST as of now.

II. QUANTUM SU(2) SCHUR SAMPLING

The studied circuits are derived from PQC—a computational model based on recoupling of angular momenta [20,22]. We hence review the basics of the angular momentum theory before introducing them. Consider n qubits indexed by $[n] := \{1, 2 \ldots n\}$. The spin of the kth qubit is defined by a triple of operators,

$$\vec{S}_k = \frac{1}{2}(X_k,\ Y_k,\ Z_k),$$

where X_k, Y_k, Z_k denote the Pauli X, Y, Z operators on the kth qubit. The *total spin operator* on a qubit subset $A \subseteq [n]$ is

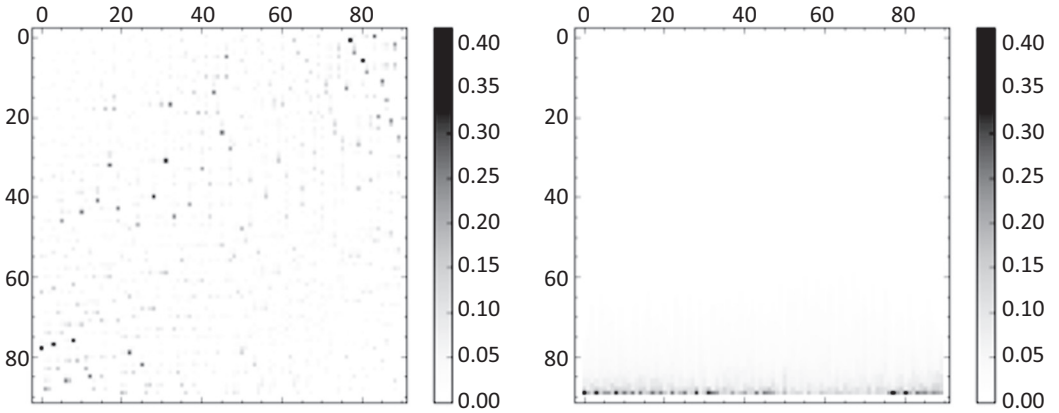


FIG. 2. Left: A part of the $|\langle x|C|y\rangle|^2$ matrix for a typical PQC instance. After normalization, most matrix elements are indistinguishable from zeros within the polynomially small approximation window. We show how to classically find the large elements. Right: The output matrix with sorted output, demonstrating that an overwhelming fraction of the probabilities are usually small.

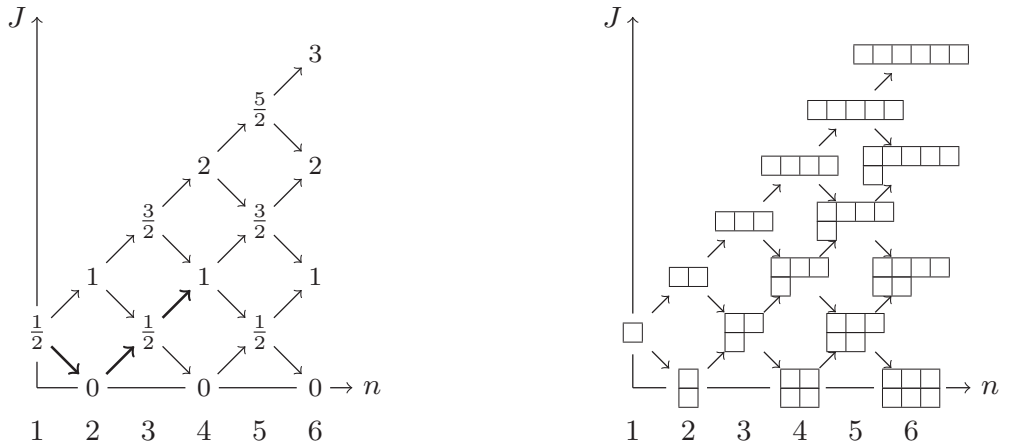


FIG. 4. The branching diagram. The highlighted path $\boldsymbol{J} = [\frac{1}{2} \to 0 \to \frac{1}{2} \to 1]$ corresponds to a set of five four-qubit quantum states: $|\boldsymbol{J}, M\rangle = |M, J = 1, j_{[3]} = \frac{1}{2}, j_{[2]} = 0\rangle$ with $M \in \{-2, -1, 0, 1, 2\}$. The path takes the following sequence of steps: $\searrow$, $\nearrow$, $\nearrow$, and corresponds to a Yamanouchi symbol 011. Yamanouchi symbols are used in representation theory of the symmetric group, which is made explicit by the diagram on the right, showing that each branching diagram node can also be labeled with the Young diagrams in two rows. Every Young diagram with n boxes has $\frac{n}{2} + J$ boxes on the top, and $\frac{n}{2} - J$ boxes in the bottom row labels a set of paths from $\mathcal{A}_n$ that end at the same J. As detailed in Appendix B, the individual paths can be shown to be bijective with standard Young tableaux with two rows. We use this to improve the sampling algorithm in Sec. IV.

given by

$$S_A^2 := \sum_{k\in A} \vec{S}_k \cdot \sum_{k'\in A} \vec{S}_{k'}.$$

We write $S^2 := S^2_{[n]}$. The operators S_A^2 and S_B^2 commute if and only if the sets A and B are disjoint or one is contained in the other. Let

$$Z_A := \frac{1}{2}\sum_{k\in A} Z_k,$$

denote the azimuthal spin operator on a qubit subset A. We again use $Z_{[n]} := Z$. The operators Z_A and S_A^2 commute for any $A \subseteq [n]$ and share an eigenspace labeled by quantum numbers j_A and m_A. The quantum number j_A is the *total spin* of qubits in A and m_A is the *azimuthal spin*. Both spin numbers are subject to constraints: The azimuthal spin m_A only takes values in integer steps between $-j_A$ and j_A, while the total spin numbers are either integer or half-integer and combine according to the angular momentum addition rules [26,27]:

$$j_{A\cup B} \in \{|j_A - j_B|,\ |j_A - j_B| + 1, \ldots,\ j_A + j_B\}. \quad (1)$$

Sets of commuting spin operators can be used to define complete orthonormal bases [20]. A particular basis is given by coupling a qubit at a time; that is, by the joint eigenstates of

$$S^2_{[2]},\ S^2_{[3]},\ \ldots S^2, Z.$$

We call it the *sequentially coupled basis* (Fig. 3). The basis states are labeled by eigenstates $j_{[2]}, j_{[3]} \ldots, j_{[n-1]}, J$ and M of the spin operators. By Eq. (1), these are subject to

$$j_{[1]} = \tfrac{1}{2}, \quad j_{[k+1]} = \left|j_{[k]} \pm \tfrac{1}{2}\right|, \quad (2)$$

which can be expressed diagrammatically by a *branching diagram* (Fig. 4). Up to the quantum number M, the sequential basis states correspond to paths in this diagram that start at $j_{[1]} = \frac{1}{2}$.

Let $\mathcal{A}_k$ be the set of all such paths on k qubits. Any path $\boldsymbol{j} \in \mathcal{A}_k$ can be labeled by a bitstring by writing 1 for any $\nearrow$ edge of the path and 0 for an $\searrow$ edge of the path $\boldsymbol{j}$ in the branching diagram. For example,

$$\left[\tfrac{1}{2} \to 1 \to \tfrac{1}{2} \to 1\right] \mapsto 101.$$

Any prefix of length $m \leqslant k-1$ in such a bitstring contains at most $\lceil \frac{m}{2} \rceil$ zeros, since the path never goes below the horizontal axis of the branching diagram. These bitstrings play a role in the representation theory of the symmetric group and are called Yamanouchi symbols [28,29]. The sets of Yamanouchi symbols with the same Hamming weight correspond to Young diagrams on two rows, which can be seen in Fig. 4. This is underpinned by the SU(2) Schur-Weyl duality, that states that the n-qubit Hilbert space decomposes into the tensor product of the symmetric group S_n modules (isomorphic to the Young diagrams in two rows) and the special unitary group SU(2) under their joint action.

See Appendix B for additional details of this correspondence and Refs. [4,18] for detailed discussion of the underlying representation theory.

