An efficient high dimensional quantum Schur transform

Hari Krovi

Quantum Engineering and Computing

Physical Sciences and Systems

Raytheon BBN Technologies, Cambridge, MA

February 5, 2019

The Schur transform is a unitary operator that block diagonalizes the action of the symmetric and unitary groups on an n fold tensor product $V^{\otimes n}$ of a vector space V of dimension d. Bacon, Chuang and Harrow [5] gave a quantum algorithm for this transform that is polynomial in n, d and $\log \epsilon^{-1}$, where ϵ is the precision. In a footnote in Harrow's thesis [18], a brief description of how to make the algorithm of [5] polynomial in $\log d$ is given using the unitary group representation theory (however, this has not been explained in detail anywhere[1]). In this article, we present a quantum algorithm for the Schur transform that is polynomial in n, $\log d$ and $\log \epsilon^{-1}$ using a different approach. Specifically, we build this transform using the representation theory of the symmetric group and in this sense our technique can be considered a "dual" algorithm to [5]. A novel feature of our algorithm is that we construct the quantum Fourier transform over the so called *permutation modules*, which could have other applications.

1 Introduction

Schur-Weyl duality is a remarkable correspondence between the irreducible representations of the symmetric group and those of the unitary group acting on an n fold tensor product of a vector space V. This correspondence allows one to construct all of the so-called polynomial representations of the unitary, general linear and special linear groups. Polynomial representations of matrix groups such as unitary groups are representations whose matrix entries can be written as polynomials in the entries of the group element i.e., $\rho(U)$ is a polynomial representation if the entries of $\rho(U)$ are polynomial in the entries of U. Schur-Weyl duality has been generalized to many other groups and algebras including quantum groups [10, 17].

Schur-Weyl duality has numerous applications in quantum information theory. It has been used to prove that the tensor product of many copies of a density operator is close to a projector. In fact, the projector is the one corresponding to a partition in Schur-Weyl duality that is closest to the spectrum of ρ [2, 26, 21, 9]. It has also been used to prove de Finetti theorems [30], which have many applications in security proofs of quantum key distribution systems. The Schur transform was first constructed for qubits in the work of Bacon, Chuang and Harrow in [4]. This has been extended to qudits by the same authors in [5]. A quantum circuit for the Schur transform also has numerous applications in quantum information theory. It has been applied to universal distortion-free entanglement concentration [32], universal compression [21, 22], encoding and decoding into decoherence-free subspaces [42, 29, 24, 3]. These applications and others are discussed in more detail in Harrow's thesis [18]. Recently, the Schur transform has been used as a primitive in an efficient algorithm for spectrum testing of a density operator [35] and in algorithms for sample optimal state tomography [34, 16] improving on previous algorithms [39, 14, 20, 25]. It is also used in a scheme for optimal

---

Hari Krovi: hkrovi@gmail.com, hari.krovi@raytheon.com

[1]Specifically, Harrow sketches a procedure to obtain the same labeling of irreps as the ones used for Gelfand-Tsetlin basis. Why this procedure leads to the Gelfand-Tsetlin basis itself (not just the same labeling scheme) could be explained in more detail.

compression [41], quantum superreplication [8, 7]. Recently, in [19], the authors present a classical algorithm to compute the transition amplitudes of the Schur sampling circuits.

There are two main ways in which Schur sampling is used: weak and strong sampling. Recall that with any kind of Fourier sampling, we block diagonalize the group action to obtain the Fourier basis from the computational basis. Strong (resp. weak) sampling refers to measuring all the registers (resp. only the label of the irreducible representation) to obtain information from the Fourier basis. Similarly, a circuit for Schur sampling block diagonalizes the unitary group (or equivalently the symmetric group) representation on the n fold tensor power of a d dimensional space. In this block diagonalization, each block essentially contains three pieces of information: (a) the irrep label, which is common to both the unitary and symmetric groups, (b) the unitary group irrep register i.e., the register holding the state that transforms under the unitary group and (c) the symmetric group irrep register holding the state that transforms under the symmetric group. Strong Schur sampling refers to measuring all three registers and weak Schur sampling refers to measuring only the irrep label. One could always measure all three registers in some basis to obtain more information about the state being transformed. However, in most of the above applications, one only needs to measure the irrep label register. In other words, one only needs weak Schur sampling and the associated probability distribution over the irrep labels to get enough information for all of the above mentioned applications. Weak Schur sampling can be performed with a quantum circuit that is polynomial in $\log d, n$ and $\log(1/\epsilon)$ using the so-called Generalized Phase Estimation (GPE) procedure. The construction of such a circuit for the weak Schur transform is described in [18].

In this paper, we present a circuit that can be used for strong Schur sampling using the representation theory of the symmetric group that runs in time polynomial in $\log d, n$ and $\log(1/\epsilon)$. In the previous approach by Bacon, Chuang and Harrow (BCH) [5], the Schur transform is constructed using the unitary group representation theory. By contrast, our algorithm uses the symmetric group and, in fact, uses transforms such as Beals' algorithm for the Fourier transform over the symmetric group [6] and the GPE algorithm mentioned above. To give an intuition for how the dual Schur transform works, we should look at how the original algorithm by BCH for the Schur transform works. BCH developed a quantum circuit for the Clebsch-Gordan (CG) decomposition problem for the unitary group, which entails block diagonalizing a tensor product of two irreps of the unitary group. Then they use this circuit to construct the Schur transform by applying it iteratively.

This gives us a clue as to how to construct the dual transform if we look at the so called Littlewood-Richardson (LR) coefficients. The LR coefficients give the number of times an irrep of the unitary group appears in the CG decomposition of two unitary group irreps. The same coefficients describe the decomposition of induced representations of the symmetric group from certain Young subgroups. This suggests that we need to investigate permutation modules of the symmetric group since they are exactly such induced representations. We can now see that block diagonalizing these induced representations gives us the dual Schur transform. We will not directly work with LR coefficients or the related Kostka numbers, but embed them in a larger space. This dual version is polynomial in $\log d, n$ and $\log(1/\epsilon)$, where as the one in [5] is polynomial in d, n and $\log(1/\epsilon)$. However, as mentioned above, in Harrow's thesis [18], a short description is given of how to make the algorithm from [5] polynomial in $\log d, n$ and $\log(1/\epsilon)$. A classical algorithm to compute the CG coefficients for $\mathsf{SU}(d)$ is given in [1].

This paper is organized as follows. In Section 2, we explain the necessary background in representation theory, namely the notions of induced representations, symmetric and unitary group representations and the structure of permutation modules of the symmetric group. In Section 3, we describe the construction of quantum Fourier transforms over induced representations. This construction was also used in [31] for Fourier transforms over quantum doubles over finite groups. Then, we apply this QFT for permutation modules which are essentially induced representations. In Section 4, we construct the dual algorithm for the Schur transform and show that the construction is $\mathrm{O}(\mathrm{poly}(n, \log d, \log 1/\epsilon))$ elementary operations. Finally, in Section 5, we present some conclusions and other potential applications of our subroutines.

2 Background in representation theory

2.1 Basics of induced representations

In this section, we briefly describe the representation theoretic concepts such as irreducible representations (irreps, for short), regular representations and induced representations. Induced representations are important in this article since the dual Schur transform is essentially a block diagonalization of induced representations. These concepts for finite groups are described in several texts such as [37, 12]. Here we follow the development in [12].

A representation of a finite group on a finite dimensional complex Hilbert space V is a homomorphism from the group G to the unitary group on the vector space $\mathsf{U}(V)$ i.e., a representation is $\rho : G \to \mathsf{U}(V)$. For every finite or compact group, any representation on V can be made unitary i.e., $\rho(g)$ is a unitary matrix for all $g \in G$. Very often, the space which the representation ρ maps to, is identified with the representation. Two representations ρ and ρ' of a group G acting on the same vector space V are considered equivalent if there exists a unitary U such that $U\rho(g)U^\dagger = \rho'(g)$ for all $g \in G$. A subspace W of V is called a sub-representation if $\rho(g)$ preserves W for all $g \in G$. In this case, the orthogonal complement of W in V is also a sub-representation and V can be viewed as a direct sum of these two sub-representations. A representation is called an irreducible representation (irrep) if it does not contain any non-trivial sub-representations. Any representation V can be broken up into a direct sum of sub-representations W and its complement as above. Continuing this process further and breaking up W and its complement into sub-representations, one can arrive at a decomposition of V into irreps: $V \cong V_1 \oplus \cdots \oplus V_n$, where some of the irreps in the decomposition may be equivalent. A special kind of irrep is the trivial irrep which acts on a one-dimensional vector space and takes all the group elements to the identity. Any finite group G has a finite number of irreducible representations whose number is equal to the number of conjugacy classes of G.

A regular representation of a finite group G acts on the vector space $\mathbb{C}[G]$, where g acts on any basis vector h by left multiplication $g : h \to gh$. The regular representation turns out to have the following interesting direct sum decomposition into irreps.

$$\mathbb{C}[G] \cong \bigoplus_i W_i^{\oplus d_i} \,, \tag{2.1}$$

where W_i is an irrep of G, i runs over all the different irreps of G and d_i is the vector space dimension of the irrep W_i. The quantum Fourier transform (QFT) over a group G usually refers to the transform that performs the above block diagonalization. This can be defined as the following basis transformation.

$$|g\rangle \to |\rho, i, j\rangle \,,$$

where ρ is the label of the irrep, i is the multiplicity space index and j is the irrep space index. From the above decomposition, we can see that the dimension of both the multiplicity space and irrep space are the same and so i and j run over the same index set labeling the basis vectors.

A type of representation that is of particular importance here is the induced representation defined as follows. Given a subgroup H of G and a representation (ρ, W) of H, we can construct a representation V of G as follows. As a vector space, it is the tensor product $\mathbb{C}[G/H] \otimes W$, where $\mathbb{C}[G/H]$ is the space of cosets of H in G. The action of G on this basis can be described using a transversal $G/H = \{t_1, t_2, \ldots, t_m\}$ for H in G, where m is the number of cosets i.e., $m = |G|/|H|$. This means that these elements form an orthonormal basis of the vector space $\mathbb{C}[G/H]$. Let $\{|w_1\rangle, \ldots, |w_d\rangle\}$ be a basis of W. We denote the induced representation by $\uparrow_H^G \rho$ or $\uparrow^G \rho$ (when H can be inferred). In this basis, $(\uparrow_H^G \rho)(g)$ is the following action on basis vectors (which can be linearly extended to other vectors).

$$(\uparrow_H^G \rho)(g) : |t\rangle \otimes |w_i\rangle \mapsto |t'\rangle \otimes \rho(h)|w_i\rangle \tag{2.2}$$

where $t' \in G/H$ and $h \in H$ are the unique elements for which $gt = t'h$.

