Efficient classical algorithms for simulating symmetric quantum systems

Eric R. Anschuetz[1], Andreas Bauer[2], Bobak T. Kiani[3], and Seth Lloyd[4,5]

[1]MIT Center for Theoretical Physics, 77 Massachusetts Avenue, Cambridge, MA 02139, USA
[2]Dahlem Centre for Complex Quantum Systems, Freie Universität Berlin, Arnimallee 14, 14195 Berlin, Germany
[3]MIT Department of Electrical Engineering and Computer Science, 77 Massachusetts Avenue, Cambridge, MA 02139, USA
[4]MIT Department of Mechanical Engineering, 77 Massachusetts Avenue, Cambridge, MA 02139, USA
[5]Turing Inc., Cambridge, MA 02139, USA

arXiv:2211.16998v4 [quant-ph] 21 Nov 2023

In light of recently proposed quantum algorithms that incorporate symmetries in the hope of quantum advantage, we show that with symmetries that are restrictive enough, classical algorithms can efficiently emulate their quantum counterparts given certain classical descriptions of the input. Specifically, we give classical algorithms that calculate ground states and time-evolved expectation values for permutation-invariant Hamiltonians specified in the symmetrized Pauli basis with runtimes polynomial in the system size. We use tensor-network methods to transform symmetry-equivariant operators to the block-diagonal Schur basis that is of polynomial size, and then perform exact matrix multiplication or diagonalization in this basis. These methods are adaptable to a wide range of input and output states including those prescribed in the Schur basis, as matrix product states, or as arbitrary quantum states when given the power to apply low depth circuits and single qubit measurements.

1 Introduction

In the physical sciences, symmetries are useful for simplifying difficult computational tasks by reducing the effective degrees of freedom of the problem. This general principle has been used to find exact solutions to many problems, such as integrable systems [1], topological fixed-point models [2], or conformal field theories [3]. There has been a hope that similar symmetries may enable the efficiency of quantum algorithms for simulating or finding the ground state of a symmetric Hamiltonian. Indeed, it is known that there exist theoretical guarantees for quantum algorithms for finding the ground state [4] and fast-forwarding quantum dynamics [5] of Hamiltonians which commute under the action of the symmetric group S_n on qubits. It has also numerically been shown that quantum algorithms are capable of finding the ground state of certain integrable systems [6, 7] even when the symmetry is not explicitly given to the quantum algorithm *a priori*. Furthermore, prior work used Lie algebraic methods to efficiently classically simulate operators restricted to a Lie algebra whose dimension is polynomially large (independent of the potentially exponentially large Hilbert space dimension) [8, 9]. Quantum machine learning models that are symmetry equivariant are also believed to be more efficiently trainable than their general counterparts [10, 11, 12, 13, 14, 15]. These quantum models are partly inspired by classical neural network models that have enjoyed much recent success [16, 17, 18]. However, restricting quantum algorithms to problems obeying many symmetries potentially allows for efficient classical algorithms which also take advantage of these same symmetries. This raises the natural question: are there efficient classical algorithms capable of performing these tasks?

This is what we investigate here. Intuitively, we show that problems constrained by large symmetry groups yield efficient classical algorithms for computing many properties of interest, as il-

Eric R. Anschuetz: eans@mit.edu
Andreas Bauer: andibauer@zedat.fu-berlin.de
Bobak T. Kiani: bkiani@mit.edu
Seth Lloyd: slloyd@mit.edu

lustrated in Figure 1(a). We first broadly discuss very general classical algorithms for finding the ground state and energy of Hamiltonians constrained by many symmetries. We also consider the problem of simulating dynamics under symmetric Hamiltonians. We then specialize to the case of systems invariant under permutations of its qubits. Finally, we dequantize an algorithm for performing binary classification problems using permutation-invariant systems on qubits.

2 Motivation and setting

Our algorithms are motivated by the fact that symmetries significantly reduce the number of degrees of freedom for a given problem. For example, consider the classical setting of Boolean functions which are invariant under arbitrary permutations of the bits. Such functions are defined up to the orbits of the Boolean cube with respect to permutations of the bits. For a Boolean function on n bits, there are $n+1$ orbits indexed by the Hamming weight of the bitstrings. Therefore, any problem over symmetric Boolean functions need only consider a given element of each of the $n+1$ orbits to cover all possible degrees of freedom. As we will later show, the symmetric group acting over n qubits similarly reduces systems to $\mathrm{O}(n^3)$ degrees of freedom. By considering the algebra of the symmetric group on the symmetric subspace of linear operators, we will show that all these degrees of freedom can be manipulated solely through classical computation.

Before proceeding, we need to introduce important functions and definitions that will be used in this setting. We first formalize the notion of symmetry by speaking of *invariant operators*, defined in the following way:

Definition 1 (Invariant operator)**.** Given a compact group G with unitary representation $R : G \to U(N)$, a linear operator $H : \mathbb{C}^N \to \mathbb{C}^N$ is invariant under $R(G)$ if

$$R(g)HR(g)^\dagger = H, \quad \forall g \in G. \tag{1}$$

Note that any invariant operator is also an *equivariant operator* [13] in the sense that it commutes with the representation of the group.

Any operator can be projected onto the symmetric subspace induced by $R(G)$ using the twirling superoperator Re_R (more commonly known as the Reynold's operator in invariant theory) [19, 20], which maps any operator onto the set of equivariant operators:

$$\mathrm{Re}_R(M) = \frac{1}{|G|}\sum_{g\in G} R(g)MR(g)^\dagger. \tag{2}$$

Invariant subspaces of a larger Hilbert space can be identified by performing an isotypic decomposition of the representation of a group. As an example, in the case of systems invariant under permutations of the qudits, the Schur decomposition maps the computational basis into blocks of invariant subspaces. We graphically visualize this phenomenon in Figure 1(b) and provide further details in Appendix A.

Throughout this study, runtime complexities are denoted as a function of the matrix multiplication exponent ω. The best known upper bound is currently $\omega = 2.37188$ [21], which implies asymptotic runtimes of $\mathrm{O}(n^{\omega+\alpha})$ for any $\alpha > 0$ for stably performing common linear algebraic routines including eigendecomposition, SVD, and matrix inversion [22].

3 Background for general symmetry groups

In this Section, we discuss the general problem of finding the ground state energy, ground state, and performing time evolution under a Hamiltonian H on a finite-dimensional Hilbert space which is invariant under some representation R of a symmetry group G. Some of these problems have previously been considered in a variety of special cases [23, 8, 9, 24, 25]. Here, we provide background on the general techniques used in solving these problems using a unified framework. We also give an equivalent formulation in Appendix B in the language of tensor networks.

We consider the *-subalgebra of operators invariant under R to which H belongs. We think of this subalgebra as a standalone *-algebra X, such that the embedding of X into the full operator algebra defines a representation A of X. The practical relevance of these considerations is when the size of the total Hilbert space grows exponentially with some scaling parameter n. The paradigmatic example is the Hilbert space of n qubits. If there are enough symmetries, it can happen that the dimension $N(n)$ of X only grows

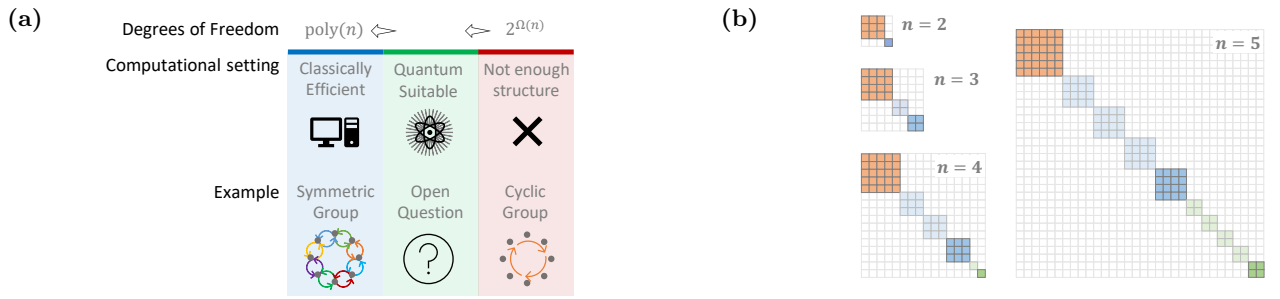


Figure 1: (a) Small groups of symmetry leave too large of an effective dimension for the problem to be tractable via quantum computation. On the contrary, very restrictive symmetries render a problem classically tractable. Between these two regions lies an area of promise where quantum computers may offer an advantage. (b) The Schur–Weyl decomposition shows that only a smaller representative subspace (indicated by darker colors) of the larger Hilbert space needs to be considered for permutation invariant operations. The size of this subspace grows as $\mathrm{O}(n^3)$ for n qubits.

polynomially with n, in which case many properties can be calculated efficiently [8]. This restriction of X to a lower-dimensional subspace may more generally happen beyond systems symmetric in the sense of Definition 1. Due to this, for now we focus explicitly on X and A, rather than on G and its representation R; we will discuss the connection of our results to G and R more specifically at the end of this Section.