For the sequentially coupled basis, the SU(2) Schur-Weyl duality gives the SU(2) QST as described in Refs. [4–6,17,18,30]. It is a sequence of the Clebsch-Gordan transformations that couple j and j' eigenspaces into a $|J, M, j, j'\rangle$ state by

$$|J, M, j, j'\rangle = \sum_{m,m'} C^{J,M}_{j,m;\ j',m'} |j, m\rangle |j', m'\rangle,$$

where the summation over m runs from $-j$ to j in integer steps (and similarly for m') and the $C^{J,M}_{j,m;\ j',m'}$ are the Clebsch-Gordan coefficients. The transform between the computational and the sequentially coupled basis is given by a cascade of the Clebsch-Gordan transforms [4,17]. For example, on

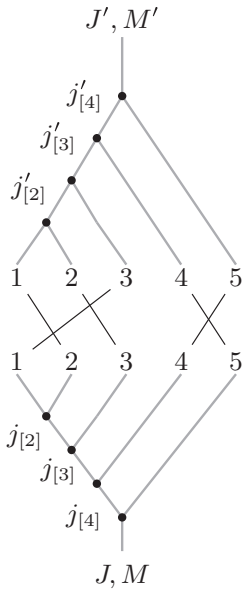


FIG. 5. Schematic representation of permutational quantum computing in the sequentially coupled basis. The applied permutation is (1,2,3)(4,5).

three qubits,

$$\begin{aligned}|J, M, j_{[2]}\rangle &= \sum_{m_{[2]},m_3} C^{J,M}_{j_{[2]},m_{[2]};\frac{1}{2},m_3}|j_{[2]}, m_{[2]}\rangle|m_3\rangle \\ &= \sum_{m_1,m_2}\sum_{m_{[2]},m_3} C^{J,M}_{j_{[2]},m_{[2]};\frac{1}{2},m_3} C^{j_{[2]},m_{[2]}}_{\frac{1}{2},m_1;\frac{1}{2},m_2}|m_1m_2m_3\rangle \\ &= \sum_{m_1m_2m_3}[U_{\mathsf{Sch}}]^{J,M,j_{[2]}}_{m_1m_2m_3}|m_1m_2m_3\rangle.\end{aligned}$$

where we omitted the $j = \frac{1}{2}$ numbers for qubits for brevity. The extension to the $n \geqslant 3$ qubit case is straightforward. We label the sequentially coupled basis states on n qubits by $|\boldsymbol{J}, M\rangle$, where $\boldsymbol{J}$ is a path in $\mathcal{A}_n$.

PQC in the sequentially coupled basis uses the permutation gate between two sequentially coupled basis states (see Fig. 5). Its transition amplitudes are

$$\langle \boldsymbol{J}, M|U_\pi|\boldsymbol{J}', M'\rangle,$$

where the permutation gate U_π is defined by its action on a computational basis state $|x_1 \ldots x_n\rangle$ as

$$U_\pi|x_1x_2x_3\ldots x_n\rangle = |x_{\pi(1)}x_{\pi(2)}x_{\pi(3)}\ldots x_{\pi(n)}\rangle.$$

Both Z and S^2 operators commute with U_π and, consequently, $M = M'$ and $J = J'$. The matrix $\langle \boldsymbol{J}, M|U_\pi|\boldsymbol{J}', M'\rangle$ block diagonalizes to J, M blocks; each of which corresponds to an irreducible representation of the symmetric group in the Young's orthogonal form. The transition amplitudes are then the matrix elements of these matrices [20]. Approximating them to polynomial precision was conjectured hard classically in Refs. [20,23] but an efficient classical algorithm was found in Ref. [21].

The methods we present here work for a broader family of quantum circuits we call the SU(2) quantum Schur sampling circuits. These have transition amplitudes,

$$\langle \boldsymbol{J}, M|W|\boldsymbol{J}', M'\rangle,$$

where W is defined by its action on a computational basis state $|x\rangle, x \in \{0, 1\}^n$,

$$W|x\rangle = |w(x)\rangle,$$

with $w : \{0, 1\}^n \to \{0, 1\}^n$ being a classical function given by a sequence of Toffoli gates—we consider only such W where this sequence is $\mathsf{poly}(n)$ long [31].

The circuits become similar in structure to Shor's algorithm in a sense of Fig. 1 if we allow for ancillary qubits. The simulation results apply also to these circuits, which we discuss in Sec. VI.

III. FINDING LARGE PROBABILITIES

We now describe an algorithm for finding large probabilities in the output of the circuits (see Fig. 2). Our approach is built on the concept of computational tractability introduced in Ref. [32].

Definition 1. An n-qubit state $|\psi\rangle$ is computationally tractable (CT) if it is possible to classically sample from the distribution

$$p(x) = \{|\langle x|\phi\rangle|^2 : x \in \{0, 1\}^n\}$$

in polynomial time and the overlaps $\langle x|\phi\rangle$ can be computed to exponential precision for a computational basis state $|x\rangle$ in polynomial time.

We proved in Ref. [21] that the sequentially coupled basis states are CT. As a corollary, we show that $|\phi\rangle = W|\boldsymbol{J}, M\rangle$ is also CT.

Lemma 1. $|\phi\rangle = W|\boldsymbol{J}, M\rangle$ is CT.

Proof. Since $\langle x|\boldsymbol{J}, M\rangle$ can be efficiently computed because $|\boldsymbol{J}, M\rangle$ is CT, so can $\langle x|W|\boldsymbol{J}, M\rangle = \langle w^{-1}(x)|\boldsymbol{J}, M\rangle$. The distribution

$$p(x) = |\langle x|W|\boldsymbol{J}, M\rangle|^2$$

can be efficiently sampled by applying the inverse of $w(x)$ to the samples drawn from $|\langle x|\boldsymbol{J}, M\rangle|^2$:

$$p(w^{-1}(x)) = |\langle w^{-1}(x)|\boldsymbol{J}, M\rangle|^2 = |\langle x|W|\boldsymbol{J}, M\rangle|^2.$$

Since W is made of polynomially many Toffoli gates, the inverse is obtained by applying the circuit in reverse to the bitstring x.

We also state Lemma 3 of Ref. [32], which is an application of the Chernoff-Hoeffding bound.

Lemma 2 (CT state overlap (ϵ, δ)-approximation [32]). An overlap $\langle\phi|\psi\rangle$ between two CT states can be approximated by $\tilde{a}$, such that

$$|\tilde{a} - \langle\phi|\psi\rangle| \leqslant \epsilon,$$

with probability $1 - \delta$ in $\mathsf{poly}(\frac{1}{\epsilon}, n, \log\frac{1}{\delta})$ time. We say that the overlap $\langle\phi|\psi\rangle$ is (ϵ, δ)-approximated by $\tilde{a}$.

We now show how to approximate a set of output probability marginals, an enabling result for extension of the techniques used by Schwarz and Van den Nest in Ref. [3]. Given a path $\boldsymbol{j} \in \mathcal{A}_k$ for $k \leqslant n$, define the *output marginal* $p(\boldsymbol{j})$,

$$\begin{aligned}p(\boldsymbol{j}) &:= \sum_{\boldsymbol{J}\supseteq\boldsymbol{j}}\sum_M p(\boldsymbol{J}, M) \\ &= \langle\phi|\sum_{\boldsymbol{J}\supseteq\boldsymbol{j};M}|\boldsymbol{J}, M\rangle\langle\boldsymbol{J}, M\,|\,\phi\rangle := \langle\phi|\Pi(\boldsymbol{j})|\phi\rangle,\end{aligned}$$

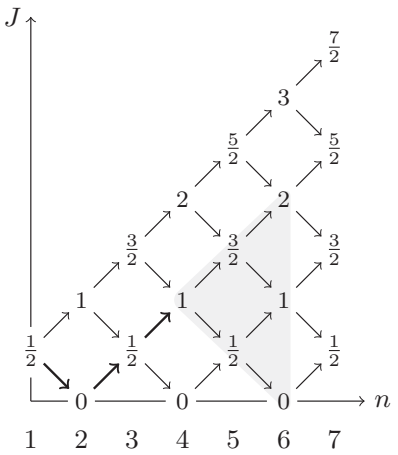


FIG. 6. The summation $\sum_{J\supseteq j}$ runs over all paths $\boldsymbol{J}\in\mathcal{A}_n$ that contain $\boldsymbol{j}\in\mathcal{A}_k$. As an example, in the diagram above, $\boldsymbol{j}=[\frac{1}{2}\to 0\to\frac{1}{2}\to 1]$ and $k=4, n=6$. The summation runs over the paths within the shaded region. It follows that (in terms of the Yamanouchi symbols) $\boldsymbol{J}\supseteq\boldsymbol{j}=\{01111, 01110, 01101, 01100\}$.

where the summation $\sum_{\boldsymbol{J}\supseteq\boldsymbol{j}}$ sums all paths $\boldsymbol{J}\in\mathcal{A}_n$ that contain $\boldsymbol{j}\in\mathcal{A}_k$ (see Fig. 6). The summation $\sum_M$ runs from $-J$ to J in integer steps. We use $\sum_{\boldsymbol{J}\supseteq\boldsymbol{j};M}$ as shorthand for $\sum_{\boldsymbol{J}\supseteq\boldsymbol{j}}\sum_M$. The projector,

$$\Pi(\boldsymbol{j}):=\sum_{\boldsymbol{J}\supseteq\boldsymbol{j};M}|\boldsymbol{J},M\rangle\langle\boldsymbol{J},M|,$$

can be simplified to (Appendix C)

$$\Pi(\boldsymbol{j})=\sum_m|\boldsymbol{j},m\rangle\langle\boldsymbol{j},m|,$$

where the sum $\sum_m$ runs over $m\in\{-j,-j+1,\dots j\}$.