2.2 Irreducible representations of symmetric and unitary groups

The symmetric group on n letters is denoted S_n and consists of all possible permutations of the n letters. There are $n!$ permutations and every element can be written as a product of transpositions, where a transposition is a swap of two letters. Here we denote a transposition between a and b as (a, b). The representation theory of the symmetric group is discussed in several books (see for example, [12, 23, 36]). The irreducible representations of S_n are labeled by Young diagrams, which are diagrams that consist of rows of boxes. A Young diagram corresponds to a partition of n, which is defined as a tuple $\lambda = (\lambda_1, \lambda_2, \ldots \lambda_k)$, where $\lambda_j \geq 0$, $\sum_j \lambda_j = n$ and $\lambda_k \geq \lambda_l$ for $k < l$. Given a partition, a Young diagram has k rows and i_j boxes in row j. In this case, we say that there are k parts in the partition λ. For example, a Young diagram corresponding to the partition $(2, 1)$ (not to be confused with a transposition) is given by ⬚⬚/⬚. A Young tableau is a Young diagram with numbers in the boxes. If the numbers are from 1 to n, increasing from left to right and increasing from top to bottom, then the Young tableau is called a standard Young tableau (SYT). For example, for the partition $(2, 1)$, an SYT could be $\begin{array}{|c|c|}\hline 1 & 3 \\\hline 2 \\\cline{1-1}\end{array}$ or $\begin{array}{|c|c|}\hline 1 & 2 \\\hline 3 \\\cline{1-1}\end{array}$. In fact, these are the only possible choices. This reflects the fact that the irrep labeled by this Young diagram is two dimensional. In general, the dimension of the irrep of S_n corresponding to a partition λ is the number of possible standard Young tableau. It is also given by the famous hook length formula

$$d_\lambda = \frac{n!}{\Pi_{i,j}\, h_\lambda(i, j)}\,, \tag{2.3}$$

where the product in the denominator is over all boxes (i, j) and $h_\lambda(i, j)$ is the hook length of a box, which is defined as the sum of the number of boxes to the right (including the box (i, j)) and the number of boxes directly below the given box (i, j). We will get back to other aspects of the symmetric group when we discuss subgroup adapted bases and permutation modules in the next section.

The unitary group $\mathsf{U}(d)$ is the group of $d \times d$ unitary matrices that is an infinite, though compact, group. The irreducible representations of this group can be labeled in several ways. Here, we describe the labeling using both Dynkin labels and Young diagrams. A Dynkin label is the set of coefficients in the so called basis of fundamental weights. Every irreducible representation has a basis whose vectors are called weight vectors and have weights associated with them. In this basis, the highest (and lowest) weight vectors are unique and, in fact, every irrep of the unitary group can be associated to a unique highest weight (and there is a one-to-one correspondence between them). It turns out that the weight vectors lie in a space (called the root space) spanned by the fundamental weights. This means that every highest weight (of any irrep) can be written as a linear combination of the fundamental weights. The weights and weight vectors of a given irrep form a special basis of the irrep of the unitary group called the Gelfand-Tsetlin basis. We will discuss this basis in more detail below. It turns out that one can use Young diagrams to label irreps of the unitary group as well. One can convert the Dynkin label representation to a Young diagram representation in the following way: if the Dynkin labels are $(l_1, \ldots l_r)$, then the corresponding partition λ has components $\lambda_i = l_i + \cdots + l_r$. Conversely, a Young diagram λ that represents an irrep can be converted to Dynkin labels by setting $l_i = \lambda_i - \lambda_{i-1}$.

2.3 Subgroup adapted bases

A subgroup adapted basis is a canonical basis for an irrep of a group G that is obtained from a tower of subgroups $G_0 = 1 \subset G_1 \subset \ldots G_n = G$ from the identity to G. To see how one obtains a canonical basis from a given subgroup tower, first consider an irrep ρ of $G = G_n$. Suppose we restrict it to the subgroup G_{n-1}, then ρ can be decomposed into irreps of G_{n-1}. Suppose that this restriction yields irreps σ_i of G_{n-1} each with multiplicity m_i, then choosing a basis for the multiplicity spaces and the irrep spaces would give us a basis for ρ. In choosing a basis for the σ_i, we can restrict to G_{n-2} and so on down the subgroup tower. Finally, we would end up with the trivial subgroup and since it has only a one-dimensional irrep, this would fix the entire basis. In the special case that each of the restrictions from G_i to G_{i-1} are multiplicity-free i.e.,

m_i are all zero or one, we get a canonical basis. In other words, there is no ambiguity in choosing the basis for the multiplicity space except for the choice of a phase for each multiplicity space.

In most applications, one makes a projective measurement in this basis, which makes this phase choice irrelevant. However, if one were to perform other quantum operations conditioned on basis vectors after the Schur transform, then the phase choice might be relevant. But in that case, one can incorporate that phase choice into the conditional operations to be performed. We will see next that there are special subgroup towers for both the symmetric and unitary groups that have multiplicity-free branching along the tower and hence lead to canonical bases.

For the symmetric group, the tower of subgroups $1 = S_1 \subset S_2 \subset \ldots S_n$, where S_i permutes the first i letters and fixes the remaining $n - i$ ones, gives a multiplicity-free branching rule from one subgroup to the next. This tower is fixed once we number the n qudit registers in some fashion. The resulting canonical basis is called Young orthonormal basis (also Young-Yamanouchi basis). This basis can be associated to Young diagrams with numbers in the boxes with the rule that the numbers are strictly increasing as one goes from left to right along a row and top to bottom along a column. As mentioned above, such numbered Young diagrams are called *standard Young tableaux* (SYT). An example is given below.

$$\lambda = \begin{array}{|c|c|c|} \hline 1 & 3 & 4 \\ \hline 2 & 5 & 6 \\ \hline 7 \\ \cline{1-1} \end{array} . \tag{2.4}$$

As is well-known, the symmetric group is generated by adjacent transpositions $(k, k+1)$. If k and $k+1$ are in different rows and columns in T, the action of any such transposition on an SYT T is given as

$$(k, k+1)|T\rangle = a_k^T |T\rangle + b_k^T |(k, k+1)T\rangle , \tag{2.5}$$

$$(k, k+1)|(k, k+1)T\rangle = b_k^T |T\rangle - a_k^T |(k, k+1)T\rangle , \tag{2.6}$$

where a_k^T is the inverse of the Manhattan distance in T between the boxes labeled k and $k+1$ and $b_k^T = \sqrt{1 - (a_k^T)^2}$. The Manhattan distance between two boxes is the number of steps needed to go up plus number of steps to the right minus number of steps to the left minus number of steps down. It can be seen easily that this does not depend on the path taken. We use the notation $|(k, k+1)T\rangle$ to denote the SYT with k and $k+1$ interchanged, which can be seen to be a SYT. If k and $k+1$ are in the same row, they must be next to each other and the action is given as

$$(k, k+1)|T\rangle = |T\rangle , \tag{2.7}$$

and if they are in the same column (and necessarily in adjacent rows) the action is

$$(k, k+1)|T\rangle = -|T\rangle . \tag{2.8}$$

For the unitary group, a subgroup tower that leads to a canonical basis is $1 = U_1 \subset U_2 \subset \ldots U_d$, where U_i is the unitary group acting on the $i \times i$ minor of the full $d \times d$ matrix. This tower is determined once we fix a basis for each qudit register. This tower, like the one for the symmetric group, gives rise to multiplicity-free branching and hence to a canonical basis. This canonical basis is called the Gelfand-Tsetlin basis. Given any irrep λ of U_d in the form of a Young diagram, one can obtain the diagrams in the restriction to U_{d-1} by removing a box from end of each column in all possible ways. If two columns have the same length in λ and the choice is to remove a box from the left column, then one must also remove the box from the right column to ensure that a valid Young diagram is obtained. An example is given below.

$$\lambda = \quad \rightarrow \quad \left\{ \quad , \quad , \quad , \quad \right\} . \tag{2.9}$$

Suppose we pick one and proceed with a choice down the tower, we would have the following possibility.

$$(2.10)$$

This sequence gives us a basis vector of the irrep λ. This can be encapsulated by putting numbers into the original irrep, which represent the stage before which the boxes are removed. For the above sequence the following numbering would hold.

$$(2.11)$$

Notice that the rows are weakly increasing i.e., the numbers either increase or stay the same as we move right and the columns are strictly increasing. Such a Young diagram is called a semi-standard Young diagram (SSYT). SSYTs with numbers taken from the set $[d]$ label basis vectors in the irrep of U_d. We will see below that SSYTs also play a role in certain induced representations of the symmetric group called permutation modules. Efficient encodings of these bases are constructed in [5] i.e., using poly($\log d$, n, $\log 1/\epsilon$) bits. For a SSYT, the tuple that contains the number of boxes labeled by a given integer is called the *content* of the SSYT. For instance, in the example above, the content is $(2, 2, 3)$ corresponding to 2 boxes numbered one, 2 boxes numbered two and 3 boxes numbered three.

SSYTs have an interesting structure that is useful in our algorithms. If we consider all the boxes containing a specific number, we find that no two of them appear in the same column. If we isolate these boxes, such a *skew* diagram is called a *horizontal strip*. An SSYT can be thought of as being composed of horizontal strips. It turns out that this composition can be made more precise as we describe in the next subsection. An example of an SSYT and the associated horizontal strips are given below.

$$(2.12)$$

In terms of horizontal strips, the above decomposition of an irrep λ of $U(d)$ into irreps μ of $U(d-1)$ can be rephrased as the set of all μ that can be obtained from λ by removing a horizontal strip in all possible ways.