For the various algorithms we now consider, we will assume that different properties of X and A are known. In the course of proving Lemma 2, we give an algorithm for finding the ground state of H. Every finite-dimensional *-algebra is isomorphic to a direct sum of irreducible blocks, and every representation is isomorphic to a direct sum of irreducible representations. That is, there is a block-diagonal orthonormal basis $|\lambda, q_\lambda, p_\lambda\rangle$ of the vector space acted upon by A, where λ labels an irrep of X, q_λ labels a basis vector internal to λ, and p_λ labels a basis vector in the multiplicity vector space of λ; this is the basis in which we compute the ground state of H (for some arbitrary and fixed multiplicity label $p_{\lambda 0}$). To prove our theorem, we assume knowledge of the matrix elements:

$$F^{i,\lambda}_{q_\lambda,q'_\lambda} := \langle \lambda, q_\lambda, p_{\lambda 0}| A_i |\lambda, q'_\lambda, p_{\lambda 0}\rangle . \tag{3}$$

In the following lemmas, we assume we are given the Hamiltonian $H \in A(X)$ as $h \in X$ expressed in the preferred basis, such that

$$H = \sum_i h_i A_i . \tag{4}$$

Though a given H may not be classically described in this exact basis, transformations into this basis are typically efficient when the number of irreps and irrep dimensions are polynomially sized. For example, a Pauli decomposition can be used to transform operators into the Pauli basis studied later in the permutation invariant setting. We first focus on the case when we are interested in finding the ground state of some representation of such a Hamiltonian, in a basis where the action of the representation is known.

Lemma 2 (Finding the ground state of symmetric Hamiltonians)**.** *Consider a subalgebra $A(X)$, and assume that the matrix elements $F^{i,\lambda}_{q_\lambda,q'_\lambda}$ are known as discussed above. Then the ground state energy and ground state of H in the $|\lambda, q_\lambda, p_{\lambda 0}\rangle$ basis can be found in time $\mathrm{O}\left(n_\lambda^2 n_q^4\right)$, where n_λ are the number of irreps of X and n_q the maximum irrep dimension.*

Proof. For each λ, we compute the operator with indices:

$$\hat{h}^\lambda_{q_\lambda,q'_\lambda} := \sum_i h_i F^{i,\lambda}_{q_\lambda,q'_\lambda} . \tag{5}$$

This takes time $\mathrm{O}\left(\dim(X)^2\right) = \mathrm{O}\left(n_\lambda^2 n_q^4\right)$. Note that in the $|\lambda, q_\lambda, p_\lambda\rangle$ basis, H has a block diagonal form. Furthermore, as p_λ labels isomorphic copies of irreps, we can find the ground state by fixing $p_{\lambda 0}$ WLOG. Namely, the ground state energy is given by:

$$\mathrm{GSE}(H) = \min_\lambda \mathrm{GSE}(\hat{h}^\lambda) , \tag{6}$$

where:

$$\mathrm{GSE}(\hat{h}^\lambda) \coloneqq \min_{|\psi\rangle} \langle\psi| \hat{h}^\lambda |\psi\rangle . \tag{7}$$

Furthermore, let

$$\lambda_{\min} \coloneqq \mathrm{argmin}_\lambda \mathrm{GSE}(\hat{h}^\lambda) \tag{8}$$

and

$$|\psi^*\rangle \coloneqq \mathrm{argmin}_{|\psi\rangle} \langle\psi| \hat{h}^{\lambda_{\min}} |\psi\rangle \ . \tag{9}$$

Then, for any p,

$$|\lambda_{\min}, \psi^*, p\rangle \tag{10}$$

is a ground state in the $|\lambda, q_\lambda, p_\lambda\rangle$ basis. The dimension of $\hat{h}^\lambda$ is $\dim_X(\lambda) \times \dim_X(\lambda)$, and thus and thus diagonalizing it takes time $\mathrm{O}\left(n_q^\omega\right)$. As we need to do this for $\mathrm{O}(n_\lambda)$ many values of λ, the total runtime for finding $|\psi^*\rangle$ is $\mathrm{O}\left(n_\lambda n_q^\omega\right)$. So the runtime of the computation in Eq. (5) dominates, and the overall runtime is $\mathrm{O}\left(n_\lambda^2 n_q^4\right)$. □

We now show that the dynamics of an initial state ρ under equivariant unitaries can be classically simulated even if $\rho \neq A(X)$. This generalizes similar approaches taken in [23] and [25] for both free fermions and fermionic systems with particle number symmetry. The Lemma as written assumes the initial state is given in a basis where the multiplicity indices have been traced out; in Section 4 we will give concrete examples on how one can compute this efficiently under a variety of input models.

Lemma 3 (Simulating equivariant dynamics)**.** *Let*

$$O = \sum_i o_i A_i \tag{11}$$

be a projective measurement and

$$U = \sum_i u_i A_i \tag{12}$$

a unitary operator. Let ρ be a quantum state, and let $\rho_{(\lambda,q_\lambda,p_\lambda),(\lambda',q'_\lambda,p'_\lambda)}$ be the coefficients of the state in the $|\lambda, q_\lambda, p_\lambda\rangle$ basis. Assume we are given the reduced density matrix

$$\tilde{\rho}^\lambda_{q_0,q_1} \coloneqq \sum_{p_\lambda} \rho_{(\lambda,q_\lambda,p_\lambda),(\lambda,q'_\lambda,p_\lambda)} . \tag{13}$$

Then,

$$\ell(\rho) = \mathrm{tr}\left(OU\rho U^\dagger\right) \tag{14}$$

can be computed in time $\mathrm{O}\left(n_\lambda^2 n_q^4\right)$.

Proof. After going to the $|\lambda, q_\lambda, p_\lambda\rangle$ basis, O and U become block diagonal, so we can compute $\mathrm{tr}\left(OU\rho U^\dagger\right)$ by restricting to fixed $p_{\lambda 0}$. To this end, we compute

$$\tilde{o}^\lambda_{q_\lambda,q'_\lambda} \coloneqq \sum_i o_i F^{i,\lambda}_{q_\lambda,q'_\lambda} . \tag{15}$$

and

$$\tilde{u}^\lambda_{q_\lambda,q'_\lambda} \coloneqq \sum_i u_i F^{i,\lambda}_{q_\lambda,q'_\lambda} . \tag{16}$$

Given the matrix elements $F^{i,\lambda}_{q_\lambda,q'_\lambda}$, this takes time $\mathrm{O}\left(\dim(X)^2\right) = \mathrm{O}\left(n_\lambda^2 n_q^4\right)$. Then we have

$$\ell(\rho) = \mathrm{tr}\left(OU\rho U^\dagger\right) = \sum_\lambda \mathrm{tr}\left(\tilde{o}^\lambda \tilde{u}^\lambda \tilde{\rho}^\lambda (\tilde{u}^\lambda)^\dagger\right) \ . \tag{17}$$

The right-hand side can be computed in time $\mathrm{O}\left(n_\lambda n_q^\omega\right)$, as it is given by the matrix multiplication of n_λ blocks each of size at most $n_q \times n_q$. □

In the above considerations, the group G and representation R do not directly enter. In practice, however, we might want to start with those two. The irreps of X are in one-to-one correspondence with those of R. By a simple corollary of the von Neumann bicommutant theorem, multiplicities of irreps in X are the dimensions of the irreps of G, and the dimensions of the irreps of G are the multiplicities of the irreps of A. We thus have that:

$$\dim(X) = \sum_\lambda \dim_X(\lambda)^2 = \sum_\lambda \mathrm{mult}_R(\lambda)^2 . \tag{18}$$

Thus, the problems discussed above become classically tractable if the number n_λ of irreps of G with non-zero multiplicity in R, as well as the maximum multiplicity n_q of an irrep λ in R, are both polynomially small.