Lemma 3. For $\boldsymbol{j}\in\mathcal{A}_k$, $p(\boldsymbol{j})$ can be classically (ϵ,δ)-approximated by $\tilde{p}(\boldsymbol{j})$ in $\mathsf{poly}(\frac{1}{\epsilon},n,\log\frac{1}{\delta})$ time.

Proof. We first show that the marginal $p(\boldsymbol{j})$ on k qubits can be written as a transition amplitude of a larger, $(n+k)$-qubit circuit as

$$\langle\phi|\boldsymbol{j},m\rangle\langle\boldsymbol{j},m|\phi\rangle=(\langle\boldsymbol{j},m|\langle\phi|)U_{\mathsf{SWAPS}}(|\phi\rangle|\boldsymbol{j},m\rangle),$$

where U_{SWAPS} is a permutation gate on $k+n$ qubits. Write symbolically $|\boldsymbol{j},m\rangle=|\psi\rangle=|\psi_1\psi_2\dots\psi_k\rangle$ and $|\phi\rangle=|\phi_1\phi_2\dots\phi_n\rangle$, so

$$\begin{aligned}\langle\phi|\boldsymbol{j},m\rangle\langle\boldsymbol{j},m|\phi\rangle&=\langle\phi|\psi\rangle\langle\psi|\phi\rangle\\&=\langle\phi_1\dots\phi_n|\psi_1\dots\psi_k\rangle\langle\psi_1\dots\psi_k|\phi_1\dots\phi_n\rangle.\end{aligned}$$

Let U_{SWAPS} swap the $(n+i)$th and $(n-k+i)$th qubits for all $1\leqslant i\leqslant k$:

$$\begin{aligned}&U_{\mathsf{SWAPS}}|\psi_1\dots\psi_k\rangle|\phi_1\dots\phi_n\rangle\\&\quad=|\phi_1\dots\phi_k\rangle|\psi_1\dots\psi_k,\phi_{k+1}\dots\phi_n\rangle.\end{aligned}$$

This gives

$$\begin{aligned}&\langle\phi|\langle\psi|U_{\mathsf{SWAPS}}|\psi\rangle|\phi\rangle\\&=\langle\phi_1\dots\phi_n|\langle\psi_1\dots\psi_k|\phi_1\dots\phi_k\rangle|\psi_1\dots\psi_k,\phi_{k+1}\dots\phi_n\rangle\\&=\langle\phi_1\dots\phi_n|\psi_1\dots\psi_k\rangle\langle\psi_1\dots\psi_k|\phi_1\dots\phi_n\rangle\\&=\langle\phi|\psi\rangle\langle\psi|\phi\rangle,\end{aligned}$$

as desired. Since both $|\boldsymbol{j},m\rangle$ and $|\phi\rangle$ states are CT and U_{SWAP} is a permutation on up to $2n$ objects,

$$(\langle\boldsymbol{j},m|\langle\phi|)U_{\mathsf{SWAPS}}(|\phi\rangle|\boldsymbol{j},m\rangle),$$

can be (ϵ,δ)-approximated by Lemma 2. Therefore

$$\begin{aligned}p(\boldsymbol{j})&=\langle\phi|\Pi(\boldsymbol{j})|\phi\rangle\\&=\sum_m\langle\phi|\boldsymbol{j},m\rangle\langle\boldsymbol{j},m|\phi\rangle\\&=\sum_m(\langle\boldsymbol{j},m|\langle\phi|)U_{\mathsf{SWAPS}}(|\phi\rangle|\boldsymbol{j},m\rangle).\end{aligned}$$

Since $\sum_m$ sums $2j+1\leqslant n+1$ terms, it follows that $p(\boldsymbol{j})$ can also be (ϵ,δ) approximated.

We now combine Lemmas 1, 2, and 3 to describe a classical algorithm that finds large elements in the output distribution of quantum Schur circuits. It is an adaptation of the Kushilevitz-Mansour algorithm [24].

Theorem 1. Let $p(\boldsymbol{J}):\mathcal{A}_n\to[0,1]$ be a probability distribution on paths. There is a classical algorithm that outputs a set $L\subseteq\mathcal{A}_n$ in $\mathsf{poly}(n,\frac{1}{\theta},\log\frac{1}{\gamma})$ time, such that for some $\theta>0$,

$$\begin{aligned}\forall\,\boldsymbol{J}\in L:\ p(\boldsymbol{J})&\geqslant\frac{\theta}{2},\\\forall\,\boldsymbol{J}\in\mathcal{A}_n:\ p(\boldsymbol{J})>\theta&\Rightarrow\boldsymbol{J}\in L,\end{aligned}\tag{3}$$

with probability at least $1-\gamma$.

Proof. See Algorithm 1.

Algorithm 1 Kushilevitz-Mansour Algorithm

1. Set $L_2=\emptyset$. Choose:
$$\delta<\frac{\theta}{2n},$$
and compute $\tilde{p}(\boldsymbol{j}_2)$ for both paths $\boldsymbol{j}_2\in\mathcal{A}_2$ by Lemma 3, such that:
$$|p(\boldsymbol{j}_2)-\tilde{p}(\boldsymbol{j}_2)|\leqslant\frac{\theta}{4}.$$
Add $\boldsymbol{j}_2\in\mathcal{A}_2$ to L_2 if $\tilde{p}(\boldsymbol{j}_2)\geqslant\frac{3}{4}\theta$.
2. Continue for $k=3,\dots,n$. Assume L_{k-1} has been found. For any path $\boldsymbol{j}_{k-1}\in L_{k-1}$, take both possible steps in the branching diagram. For any $\boldsymbol{j}_{k-1}$, there are at most two paths $\boldsymbol{j}_k\in\mathcal{A}_k$ that coincide with $\boldsymbol{j}_{k-1}$ for the first $k-1$ steps and end at $j_k=|j_{k-1}\pm\frac{1}{2}|$. For every such $\boldsymbol{j}_k$, compute the approximation $\tilde{p}(\boldsymbol{j}_k)$ such that:
$$|\tilde{p}(\boldsymbol{j}_k)-p(\boldsymbol{j}_k)|\leqslant\frac{\theta}{4}.$$
If $\tilde{p}(\boldsymbol{j}_k)\geqslant\frac{3}{4}\theta$, add the path $\boldsymbol{j}_k$ to L_k.
3. In every step of the computation, check if $|L_k|>\frac{2}{\theta}$. If true, halt and output $\emptyset$.
The algorithm never halts if all approximation steps succeed.
4. Output $L=L_n$.

The algorithm runs in n steps, each of which succeeds with probability at least $(1-\delta)^{|L_k|}$. Since $|L_k|\leqslant\frac{2}{\theta}$, the success probability is at least

$$(1-\delta)^{2n/\theta}\geqslant1-\frac{2\delta n}{\theta}:=1-\gamma.$$

Thanks to $\delta < \frac{\theta}{2n}$, it follows that $1-\gamma > 0$. The algorithms terminates in $\mathsf{poly}(n, \frac{1}{\theta}, \log\frac{1}{\gamma})$ time as it halts whenever the number of elements in any list exceeds $\frac{2}{\theta}$. Since $p(\boldsymbol{j}_k) \geqslant \frac{\theta}{2}$ for each $\boldsymbol{j}_k \in L_k$, the final list L contains at most $\frac{2}{\theta}$ elements by normalization. So if all approximation steps succeed, the algorithm does not halt before it outputs L.