2.4   RSK algorithm and composition of Young tableaux

The discussion in this section is taken from the book by Fulton [11]. This procedure is used in the algorithm **DualSchur** (step 4). For a more detailed explanation of how to obtain SYTs with permuted content, see [11] chapter 4. The RSK (Robinson-Schensted-Knuth) algorithm establishes a correspondence between pairs of words and pairs of tableaux. The main part of the RSK is a procedure called row insertion that lets one insert a letter into a tableau such that the resulting tableau is semi-standard with one more box. This correspondence has several applications, but the primary application here is to produce semi-standard Young tableaux where the content is permuted. In this subsection, we briefly describe this algorithm. The row insertion procedure takes as input a tableau T and an integer x and outputs a tableau with one more box than T. The procedure is as follows

1. Find the number in the first row of T that is greater than x.

2. If there is none, then place x in a box at the end of the first row.

3. If there are numbers greater than x in the first row, let y be the smallest among them. Place x in y's position (x 'bumps' y).

4. Repeat the previous steps with y (in the place of x) and starting with the second row.

The RSK algorithm uses the above row bumping procedure. It takes a pair of words, say $u = u_1 u_2 \ldots u_r$ and $v = v_1 v_2 \ldots v_r$ that has the following two properties as input. The first is that u is weakly increasing and second if $u_{k-1} = u_k$, then $v_{k-1} \le v_k$. Given such pairs as input the procedure produces a pair of tableaux (P, Q) iteratively as follows. Start with the base tableaux $\boxed{x}$ and $\boxed{y}$, where $x = v_1$ and $y = u_1$. Then from any pair (P_{k-1}, Q_{k-1}), row insert v_k into P_{k-1} getting P_k. Then add a box to Q_{k-1} in the position where the new box is in P_k and put u_k in this box.

This procedure allows us to define a product or composition of tableaux mentioned in the previous subsection. Suppose S and T are two tableaux, then $S \cdot T$ is defined as follows. If T consists of only one box, then the product of S and T is the result of row insertion into S. If T contains more than one box, then we row insert them one by one into S starting from the bottom left box and moving left to right along each row and upwards along the rows.

In this paper, we will need this procedure (in the algorithm **DualSchur** step 4) to create an SSYT V from another SSYT U with the content permuted. In order to describe it for any permutation, we only need to show it for a single transposition. As noted above, an SSYT consists of horizontal strips each having the same number. Now suppose that the transposition is $(k, k+1)$ i.e., if the original SSYT U contains n_k boxes numbered k and n_{k+1} boxes numbered $k+1$, then the new SSYT V should contain n_{k+1} boxes numbered k and n_k boxes numbered $k+1$. This is done by using the product defined above. It turns out [11] that we can write $U = A \cdot B \cdot C$, where A is a SSYT that contains only boxes numbered 1 through k, B is an SSYT that contains boxes numbered k and $k+1$ and C is an SSYT that contains the remaining numbers. Since B contains only two labels, it must be of the form
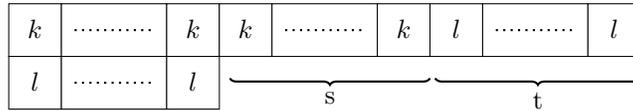
where $l = k + 1$ for brevity in the figure and the overhang consists of s boxes numbered k and t boxes numbered l. When we swap the number of ks and ls, we obtain a similar diagram where the overhang contains t ks and s ls. Denote this diagram by B'.
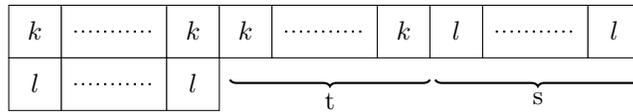
Now the new SSYT V is obtained by composing $A \cdot B' \cdot C$ using the RSK algorithm.

2.5 Gelfand-Tsetlin bases

An alternate way of representing the basis vectors of the unitary group is the so called Gelfand-Tsetlin (GT) patterns. GT patterns are useful in certain applications, although one can easily convert an SSYT to a GT pattern and vice versa. The power of GT patterns comes from the fact that in the GT basis, one can write the matrix elements of the Lie algebra $\mathsf{SU}(d)$ as derived by Gelfand and Tsetlin [13]. The formulae in this section can be found in the book by Vilenkin and Klimyk [40].

A GT pattern M is a triangle of numbers such as the one below.

$$
M = \begin{pmatrix}
m_{1,d} & & m_{2,d} & & \cdots & & m_{d,d} \\
& m_{1,d-1} & & & \cdots & m_{d-1,d-1} & \\
& & \ddots & & \iddots & & \\
& & m_{1,2} & & m_{2,2} & & \\
& & & m_{1,1} & & &
\end{pmatrix} . \tag{2.13}
$$

These numbers satisfy the *in betweenness* condition

$$
m_{k,l} \ge m_{k,l-1} \ge m_{k+1,l}, \qquad 1 \le k < l \le d . \tag{2.14}
$$

The numbers in the first row of the GT pattern correspond to the number of boxes in each row of the corresponding SSYT. The number of boxes in the SSYT with the number l in row k is $m_{k,l} - m_{k,l-1}$. The total number of boxes with the number l is therefore the difference of the row sums $\sum_k(m_{k,l} - m_{k,l-1})$ (where $m_{k,l} = 0$ if $k > l$). More systematically, in order to convert a GT pattern to an SSYT, we start from bottom-most row of the GT pattern and create a partial SSYT with one row of $m_{1,1}$ boxes labelled 1. Next, we add $m_{1,2} - m_{1,1}$ boxes to this row labelled 2 and put $m_{2,2}$ boxes in the second row labelled 2. Continuing in this way, we add $m_{1,l} - m_{1,l-1}$ boxes to the first row labelled l, $m_{2,l} - m_{2,l-1}$ boxes in row two labelled l etc. The in-betweenness conditions guarantee that the skew tableau with boxes labelled l is a *horizontal strip* i.e., a skew tableau with at most one box in each of its columns. Now, to convert from an SSYT to a GT pattern, we can use the fact that the number of boxes labelled l in the kth row is $m_{k,l} - m_{k,l-1}$ and fill in the kth diagonal.

The generators of the Lie algebra of $\mathsf{SU}(d)$ are defined as follows.

$$J_0^{(l)} = \frac{1}{2}(E^{l,l} - E^{l+1,l+1}), \tag{2.15}$$

$$J_+^{(l)} = E^{l,l+1}, \tag{2.16}$$

$$J_-^{(l)} = E^{l+1,l}, \tag{2.17}$$

where $E^{k,l}$ is the matrix with a one in the $(k,l)^{th}$ position and zeros everywhere else and $1 \leq l \leq d-1$. The matrix elements of these generators can be expressed in the GT basis. The action of these elements on a basis vector corresponding to a GT pattern $|M\rangle$ is given by

$$J_0^{(l)}|M\rangle = \left[\sum_{k=1}^{l} m_{k,l} - \frac{1}{2}(\sum_{k=1}^{l+1} m_{k,l+1} + \sum_{k=1}^{l-1} m_{k,l-1})\right]|M\rangle, \tag{2.18}$$

$$\langle M + \delta_{k,l}|J_+^{(l)}|M\rangle = \left(-\frac{\Pi_{k'=1}^{l+1}(m_{k',l+1} - m_{k,l} + k - k')\Pi_{k'=1}^{l-1}(m_{k',l-1} - m_{k,l} + k - k' - 1)}{\Pi_{k'=1,k'\neq k}^{l}(m_{k',l} - m_{k,l} + k - k')(m_{k',l} - m_{k,l} + k - k' - 1)}\right)^{1/2},$$

$$\langle M - \delta_{k,l}|J_-^{(l)}|M\rangle = \left(-\frac{\Pi_{k'=1}^{l+1}(m_{k',l+1} - m_{k,l} + k - k' + 1)\Pi_{k'=1}^{l-1}(m_{k',l-1} - m_{k,l} + k - k')}{\Pi_{k'=1,k'\neq k}^{l}(m_{k',l} - m_{k,l} + k - k' + 1)(m_{k',l} - m_{k,l} + k - k')}\right)^{1/2}.$$

Here $\delta_{k,l}$ is a triangle of numbers like a GT pattern with zeros everywhere and a one in the kth diagonal and l^{th} row. It is not a valid GT pattern on its own. In the above formulae, only those $M \pm \delta_{k,l}$ are considered that are valid GT patterns.

2.6 Schur-Weyl duality

Schur-Weyl duality refers to the fact that the actions of the symmetric group and the unitary group on $V^{\otimes n}$ are full centralizers of each other. A good description of Schur-Weyl duality can be found in the book by Goodman and Wallach [15]. In terms of representations, this can be written as follows. Suppose we pick the representation of the symmetric group on $V^{\otimes n}$ and block diagonalize it to obtain irreps as follows

$$V^{\otimes n} \cong \bigoplus_\lambda W_\lambda \otimes V_\lambda, \tag{2.19}$$

where λ runs over all the irreps of the symmetric group, V_λ is the irrep space of the irrep λ and W_λ is the multiplicity space on which the symmetric group acts trivially. Now consider the following action of the unitary group: $U^{\otimes n}$ i.e., the diagonal action where the same unitary acts on each copy of the vector space V. This action clearly commutes with the action of the symmetric group. This means that in terms of representations, W_λ is a representation of the unitary group for each λ. Schur-Weyl duality essentially asserts that W_λ is not just a representation, but rather an *irreducible representation* of the unitary group. Stated in yet another way, the same unitary transformation that block diagonalizes the symmetric group

representation into irreps also block diagonalizes the unitary group representation into irreps. In order to write this in terms of a basis, let us first pick a basis for V as $\{|1\rangle, \ldots, |d\rangle\}$. Then a basis for $V^{\otimes n}$ is the set $\{|i_1, \ldots, i_n\rangle\}$, where $i_k \in [d]$. The basis after block diagonalization can be written as $|\lambda, i, j\rangle$, where λ labels the symmetric (or unitary) group irrep, i is an index for a basis of the symmetric group irrep and j indexes the unitary group irrep basis. In terms of this, the (strong) Schur transform can be defined as the unitary transformation that changes the basis from the computational one i.e., $|i_1, \ldots, i_n\rangle$ to the block diagonal one $|\lambda, i, j\rangle$.

The label i of the unitary group irrep is essentially a GT pattern or equivalently a SSYT and the label j is a SYT since it corresponds to the Young-Yamanouchi basis element of the symmetric group. The GT basis of the unitary group irrep λ consists of a highest weight given by $(\lambda_1, \ldots, \lambda_d)$ and the highest weight vector is represented by the SSYT of shape λ and content is also λ i.e., its first row contains all ones, second row is all twos etc. In fact, the weight of any basis vector is the content of the SSYT. Therefore, all the basis vectors corresponding to SSYTs of a fixed content μ are degenerate and the multiplicity of this weight space is $K_{\lambda,\mu}$. The correspondence between the Kostka number $K_{\lambda,\mu}$ and the multiplicity of the weight space with content μ can be shown combinatorially (see for instance the book by Stanley [38]).