4 Permutation invariance on qubits

We will now apply the general algorithms of Section 3 to the case where G is given by S_n and R is the representation on n qubits acting by permutations,

$$\begin{aligned} &R(\pi) |i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_n\rangle \\ &\quad = |i_{\pi^{-1}1}\rangle \otimes |i_{\pi^{-1}2}\rangle \otimes \cdots \otimes |i_{\pi^{-1}1}\rangle \ . \end{aligned} \tag{19}$$

A straightforward basis for the algebra of invariant operators can be obtained by applying the

Reynold's operator in Eq. (2) to the Pauli basis. Normalizing such that all operators A_i are sums of unit norm Pauli terms, we obtain

$$A_{\boldsymbol{i}} = \frac{1}{i_1! i_x! i_y! i_z!} \cdot \sum_{\pi \in \mathrm{S}_n} R(\pi) \left(\sigma_1^{\otimes i_1} \otimes \sigma_x^{\otimes i_x} \otimes \sigma_y^{\otimes i_y} \otimes \sigma_z^{\otimes i_z} \right) R^{-1}(\pi) , \tag{20}$$

for every 4-tuple of positive integers

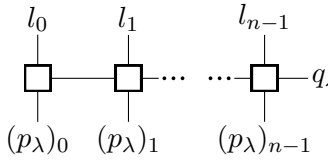
$$\boldsymbol{i} = (i_1, i_x, i_y, i_z) : i_1 + i_x + i_y + i_z = n , \tag{21}$$

where σ denote Pauli operators, and $\sigma_1 = \mathrm{id}_2$.

Enumerating such 4-tuples shows that the dimension of the algebra X is of order $\mathrm{O}(n^3)$. Therefore, the algorithms of Section 3 would reduce the task of finding the ground state (energy) of a Hamiltonian to a polynomial runtime in n, compared to exact diagonalization of the full Hamiltonian with a runtime exponential in n. Let us start with computing the ground state energy and ground state of a permutation-invariant Hamiltonian H given as a vector h_i in the basis of symmetrized Pauli monomials.

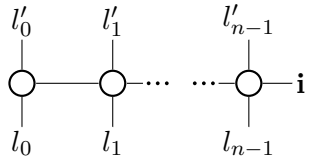
Theorem 4 (Efficient classical computation of the ground state of a permutation-invariant Hamiltonian)**.** *The ground state and ground state energy of a permutation-invariant Hamiltonian on n qubits, given as h_i in the basis of symmetrized Pauli monomials above, can be computed in time $\mathrm{O}(n^7)$ via Lemma 2. When applied to a single Hamiltonian for each system size n, the bottleneck of the algorithm is the computation of the matrix elements $F^{i,\lambda}_{q_\lambda, q'_\lambda}$ that takes $\mathrm{O}(n^7)$, but is independent on h_i. Once the matrix elements are computed, application of Lemma 2 only takes time $\mathrm{O}(n^6)$. The output ground state is given in the $|\lambda, q_\lambda, p_{\lambda 0}\rangle$ Schur basis for a fixed $p_{\lambda 0}$.*

Proof. The block-diagonal basis of the permutation representation R of $G = S_n$ is known as *Schur basis*, which we will review in Appendix A. From this it is easy to see that the number n_λ of irreps with nonzero multiplicity is $n_\lambda = \mathrm{O}(n)$, and the maximal dimension n_q of these irreps is $n_q = \mathrm{O}(n)$. Direct application of the algorithm in Lemma 2, then leads to a runtime of $\mathrm{O}(n^6)$ for computing the ground state of S_n-equivariant Hamiltonians in the Schur basis. All that remains is the computation of the $\mathrm{O}(n^6)$ many matrix elements $F^{i,\lambda}_{q_\lambda, q'_\lambda}$ for a given n. To this end, we make use of the fact that the Schur basis $|\lambda, q_\lambda, p_{\lambda 0}\rangle$ as well as the representation A both have *matrix product operator* (MPO) representations. We will here only give a sketch of the tensor-network computation, and refer to Appendix C for the details. The representations are of the form

$\langle \lambda, q_\lambda, \vec{p}_\lambda | \vec{l} \rangle =$ 

, (22)

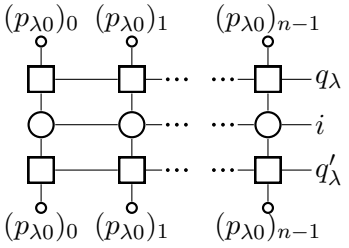
of bond dimension $\mathrm{O}(n)$, for each fixed value of λ, and

$\langle \vec{l} | A_{\mathbf{i}} | \vec{l}' \rangle =$ 

(23)

of bond dimension $\mathrm{O}(n^3)$. With this, the overlap in Eq. (3) in the computation of the matrix elements of F can be expressed as a tensor-network diagram:

$F^{i,\lambda}_{q_\lambda, q'_\lambda} = \langle \lambda, q_\lambda, p_{\lambda 0} | A_i | \lambda, q'_\lambda, p_{\lambda 0} \rangle$

$=$ 

. (24)

Contracting this tensor network from the left to the right (for a fixed λ and $\vec{p}_{\lambda 0}$) is the same as a sequence of $n-1$ vector-matrix multiplications of dimension $\mathrm{O}(n^5)$. This naive contraction already yields a polynomial runtime, namely $\mathrm{O}(n^{10})$ for each of $\mathrm{O}(n)$ contractions for each of $\mathrm{O}(n)$ values of λ, yielding $\mathrm{O}(n^{12})$ in total. However, in Appendix C, we make use of a band-diagonal structure of the matrices to reduce the cost of a single contraction from $\mathrm{O}(n^{10})$ to $\mathrm{O}(n^5)$, yielding $\mathrm{O}(n^7)$ in total. In Appendix E, we also give an alternative combinatorial way of calculating the matrix entries of F which has a slightly worse runtime of $\mathrm{O}(n^{10})$ in total but is better parallelizable. □

Remark 5. The output of the classical algorithm in Theorem 4 is the ground state in the Schur basis, which might not always be a useful description. However, from this output, we can be

efficiently construct the ground state on n qubits with a quantum computer via the Schur transform [26, 27, 28]. Furthermore, we can construct an efficient classical representation of a ground state as a low-bond-dimension MPS as we discuss in Section 4.2 and Appendix D.

We now consider an application of Lemma 3 to the symmetric group case.

Theorem 6 (Efficient classical simulation of permutation-equivariant dynamics)**.** *Consider the classical evaluation of $\ell(\rho)$ as in Eq. (14) using the same assumptions as Lemma 3 with symmetry group S_n. $\ell(\rho)$ can be calculated in time $\mathrm{O}(n^7)$.*

Proof. As for the proof of Theorem 4, we can compute the matrix elements $F^{i,\lambda}_{q_\lambda,q'_\lambda}$ in time $\mathrm{O}(n^7)$. The remainder is straight-forward application of Lemma 3 with $n_\lambda, n_q = \mathrm{O}(n)$. □

Theorems 4 and 6 provide efficient end-to-end classical algorithms if the output ground state, or the input state (for computing the dynamics) are given in the Schur basis. Prior work has also shown similar results for Dicke states [29, 24]. It may be desirable and more natural to process inputs and outputs of the algorithms that do not explicitly depend on the Schur basis. In the following two subsections, we will discuss the algorithms for two different input and output formats, namely first as a quantum state, and second as a matrix product state. These results are summarized in Table 1

4.1 Input and output as Quantum State

In this subsection, we will describe how Theorems 4 and 6 can be applied if the input or output is directly given as a quantum state on n qubits. Of course, this requires applying some quantum gates, measurements, and state preparations. However, the necessary quantum computation is of shallow depth and its function is merely to prepare the state or to measure a classical shadow of the state, whereas the core of the computation is classical.

Specifically, we use the shallow-depth quantum circuit from Ref. [27, 26] implementing the Schur transformation V_{STO} defined by:

$$V_{STO}\left|\lambda, q_\lambda, p_\lambda\right\rangle = \left|\boldsymbol{\lambda}\right\rangle\left|\boldsymbol{q_\lambda}\right\rangle\left|\boldsymbol{p_\lambda}\right\rangle. \tag{25}$$

Here, $\left|\boldsymbol{\lambda}\right\rangle, \left|\boldsymbol{q_\lambda}\right\rangle, \left|\boldsymbol{p_\lambda}\right\rangle$ are bitstring encodings of $\lambda, q_\lambda, p_\lambda$, respectively, in the computational basis, labeling the $\boldsymbol{\lambda}$ register, $\boldsymbol{q}$ register, and $\boldsymbol{p}$ register, respectively.