Algorithm 1 has an interesting consequence: Since it runs in polynomial time whenever $\theta = 1/\mathsf{poly}(n)$, paths with polynomially small $p(\boldsymbol{J}) = \sum_M p(\boldsymbol{J}, M)$ can be found in polynomial time. When such path $\boldsymbol{J}$ is found, it is possible to approximate $p(\boldsymbol{J}, M)$ for all M; since there are $2J+1 \leqslant n+1$ distinct values of M, it has to be approximated for by Lemma 2. Such approximation of transition amplitudes has the same precision as if when polynomially many samples were taken with a quantum computer. The SU(2) quantum Schur sampling circuits therefore cannot encode classically hard-to-approximate quantities in amplitudes that could be resolved by sampling because any such quantity could be found by the presented algorithm and then approximated by Lemma 2.

IV. APPROXIMATE SAMPLING

Following Schwarz and Van den Nest [3], we use the above algorithm to approximately sample the quantum Schur circuits under additional sparsity constraint on their output distribution:

Definition 2 (ϵ-approximate t-sparsity). A probability distribution $p(\boldsymbol{J}, M)$ is t-sparse if it has at most t nonzero elements $p(\boldsymbol{J}, M)$. A probability distribution $\tilde{p}(\boldsymbol{J}, M)$ is ϵ-approximately t-sparse if there exists a t-sparse distribution $p(\boldsymbol{J}, M)$ such that

$$\|p - \tilde{p}\|_1 \leqslant \epsilon.$$

We also adapt a technical lemma from Ref. [24].

Lemma 4. Let $p(\boldsymbol{J}, M)$ be an ϵ-approximate t-sparse distribution. Let S be the set of all $(\boldsymbol{J}, M)$ for which $p(\boldsymbol{J}, M)$ is greater than $\frac{\epsilon}{t}$. Then

$$\sum_{\boldsymbol{J}\notin S;M} p(\boldsymbol{J}, M) \leqslant 2\epsilon.$$

Proof. Let $p^t(\boldsymbol{J}, M)$ be a t-sparse distribution that is ϵ-close to $p(\boldsymbol{J}, M)$. Define T to be the set of all $(\boldsymbol{J}, M)$ for which p^t is nonzero, i.e., the support of p^t. Trivially, $S \cap T \subseteq S$, which implies that

$$\sum_{\boldsymbol{J}\notin S;M} p(\boldsymbol{J}, M) \leqslant \sum_{\boldsymbol{J}\notin S\cap T;M} p(\boldsymbol{J}, M).$$

Define the indicator $I_A : A \to \{0, 1\}$ on the set A as follows:

$$I_A(a) := \begin{cases} 1, & \text{if } a \in A, \\ 0, & \text{otherwise,} \end{cases}$$

so that

$$\begin{aligned} \sum_{\boldsymbol{J}\notin S\cap T;M} p(\boldsymbol{J}, M) &= \sum_{\boldsymbol{J}\in\mathcal{A}_n;M} p(\boldsymbol{J}, M)(I_{S\cap T}(\boldsymbol{J}, M) - 1) \\ &= \|pI_{S\cap T} - p\|_1. \end{aligned}$$

By the triangle inequality:

$$\|pI_{S\cap T} - p\|_1 \leqslant \|p - pI_T\|_1 + \|pI_{S\cap T} - pI_T\|_1.$$

Since p is ϵ-approximate t-sparse, it follows that

$$\|p - pI_T\|_1 \leqslant \|p - p^t\|_1 \leqslant \epsilon.$$

We also have that

$$\begin{aligned} \|pI_{S\cap T} - pI_T\|_1 &= \sum_{\boldsymbol{J}\in T;M} p(\boldsymbol{J}, M) I_{S\cap T}(\boldsymbol{J}, M) \\ &= \sum_{\boldsymbol{J}\in T/S;M} p(\boldsymbol{J}, M) \leqslant \frac{\epsilon}{t} t = \epsilon, \end{aligned} \tag{4}$$

because all elements in T/S are $\leqslant \frac{\epsilon}{t}$ and there is at most t of them. This gives

$$\sum_{\boldsymbol{J}\notin S;M} p(\boldsymbol{J}, M) \leqslant 2\epsilon.$$

Theorem 1 and Lemma 4 can be combined to define a probability distribution close to the quantum output that can be sampled from in $\mathsf{poly}(t, \frac{1}{\epsilon}, n)$ time. Assume that $p(\boldsymbol{J}, M)$ is ϵ-approximately t-sparse. Let $L \subseteq \mathcal{A}_n$ be the set of paths generated by the Kushilevitz-Mansour algorithm with threshold $\theta = \frac{\epsilon}{t}$. Choose

$$\epsilon' = \min\left(\frac{\epsilon}{(n+1)|L|}, \frac{\epsilon}{4t}\right), \tag{5}$$

and compute ϵ' approximations $\tilde{p}(\boldsymbol{J}, M)$ for all $\boldsymbol{J} \in L$ and M by Lemma 2. Define a normalization factor α as

$$\alpha = \frac{1}{2^n - \sum_{\boldsymbol{J}\in L}(2J+1)},$$

such that $\sum_{\boldsymbol{J}\notin L;M} \alpha = 1$ (see Appendix A 3). Use the ϵ'-approximations $\tilde{p}(\boldsymbol{J}, M)$ to define

$$\tilde{p} = \begin{cases} \tilde{p}(\boldsymbol{J}, M) & \text{for } \boldsymbol{J} \in L, \\ \tilde{p}_\circ := \alpha(1 - \sum_{\boldsymbol{J}\in L;M} \tilde{p}(\boldsymbol{J}, M)) & \text{otherwise,} \end{cases}$$

so $\tilde{p}$ becomes uniform on all $\boldsymbol{J}$ outside L. The constant $\tilde{p}_\circ$ is chosen so $\tilde{p}$ is normalized. Then

$$\begin{aligned} &\|\tilde{p} - p\|_1 \\ &= \sum_{\boldsymbol{J}\in L;M} |\tilde{p}(\boldsymbol{J}, M) - p(\boldsymbol{J}, M)| + \sum_{\boldsymbol{J}\notin L;M} |\tilde{p}(\boldsymbol{J}, M) - p(\boldsymbol{J}, M)| \\ &\leqslant \epsilon + \sum_{\boldsymbol{j}\notin L;M} |\tilde{p}(\boldsymbol{J}, M) - p(\boldsymbol{J}, M)|, \end{aligned}$$

since

$$\begin{aligned} \sum_{\boldsymbol{J}\in L;M} |\tilde{p}(\boldsymbol{J}, M) - p(\boldsymbol{J}, M)| &\leqslant \sum_{\boldsymbol{J}\in L;M} \epsilon' \\ &\leqslant (n+1)|L|\epsilon' \leqslant \epsilon. \end{aligned}$$

Define also

$$p_\circ = \alpha\left(1 - \sum_{\boldsymbol{J}\in L;M} p(\boldsymbol{J}, M)\right),$$

and notice that

$$\sum_{\boldsymbol{J}\notin L;M} |p_\circ - \tilde{p}_\circ| \leqslant \left| \sum_{\boldsymbol{J}\in L;M} p(\boldsymbol{J},M) - \sum_{\boldsymbol{J}\in L;M} \tilde{p}(\boldsymbol{J},M) \right| \leqslant \sum_{\boldsymbol{J}\in L;M} |p(\boldsymbol{J},M) - \tilde{p}(\boldsymbol{J},M)| \leqslant \epsilon.$$

By the triangle inequality:

$$\sum_{\boldsymbol{J}\notin L;M} |\tilde{p}(\boldsymbol{J},M) - p(\boldsymbol{J},M)| = \sum_{\boldsymbol{J}\notin L;M} |\tilde{p}_\circ - p(\boldsymbol{J},M)| \leqslant \sum_{\boldsymbol{J}\notin L;M} |p_\circ - \tilde{p}_\circ| + \sum_{\boldsymbol{J}\notin L;M} |p_\circ - p(\boldsymbol{J},M)| \leqslant \epsilon + \sum_{\boldsymbol{J}\notin L;M} |p_\circ - p(\boldsymbol{J},M)|.$$

We now use the set S from from Lemma 4. $S \subseteq L$ by the defining property of L. It follows that

$$\sum_{\boldsymbol{J}\notin L;M} p(\boldsymbol{J},M) \leqslant \sum_{\boldsymbol{J}\notin S;M} p(\boldsymbol{J},M) \leqslant 2\epsilon.$$

Notice that

$$\sum_{\boldsymbol{J}\notin L;M} p_\circ = \sum_{\boldsymbol{J}\notin L;M} \left(\alpha \sum_{\boldsymbol{J}'\notin L;M'} p(\boldsymbol{J}',M') \right) = \sum_{\boldsymbol{J}'\notin L;M'} p(\boldsymbol{J}',M').$$

This gives

$$\sum_{\boldsymbol{J}\notin L;M} |p_\circ - p(\boldsymbol{J},M)| \leqslant \sum_{\boldsymbol{J}\notin L;M} p_\circ + \sum_{\boldsymbol{J}\notin L;M} p(\boldsymbol{J},M) = 2 \sum_{\boldsymbol{J}\notin L;M} p(\boldsymbol{J},M) \leqslant 4\epsilon,$$

which leads to

$$\sum_{\boldsymbol{J}\notin L;M} |\tilde{p}(\boldsymbol{J},M) - p(\boldsymbol{J},M)| \leqslant 5\epsilon,$$

and

$$\|\tilde{p} - p\|_1 \leqslant 6\epsilon.$$

We now show how to classically sample $\tilde{p}$.