2.7 Permutation modules of the symmetric group

Now, let us look at so called *permutation modules* of the symmetric group, which are useful in understanding the structure of the S_n representation on the space $V^{\otimes n}$. For a more careful treatment, see the book by Sagan [36]. First, define the *type* of any n-tuple $E = (e_1, e_2, \ldots, e_n)$ with $e_i \in [d]$ to be an n-tuple $\mathcal{T}(E) = (t_1, \ldots, t_n)$, where t_i is the number of occurrences of i in E. Clearly, $\sum_i t_i = n$ and so corresponding to any E is a partition $\mu(E)$. Given a type $\mathcal{T}$, denote by $W(\mathcal{T})$, the set of all n-tuples of that type. This set can be obtained by starting with the tuple $E_0(\mathcal{T}) = (1, \ldots, 1, 2, \ldots, 2, \ldots, n, \ldots, n)$, where there are t_i elements labeled i and then applying all possible permutations to it. Now, a permutation module corresponding to $\mathcal{T}$ is the representation of S_n on the vector space with basis as the set $W(\mathcal{T})$. This basis comprises of vectors of the form $|E\rangle = |e_1, e_2, \ldots, e_n\rangle$, where $\mathcal{T}(E) = \mathcal{T}$. Let $\mu = \mu(\mathcal{T})$ denote the associated partition i.e., the tuple obtained by arranging the non-zero elements of $\mathcal{T}$ in decreasing order.

It turns out that the representation of S_n described above is an induced representation. It is induced from the trivial representation of a particular subgroup to the full group S_n. This subgroup is denoted $Y_{\mathcal{T}}$ and called a *Young* subgroup. The Young subgroup is the stabilizer of E_0 i.e., all possible permutations of S_n that preserve E_0. So the permutation module is $P(\mathcal{T}) = \uparrow_{Y_{\mathcal{T}}}^{S_n} \mathbf{1}$. It turns out that the representation of S_n on $V^{\otimes n}$ is just the direct sum of the permutation modules $P(T)$, where the sum is over all possible types $\mathcal{T}$. These permutation modules are reducible in general and decompose into irreps λ of S_n. However, not all irreps appear in this decomposition. Only those λ which dominate $\mu(\mathcal{T})$ (in a certain ordering defined below) appear in the decomposition. Their multiplicities are called Kostka numbers and are denoted $K_{\lambda\mu}$. The dominance order on the irreps or Young diagrams is the following. A Young diagram or a partition λ is said to dominate μ if $\lambda_1 + \cdots + \lambda_k \geq \mu_1 + \cdots + \mu_k$ for all $k \geq 1$.

Let us now look at the structure of the multiplicity space of any irrep in a permutation module. We would like to understand this space and its basis since, in the dual version of the Schur transform, this space leads to the irrep space of the unitary group. As mentioned earlier, the dimension of the multiplicity of λ in the permutation module of the partition μ is $K_{\lambda\mu}$. This space has a basis in terms of semi-standard Young tableau (SSYT) of shape λ and content μ (both of which are partitions of n) i.e., a Young diagram of shape λ filled with μ_1 ones, μ_2 twos etc., such that the numbers are strictly increasing in the columns and weakly increasing in the rows. As a special case, when $\lambda = \mu$, we have $K_{\lambda,\lambda} = 1$. In other words, there is only one SSYT with content and shape given by the same Young diagram. For $\lambda = (2,2)$, it is $\begin{array}{|c|c|} \hline 1 & 1 \\ \hline 2 & 2 \\ \hline \end{array}$ and it turns out that such SSYTs lead to highest weight vectors in the Gelfand-Tsetlin basis of the unitary group.

3 Quantum Fourier transforms

3.1 Precision of quantum transforms

The precision of a unitary operator can be defined as follows. Given a target unitary V, U is called an approximation to a precision ϵ if

$$\sup_{|\psi\rangle \neq 0} \frac{||(U - V)|\psi\rangle||}{|||\psi\rangle||} \leq \epsilon, \tag{3.1}$$

where $|||\psi\rangle||$ is the norm of the state $|\psi\rangle$. It can be shown [27] that a computation consisting of a sequence of L ϵ-approximate unitaries followed by a measurement that has a error probability δ has an overall error probability $\leq \delta + 2L\epsilon$.

When one has a $m \times m$ unitary matrix whose entries can be efficiently computed, one can use the Solovay-Kitaev theorem to ϵ-approximate it by a sequence of gates from a universal gate set using $O(m^2 \log^c(m^2/\epsilon))$ elementary operations. For a constant sized m, this is efficient in $\log(1/\epsilon)$. As we will see below, the QFT over the symmetric group S_n can be done in time $\mathrm{O}(\mathrm{poly}(n, \log(1/\epsilon)))$.

3.2 QFT over the symmetric group

Although implicit in steps of Beals' algorithm [6], the dependence on the precision ϵ is not written explicitly. Here we show that it is $\mathrm{poly}(\log(1/\epsilon))$. We briefly explain the steps in Beals' algorithm for a quantum Fourier transform over $\mathbb{C}(S_n)$ and the labeling of the Fourier basis used in it. The algorithm proceeds by reducing each element of the symmetric group into a product of adjacent transpositions. The set of adjacent transpositions $\{(12), (23), \ldots, (n-1\,n)\}$ generate the group and hence any element can be written as a product of adjacent transpositions. Beals' algorithm uses subgroup adapted bases, strong generating sets with small adapted diameter (techniques that have been generalized to several other groups in [33]). By inductively constructing the Fourier transform on the subgroup tower $\{S_n, S_{n-1}, \ldots, S_1\}$, the algorithm converts from the group basis $|g\rangle$ to the basis $|\lambda, i, j\rangle$. The indices i and j label the multiplicity space and irrep space, which are of the same dimension since this is the regular representation. Therefore, they are both labeled by SYTs defined above.

They can also be labeled by paths in the *Bratelli* diagram, which is a rooted tree with nodes at each level n corresponding to all the inequivalent irreps of S_n. In this tree, there are edges between a node or irrep at level n (say ρ) and a node or irrep at level $n-1$ (say σ) if σ is contained in the restriction of ρ to S_{n-1}. The multiplicity of this edge is equal to the multiplicity of σ in ρ. We will use this algorithm as a subroutine below. We will denote a QFT over S_n as QFT(S_n) in the following and a QFT over any Young subgroup $Y_{\mathcal{T}}$ by QFT$(Y_{\mathcal{T}})$.

The main steps in the algorithm can be summarized as proceeding from S_1 to S_n along the tower,

1. Embed an irrep of S_k into an irrep of S_{k+1}.

2. Apply the unitary $\rho(t)$, where ρ is an irrep of S_{k+1} and t is a transversal i.e., an element of S_{k+1}/S_k.

3. Sum over all cosets of S_k in S_{k+1}.

Each of these steps involves applying a unitary transform that is sparse (as shown in [6]) with only a constant number of non-zero entries that can be calculated efficiently. Using standard results described in the previous subsection, we can approximate the QFT in $\mathrm{poly}(\log(1/\epsilon))$.

3.3 Fourier transform over induced representations

As mentioned earlier, the usual Fourier transform is a unitary operator, which changes the basis from the basis of group elements $\{|g\rangle \mid g \in G\}$ to the block diagonal form given in (2.1). In this subsection, we use this transform to construct a Fourier transform for induced representations i.e., a transform that block

diagonalizes induced representations. It turns out that the Fourier transform for induced representations allows us to construct the dual Schur transform.

Suppose we have an induced representation from H to G of an irreducible representation σ of H i.e., we have $\uparrow_H^G \sigma$. The computational basis for this space can be written as $|t, v\rangle$, where t is an element of the transversal and v is a basis vector in the representation space of σ. This induced representation is in general reducible as a representation of G and can be decomposed as a sum of irreducible representations. The multiplicity of each irreducible representation ρ in the decomposition is equal to, using Frobenius reciprocity, the multiplicity of σ in the restriction of ρ to H. We now describe how one can perform this transform.

The change of basis we want to implement is from the basis labeled $|t, v\rangle$ to a block diagonal basis labeled, say $|\lambda, i, j\rangle$. Here λ labels the irreps that appear in the decomposition and i and j are the multiplicity and the irrep space basis vectors.

1. First, we append an ancilla register of size $|H|/\dim(\sigma)$ (which is an integer) to the initial state so that we have a register of size H (excluding the transversal register). We now embed $|v\rangle$ into this register of size H as $|\sigma, u, v\rangle$, where u labels the multiplicity space of dimension $\dim(\sigma)$.

2. Next, perform the inverse QFT over H to get the group basis $|h\rangle$. So including the transversal register, we have $|t, h\rangle$. Now, write this as $|g\rangle$. For the symmetric group, it turns out that this is easy since both t and h are specified in terms of adjacent transpositions. However, if the group basis for certain groups is defined in a complicated way, then this transform might be non-trivial.

3. Now perform the QFT over G to obtain the basis $|\lambda, i, j\rangle$. The label i is the multiplicity index, which runs over the entire dimension of λ. However, for an induced representation, it would run over a smaller set in general. Similarly for the irrep label. If the irrep and the multiplicity index labels can be ordered in such a way that the ones that occur in the decomposition are higher in the ordering, then we can easily return the ancilla register and obtain a clean transform. For the goal of block diagonalization, this ordering is not so important. However, we will see that this can be done for the case of the symmetric group.

3.4 Quantum Fourier transform over permutation modules

As explained above, permutation modules are induced representations from the trivial representation of some Young subgroup $Y_{\mathcal{T}}$ to S_n. The Young subgroup $Y_{\mathcal{T}}$ can be regarded as the stabilizer group of some n tuple $E = (e_1, \ldots, e_n)$ of type T with entries $e_i \in [d]$. This representation space is spanned by all possible permutations of E. In order to construct the Fourier transform over this space, we first re-write these vectors in a way that reflects the structure of the induced representation. In other words, we choose a transversal and think of elements of the transversal $|t\rangle$ as basis vectors. Now, we can use the algorithm to construct Fourier transforms over induced representations. For clarity, we make the steps involved more explicit here.