Theorem 7 (Efficient classical preparation of the ground state of a permutation-invariant Hamiltonian)**.** *Given a Hamiltonian in the symmetrized Pauli basis as in Theorem 4, we can prepare one of its ground states as a quantum state to an accuracy ϵ in classical runtime $\mathrm{O}(n^7)$ using a quantum circuit of depth $\tilde{\mathrm{O}}(n \operatorname{poly}\log(\epsilon^{-1}))$.*

Proof. We first use Theorem 4 to get a representation of the ground state in the $|\lambda, q_\lambda, p_\lambda\rangle$-basis, given by $\lambda_{\min}$ and $|\psi^*\rangle$. We then prepare the quantum state

$$\left|\boldsymbol{\lambda}_{\min}\right\rangle\left(\sum_{q_{\lambda_{\min}}} \psi^*_{q_{\lambda_{\min}}}\left|\boldsymbol{q_{\lambda_{\min}}}\right\rangle\right)\left|\boldsymbol{p_{\lambda_{\min}}0}\right\rangle \tag{26}$$

for some $\boldsymbol{p_{\lambda_{\min}0}}$ of choice. This preparation takes time $\mathrm{O}(n)$. The quantum ground state is then obtained by applying V_{STO} to the above state, which can be implemented in time $\tilde{\mathrm{O}}(n \operatorname{poly}\log(\epsilon^{-1}))$ on a quantum computer [27]. □

Theorem 8 (Efficient classical simulation of permutation-equivariant dynamics on a given quantum state)**.** *Consider the classical evaluation of $\ell(\rho)$ as in Eq. (14) using the same assumptions as Lemma 3 with symmetry group S_n, except instead of $\tilde{\rho}$ one is given ρ as a quantum state. Then, $\ell(\rho)$ can be estimated to additive error ϵ with probability $1-\delta$ via $\tilde{\mathrm{O}}\left(\|O\|_\infty^2 \epsilon^{-2} n^4 \log(\delta^{-1})\right)$ calls to a quantum computer each of depth $\tilde{\mathrm{O}}(n \operatorname{poly}\log(\epsilon^{-1}))$, up to an additional time $\mathrm{O}(n^7)$ in classical processing.*

Proof. This follows from using the Schur transform [26, 27] to prepare $\tilde{\rho}$ (as defined in Eq. (13)) from ρ, which can be done in time $\tilde{\mathrm{O}}(n \operatorname{poly}\log(\epsilon^{-1}))$ on a quantum computer. Note that $\tilde{\rho}$ is a $\lceil \log_2(n^2) \rceil$-qubit state as it only acts on the registers labeled by irreps λ and their basis vectors q_λ, i.e., the multiplicity register p_λ has been traced out. Given a source of ρ, then, a quantum computer can efficiently use classical shadow techniques [30] to output a classical description of $\tilde{\rho}$ capable of estimating $\ell(\rho)$ to addi-

Task	Runtime	Ref.	Notes
Finding ground state:			
Output specified in Schur basis	$\mathrm{O}(n^7)$	Thm. 4	$\mathrm{O}(n^6)$ if F matrix is precomputed.
Output matrix product state*	$\mathrm{O}(n^7)$	Rem. 11	$\mathrm{O}(n^6)$ if F matrix is precomputed.
Hamiltonian simulation:			
Any input quantum state**	$\tilde{\mathrm{O}}\left(\lvert O\rvert_\infty^2 \epsilon^{-2} n^4 \log\left(\delta^{-1}\right)\right)$	Thm. 8	Added $\mathrm{O}(n^6)$ time to process observable.
Input specified in Schur basis***	$\mathrm{O}(n^6)$	Thm. 6	
Matrix product state input	$\mathrm{O}(\chi^\omega n^3 + n^7)$	Thm. 10	χ is the bond dimension.

*Bond dimension of output $O(n)$.

**Requires low-depth circuit and single qubit measurements to measure the quantum state.

***Previous work provides efficient algorithms for symmetric Dicke states [24, 29].

Table 1: Summary of runtimes and results for finding the ground state and simulating permutation invariant Hamiltonians on inputs of different forms. Throughout, n refers to the number of qubits, δ is the probability of success and ϵ is the error of the outcome.

tive error ϵ with probability $1-\delta$ using

$$N = \tilde{\mathrm{O}}\left(\left\|\tilde{u}^\dagger \tilde{o} \tilde{u}\right\|_{\mathrm{shadow}}^2 \epsilon^{-2} \log\left(\delta^{-1}\right)\right) \tag{27}$$

samples of the state. Here, $\|\cdot\|_{\mathrm{shadow}}$ is the shadow norm, and $\tilde{o}$ and $\tilde{u}$ are defined as in Eqs. (15) and (16), respectively. As $\tilde{u}^\dagger \tilde{o} \tilde{u}$ is on $\lceil \log_2(n^2) \rceil$ qubits we have the general bound [30]

$$\left\|\tilde{u}^\dagger \tilde{o} \tilde{u}\right\|_{\mathrm{shadow}}^2 \leq n^4 \left\|\tilde{u}^\dagger \tilde{o} \tilde{u}\right\|_\infty^2 = n^4 \|O\|_\infty , \tag{28}$$

where $\|\cdot\|_\infty$ is the operator norm. We have then reduced the problem to a purely classical one, for which the classical runtime is given by Theorem 6. □

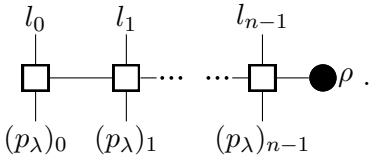
The sample complexity of this procedure could potentially be improved to $\tilde{\mathrm{O}}(\|O\|_\infty^2 \epsilon^{-2} n^3 \log(\delta^{-1}))$ as $\tilde{\rho}$ only has n^3 degrees of freedom. However, the block diagonal structure over irreps is lost when transforming to the bitstring encoding $|\boldsymbol{\lambda}\rangle |\boldsymbol{q}\rangle |\boldsymbol{p}\rangle$ via the Schur transform, and thus we arrive at the sample complexity given.

4.2 Input and Output as Matrix Product States

The previous subsection implemented algorithms with quantum states as outputs and inputs. In that particular setting, the readout of a quantum state requires some form of quantum computation of low depth. Here we describe alternative algorithms where we give the inputs and outputs purely classically as low-bond dimension MPS, which are discussed in more detail in Appendix D. Let us start by giving the output of Theorem 4 as an MPS.

Theorem 9. *The output of Theorem 4 can be given purely classically as an MPS of bond dimension* $\mathrm{O}(n)$.

Proof. The output of Theorem 4 is a value of λ and a vector ρ_{q_λ}. An MPO representation of the ground state can be obtained by contracting ρ with the MPO for the Schur basis in Eq. (22),



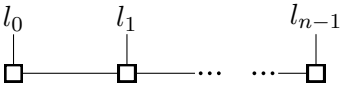
(29)

As we mentioned around Eq. (22), the bond dimension of this MPO is $\mathrm{O}(n)$. We consider this MPS representation in more detail in Appendix D. □

Next, let us consider Theorem 6 with an MPS as classical input.

Theorem 10. *Consider a state* ψ *represented as an MPS of bond dimension* χ. *The time evolved expectation value* $\ell(\rho)$ *with* $\rho = |\psi\rangle\langle\psi|$ *as in Theorem 6 can be computed classically in runtime* $\mathrm{O}(\chi^\omega n^3 + n^7)$.

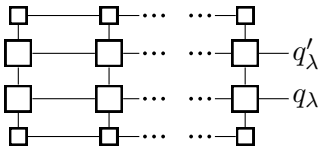
Proof. Denoting the MPS as tensor-network diagram,



(30)

the reduced density matrix $\tilde{\rho}^\lambda$ as defined in Eq. (13) is given by the overlap with twice

Eq. (22):



(31)

The naive contraction of this tensor network from left to right would need runtime $\mathrm{O}(n^8\chi^4)$. However, using that the MPO in the middle is band diagonal, the runtime reduces to $\mathrm{O}(n^4\chi^4)$. We discuss this contraction in more detail in Appendix D. After we have obtained ρ, we can proceed as in Theorem 6. □

Remark 11. If the matrix F in Theorem 4 is pre-computed, then this runtime is reduced to $\mathrm{O}(\chi^\omega n^3 + n^6)$.

4.3 Applications to machine learning

Though it is widely believed that general quantum machine learning models are more expressive than their classical counterparts [31], it is known that generic quantum machine learning models are untrainable [32, 33, 34, 7, 35, 36, 37, 11]. This gives a strong motivation for constructing quantum machine learning models from symmetry-equivariant time dynamics [13] which are believed to be efficiently trainable [36, 37]. This was formally proven for S_n-equivariant models in Ref. [4].

As a specific application of our results, we now consider one of the learning problems for which a variational quantum algorithm was given in Ref. [4]. We emphasize that here, just as in Lemma 3, we do not require that the input states ρ_i respect the symmetries of the model. For simplicity we assume the quantum input setting of Theorem 8, but a similar theorem can be given for MPS inputs.