Theorem 2. Assume that $p(\boldsymbol{J},M)$ is ϵ-approximate t-sparse. It can be sampled classically in $\mathsf{poly}(n, \frac{1}{\epsilon}, t)$ time to 6ϵ error in the total variational distance.

Proof. Use the Kushilevitz-Mansour algorithm in Theorem 1 with threshold $\theta = \frac{\epsilon}{t}$ to find L. Compute $b = \sum_{\boldsymbol{J}\in L;M} \tilde{p}(\boldsymbol{J},M)$. Flip a coin with a bias b.

(1) With probability b, output a sample drawn from $\tilde{p}(\boldsymbol{J},M)/b$ for $\boldsymbol{J} \in L$ and corresponding M.

(2) With probability $1-b$, output $(\boldsymbol{J},M)$ for $\boldsymbol{J} \notin L$ uniformly randomly.

To sample $(\boldsymbol{J},M)$ uniformly randomly, generate a random bitstring with $n-1$ bits and check if it encodes a Yamanouchi symbol. This can be verified by checking that any prefix of $m \leqslant n-1$ bits has at most $\lceil \frac{m}{2} \rceil$ zeros. Once found, generate a random integer M' from $[n+1]$. Check if $M' \leqslant 2J+1$. If yes, define $M = (M' - J - 1)$ and output $(\boldsymbol{J},M)$. Otherwise repeat. A valid Yamanouchi symbol will be found in $\mathsf{poly}(n)$ trials by a dimensionality argument (Appendix A 3). This procedure samples the probability distribution $\tilde{p}$ defined above, which has been shown 6ϵ close in the total variational distance to p.

While the above algorithm runs in $\tilde{p}$ in $\mathsf{poly}(\frac{1}{\epsilon}, n, t)$ time, it discards a significant amount of paths during the uniform sampling of paths, which may be an unnecessary bottleneck for the eventual implementation. We explain how to avoid this problem by an alternative algorithm for sampling the paths, based on the Greene-Nijenhuis-Wilf algorithm [33] in Appendix B.

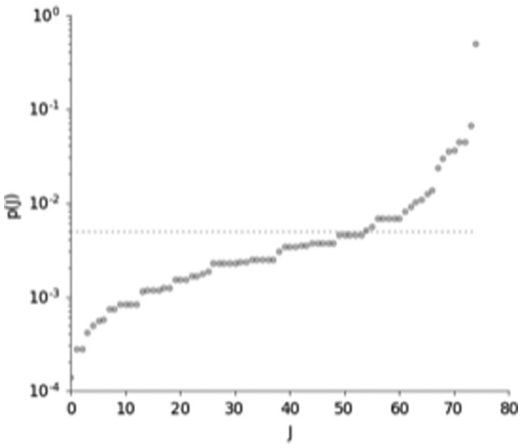


FIG. 7. An output distribution of permutational quantum computing in the sequentially coupled basis that does not satisfy the sufficient sparsity condition for $n = 10$. The horizontal line labels the $(2n^2)^{-1}$ threshold. The distribution is actually $1/10$-approximate 21-sparse and 'fools' the proxy criteria by having a single overwhelmingly large element. The p-axis is logarithmic.

V. HOW SPARSE IS THE OUTPUT?

We consider the range of applicability of the outlined algorithm. Since the set of classical gates W is large, we limit this analysis to PQC in the sequentially coupled basis and study the output distributions for $n \leqslant 10$ qubits. We randomly chose five paths and consider ten random permutations for each. This gives 50 sets of output distributions with dimension d determined by J of each path. Recall that d can be exponentially large in n.

All chosen distributions contained an element greater than $\frac{1}{2n}$. As a sufficient condition for $\frac{1}{n}$-approximate $2n^2$-sparsity by Lemma 4, we checked if the sum of all elements less than $\frac{1}{2n^2}$ is less than $\frac{1}{2n}$. Distributions for permutations on four to nine qubits all have this property, while the fraction that do not have it for $n = 10$ qubits was estimated to be less than 0.1%. Being a sufficient condition, some of these distributions are nevertheless very far from flat—an example is shown in Fig. 7.

We also consider a stricter sufficient condition: For all J-blocks with dimension $d > n$, we computed the fraction of output distributions for which the sum of all elements except for the largest $C(\log_2 d)^D$ ones is less than $1/\log_2 d$ for some constants C and D. Since $d < 2^n$, this condition suffices for $2n$-approximate (Cn^D)-sparsity of the output. Almost all of the distributions, with the exception of about 0.4% of those for $n = 9$, were $2\log(d)$-approximate $\log(d)^2$-sparse. While we were not able to prove that a significant fraction of the output distributions are ϵ-approximate t-sparse for some $t =$

$\mathsf{poly}(n)$ and $\epsilon = 1/\mathsf{poly}(n)$, the results give some indication that close-to-sparse output distributions could be common for the relevant regime of PQC.

VI. CIRCUITS WITH ANCILLAS

The proposed simulation technique extends to quantum Schur sampling circuits with ancilla qubits, with transition amplitudes given by

$$(\langle 0|^{k'}\langle \boldsymbol{J}', M'|)W(|\boldsymbol{J}, M\rangle|0\rangle^{k}),$$

for $\boldsymbol{J}' \in \mathcal{A}_{n'}$ and $\boldsymbol{J} \in \mathcal{A}_n$, such that $k' + n' = k + n$. Note that $W(|\boldsymbol{J}, M\rangle|0\rangle^k)$ is CT. Since the marginal approximation of Lemma 3 relies only on approximating overlaps of the form

$$(\langle \boldsymbol{j}, m|\langle\phi|)U_{\mathsf{SWAPS}}(|\phi\rangle|\boldsymbol{j}, m\rangle),$$

where $|\phi\rangle$ is a CT state, it also extends to marginals,

$$(\langle 0|^{k'}\langle \boldsymbol{j}, m|\langle\phi|)U_{\mathsf{SWAPS}}(|\phi\rangle|\boldsymbol{j}, m\rangle|0\rangle^{k'}),$$

since $|\phi\rangle|\boldsymbol{j}, m\rangle|0\rangle^{k'}$ is CT (see Ref. [32] for details).