- First, take an ancilla of size $|Y_{\mathcal{T}}|$ (more precisely, $m = \lceil \log |Y_{\mathcal{T}}| \rceil$ qubits) with all the qubits in the state $|0\rangle$. Here $|Y_{\mathcal{T}}|$ stands for the number of elements in the subgroup $Y_{\mathcal{T}}$. The state $|0\rangle^m$ is taken to be the label of the trivial irrep of $Y_{\mathcal{T}}$.

- Then perform the inverse Fourier transform over $Y_{\mathcal{T}}$ to obtain the equal superposition over all the elements of $Y_{\mathcal{T}}$. This can now be thought of as a subspace of $\mathbb{C}(S_n)$ spanned by equal superpositions over elements of cosets of $Y_{\mathcal{T}}$ in S_n.

- On this space, we can perform the quantum Fourier transform over S_n. This produces the basis $|\lambda, i, j\rangle$ with i and j labeled by SYTs.

The set of irreps λ that appear in the decomposition are the ones that dominate the Young diagram corresponding to $Y_{\mathcal{T}}$. Similarly, the multiplicity space, although embedded inside a space of dimension d_λ, does not have support on all the vectors. It will be supported only on $K_{\lambda\mu}$ many vectors. This space is the multiplicity of the trivial representation of the subgroup $Y_{\mathcal{T}}$ when λ is restricted to $Y_{\mathcal{T}}$.

While the algorithm above does an essential block diagonalization, we would ideally like to have the multiplicity index label basis vectors of the right subspace rather than have it label the basis of a larger space. In order to do this, we need to change the basis inside multiplicity spaces to correspond to the trivial space under the action of the subgroup $Y_\mathcal{T}$. This follows from Frobenius reciprocity as mentioned earlier. It turns out this is also important to get to the Gelfand-Tsetlin basis in the dual Schur transform that we construct in the next section. To perform this base change in the multiplicity spaces, we can use generalized phase estimation (GPE) [18]. GPE is a generalization of Kitaev's phase estimation technique [28]. In this technique, one can block diagonalize any representation $\rho(g)$ if one can perform $\rho(g)$ with g as control. The main reason why we need the GPE is to organize the multiplicity space into a basis that consists of vectors that transform trivially under $Y_\mathcal{T}$. We can use this primitive to re-organize the multiplicity space.

The action of $Y_\mathcal{T}$ and S_n in the multiplicity space is the so called *right regular representation R*. This acts on Young tableau according to the Young orthonormal representation since for the right action of S_n, the multiplicity space is an irrep. The result of performing GPE on the basis vector $\begin{array}{|c|c|}\hline 1 & 2 \\ \hline 3 & 4 \\ \hline\end{array}$ (say) for $Y_\mathcal{T} = S_2 \times S_2$ would be to stabilize it. In other words, performing (12) or (34) does not affect the vector. This would then correspond to the SSYT $\begin{array}{|c|c|}\hline 1 & 1 \\ \hline 2 & 2 \\ \hline\end{array}$. Some vectors do not appear in the multiplicity space of the trivial irrep of $Y_\mathcal{T}$ in GPE. For example, the vector $\begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 & 4 \\ \hline\end{array}$ would be taken to zero by the action of $Y_\mathcal{T}$. These statements are proved later in generality. In order to attain this change of basis, we need to perform GPE. At the end of GPE, instead of standard Young tableau labeling the basis vectors of the multiplicity space, we would have $\lambda(Y_\mathcal{T})|T\rangle$, where T is a SYT. Here

$$\lambda(Y_\mathcal{T}) = \frac{1}{|Y_\mathcal{T}|} \sum_{g \in Y_\mathcal{T}} \lambda(g), \tag{3.2}$$

and $\lambda(g)$ is the representation corresponding to the partition λ evaluated at the group element g.

In order to have a controlled application inside the multiplicity space, we can first apply a controlled right multiplication and then apply the quantum Fourier transform. We can just combine GPE with the algorithm described above to get a decomposition of the multiplicity space. However, for completeness, we explicitly list out all the steps below. The algorithm for GPE and its performance guarantee are as follows [18].

GPE$_\rho(G)$

Inputs: A quantum state $|\psi\rangle$ in the representation space of ρ of a group G.

Blackbox: The ability to perform controlled multiplication in the representation ρ.

Outputs: The outcome λ of an irrep of G with probability $p_\lambda = \langle\psi|\Pi_\lambda|\psi\rangle$.

Runtime: $2T_{QFT(G)} + T_{C_\rho}$, where $T_{QFT(G)}$ is the time to perform a QFT over the group G and T_{C_ρ} is the time to perform controlled multiplication in the representation ρ.

1. Take an ancilla register consisting of $\lceil \log |G| \rceil$ qubits and create an equal superposition over elements of G.

2. Perform $C_\rho = \sum_g |g\rangle\langle g| \otimes \rho(g)$.

3. Perform a QFT(G) to the ancilla register.

4. Measure the irrep label of the ancilla register.

The black box in the above algorithm can be made explicit in the following algorithm. The representation ρ turns out to be the usual right regular representation and can be efficiently implemented. The overall algorithm to block diagonalize permutation modules is the following. This algorithm is potentially applicable
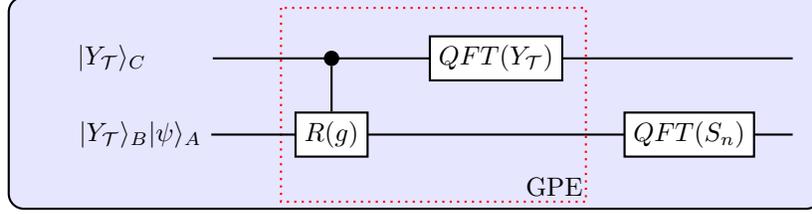
Figure 1: Circuit diagram of **QFTPermMod**. Here $|Y_\mathcal{T}\rangle$ stands for an equal superposition over the elements of the subgroup $Y_\mathcal{T}$, the registers B and C contain the state $|Y_\mathcal{T}\rangle$, register A contains an arbitrary state $|\psi\rangle$ with support on the transversal. The dotted red box shows the part that is GPE.

to other problems where one needs to block diagonalize induced representations and could be of independent interest. In the following, we will use **GPE** as a unitary module without the measurement step (step 4 above). The algorithm **QFTPermMod** described next is essentially the quantum part of the algorithm **DualSchur** described in the next section. To make things clear, we have included a block diagram of the algorithm **QFTPermMod** in Fig. 1.

QFTPermMod$(Y_\mathcal{T})$

Inputs: A quantum register A with the computational basis given by elements of the transversal of $Y_\mathcal{T}$ in S_n.

Outputs: Quantum registers $|\lambda, i, j\rangle$ corresponding to the block diagonalization of the induced representation of the trivial irrep of $Y_\mathcal{T}$ to S_n.

Runtime: $O(\text{poly}(n, \log \epsilon^{-1}))$.

1. Take an ancilla register B consisting of $\lceil \log |Y_\mathcal{T}| \rceil$ qubits and create equal superposition over elements of $Y_\mathcal{T}$.

2. Perform $\text{GPE}_R(Y_\mathcal{T})$ on register AB where R is the right regular representation of S_n.

3. Perform a $\text{QFT}(S_n)$ on the registers AB.

Theorem 1. *The above algorithm* **QFTPermMod** *performs a block diagonalization of the permutation module in time $O(poly(n, \log \epsilon^{-1}))$.*

Proof. The proof can be broken into three parts.

1. We show that for states of the form $\sum_t a_t |t\rangle$, performing GPE and measuring the irrep of $Y_\mathcal{T}$ would always give the trivial irrep.

2. We show that when the trivial irrep of $Y_\mathcal{T}$ appears in the measurement, the computational basis of the multiplicity space is rotated from SYTs to (the normalized version of) $\lambda(Y_\mathcal{T})|T\rangle$, where T is a SYT.

3. Then we show that the states $\lambda(Y_\mathcal{T})|T\rangle$ are in one-to-one correspondence to SSYTs whenever the states are non-zero.

   Part (i)

   To show that we always get the trivial irrep of $Y_\mathcal{T}$ for states of the form $\sum_t a_t |t\rangle$, where t is the transversal of $Y_\mathcal{T}$ in S_n, we track the state through the steps of the algorithm.

1. Suppose we had the state $\sum_t a_t |t\rangle^A$, where t is an element of the transversal of $Y_\mathcal{T}$ in S_n. We take an ancilla register B consisting of $\log |Y_\mathcal{T}|$ qubits to get $\sum_t a_t |t\rangle^A |0\rangle^B$.

13

2. Perform inverse QFT over $Y_{\mathcal{T}}$ on the B register to obtain an equal superposition over group elements of $Y_{\mathcal{T}}$. This allows us to view the registers A and B together as the group basis of S_n. This step takes $O(\text{polylog}\,|Y_{\mathcal{T}}|)$ time since the QFT over $Y_{\mathcal{T}}$ can be done efficiently for Young subgroups (as they are direct products of symmetric groups). We now have the state $\sum_{t,h} b_t |t\rangle^A |h\rangle^B$. Here h runs over all the elements of $Y_{\mathcal{T}}$ and $b_t = a_t/\sqrt{|Y_{\mathcal{T}}|}$.

3. Perform GPE, which consists of the following steps. This takes $O(\text{polylog}\,|Y_{\mathcal{T}}| + \text{poly}(n))$ time since controlled R operations can be done in poly n time and there are two QFTs over $Y_{\mathcal{T}}$.

   - Take a register C of $\log|Y_{\mathcal{T}}|$ qubits initialized to $|0\rangle$ and obtain an equal superposition over $Y_{\mathcal{T}}$ in it. This gives us the state

   $$\sum_{t,h_1,h_2} c_t |t\rangle^A |h_1\rangle^B |h_2\rangle^C , \tag{3.3}$$

   where h_1 and h_2 run over all elements of $Y_{\mathcal{T}}$ and $c_t = b_t/\sqrt{|Y_{\mathcal{T}}|}$.

   - Conditioned on the register C, perform a controlled right multiplication on the group basis of S_n in registers AB i.e., perform

   $$\sum_{h \in Y_{\mathcal{T}}} |h\rangle\langle h|^C \otimes R(h)^{AB} , \tag{3.4}$$

   where R is the right multiplication in S_n. This gives the state

   $$\sum_{t,h_1,h_2} c_t\, R(h_2)^{AB} \left(|t\rangle^A |h_1\rangle^B \right) |h_2\rangle^C . \tag{3.5}$$

   This state can be rewritten as

   $$\sum_{t,h_1,h_3} c_t \left(|t\rangle^A |h_3\rangle^B \right) |h_1^{-1} h_3\rangle^C , \tag{3.6}$$

   where we have replaced $h_1 h_2$ as h_3.