Corollary 12 (Efficient classical simulation of permutation-invariant models)**.** *Consider a binary classification problem with labels* $y_i \in \{-1, 1\}$ *and empirical loss*

$$\hat{\mathcal{L}}(\boldsymbol{\theta}) = -\frac{1}{M}\sum_{i=1}^{M} y_i \ell_{\boldsymbol{\theta}}(\rho_i), \tag{32}$$

where $\ell_{\boldsymbol{\theta}}(\rho_i)$ *is as in Eq.* (14) *with a* $\boldsymbol{\theta}$*-dependent* U *and otherwise the same assumptions as in Theorem 8.* $\hat{\mathcal{L}}$ *can be estimated to additive error* ϵ *at* P *points in time*

$$\tilde{\mathrm{O}}\left(M\|O\|_\infty^2 \epsilon^{-2} n^5 \log\left(\frac{P}{\delta}\right) + MPn^{\omega+1} + n^7\right) \tag{33}$$

with total probability of success at least $1-\delta$*.*

Proof. This follows immediately from Theorem 8 with $\delta \to \frac{\delta}{P}$ by the union bound. □

Corollary 12 implies that the loss of these models can be estimated completely classically when the states ρ_i are given as certain classical shadows or matrix product state descriptions, even if they do not respect the symmetries of the model. As a point of comparison, consider the runtime of using a variational quantum algorithm to perform this binary classification task. Assume the variational circuits are of depth $\Omega(n^3)$ as required in Theorem 3 of Ref. [4] to ensure convergence. Then—taking $\Omega\left(\|O\|_\infty^2 \epsilon^{-2}\right)$ samples for each measurement to achieve an overall shot noise of $\mathrm{O}(\epsilon)$—this yields an overall runtime of $\Omega\left(MP\|O\|_\infty^2 \epsilon^{-2} n^3\right)$. For P sufficiently large, compare this to the time $\mathrm{O}\left(MPn^{\omega+1}\right)$ algorithm found for the classical algorithm where, even if quantum states are given as input, a classical shadow representation can be measured in quantum depth only $\tilde{\mathrm{O}}\left(n \operatorname{poly}\log\left(\epsilon^{-1}\right)\right)$. Unlike the quantum algorithm, this algorithm can be parallelized over irreps (i.e., over n_λ) easily, giving an effective runtime $\mathrm{O}\left(MPn^\omega\right) = \mathrm{o}\left(MPn^3\right)$. Even for P small, given many QPUs capable of running depth $\sim n$ quantum circuits, the classical algorithm parallelizes more effectively than the quantum algorithm as the required shadow tomography can be straightforwardly parallelized over shots.

5 Conclusion

We have specified a general framework for classically simulating highly symmetric quantum systems. Specializing to the symmetric group, we showed that these techniques yield an efficient classical algorithm for finding the ground state of quantum systems obeying an S_n symmetry, evaluating dynamics, and simulating S_n-equivariant quantum machine learning models. We hope that this framework sets the foundations for the future study of classical characterizations of symmetric quantum systems. Potential applications include

the physical analysis of symmetric quantum systems [9, 38, 39, 40] and implementation of algorithmic primitives for measuring entanglement [41], performing measurements with distributed sensors [42], and learning or testing quantum states [43, 44, 45].

A major remaining open question is to what extent the classical algorithms we have proposed for the case of qubit permutation invariance carry over to the more general highly symmetric many-body quantum systems. The approach we presented in Section 3 already works for general symmetry groups and representations. However, the computation of the matrix elements of F, as well as the representations of ground and initial states as matrix product states or quantum states was only described for the case of qubit permutation invariance. The perhaps most obvious generalization to other symmetry groups of representations is to consider permutation-invariant systems consisting of n d-dimensional qudits instead of qubits. We sketch this generalization in Appendix G. As it turns out, all algorithms we have proposed generalize with polynomial runtimes, however, the exponents become very large for large d. Specifically, the computation of the matrix elements F, which is the bottleneck for many of our algorithms, takes time $\mathrm{O}(n^{2d^2-1})$ in this case.

It should be noted that the rather large exponents of the polynomial runtimes ($\mathrm{O}(n^7)$ for qubits and $\mathrm{O}(n^{2d^2-1})$ for qudits) do not come from the method we use, but from the way we phrase the problem itself: Specifying a symmetric Hamiltonian alone needs $\mathrm{O}(n^3)$ data for qubits, and $\mathrm{O}(n^{d^2-1})$ data for qudits, so when measuring the runtime in the size of the input data, the scaling is just a little more than quadratic. For more efficient algorithms we would need to restrict the input to a subset of permutation-invariant Hamiltonians/operators, for example to Hamiltonians that are sparse in the symmetrized Pauli basis, or k-local Hamiltonians for constant k.

Our results indicate that care must be taken when constructing quantum algorithms to solve problems described by strongly symmetric quantum systems to ensure no equally efficient classical algorithm exists, as was demonstrated in Corollary 12. One approach to avoid classical simulation methods is to consider smaller symmetry groups as illustrated in Figure 1(a). Indeed, complexity theory arguments [46, 47] suggest that solving the ground state problem for systems with sufficiently small symmetry groups—such as translational invariance—is expected to be difficult. It is also reasonable to expect quantum algorithms for strongly symmetric quantum systems to offer polynomial speedups in appropriately selected scenarios, such as has been demonstrated in Ref. [48]. Notably, the tensor contraction algorithms outlined in Theorem 4 currently have a runtime of $\mathrm{O}(n^7)$, which in terms of the number of qubits n is not as optimal as quantum algorithms. We hope this work sparks further investigation into how problems involving symmetric quantum systems may produce a practical use case for quantum algorithms.

Acknowledgments

Code implementing some of the algorithms discussed is available at `https://github.com/bkiani/symmetric_hamiltonians`. We thank Quynh T. Nguyen for pointing out connections to tensor networks. We thank Marco Cerezo, Martin Larocca, Frederic Sauvage, and Louis Schatzki for their helpful feedback on the first draft of this manuscript. E.R.A. is supported by STAQ under award NSF Phy-1818914. A.B., B.T.K., and S.L. were supported by DARPA under the RQMLS program. A.B. is supported by the DFG (CRC 183), BMWK (PlanQK, EniQmA), and the Munich Quantum Valley (K-8).

References

[1] Hans Bethe. "Zur theorie der metalle". Z. Phys. **71**, 205–226 (1931).

[2] M. A. Levin and X.-G. Wen. "String-net condensation: A physical mechanism for topological phases". Phys. Rev. B **71**, 045110 (2005).

[3] A.A. Belavin, A.M. Polyakov, and A.B. Zamolodchikov. "Infinite conformal symmetry in two-dimensional quantum field theory". Nucl. Phys. B **241**, 333–380 (1984).

[4] Louis Schatzki, Martin Larocca, Quynh T. Nguyen, Frederic Sauvage, and M. Cerezo. "Theoretical guarantees for permutation-equivariant quantum neural networks" (2022). arXiv:2210.09974.

[5] Shouzhen Gu, Rolando D. Somma, and Burak Şahinoğlu. "Fast-forwarding quantum evolution". Quantum **5**, 577 (2021).

[6] Roeland Wiersema, Cunlu Zhou, Yvette de Sereville, Juan Felipe Carrasquilla, Yong Baek Kim, and Henry Yuen. "Exploring entanglement and optimization within the Hamiltonian variational ansatz". PRX Quantum **1**, 020319 (2020).

[7] Eric Ricardo Anschuetz. "Critical points in quantum generative models". In International Conference on Learning Representations. (2022). url: https://openreview.net/forum?id=2f1z55GVQN.

[8] Rolando Somma, Howard Barnum, Gerardo Ortiz, and Emanuel Knill. "Efficient solvability of Hamiltonians and limits on the power of some quantum computational models". Phys. Rev. Lett. **97**, 190501 (2006).

[9] Robert Zeier and Thomas Schulte-Herbrüggen. "Symmetry principles in quantum systems theory". J. Math. Phys. **52**, 113510 (2011).

[10] Xuchen You, Shouvanik Chakrabarti, and Xiaodi Wu. "A convergence theory for over-parameterized variational quantum eigensolvers" (2022). arXiv:2205.12481.

[11] Eric R. Anschuetz and Bobak T. Kiani. "Quantum variational algorithms are swamped with traps". Nat. Commun. **13**, 7760 (2022).

[12] Grecia Castelazo, Quynh T. Nguyen, Giacomo De Palma, Dirk Englund, Seth Lloyd, and Bobak T. Kiani. "Quantum algorithms for group convolution, cross-correlation, and equivariant transformations". Phys. Rev. A **106**, 032402 (2022).