We give some evidence that these circuits can give rise to computationally interesting structures, largely inspired by Ref. [23]. Prepare

$$U_{\mathsf{Sch}}H^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}}\sum_{\boldsymbol{J},M}|\boldsymbol{J}, M\rangle$$

and consider a classical circuit W that encodes the path $\boldsymbol{J}$ a the Yamanouchi symbol x into an ancilla register. This should done *before* the Schur transform as the W gate is generally controlled in the computational basis. A way to implement this is to use the form of QST which encodes the information about irreps explicitly into the computational basis input at the expense of logarithmic overhead in number of qubits [4,18]. Additionally, compute the value of J to another ancilla register of $\lceil \log n - 1 \rceil$ qubits, giving the state

$$\frac{1}{\sqrt{2^n}}\sum_{\boldsymbol{J},M}|\boldsymbol{J}, M\rangle|x(\boldsymbol{J})\rangle|J\rangle.$$

Apply the permutation gate U_π to the first register. After applying the gate sequence $H^{\otimes n}U_{\mathsf{Sch}}^{\dagger}$ and measuring the first n qubits and the J register, we have

$$p(\underbrace{0\ldots0}_{n}, J) = \frac{1}{4^n}\left|\sum_{\boldsymbol{J}}\langle \boldsymbol{J}|U_\pi|\boldsymbol{J}\rangle\right|^2,$$

where $\sum_{\boldsymbol{J}}$ runs over all paths that end at J. Here $T(\pi) = \sum_{\boldsymbol{J}}\langle \boldsymbol{J}|U_\pi|\boldsymbol{J}\rangle$ is the trace of U_π over the J-block, which is (up to a sign) the square of the character of the conjugacy class of π of the S_n irrep. defined by J. This quantity was shown to be #P-hard in Ref. [34], so we know that there exist $\pi \in S_n$ for which *exact* computation of $T(\pi)$ becomes intractable under the standard complexity theoretic assumptions. Despite the fact that an efficient classical method for computing *additive* approximations to this quantity was given by Ref. [23] (its existence is in fact a consequence of Theorem 1), it is still possible that its multiplicative approximation retains hardness. This could lead to another class of probability distributions unlikely to be sampled from classically akin to Refs. [35–37]. On the discouraging side, limitations of the quantum "Fourier-Schur" sampling in the context of addressing the hidden subgroup problem were identified in Ref. [19].

VII. DISCUSSION

Circuits using the QST underpin a diverse range of protocols in quantum information processing, from state discrimination to computational models such as PQC. While studying the computational power of the transform, we singled out a class of circuits with QST blocks that extend a computationally interesting regime of PQC. The key result that enabled this analysis was the efficient approximation of quantum Schur sampling circuits studied in Ref. [21] as means to characterize its computational power. Building on the work of Schwarz and Van den Nest [3,32], we showed that large elements of the output distributions can be efficiently found, which precludes the possibility that the circuits could encode quantities that would be hard to classically approximate by taking polynomial number of samples. We subsequently proved that these circuits can be classically efficiently approximately sampled from if their output distribution becomes sufficiently close to a sparse one.

Our algorithm is a random walk on the angular momentum branching diagram associated with the computation. One distinctive feature of the algorithm is then that it is not limited to the angular momentum and can be extended to other branching diagrams. It will remain efficient as long as the counterparts of the Clebsch-Gordan coefficients remain efficiently computable to high precision and the out degree of any vertex of the branching diagram will be bounded by a constant (see also the discussion in Ref. [21]). One of the interesting cases where our techniques could apply with some adaptation is the case of q-deformations of the SU(2) branching diagrams, applied in the study of topological phases of matter [38,39].

Circuits using a similar structure but using an SU(d) Schur-Weyl transformation for $d > 2$ were recently applied in study of boson sampling with partially distinguishable bosons in the first quantization [40]. The possibility of leveraging the simulation techniques proposed here in this context remains open.

ACKNOWLEDGMENTS

V.H. was supported by Keble de Breyne and Clarendon scholarship at the University of Oxford. S.S. was supported by the Leverhulme Early Career Fellowship. K.T. acknowledges support from the IBM Research Frontiers Institute. Authors are grateful to Richard Jozsa and Greg Kuperberg for reading the paper and acknowledge discussions with Will Kirby, Alex Moylett, and Peter Turner that helped to improve this manuscript.

APPENDIX A

1. $A \supseteq B$, then S_A and S_B commute

For sets A, B of qubits, S_A^2 and S_B^2 commute if A and B are disjoint or one is subset of the other. Setting $A \supseteq B$, we have

$$\sum_{k\in A}\vec{S}_k = \overbrace{\sum_{k\in B}\vec{S}_k}^{\vec{\alpha}} + \overbrace{\sum_{k\in A/B}\vec{S}_k}^{\vec{\beta}}.$$

Note that

$$[\vec{\alpha}, \vec{\beta}] = 0, \quad [\vec{\alpha}^2, \vec{\beta}] = \vec{0}.$$

This gives

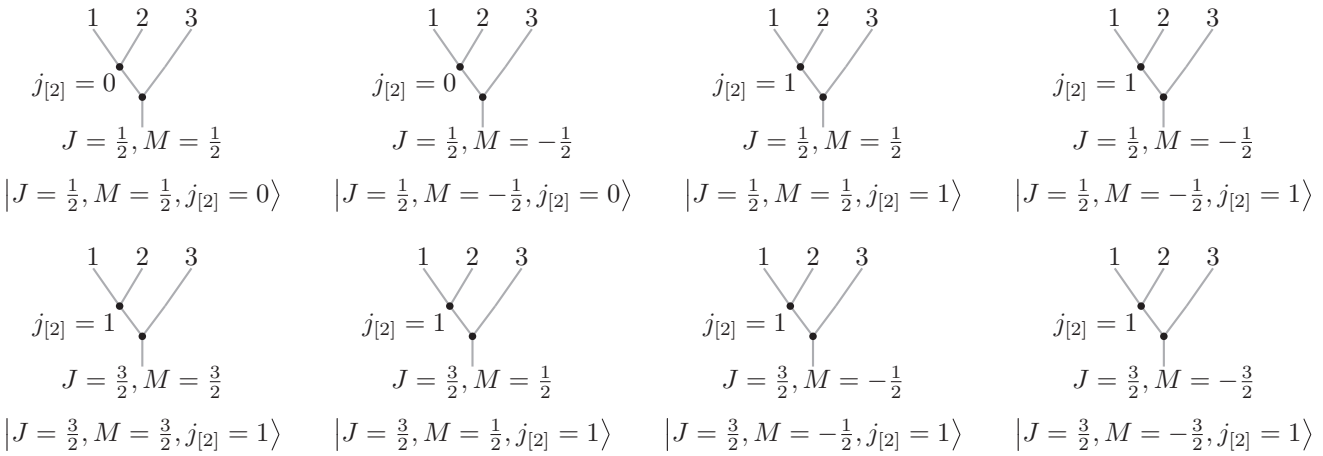
$$\left[S_A^2, S_B^2\right] = 2[\vec{\alpha} \cdot \vec{\beta}, \vec{\alpha}^2] = 2[\vec{\alpha} \cdot, \vec{\alpha}^2]\vec{\beta} = 0.$$

2. Z_A and S_B^2 commute for $A \supseteq B$

This can be seen by

$$\begin{aligned} 8\left[Z_A, S_B^2\right] = 8\left[Z_B, S_B^2\right] &= \sum_{k,l,m\in B} [Z_k, (X_l X_m + Y_l Y_m + Z_l Z_m)] \\ &= \sum_{k,l,m\in B} [Z_k, X_l]X_m + X_l[Z_k, X_m] + [Z_k, Y_l]Y_m + Y_l[Z_k, Y_m] \\ &= 2i \sum_{k,l\in B} Y_l X_k + X_l Y_k - Y_l X_k - X_l Y_k = 0. \end{aligned}$$

3. Diagrammatic representation of the spin-basis states



APPENDIX B: COMPLETENESS OF THE SEQUENTIALLY COUPLED BASIS

The argument comes from Ref. [28]. Denote the number of paths in $\mathcal{A}_k$ that end at $j_{[k]}$ by $d(j_{[k]})$. It follows from Eqs. (2) that such $j_{[k]}$ can be reached by taking a step in a path $\boldsymbol{j}_{k-1} \in \mathcal{A}_{k-1}$ that ends either at $j_{[k-1]} + \frac{1}{2}$ or $j_{[k-1]} - \frac{1}{2}$. This gives a recurrence,

$$d(j_{[k]}) = d\left(j_{[k-1]} - \tfrac{1}{2}\right) + d\left(j_{[k-1]} + \tfrac{1}{2}\right),$$

which is solved by

$$d(J) = \binom{n}{\frac{1}{2}n - J} - \binom{n}{\frac{1}{2}n - J - 1}.$$

The J eigenspaces are $2J + 1$-degenerate due to possible values of the M number. We then have that

$$\sum_{\boldsymbol{J}\in\mathcal{A}_n} (2J + 1) = \sum_{J=0}^{n} (2J + 1) d(J) = 2^n. \tag{B1}$$

It follows that eigenstates of $\mathcal{S}_n$ span the n-qubit Hilbert space. This also implies that there exist exponentially large blocks for fixed J that asymptotically scale as 2^n, since the summation in Eq. (B1) runs only over polynomially many J. In particular, this makes the sampling algorithm of Theorem 3 run in polynomial time.