   - Perform QFT$(Y_{\mathcal{T}})$ on C. This gives us the following state

   $$\sum_{\mu,k,l,h_1,h_3,t} \frac{\sqrt{d_\mu}}{|Y_{\mathcal{T}}|} [\mu(h_1^{-1} h_3)]_{k,l}\, c_t\, |t, h_3\rangle^{AB} |\mu, k, l\rangle^C , \tag{3.7}$$

   where μ runs over all the irreps of $Y_{\mathcal{T}}$ and k and l run over its dimension. The sum over h_1 forces μ to be the trivial irrep of $Y_{\mathcal{T}}$. Thus, for states of the form $\sum_t a_t |t\rangle$, we will always get the trivial irrep when we measure μ.

   Part (ii)

If we had done a QFT over S_n and not performed GPE before that, we would have had the basis of the multiplicity space labeled by SYTs. Here we show that, after performing GPE and when μ (the irrep label of $Y_{\mathcal{T}}$) is trivial, the basis of SYTs is rotated to $\lambda(Y_{\mathcal{T}})|T\rangle$, where T is a SYT of shape λ. Starting with a state of the form $|th_1\rangle$ and performing GPE would give us the state

$$\sum_{\mu,k,l,h_2} \frac{\sqrt{d_\mu}}{|Y_{\mathcal{T}}|} [\mu(h_2)]_{k,l}\, R(h_2)|t, h_1\rangle^{AB} |\mu, k, l\rangle^C , \tag{3.8}$$

Next we perform a QFT over S_n on this state to get the following state

$$\sum_{h_2,\lambda,\mu,T_1,T_2,k,l} \frac{\sqrt{d_\mu d_\lambda}}{|Y_{\mathcal{T}}|n!} [\lambda(th_1)]_{T_1,T_2} [\mu(h_2)]_{k,l} (|\lambda\rangle \otimes \lambda(h_2)|T_1\rangle \otimes |T_2\rangle)^{AB} |\mu, k, l\rangle^C , \tag{3.9}$$

14

where T_1 and T_2 are SYTs. Since the right regular representation acts only on the first register, we have the $\lambda(h_2)$ acting only on $|T_1\rangle$. When μ is trivial the basis state $|T_1\rangle$, which corresponds to a SYT gets taken to $\lambda(Y_{\mathcal{T}})|T_1\rangle$.

Part (iii)

We show next that $\lambda(Y_{\mathcal{T}})|T\rangle$ can be identified with semistandard Young tableau of shape λ and content defined by $Y_{\mathcal{T}}$. In other words, if $Y_{\mathcal{T}} = S_{X_1} \times S_{X_2} \times \cdots \times S_{X_k}$ for some k, then the content is $|X_1|$ 1s, $|X_2|$ 2s etc. Also note that the sets X_i consist of consecutive integers. The SSYT associated with $\lambda(Y_{\mathcal{T}})|T\rangle$ is the one where all the integers in X_i are replaced by i. This is a valid SSYT if no two integers in X_i are in the same column or equivalently, every column has at most one element of X_i. If we isolate the boxes in the SYT labeled by elements of X_i and they have the property above, such a skew Young diagram is called a *horizontal strip* as described in section 2.3.

We now only need to show that for every X_i that if the boxes numbered with elements of X_i form a horizontal strip, then $\lambda(Y_{\mathcal{T}})|T\rangle$ is non-zero and it is zero otherwise. To show this, we focus on a single X_i and show that if there are two elements of X_i in the same column, then $\lambda(X_i)|T\rangle = 0$. This is done next in lemma 2. Finally, the claim of the dependence on ϵ follows from the fact the each of the steps in the algorithm (including the group multiplications and quantum Fourier transforms) can be done with $O(\mathrm{polylog}\,\epsilon^{-1})$ elementary gates based on the results described in section 3.1 and 3.2.

$\square$

Lemma 2. *Let A be a set of consecutive integers $\{a_1, \ldots a_k\}$, S_A be the symmetric group of size $|A|!$ permuting the elements of A. Let $|T\rangle$ be a SYT of shape λ with entries that include the set A. Let $\lambda(S_A) = \sum_\pi \lambda(\pi)$, where π runs over elements of S_A. Assume that T contains two elements of A in the same column. Then, $\lambda(S_A)|T\rangle = 0$, where $\lambda(\pi)$ is the Young orthogonal representation on SYTs.*

Proof. Since the elements of A are consecutive integers, we can assume without loss of generality that there are two elements of A that appear in the same column and in consecutive rows. Let the elements be i and j with $j > i$. We first show that if $j = i + 1$, then $\lambda(S_A)|T\rangle = 0$ and then reduce the general case to this one.

So assume now that i and $i+1$ are in the same column (they have to be in consecutive rows). The action of the transposition $(i, i+1)$ is $\lambda((i, i+1))|T\rangle = -|T\rangle$. Therefore, $\lambda(e + (i, i+1))|T\rangle = 0$, where e is the identity element. It is easy to see that for any set $K = \{i_1, i_2 \ldots i_r\}$, the symmetric group algebra element that is a sum of all possible permutations of the elements of K can be written as follows.

$$S_K = ((i_r, i_1) + \cdots + (i_r, i_{r-1})) \ldots ((i_3, i_1) + (i_3, i_2) + e)((i_2, i_1) + e) \tag{3.10}$$

While this factorization is dependent on the ordering of the elements, the overall group algebra element S_K is independent of it. Using this and writing $A = \{i, i+1, i+2, \ldots, a_k, a_1, a_2, \ldots, i-1\}$, we obtain

$$S_A = S'_A((i, i+1) + e), \tag{3.11}$$

where in S'_A, we collect the rest of the terms i.e., S'_A is a product of sums of transpositions coming from the factorization above. It is now easy to see that $S_A|T\rangle = 0$ if T contains i and $i+1$ in the same column. For the general case, assume inductively that when i and $j-1$ are in the same column, then $S_A|T\rangle = 0$. Now, suppose that i and j are in the same column and in consecutive rows.

Consider the element k, where k is the largest element between i and j such that k is in a different row from j and all elements between k and j are in the same row as j. For example, if $j-1$ is in a different row from j, then $k = j - 1$. This, in particular, means that k and $k+1$ are in different rows. If they are in the same column, then we are done. So assume that they are not in the same column. Then we have Let $(k, k+1)$ be a transposition and let $|(k, k+1)T\rangle$ be the SYT with k and $k+1$ interchanged. Since they are not in the same row or column, $(k, k+1)T$ is also a SYT. We have

$$(k, k+1)|T\rangle = a_k^T|T\rangle + b_k^T|(k, k+1)T\rangle, \tag{3.12}$$

$$(k, k+1)|(k, k+1)T\rangle = b_k^T|T\rangle - a_k^T|(k, k+1)T\rangle, \tag{3.13}$$

where a_k^T is the inverse of the Manhattan distance in T between k and $k+1$ and $b_k^T = \sqrt{1 - (a_k^T)^2}$. Using these equations, we have

$$|T\rangle = (Ae + B(k, k+1))|(k, k+1)T\rangle, \tag{3.14}$$

where $A_k = \frac{1}{b_k^T}$ and $B_k = \frac{a_k^T}{b_k^T}$. Therefore, we have

$$S_A|T\rangle = S_A(A_ke + B_k(k, k+1))|(k, k+1)T\rangle = (A_ke + B_k(k, k+1))S_A|(k, k+1)T\rangle, \tag{3.15}$$

where the last equality follows from the fact that both e and $(k, k+1)$ commute with S_A. Now note that in $(k, k+1)T$, $k+1$ and $k+2$ are not in the same row or column. Continuing this process we get

$$\begin{aligned} S_A|T\rangle =&(A_ke + B_k(k, k+1))(A_{k+1}e + B_{k+1}(k+1, k+2))\dots(A_{j-1}e + B_{j-1}(j-1, j)) \\ &S_A|(j-1, j)\dots(k, k+1)T\rangle. \end{aligned} \tag{3.16}$$

It is easy to see that $|(j-1, j)\dots(k, k+1)T\rangle$ is a SYT where i and $j-1$ are in the same column and consecutive rows. By the induction hypothesis, we have $S_A|(j-1, j)\dots(k, k+1)T\rangle = 0$.

$\square$

4 Dual algorithm for the Schur transform

We are now ready to describe our dual algorithm for the Schur transform using the above tools. It involves essentially two main steps. The first is a block diagonalization into permutation modules and the second is a block diagonalization of each permutation module into irreps using the above transform. The algorithm is as follows.

DualSchur(n, d, ϵ)

Inputs: A quantum register A with the computational basis given by n-tuples $|e_1, \dots, e_n\rangle$, where $e_j \in [d]$.

Outputs: Quantum registers $|\lambda, i, j\rangle$, where λ is the irrep label of the symmetric or unitary groups, i is the irrep label of the symmetric group in the Young orthonormal basis and j is the irrep register of the unitary group in the Gelfand-Tsetlin basis.

Runtime: $\mathrm{O}(\mathrm{poly}(n, \log d, \log \epsilon^{-1}))$.

1. Map the entries of any basis vector that are greater than n to entries inside n and keep track of this mapping. For example, $|e\rangle = |e_1, \dots, e_n\rangle$ is now $|p_e, \tilde{e}\rangle$, where $\tilde{e}$ is a vector with entries in $[n]$ and p_e is the map.

2. Convert $|\tilde{e}\rangle$ to $|\mathcal{T}, t\rangle$, where $\mathcal{T}$ is the type and t is the transversal element of the subgroup $Y_\mathcal{T}$ in S_n.

3. Conditioned on $\mathcal{T}$, apply **QFTPermMod** to $|t\rangle$ and obtain the basis $|\lambda, i, j\rangle$, where j is an SSYT with entries in $[d]$ and i is an SYT.

4. The basis at this point is $|\lambda, i, (\mathcal{T}, j)\rangle$. Use RSK to convert the pair $(\mathcal{T}, j)$ to a SSYT as described in section 2.4.