[13] Johannes Jakob Meyer, Marian Mularski, Elies Gil-Fuster, Antonio Anna Mele, Francesco Arzani, Alissa Wilms, and Jens Eisert. "Exploiting symmetry in variational quantum machine learning" (2022). arXiv:2205.06217.

[14] Martín Larocca, Frédéric Sauvage, Faris M. Sbahi, Guillaume Verdon, Patrick J. Coles, and M. Cerezo. "Group-invariant quantum machine learning". PRX Quantum **3**, 030341 (2022).

[15] Michael Ragone, Paolo Braccia, Quynh T Nguyen, Louis Schatzki, Patrick J Coles, Frederic Sauvage, Martin Larocca, and M Cerezo. "Representation theory for geometric quantum machine learning" (2022). arXiv:2210.07980.

[16] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. "Geometric deep learning: Going beyond Euclidean data". IEEE Signal Process. Mag. **34**, 18–42 (2017).

[17] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. "A comprehensive survey on graph neural networks". IEEE Trans. Neural Netw. Learn. Syst. **32**, 4–24 (2021).

[18] Taco Cohen and Max Welling. "Group equivariant convolutional networks". In Maria Florina Balcan and Kilian Q. Weinberger, editors, Proceedings of The 33rd International Conference on Machine Learning. Volume 48 of Proceedings of Machine Learning Research, pages 2990–2999. New York, New York, USA (2016). PMLR. url: https://proceedings.mlr.press/v48/cohenc16.html.

[19] Peter J. Olver. "Classical invariant theory". London Mathematical Society Student Texts. Cambridge University Press. Cambridge, UK (1999).

[20] Bernd Sturmfels. "Algorithms in invariant theory". Texts & Monographs in Symbolic Computation. Springer Vienna. Vienna, Austria (2008).

[21] Ran Duan, Hongxun Wu, and Renfei Zhou. "Faster matrix multiplication via asymmetric hashing" (2022). arXiv:2210.10173.

[22] James Demmel, Ioana Dumitriu, and Olga Holtz. "Fast linear algebra is stable". Numer. Math. **108**, 59–91 (2007).

[23] Barbara M. Terhal and David P. DiVincenzo. "Classical simulation of noninteracting-fermion quantum circuits". Phys. Rev. A **65**, 032325 (2002).

[24] Nathan Shammah, Shahnawaz Ahmed, Neill Lambert, Simone De Liberato, and Franco Nori. "Open quantum systems with local and

collective incoherent processes: Efficient numerical simulations using permutational invariance". Phys. Rev. A **98**, 063815 (2018).

[25] Guang Hao Low. "Classical shadows of fermions with particle number symmetry" (2022). arXiv:2208.08964.

[26] Dave Bacon, Isaac L. Chuang, and Aram W. Harrow. "Efficient quantum circuits for Schur and Clebsch-Gordan transforms". Phys. Rev. Lett. **97**, 170502 (2006).

[27] Dave Bacon, Isaac L. Chuang, and Aram W. Harrow. "The quantum Schur transform: I. efficient qudit circuits" (2006). arXiv:quant-ph/0601001.

[28] William M. Kirby and Frederick W. Strauch. "A practical quantum algorithm for the Schur transform". Quantum Info. Comput. **18**, 721–742 (2018). url: https://dl.acm.org/doi/10.5555/3370214.3370215.

[29] Michael Gegg and Marten Richter. "Efficient and exact numerical approach for many multi-level systems in open system CQED". New J. Phys. **18**, 043037 (2016).

[30] Hsin-Yuan Huang, Richard Kueng, and John Preskill. "Predicting many properties of a quantum system from very few measurements". Nat. Phys. **16**, 1050–1057 (2020).

[31] Yunchao Liu, Srinivasan Arunachalam, and Kristan Temme. "A rigorous and robust quantum speed-up in supervised machine learning". Nat. Phys. **17**, 1013–1017 (2021).

[32] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. "Barren plateaus in quantum neural network training landscapes". Nat. Commun. **9**, 4812 (2018).

[33] Marco Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J Coles. "Cost function dependent barren plateaus in shallow parametrized quantum circuits". Nat. Commun. **12**, 1791–1802 (2021).

[34] Carlos Ortiz Marrero, Mária Kieferová, and Nathan Wiebe. "Entanglement-induced barren plateaus". PRX Quantum **2**, 040316 (2021).

[35] John Napp. "Quantifying the barren plateau phenomenon for a model of unstructured variational ansätze" (2022). arXiv:2203.06174.

[36] Martin Larocca, Piotr Czarnik, Kunal Sharma, Gopikrishnan Muraleedharan, Patrick J. Coles, and M. Cerezo. "Diagnosing barren plateaus with tools from quantum optimal control". Quantum **6**, 824 (2022).

[37] Martin Larocca, Nathan Ju, Diego García-Martín, Patrick J. Coles, and M. Cerezo. "Theory of overparametrization in quantum neural networks" (2021). arXiv:2109.11676.

[38] Bradley A. Chase and J. M. Geremia. "Collective processes of an ensemble of spin-1/2 particles". Phys. Rev. A **78**, 052101 (2008).

[39] Peter Kirton and Jonathan Keeling. "Superradiant and lasing states in driven-dissipative Dicke models". New J. Phys. **20**, 015009 (2018).

[40] Athreya Shankar, John Cooper, Justin G. Bohnet, John J. Bollinger, and Murray Holland. "Steady-state spin synchronization through the collective motion of trapped ions". Phys. Rev. A **95**, 033423 (2017).

[41] Ryszard Horodecki, Paweł Horodecki, Michał Horodecki, and Karol Horodecki. "Quantum entanglement". Rev. Mod. Phys. **81**, 865–942 (2009).

[42] Zheshen Zhang and Quntao Zhuang. "Distributed quantum sensing". Quantum Sci. Technol. **6**, 043001 (2021).

[43] Robert Alicki, Sławomir Rudnicki, and Sławomir Sadowski. "Symmetry properties of product states for the system of N n-level atoms". J. Math. Phys. **29**, 1158–1162 (1988).

[44] Ryan O'Donnell and John Wright. "Learning and testing quantum states via probabilistic combinatorics and representation theory". Curr. Dev. Math. **2021**, 43–94 (2021).

[45] Andrew M. Childs, Aram W. Harrow, and Paweł Wocjan. "Weak Fourier-Schur sampling, the hidden subgroup problem, and the quantum collision problem". In Wolfgang Thomas and Pascal Weil, editors, STACS 2007. Pages 598–609. Berlin (2007). Springer Berlin Heidelberg.

[46] Dorit Aharonov and Sandy Irani. "Hamiltonian complexity in the thermodynamic limit". In Stefano Leonardi and Anupam

Gupta, editors, Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing. Pages 750–763. STOC 2022New York (2022). Association for Computing Machinery.

[47] James D. Watson and Toby S. Cubitt. "Computational complexity of the ground state energy density problem". In Stefano Leonardi and Anupam Gupta, editors, Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing. Pages 764–775. STOC 2022New York (2022). Association for Computing Machinery.

[48] Eric R. Anschuetz, Hong-Ye Hu, Jin-Long Huang, and Xun Gao. "Interpretable quantum advantage in neural sequence learning". PRX Quantum **4**, 020338 (2023).

[49] Jin-Quan Chen, Jialun Ping, and Fan Wang. "Group representation theory for physicists". World Scientific Publishing. Singapore (2002). 2nd edition.

[50] OEIS Foundation Inc. "The On-Line Encyclopedia of Integer Sequences" (2022). Published electronically at http://oeis.org, Sequence A000292.

[51] William Fulton. "Young tableaux: With applications to representation theory and geometry". London Mathematical Society Student Texts. Cambridge University Press. Cambridge, UK (1996).

[52] Kenneth R Davidson. "C*-algebras by example". Volume 6 of Fields Institute Monographs. American Mathematical Society. Ann Arbor, USA (1996). url: https://bookstore.ams.org/fim-6.

[53] Giulio Racah. "Theory of complex spectra. II". Phys. Rev. **62**, 438–462 (1942).

[54] Vojt ěch Havlíček and Sergii Strelchuk. "Quantum Schur sampling circuits can be strongly simulated". Phys. Rev. Lett. **121**, 060505 (2018).

[55] R. H. Dicke. "Coherence in spontaneous radiation processes". Phys. Rev. **93**, 99–110 (1954).

[56] Andreas Bärtschi and Stephan Eidenbenz. "Deterministic preparation of Dicke states". In Leszek Antoni Gąsieniec, Jesper Jansson, and Christos Levcopoulos, editors, Fundamentals of Computation Theory. Pages 126–139. Cham (2019). Springer International Publishing.