APPENDIX C: PATHS TO YOUNG TABLEAUX

Here we show that the paths are one to one with the standard Young tableaux on two rows, which we use to give an improved sampling method in Appendix C. Let $\boldsymbol{J} \in \mathcal{A}_n$ be a path and let

$$x = x_1 x_2 \ldots x_{n-1} \in \{0, 1\}^{n-1}$$

be its Yamanouchi symbol. The shape of the corresponding standard Young tableau is determined by J and n—it will have $\frac{n}{2} + J$ boxes in the first row and $\frac{n}{2} - J$ boxes in the second row. Write 1 to the first box in the upper row, then read the Yamanouchi symbol x from left to right. If the ith bit $x_i = 0$, add an element $i + 1$ to the leftmost empty box in the *lower* row. Conversely, if $x_i = 1$, add $i + 1$ to the leftmost empty box in the *upper* row. The resulting Young tableau is in

the standard form (its elements are increasing both along its rows and columns). The elements in each row are increasing by construction. The elements in each column also increase, which can be seen from the property that any prefix of length $m \leqslant n-1$ of the Yamanouchi bitstring contains at most $\lceil \frac{m}{2} \rceil$ zeros—in other words, the upper row will always be filled faster than the lower one. Paths are also *onto* the standard two-row Young tableaux, which can be proved by converting the tableaux to bitstrings by reversing the above algorithm and checking the defining property of the Yamanouchi symbol.

As an example, take the sequentially coupled basis state on $n = 3$ qubits,

$$\left|J = \tfrac{1}{2}, M = \tfrac{1}{2}, j_{[2]} = 0\right\rangle = |\boldsymbol{J}, M\rangle,$$

with $\boldsymbol{J} = [\frac{1}{2} \to 0 \to \frac{1}{2}]$. The path ends at $J = \frac{1}{2}$, which means that the corresponding Young diagram will have two boxes in the upper and one in the lower rows:

The path for this state is $[\frac{1}{2} \to 0 \to \frac{1}{2}]$, which gives a Yamanouchi symbol $\searrow\nearrow = 01$. It also gives a prescription to fill the Young diagram by the above algorithm, giving the tableau

$$01 \cong \begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 \\ \cline{1-1}\end{array},$$

so the quantum state can be equivalently labeled as:

$$\left|M = \frac{1}{2}, \begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 \\ \cline{1-1}\end{array}\right\rangle.$$

There is a one-to-one correspondence between the semistandard Young tableaux of the same shape filled with $\uparrow, \downarrow$, and M—see Ref. [18] for discussion of this. However, since M and n completely determine the filling in this case, there is no need to use this here.

APPENDIX D: SAMPLING WITH THE GREENE-NIJENHUIS-WILF ALGORITHM

We now describe how to sample the paths with n steps uniformly randomly using the algorithm proposed by Greene *et al.* in Ref. [33]. First, fix an endpoint of the path by sampling J from the distribution $\Pi(J) = \frac{2J+1}{2^n} d(J)$, where $d(J)$ is the number of paths that end at J, as defined in Appendix B. Take a two-row Young diagram with $\frac{n}{2} + J$ boxes in the upper and $\frac{n}{2} - J$ in the lower row and use the GNW algorithm to uniformly generate a standard Young tableaux of this shape—every such tableau is sampled with probability $\frac{1}{d(J)}$ and the sampling algorithm runs in $O(n^2)$ time. Convert the Young tableau to the Yamanouchi symbol and the corresponding path $\boldsymbol{J}$ using Appendix A 3. Lastly, choose $M \in \{-J, -J+1, \ldots J\}$ uniformly randomly. The probability of choosing a specific $(\boldsymbol{J}, M)$ is then given by

$$\Pi(J) \frac{1}{(2J+1)d(J)} = \frac{1}{2^n},$$

as wanted. The sampling procedure is then the following:

Theorem 3. Assume that $p(\boldsymbol{J}, M)$ is ϵ-approximate t-sparse. It can be sampled classically in $\mathsf{poly}(n, \frac{1}{\epsilon}, t)$ time to 6ϵ error in the total variational distance.

Proof. Use the Kushilevitz-Mansour algorithm in Theorem 1 with threshold $\theta = \frac{\epsilon}{t}$ to find L and compute $b = \sum_{\boldsymbol{J} \in L; M} \tilde{p}(\boldsymbol{J}, M)$. Flip a coin with a bias b.

(1) With probability b, output a sample drawn from $\tilde{p}(\boldsymbol{J}, M)/b$ for $\boldsymbol{J} \in L$ and corresponding M.

(2) With probability $1 - b$, output $(\boldsymbol{J}, M)$ for $\boldsymbol{J} \notin L$ uniformly randomly.

To sample $(\boldsymbol{J}, M)$ uniformly randomly, use the above algorithm to uniformly randomly generate a $(\boldsymbol{J}, M)$ and check if $\boldsymbol{J} \notin L$. If yes, output. If no, sample again. ■

APPENDIX E: SIMPLIFICATION OF THE MARGINAL PROJECTOR

The aim of this section is to simplify the marginal projector expression as

$$\Pi(\boldsymbol{j}) = \sum_M \sum_{\boldsymbol{J} \supseteq \boldsymbol{j}} |\boldsymbol{J}, M\rangle\langle \boldsymbol{J}, M| = \sum_m |\boldsymbol{j}, m\rangle\langle \boldsymbol{j}, m|,$$

for $\boldsymbol{j} \in \mathcal{A}_k$ and $\sum_{\boldsymbol{J} \supseteq \boldsymbol{j}}$ runs over all paths $\boldsymbol{J} \in \mathcal{A}_n$ that contain $\boldsymbol{j}$. The sum $\sum_m$ runs over $m \in \{-j, -j+1, \ldots j\}$.

To do so, we repeatedly use the Clebsch-Gordan orthogonality:

$$\sum_{JM} C^{JM}_{jm, j'm_2} C^{JM}_{jm', j'm'_2} = \delta_{m_2, m'_2} \delta_{mm'}.$$

As we study coupling in the sequential basis, we have that

$$\sum_{JM} C^{JM}_{jm, m_2} C^{JM}_{jm', m'_2} = \delta_{m_2, m'_2} \delta_{mm'}.$$

We have for the projector $\Pi(\boldsymbol{j})$ that

$$\begin{aligned}\Pi(\boldsymbol{j}) &= \sum_M \sum_{\boldsymbol{J} \supseteq \boldsymbol{j}} |\boldsymbol{J}, M\rangle\langle \boldsymbol{J}, M| \\ &= \sum_{\boldsymbol{J}_{n-1} \supseteq \boldsymbol{j}} \sum_{J, M} \sum_{m_n, m'_n} \sum_{M_{n-1}, M'_{n-1}} C^{J,M}_{J_{n-1} M_{n-1}; m_n} |m_n\rangle \\ &\quad \times |\boldsymbol{J}_{n-1}, M_{n-1}\rangle\langle \boldsymbol{J}_{n-1}, M'_{n-1}| \langle m'_n| C^{J,M}_{J_{n-1}, M'_{n-1}; m'_n},\end{aligned}$$

where $\sum_{\boldsymbol{J}_{n-1} \supseteq \boldsymbol{j}}$ runs over all $\boldsymbol{J}_{n-1} \in \mathcal{A}_{n-1}$ that contain $\boldsymbol{j}$. The $\sum_J$ runs over all allowed J. The Clebsch-Gordan coefficients are only nonzero for $J = |J_{n-1} \pm \frac{1}{2}|$. Using the CG orthogonality, this evaluates to

$$\begin{aligned}\Pi(\boldsymbol{j}) &= \sum_{\boldsymbol{J}_{n-1} \supseteq \boldsymbol{j}} \sum_{m_n, m'_n} \sum_{M_{n-1}, M'_{n-1}} \delta_{m_n, m'_n} \delta_{M_{n-1}, M'_{n-1}} |m_n\rangle \\ &\quad \times |\boldsymbol{J}_{n-1}, M_{n-1}\rangle \big\langle \boldsymbol{J}_{n-1}, M'_{n-1} \big| \langle m'_n| \\ &= \sum_{\boldsymbol{J}_{n-1} \supseteq \boldsymbol{j}} \sum_{M_{n-1}} \underbrace{\sum_{m_n} |m_n\rangle\langle m_n|}_{\mathbb{I}_{2\times 2}} \otimes |\boldsymbol{J}_{n-1}, M_{n-1}\rangle\langle \boldsymbol{J}_{n-1}, M_{n-1}| \\ &= \sum_{M_{n-1}} \sum_{\boldsymbol{J}_{n-1} \supseteq \boldsymbol{j}} |\boldsymbol{J}_{n-1}, M_{n-1}\rangle\langle \boldsymbol{J}_{n-1}, M_{n-1}|.\end{aligned}$$