Theorem 3. *Given an n fold tensor product of d dimensional Hilbert spaces and accuracy ϵ, the quantum algorithm **DualSchur** runs in time $O(\text{poly} \log d, n, \log 1/\epsilon)$ and performs the strong Schur transform i.e., performs the change of basis from the computational basis to the block diagonal basis $|\lambda, i, j\rangle$, where λ is the symmetric or unitary group irrep, i labels the Young-Yamanouchi basis vector in the symmetric group and j labels the Gelfand-Tsetlin basis vector in the unitary group irrep.*

Proof. To prove the runtime claim, we describe the steps in more detail with an example and bound the run time.

1. Given a basis vector in the computational basis, first we map the entries of the vector that are greater than n to entries inside n and keep track of the map. So a basis vector $|e\rangle = |e_1, \ldots, e_n\rangle$ becomes $|p_e, \tilde{e}\rangle$, where $\tilde{e}$ is a vector with entries in $[n]$ and p_e is the map. This map need not be global and can be specific to the vector $|e\rangle$. This can be done in poly(n) steps. As a running example, let us consider $n = 5$, $d = 10$ and take the vector $|5, 5, 10, 3, 9\rangle$. In the first step, this gets mapped to $|10 \rightarrow 1, 9 \rightarrow 2\rangle \otimes |5, 5, 1, 3, 2\rangle$. This can be done in O(poly($n, \log d$)) steps.

2. Compute the type of the vector $\tilde{e}$ and the symmetric group element (as a product of a set of transpositions) needed to convert the standard basis vector i.e., where the entries appear in ascending order to $\tilde{e}$. The basis vector $|e\rangle$ is converted to $|p_e, \mathcal{T}, t\rangle$, where $\mathcal{T}$ is the type and t is the transversal element of the subgroup $Y_{\mathcal{T}}$ in S_n. For the example, we would have the standard basis vector as $|1, 1, 2, 3, 4\rangle$, the type is $(1, 1, 1, 0, 2)$ i.e., one 1, one 2, one 3, zero 4 and two 5s. The transversal as a product of transpositions would be $(15)(52)(53)$. This takes poly(n) time.

3. Recall that the register with $|t\rangle$ can be viewed as the induced representation of $Y_{\mathcal{T}}$ in S_n i.e., a permutation module. Conditioned on the type $\mathcal{T}$, we can apply the Fourier transform for permutation modules to obtain the basis $|p_e, \mathcal{T}, \lambda, i, j\rangle$. This step takes poly($n, \log \epsilon^{-1}$) time.

4. Rewriting this as $|\lambda, (p_e, \mathcal{T}, i), j\rangle$, we can convert $p_e, \mathcal{T}, i$ into a SSYT by using the information in p_e and $\mathcal{T}$ and rewriting the basis i, which consists of SSYT of shape λ and content $\mathcal{T}$, to match the GT basis (we prove below that this is gives the GT basis).

The claim of the dependence on ϵ follows from the fact the each of the steps in the algorithm can be done with O(polylog ϵ^{-1}) elementary gates based on the results described in section 3.1 and 3.2. This shows that the overall runtime is O(poly($\log d, n, \log \epsilon^{-1}$)).

To prove the theorem, we now need to prove that the basis obtained in the last step (labeled by SSYTs) is exactly the GT basis. The (unnormalized) basis states can be written in the form (from Eq. 3.9)

$$|\lambda, p_e, \mathcal{T}, i, j\rangle = \sqrt{\frac{d_\lambda}{|G|/|Y_{\mathcal{T}}|}} \sum_g [\lambda(gY_{\mathcal{T}})]_{i,j} |gY_{\mathcal{T}}\rangle, \tag{4.1}$$

where G is the symmetric group S_n, $Y_{\mathcal{T}}$ is a Young subgroup of the form $(S_{\mu_1} \times S_{\mu_2} \times \ldots S_{\mu_k})$ for some $k \leq d$. This follows from the next lemma. $\qquad\square$

Lemma 4. *Let G be the symmetric group S_n and $Y_{\mathcal{T}}$ be a Young subgroup corresponding to the type $\mathcal{T}$. Consider the unnormalized basis states written as*

$$|\lambda, p_e, \mathcal{T}, i, j\rangle = \sqrt{\frac{d_\lambda}{|G|/|Y_{\mathcal{T}}|}} \sum_g [\lambda(gY_{\mathcal{T}})]_{i,j} |gY_{\mathcal{T}}\rangle, \tag{4.2}$$

where the sum over g is over the cosets of $Y_{\mathcal{T}}$ in G. When $\mathcal{T}$ runs over all possible types and p_e runs over all possible subsets of n letters from the alphabet $\{1, \ldots, d\}$, the above set of vectors form the Gelfand-Tsetlin basis.

Proof. In order to prove this, we will verify that this basis is the subgroup adapted basis of the tower of subgroups $U(d) \supset U(d-1) \supset \cdots \supset U(1)$. Equivalently, we will show that this basis makes the following decomposition block diagonal.

$$V_\lambda \downarrow_{U(d-1)}^{U(d)} \simeq \bigoplus_\mu V_\mu, \tag{4.3}$$

where V_λ is the irrep of the unitary group $U(d)$ that corresponds to the shape λ and V_μ is an irrep of $U(d-1)$ that corresponds to the shape μ. The direct sum runs over all the shapes μ that differ from λ by a horizontal strip. The shapes μ are all obtained from λ by considering the various SSYTs that constitute the basis of

the irrep space of the unitary group and removing the horizontal strips labelled d. This corresponds to all unitaries in $U(d)$ that fix the label d.

The (unnormalized) basis states can be written in the form

$$|\lambda, i, j\rangle = \sqrt{\frac{d_\lambda}{|G|/|H|}} \sum_g [\lambda(gH)]_{i,j} |gH\rangle, \qquad (4.4)$$

where G is the symmetric group S_n, H is a Young subgroup of the form $(S_{\mu_1} \times S_{\mu_2} \times \ldots S_{\mu_k})$ for some $k \leq d$. To avoid clutter, we replace $Y_{\mathcal{T}}$ above by H. We will call the tuple μ, which can be permuted to a valid Young diagram, the content corresponding to H since it is the content of the SSYTs that we obtain. The sum over g runs over elements of the transversal of H in G. With the embedding of the permutation module into the group algebra of S_n done in the previous section, we can interpret $|H\rangle$ as the computational basis state $|11\ldots12\ldots2\ldots k\ldots k\rangle$ where there are μ_1 1s, μ_2 2s etc. The state $|gH\rangle$ as the basis state which is permuted according to the transversal element g. In the proof of theorem 1, it was also shown the action of H on the SYT j leads to a SSYT. In order to show the above block diagonalization, we need to show that the action of $U(d-1)$ leaves the horizontal strip labeled d intact. The action of the Lie algebra of $U(d-1)$ is generated by the operators $J_0^{(l)}$, $J_+^{(l)}$ and $J_-^{(l)}$ for $l = 1, 2, \ldots, d-2$. These operators were defined in section 2.5. We show that these operators acting on any $|\lambda, i, j\rangle$ leave the horizontal strip labeled d intact. In other words, they give superpositions of states of the form $|\lambda, i, j'\rangle$, where j' is a SSYT of shape λ and whose horizontal strip labeled d is the same as the one in j.

To show this, let us look at the action of $J_+^{(l)}$ for some l in $[d-2]$. The action of $J_-^{(l)}$ is the Hermitian conjugate and so it suffices to look at $J_+^{(l)}$. We work with the unnormalized states given above.

$$J_+^{(l)} |\lambda, i, j\rangle = J_+^{(l)} \sqrt{\frac{d_\lambda}{|G|/|H|}} \sum_g [\lambda(gH)]_{i,j} |gH\rangle = \sqrt{\frac{d_\lambda}{|G|/|H|}} \sum_{g'} c_{g'} |g'H'\rangle, \qquad (4.5)$$

where g' runs over the transversal of H' in G and H' is the Young subgroup that differs from H in the number of ls and $l+1$s with one more l and one less $l+1$ than H i.e., if the content corresponding to H is $\mu = (\mu_1, \ldots, \mu_k)$, then that of H' is $\mu' = (\mu_1, \ldots, \mu_l + 1, \mu_{l+1} - 1, \ldots, \mu_k)$. The coefficient $c_{g'}$ can be calculated to be

$$c_{g'} = [\lambda(g'JH)]_{i,j}, \qquad (4.6)$$

where the operator J is defined as

$$J = \sum_{m=\sigma_l+1}^{\sigma_{l-1}+1} (\sigma_l + 1, m), \qquad (4.7)$$

where $\sigma_l = \mu_1 + \cdots + \mu_l$ and similarly σ_{l+1}. This shows that the new SSYT is a sum SSYTs obtained by taking a SYT j and applying H and an element of J. This means that from the SSYT $\tilde{j}$, we get a sum that involves SSYTs obtained by replacing some box labeled $l+1$ by l as long as it gives a valid SSYT (since if two boxes labeled l are in the same column, then by lemma 2, that SSYT is taken to zero).

Now we show that the action of $J_0^{(l)}$ is as described in 2.5. To see this note that $E^{l,l}$ acting on the computational basis counts the number of ls in the basis vector, which means that it counts the number of boxes labeled l in the SSYT basis. Therefore since $J_0^{(l)} = (1/2)(E^{l,l} - E^{l+1,l+1})$, the action of $J_0^{(l)}$ on our basis is $(\mu_l - \mu_{l+1})/2$. Using the translation from SSYT to GT patterns described in section 2.5, for a GT pattern M defined as

$$M = \begin{pmatrix} m_{1,d} & & m_{2,d} & & \cdots & & m_{d,d} \\ & m_{1,d-1} & & & \cdots & m_{d-1,d-1} & \\ & & \ddots & & & \ddots & \\ & & m_{1,2} & & m_{2,2} & & \\ & & & m_{1,1} & & & \end{pmatrix}, \qquad (4.8)$$

18

we have

$$\mu_l = \sum_{k=1}^{l}(m_{k,l} - m_{k,l-1}).$$ (4.9)

Thus the action of $J_0^{(l)}$ can be seen to be

$$J_0^{(l)}|M\rangle = \left[\sum_{k=1}^{l} m_{k,l} - \frac{1}{2}(\sum_{k=1}^{l+1} m_{k,l+1} + \sum_{k=1}^{l-1} m_{k,l-1})\right]|M\rangle.$$ (4.10)

$\square$

5 Conclusions

We have presented an efficient algorithm for a high dimensional Schur transform that runs in time $O(\text{poly}(n, \log d, \log 1/\epsilon))$. This improves exponentially in the dimension over the prior work of Bacon, Chuang and Harrow [5]. As mentioned above, Harrow's thesis [18] contains a way to make the unitary group approach of [5] polynomial in $\log d$. Our algorithm is novel in that it uses the representation theory of the symmetric group rather than that of the unitary group. Another interesting feature is that it uses only the quantum Fourier transform and generalized phase estimation (which is also based on the QFT) and essentially no new tools. A potentially useful feature of this algorithm that could be a primitive for other problems is the circuit for a Fourier transform over induced representations. Several permutation modules, which are induced representations encode important problems that include element distinctness and collision finding. The subroutines to block diagonalize permutation modules could provide Fourier analytic algorithms to these problems and generalize to solve other problems which have permutational symmetry.