[57] N. J. Vilenkin and A. U. Klimyk. "Representation of Lie groups and special functions". Volume 3. Springer Dordrecht. Dordrecht, Netherlands (1992).

A The Schur basis

In the presence of permutation invariance, the action of operations can be fully understood by analyzing a much smaller subspace of the larger Hilbert space. To precisely understand the form of that subspace, we turn to the Schur–Weyl decomposition of n qubits into subspaces corresponding to irreducible representations of the symmetric and unitary groups labeled by Young diagrams. Schur–Weyl duality offers a means to perform this decomposition by considering the natural representations of the permutation group and n-fold unitary group acting on n qubits [27, 49]. To describe the Schur basis and the resulting Schur transform, first we note the natural action of a permutation operation $R(\pi)$ acting on qubits:

$$R(\pi)\left|i_1\right\rangle \otimes \left|i_2\right\rangle \otimes \cdots \otimes \left|i_n\right\rangle = \left|i_{\pi^{-1}1}\right\rangle \otimes \left|i_{\pi^{-1}2}\right\rangle \otimes \cdots \otimes \left|i_{\pi^{-1}1}\right\rangle \tag{34}$$

as in the main text.

Similarly, a unitary $U \in \mathcal{U}(2)$ acting as the n-fold product $Q(U)$ takes the form

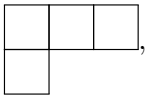
$$Q(U)\left|i_1\right\rangle \otimes \left|i_2\right\rangle \otimes \cdots \otimes \left|i_n\right\rangle = U\left|i_1\right\rangle \otimes U\left|i_2\right\rangle \otimes \cdots \otimes U\left|i_n\right\rangle . \tag{35}$$

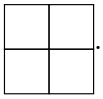
Schur–Weyl duality takes advantage of the fact that $Q(\cdot)$ and $R(\cdot)$ are each others' commutants, stating that the subspace of $(\mathbb{C}^2)^{\otimes n}$ decomposes as

$$Q(U)R(\pi) \cong \bigoplus_{\lambda} \rho_\lambda(U) \otimes \sigma_\lambda(\pi), \tag{36}$$

where λ runs over the set of partitions of n into at most two elements, and $\rho_\lambda(\cdot)$ and $\sigma_\lambda(\cdot)$ are irreducible representations of the unitary group $\mathcal{U}(2)$ and the symmetric group S_n, respectively. Note that irreps of both of these groups are indexed by partitions. More generally, for the space $(\mathbb{C}^d)^{\otimes n}$ of n qudits of dimension d, the λ would span over partitions of n into at most d elements. Partitions can equivalently be enumerated by Young diagrams. For example for the setting of 4 qubits, we have the 3 Young diagrams below that appear in the decomposition above:

$\lambda = (4,0)$: ,

$\lambda = (3,1)$:  ,

$\lambda = (2,2)$:  .

A consequence of the above is that there exists a basis indexed by $|\lambda, q_\lambda, p_\lambda\rangle$ called the *Schur basis* where the actions of $Q(\cdot)$ and $R(\cdot)$ are separated [27]:

$$Q(U)\left|\lambda, q_\lambda, p_\lambda\right\rangle = \rho_\lambda(U)\left|\lambda, q_\lambda, p_\lambda\right\rangle , \tag{37}$$

$$R(\pi)\left|\lambda, q_\lambda, p_\lambda\right\rangle = \sigma_\lambda(\pi)\left|\lambda, q_\lambda, p_\lambda\right\rangle , \tag{38}$$

where we have implicitly projected onto the subspace indexed by λ. Here, $\rho_\lambda(U)$ and $\sigma_\lambda(\pi)$ act only on the q_λ and p_λ space, respectively. $\rho_\lambda(U)$ and $\sigma_\lambda(\pi)$ are respectively the linear transformations corresponding to the irreducible representations of U_d and S_n for the irreducible representation indexed by λ. The above also presents a useful fact about permutation invariance. Namely, such an operation will act invariantly on the permutation register $|p_\lambda\rangle$ thus significantly reducing the degrees of freedom of a problem. The Schur transform U_{Sch} is a unitary transformation that acts as a change of basis from the computational to the Schur basis described above. The Schur transform can be efficiently

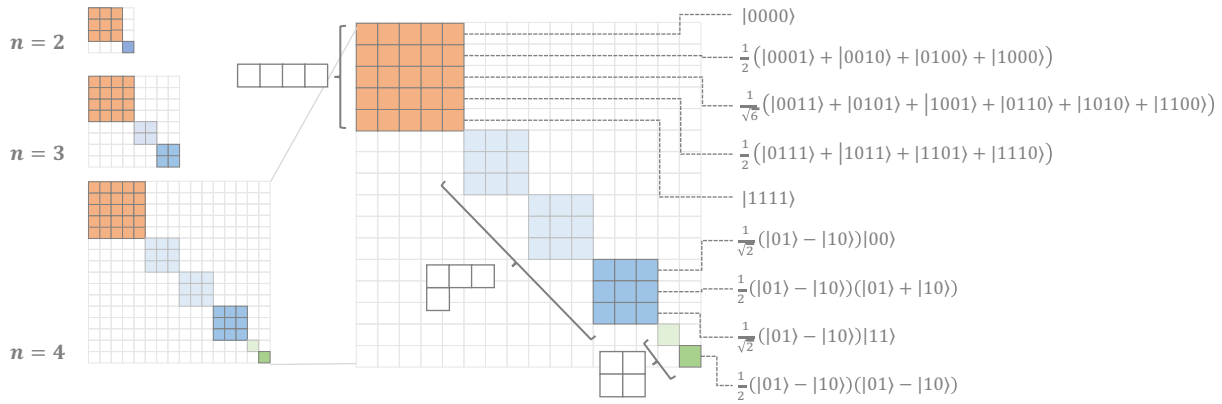


Figure 2: Graphical depiction of Schur decomposition for $n = 4$ qubits. There are three Young diagrams of at most two rows for 4 qubits. Due to the presence of permutation invariance, we can restrict attention to the darker colored subspaces which correspond to a single subspace over the multiplicity of the permutation irreps. To project onto this darker colored subspace, we use the Young symmetrizer (Eq. (41)).

implemented on a quantum computer running in time $\tilde{\mathrm{O}}\left(n\mathrm{poly}(d, \log n, 1/\epsilon)\right)$ for error ϵ on qudit systems of local dimension d [27]. We follow the notation of Ref. [27]:

$$|\lambda, q_\lambda, p_\lambda\rangle = \sum_{i_1,i_2,\ldots,i_n=0}^{d-1} [U_{Sch}]^{\lambda,q_\lambda,p_\lambda}_{i_1,i_2,\ldots,i_n} |i_1\rangle |i_2\rangle \cdots |i_n\rangle \,. \tag{39}$$

As noted in the main text, the total degrees of freedom reduces to $\binom{n+3}{3}$ in settings with permutation invariance. To see this, note that the dimension of the $|q_\lambda\rangle$ register for a partition (a, b) is equal to $a - b + 1$. Therefore, we have

$$\mathrm{DOF} = \sum_{k=0}^{\lfloor n/2 \rfloor} \left(2k + 1 + n - 2\left\lfloor \frac{n}{2} \right\rfloor\right)^2 = \binom{n+3}{3} \tag{40}$$

degrees of freedom. A similar calculation can be performed via a stars-and-bars counting argument. The above is also enumerated by the tetrahedral numbers [50].

To expand and manipulate individual basis states indexed by the $|q_\lambda\rangle$ register, one can use the Young symmetrizer Π_{p_λ} to project onto an explicit basis for each λ [27, 51]. Here, p_λ is a particular Young tableau for the Young diagram λ. The Young symmetrizer projects onto a subspace isomorphic to the subspace spanned by $|q_\lambda\rangle$:

$$\Pi_{p_\lambda} = \frac{\dim(\lambda)}{n!} \left(\sum_{c \in \mathrm{Col}(p_\lambda)} \mathrm{sgn}(c) R(c)\right) \left(\sum_{r \in \mathrm{Row}(p_\lambda)} R(r)\right), \tag{41}$$

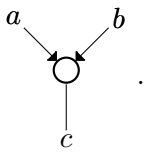
where $\mathrm{Row}(p_\lambda)$ and $\mathrm{Col}(p_\lambda)$ are the set of permutations which permute integers within only rows and columns of the Young tableau p_λ, respectively [27, 51]. An example of the basis found via application of the Young symmetrizer is shown in Figure 2. Throughout our study, we consider the Young tableau formed by filling entries in order first column-wise and then row-wise to be the "canonical" basis that we study. As an example, for 4 qubits, there are the following Young tableaus in our "canonical" basis:

$$\begin{array}{|c|c|c|c|}\hline 1 & 2 & 3 & 4 \\ \hline \end{array}, \quad \begin{array}{|c|c|c|}\hline 1 & 3 & 4 \\ \hline 2 \\ \cline{1-1} \end{array}, \quad \begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 & 4 \\ \hline \end{array}. \tag{42}$$

B General formalism using tensor-network diagrams

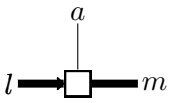
In this and the two subsequent appendices, we discuss the tensor-network based methods used in Section 4 of the main text in more detail. As a preparation, we will here recap the general formalism from Section 3 in a tensor-network language.