This has the same form as the initial expression, but the projector is now defined by summing over paths with $n-1$

steps. It is possible to continue recursively and write

$$\Pi(\boldsymbol{j}) = \sum_{M_{n-i}} \sum_{\boldsymbol{J}_{n-i} \supseteq \boldsymbol{j}} |\boldsymbol{J}_{n-i}, M_{n-i}\rangle\langle \boldsymbol{J}_{n-i}, M_{n-i}|$$

for any integer $0 \leqslant i \leqslant n-k$. For $i = n-k$, one obtains that

$$\Pi(\boldsymbol{j}) = \sum_{M_k} \sum_{\boldsymbol{J}_k \supseteq \boldsymbol{j}} |\boldsymbol{J}_k, M_k\rangle\langle \boldsymbol{J}_k, M_k|.$$

Since $\boldsymbol{j} \in \mathcal{A}_k$, there is only one path contributing to $\sum_{\boldsymbol{J}_k \supseteq \boldsymbol{j}}$, the path $\boldsymbol{j}$ itself. It follows that

$$\Pi(\boldsymbol{j}) = \sum_{M_k} |\boldsymbol{j}, M_k\rangle\langle \boldsymbol{j}, M_k|.$$

We can write

$$\Pi(\boldsymbol{j}) = \sum_{m} |\boldsymbol{j}, m\rangle\langle \boldsymbol{j}, m|,$$

where the final summation runs over $m \in \{-j, -j+1, \ldots j\}$.

[1] P. W. Shor, Algorithms for quantum computation: Discrete logarithms and factoring, in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, SFCS '94, pages 124–134, Washington, DC, USA, 1994 (IEEE Computer Society).

[2] A. Y. Kitaev, Quantum measurements and the Abelian stabilizer problem, arXiv:quant-ph/9511026.

[3] M. Schwarz and M. Van den Nest, Simulating quantum circuits with sparse output distributions, Electron. Colloquium Comput. Complexity **20**, 154 (2013).

[4] A. W. Harrow, Applications of coherent classical communication and the Schur transform to quantum information theory, Ph.D thesis, Massachusetts Institute of Technology, Cambridge, MA, 2005.

[5] D. Bacon, I. L. Chuang, and A. W. Harrow, Efficient Quantum Circuits for Schur and Clebsch-Gordan Transforms, Phys. Rev. Lett. **97**, 170502 (2006).

[6] D. Bacon, I. L. Chuang, and A. W. Harrow, The quantum Schur and Clebsch-Gordan transforms: I. Efficient qudit circuits, in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '07, pages 1235–1244, Philadelphia, PA, USA, 2007* (Society for Industrial and Applied Mathematics).

[7] R. D. Gill and S. Massar, State estimation for large ensembles, Phys. Rev. A **61**, 042312 (2000).

[8] M. Keyl and R. F. Werner, Estimating the spectrum of a density operator, Phys. Rev. A **64**, 052311 (2001).

[9] M. Hayashi and K. Matsumoto, Variable length universal entanglement concentration by local operations and its application to teleportation and dense coding, arXiv:quant-ph/0109028.

[10] M. Hayashi and K. Matsumoto, Quantum universal variable-length source coding, Phys. Rev. A **66**, 022311 (2002).

[11] M. Hayashi and K. Matsumoto, Simple construction of quantum universal variable-length source coding, Quantum Inform. Comp. **2**, 519 (2002).

[12] M. Hayashi and K. Matsumoto, Universal distortion-free entanglement concentration, arXiv:quant-ph/0209030.

[13] J. Kempe, D. Bacon, D. A. Lidar, and K. B. Whaley, Theory of decoherence-free fault-tolerant universal quantum computation, Phys. Rev. A **63**, 042307 (2001).

[14] S. D. Bartlett, T. Rudolph, and R. W. Spekkens, Classical and Quantum Communication Without a Shared Reference Frame, Phys. Rev. Lett. **91**, 027901 (2003).

[15] S. D. Bartlett, T. Rudolph, and R. W. Spekkens, Reference frames, superselection rules, and quantum information, Rev. Mod. Phys. **79**, 555 (2007).

[16] A. Hayashi, T. Hashimoto, and M. Horibe, Extended quantum color coding, Phys. Rev. A **71**, 012326 (2005).

[17] W. M. Kirby and F. W. Strauch, A practical quantum algorithm for the Schur transform, Quantum Inf. Comput. **18**, 09 (2018).

[18] W. Kirby, A practical quantum Schur transform, undergraduate thesis, Williams College, 2017.

[19] A. M. Childs, A. W. Harrow, and P. Wocjan, Weak Fourier-Schur sampling, the hidden subgroup problem, and the quantum collision problem, in *STACS 2007*, edited by W. Thomas and P. Weil (Springer, Berlin, 2007), pp. 598–609.

[20] S. P. Jordan, Permutational quantum computing, Quantum Info. Comput. **10**, 470 (2010).

[21] V. Havlíček and S. Strelchuk, Quantum Schur Sampling Circuits can be Strongly Simulated, Phys. Rev. Lett. **121**, 060505 (2018).

[22] A. Marzuoli and M. Rasetti, Computing spin networks, Ann. Phys. **318**, 345 (2005).

[23] S. P. Jordan, Fast quantum algorithms for approximating some irreducible representations of groups, arXiv:0811.0562.

[24] E. Kushilevitz and Y. Mansour, Learning decision trees using the fourier spectrum, SIAM J. Comput. **22**, 1331 (1993).

[25] O. Goldreich and L. A. Levin, A hard-core predicate for all one-way functions, in *Proceedings of the Twenty-first Annual ACM Symposium on Theory of Computing, STOC '89*, (ACM, New York, NY, USA, 1989), pp. 25–32.

[26] P. Woit, *Quantum Theory, Groups and Representations: An Introduction* (Springer International Publishing, 2017).

[27] J. J. Sakurai, *Modern Quantum Mechanics,* rev. ed. (Addison-Wesley, Reading, MA, 1994).

[28] R. Pauncz, *Alternant Molecular Orbital Method*, Studies in Physics and Chemistry, No. 4 (Saunders, 1967).

[29] A. J. Coleman, *The Symmetric Group Made Easy*, Advances in Quantum Chemistry, Vol. 4 (Academic Press, 1968), pp. 83–108.

[30] H. Krovi, An efficient high dimensional quantum Schur transform, Quantum **3**, 122 (2019).

[31] These circuits extend the notion of quantum Schur sampling circuits introduced in Ref. [21], where we only considered circuits of the form $\langle \boldsymbol{J}, M|\Lambda|\boldsymbol{J}', M'\rangle$ with Λ being a Z-diagonal gate with efficiently computable elements. Our technique works for these circuits as well.

[32] M. Van den Nest, Simulating quantum computers with probabilistic methods, Quantum Info. Comput. **11**, 784 (2011).

[33] C. Greene, A. Nijenhuis, and H. S. Wilf, A probabilistic proof of a formula for the number of young tableaux of a given shape, Adv. Math. **31**, 104 (1979).

[34] C. T. Hepler, On the Complexity of Computing Characters of Finite Groups, Master thesis, University of Calgary, 1994.

[35] M. J. Bremner, R. Jozsa, and D. J. Shepherd, Classical simulation of commuting quantum computations implies collapse

of the polynomial hierarchy, Proc. R. Soc. London A **467**, 459 (2011).

[36] M. J. Bremner, A. Montanaro, and D. J. Shepherd, Average-Case Complexity Versus Approximate Simulation of Commuting Quantum Computations, Phys. Rev. Lett. **117**, 080501 (2016).

[37] S. Aaronson and A. Arkhipov, The computational complexity of linear optics, in *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing, STOC '11* (ACM, New York, NY, USA, 2011), pp. 333–342.

[38] L. Hormozi, Topological quantum compiling, Ph.D thesis, Florida State University, 2007.

[39] R. Fern, J. Kombe, and S. H. Simon, How SU(2)_4 anyons are Z_3 parafermions, SciPost Phys. **3**, 037 (2017).

[40] A. E. Moylett and P. S. Turner, Quantum simulation of partially distinguishable boson sampling, Phys. Rev. A **97**, 062329 (2018).