6 Acknowledgments

References

[1] Arne Alex, Matthias Kalus, Alan Huckleberry, and Jan von Delft. A numerical algorithm for the explicit calculation of su (n) and sl (n, c) clebsch–gordan coefficients. *Journal of Mathematical Physics*, 52(2): 023507, 2011. DOI: 10.1063/1.3521562. 2

[2] Robert Alicki, Slawomir Rudnicki, and Slawomir Sadowski. Symmetry properties of product states for the system of n n-level atoms. *Journal of Mathematical Physics*, 29:1158–1162, 1988. DOI: 10.1063/1.527958. 1

[3] D Bacon. *Decoherence, control, and symmetry in quantum computers*. PhD thesis, University of California, Berkeley, 2001. 1

[4] Dave Bacon, Isaac L. Chuang, and Aram W. Harrow. Efficient quantum circuits for schur and clebsch-gordan transforms. *Phys. Rev. Lett.*, 97:170502, Oct 2006. DOI: 10.1103/PhysRevLett.97.170502. URL http://link.aps.org/doi/10.1103/PhysRevLett.97.170502. 1

[5] Dave Bacon, Isaac L. Chuang, and Aram 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. ISBN 978-0-898716-24-5. URL http://dl.acm.org/citation.cfm?id=1283383.1283516. 1, 1, 1, 1, 2, 2, 2, 6, 19, 19

[6] Robert Beals. Quantum computation of fourier transforms over symmetric groups. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 48–53. ACM, 1997. DOI: 10.1145/258533.258548. 2, 10, 10

[7] G Chiribella, Y. Yang, and C. Huang. Universal super-replication of unitary gates. *Phys. Rev. Lett.*, 114(120504), 2015. DOI: 10.1103/PhysRevLett.114.120504. 2

[8] Giulio Chiribella and Yuxiang Yang. Quantum superreplication of states and gates. *Frontiers of Physics*, 11(3):110304, 2016. DOI: 10.1007/s11467-016-0556-7. 2

[9] Matthias Christandl and Graeme Mitchison. The spectra of quantum states and the kronecker coefficients of the symmetric group. *Communications in mathematical physics*, 261(3):789–797, 2006. DOI: 10.1007/s00220-005-1435-1. 1

[10] Richard Dipper, Stephen Doty, and Jun Hu. Brauer algebras, symplectic schur algebras and schur-weyl duality. *Transactions of the American Mathematical Society*, 360(1):189–213, 2008. DOI: 10.1090/S0002-9947-07-04179-7. 1

[11] William Fulton. *Young tableaux: with applications to representation theory and geometry*, volume 35. Cambridge University Press, 1997. DOI: 10.1017/CBO9780511626241. 6, 6, 7

[12] William Fulton and Joe Harris. *Representation theory*, volume 129. Springer Science and Business Media, 1991. DOI: 10.1007/978-1-4612-0979-9. 3, 3, 4

[13] Israel M. Gelfand and Michael .L. Zetlin. Finite-dimensional representations of the group of unimodular matrices. *Dokl. Akad. Nauk Ser. Fiz.*, 71:825–828, 1950. 7

[14] Richard D Gill and Serge Massar. State estimation for large ensembles. *Physical Review A*, 61(4): 042312, 2000. DOI: 10.1103/PhysRevA.61.042312. 1

[15] Roe Goodman and Nolan R Wallach. *Representations and invariants of the classical groups*, volume 68. Cambridge University Press, 2000. DOI: 10.1007/978-0-387-79852-3. 8

[16] J. Haah, A. W. Harrow, Z. Ji, X. Wu, and N. Yu. Sample-optimal tomography of quantum states. *IEEE Transactions on Information Theory*, 63(9):5628–5641, Sep. 2017. ISSN 0018-9448. DOI: 10.1109/TIT.2017.2719044. 1

[17] Tom Halverson and Arun Ram. q-rook monoid algebras, hecke algebras, and schur–weyl duality. *Journal of Mathematical Sciences*, 121(3):2419–2436, 2004. DOI: 10.1023/B:JOTH.0000024623.99412.13. 1

[18] Aram W. Harrow. *Applications of coherent classical communication and the Schur transform to quantum information theory*. PhD thesis, Massachusetts Institute of Technology, 2005. URL http://arxiv.org/abs/quant-ph/0512255. 1, 1, 2, 2, 12, 12, 19

[19] Vojtech Havlicek and Sergii Strelchuk. Quantum schur sampling circuits can be strongly simulated. *Phys. Rev. Lett.*, 121(060505), 2018. DOI: 10.1103/PhysRevLett.121.060505. 2

[20] Masahito Hayashi. Optimal sequence of quantum measurements in the sense of stein's lemma in quantum hypothesis testing. *Journal of Physics A: Mathematical and General*, 35(50):10759, 2002. DOI: 10.1088/0305-4470/35/50/307. 1

[21] Masahito Hayashi and Keiji Matsumoto. Quantum universal variable-length source coding. *Physical Review A*, 66(2):22311, 2002. DOI: 10.1103/PhysRevA.66.022311. 1, 1

[22] Masahito Hayashi and Keiji Matsumoto. Simple construction of quantum universal variable-length source coding. *Quantum Information & Computation*, 2(7):519–529, 2002. DOI: 10.1109/ISIT.2003.1228476. 1

[23] James and Kerber. *The Representation Theory of the Symmetric Group*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1984. DOI: 10.1017/CBO9781107340732. 4

[24] Julia Kempe, Dave Bacon, Daniel A Lidar, and K Birgitta Whaley. Theory of decoherence-free fault-tolerant universal quantum computation. *Physical Review A*, 63(4):042307, 2001. DOI: 10.1103/PhysRevA.63.042307. 1

[25] Michael Keyl. Quantum state estimation and large deviations. *Reviews in Mathematical Physics*, 18 (01):19–60, 2006. DOI: 10.1142/S0129055X06002565. 1

[26] Michael Keyl and Reinhard F Werner. Estimating the spectrum of a density operator. *Physical Review A*, 64(5):052311, 2001. DOI: 10.1103/PhysRevA.64.052311. 1

[27] A Yu Kitaev. Quantum measurements and the abelian stabilizer problem. *arXiv preprint quant-ph/9511026*, 1995. 10

[28] Alexei Yu Kitaev, Alexander Shen, and Mikhail N Vyalyi. *Classical and quantum computation*, volume 47. American Mathematical Society Providence, 2002. DOI: 10.1090/gsm/047. 12

[29] Emanuel Knill, Raymond Laflamme, and Lorenza Viola. Theory of quantum error correction for general noise. *Physical Review Letters*, 84(11):2525, 2000. DOI: 10.1103/PhysRevLett.84.2525. 1

[30] Robert Koenig and Graeme Mitchison. A most compendious and facile quantum de finetti theorem. *Journal of Mathematical Physics*, 50(1):012105, 2009. DOI: 10.1063/1.3049751. 1

[31] Hari Krovi and Alexander Russell. Quantum fourier transforms and the complexity of link invariants for quantum doubles of finite groups. *Communications in Mathematical Physics*, 334(2):743–777, 2015. DOI: 10.1007/s00220-014-2285-5. 2

[32] Keiji Matsumoto and Masahito Hayashi. Universal distortion-free entanglement concentration. *Physical Review A*, 75(6):062338, 2007. DOI: 10.1103/PhysRevA.98.032326. 1

[33] Cristopher Moore, Daniel Rockmore, and Alexander Russell. Generic quantum fourier transforms. *ACM Transactions on Algorithms (TALG)*, 2(4):707–723, 2006. DOI: 10.1145/1198513.1198525. 10

[34] Ryan O'Donnell and John Wright. Quantum spectrum testing. In *Proceedings of the Forty-seventh Annual ACM Symposium on Theory of Computing*, STOC '15, pages 529–538, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3536-2. DOI: 10.1145/2746539.2746582. URL http://doi.acm.org/10.1145/2746539.2746582. 1

[35] Ryan O'Donnell and John Wright. Efficient quantum tomography. In *Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing*, STOC '16, pages 899–912, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4132-5. DOI: 10.1145/2897518.2897544. URL http://doi.acm.org/10.1145/2897518.2897544. 1

[36] Bruce Sagan. *The symmetric group: representations, combinatorial algorithms, and symmetric functions*, volume 203. Springer Science & Business Media, 2013. DOI: 10.1007/978-1-4757-6804-6. 4, 9

[37] Jean-Pierre Serre. Linear representations of finite groups, translated from the second french edition by leonard l. scott. *Graduate Texts in Mathematics*, 42, 1977. DOI: 10.1007/978-1-4684-9458-7. 3

[38] Richard P Stanley. *Enumerative Combinatorics*, volume 1. Cambridge University Press, 2011. DOI: 10.1017/CBO9781139058520. 9

[39] G Vidal, JI Latorre, P Pascual, and Rolf Tarrach. Optimal minimal measurements of mixed states. *Physical Review A*, 60(1):126, 1999. DOI: 10.1103/PhysRevA.60.126. 1

[40] N. Ja. Vilenkin and Anatoliǐ Ulianovich Klimyk. *Representation of Lie Groups and Special Functions: Volume 3: Classical and Quantum Groups and Special Functions*, volume 3. Springer Science and Business Media, 2013. DOI: 10.1007/978-94-017-2881-2. 7

[41] Yuxiang Yang, Giulio Chiribella, and Masahito Hayashi. Optimal compression for identically prepared qubit states. *Phys. Rev. Lett*, 117(090502), 2016. DOI: 10.1103/PhysRevLett.117.090502. 2

[42] Paolo Zanardi and Mario Rasetti. Error avoiding quantum codes. *Modern Physics Letters B*, 11(25): 1085–1093, 1997. DOI: 10.1142/S0217984997001304. 1