An associative algebra on a vector space B is determined by a linear multiplication map $B \otimes B \to B$, whose structure coefficients form a 3-index tensor,



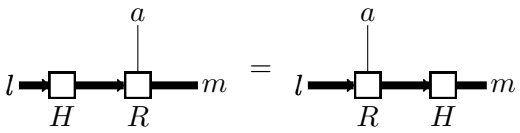
. (43)

Here, a and b label basis vectors at the input of the multiplication map, and c is at the output. In our setting, there are three different relevant algebras, namely X, the group algebra of S_n, and the group algebra of $SU(2)$. A representation of the algebra on a vector space V is determined by a linear map $V \otimes B \to V$, whose structure coefficients are again a 3-index tensor,



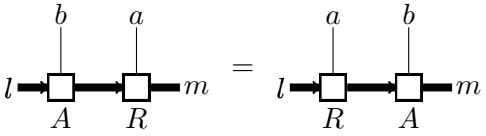
. (44)

As depicted, we will often use different line styles for different vector spaces such as B and V; here we made the V-indices thick. Again, there are three different relevant representations: The representation A of X, the representation R of S_n, and the representation $(\frac{1}{2})^{\otimes n}$ of $SU(2)$, all of which act on the n-qubit Hilbert space. The statement that a Hamiltonian (or an arbitrary operator) H is invariant under the symmetry representation R can be written as



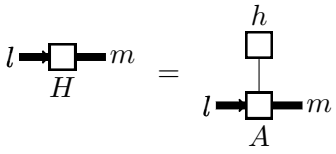
. (45)

The commutativity of the representation A of X with the representation R of S_n reads



. (46)

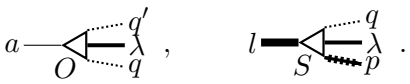
Since by construction A spans the whole commutant of R, H can be parametrized by some X-element h as



. (47)

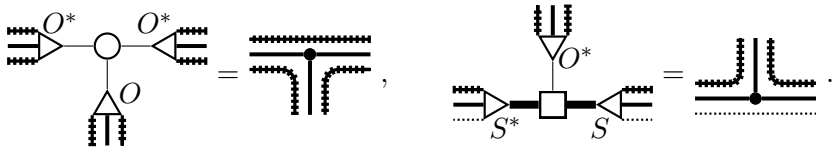
Eq. (45) then follows through Eq. (46).

Next, let us describe how to obtain the ground state energy and ground state via Theorem 4 from the main text using the matrix F. To this end, we use a central theorem of representation theory [52]: 1) Every finite-dimensional *-algebra (which includes group algebras and the algebra X) is isomorphic to a direct sum of full matrix algebras, and 2) every (unitary) representation is isomorphic to a direct sum of irreducible representations, which are projections onto one of the matrix-algebra summands. The two isomorphisms, which we call O and S, respectively, are unitary matrices,



(48)

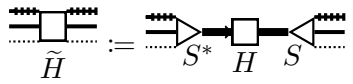
Here, the index a labels basis elements of the algebra, the *irrep index* λ labels irreducible representations of the algebra, *internal indices* like q and q' label basis vectors within an irreducible representation λ, and the *multiplicity index* p labels the different copies of a fixed irreducible representation λ, if λ occurs multiple times in the representation. Note that indices with different bond dimensions were drawn with different line styles, such as thick for irrep, dotted for internal, and ticked for multiplicity indices. Note that the bond dimension of the internal and multiplicity indices can depend on the value λ of the nearby irrep index, so the diagrams are not classical tensor-network notation but a slightly generalized version thereof. In this notation, the isomorphism can be written as



(49)

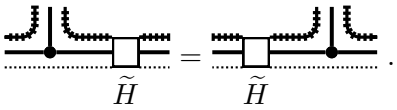
The small black dot on both right-hand sides is a δ-tensor, which is 1 if the irrep labels at all indices are equal, and 0 otherwise. The free lines represent identity matrices. The three identity matrices on the right-hand side of the left equation are matrix multiplication: Each matrix is represented by a double-index, and the second index of the first matrix is contracted with the first index of the second matrix.

We now use the isomorphism O of the symmetry group algebra, and the isomorphism S of its representation. Let us quickly recall what this means for the symmetry group S_n and its qubit permutation representation R, even though the discussion here in principle works for general symmetries. As described in the first appendix, the irreps λ are Young diagrams and the internal indices label Young tableaux p_λ. The dimension of the multiplicity index q is only non-zero for Young diagrams with one or two rows. In this case, we can label the irrep by a half-integer $0 \leq \lambda \leq \frac{n}{2}$ corresponding to the Young diagram with two rows of lengths $(n/2+\lambda, n/2-\lambda)$, but also to a $SU(2)$ representation. We now let $\widetilde{H}$ denote the Hamiltonian (or an arbitrary invariant operator) H conjugated with S,



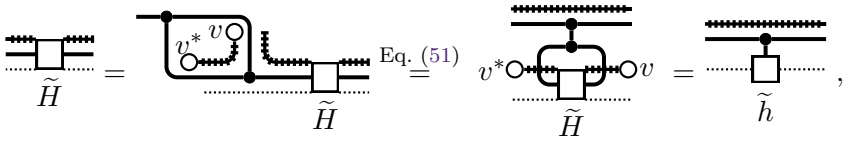
(50)

Eq. (45) in this block-diagonal basis then becomes



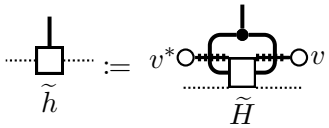
(51)

By contracting the left two open internal indices on the left-hand side with copies of a normalized vector v and applying a δ-tensor to the left two irrep indices, we find



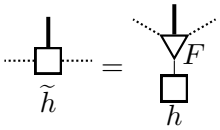
(52)

where



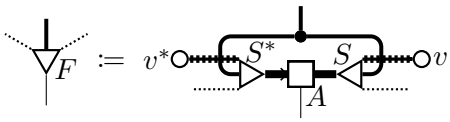
(53)

Note that v is actually a collection of vectors, depending on the value λ of the irrep. Plugging in Eq. (50) and Eq. (47), we find



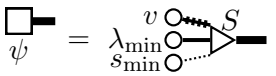
(54)

with



(55)

F is in fact the isomorphism O for the algebra X. The tensor $\widetilde{h}$ can be interpreted as a collection of matrices $\widetilde{h}^\lambda$ for different irreps λ. As can be easily seen from Eq. (52), the ground state energy of H is the minimum over λ of the ground state energies of $\widetilde{h}^\lambda$. Furthermore, the ground states are given by

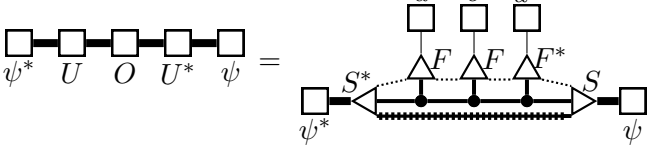


(56)

where $s_{\min}$ is the ground state of $\widetilde{h}^{\lambda_{\min}}$, and v is an arbitrary vector. Of course, there cannot be an efficient classical description which works for all ground states, since the dimension of v (and therefore the ground state degeneracy) is exponentially large in n. However, as we will later give a low-bond dimension MPS description for a full basis of ground states in Eq. (67).

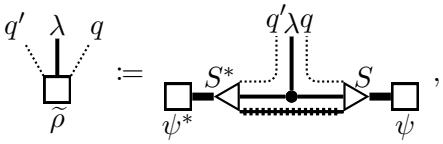
Finally, let us consider the time evolution of a pure state $|\psi\rangle$ under an invariant unitary U. More precisely, we want to calculate the time-evolved expectation value $\langle\psi| UOU^\dagger |\psi\rangle$ of an invariant operator O. We find

$\langle\psi| UOU^\dagger |\psi\rangle =$



(57)

As usual, we cannot efficiently contract this tensor network, since the internal p-index and the n-qubit indices have an exponential bond dimension in n. However, once we have the reduced density matrix



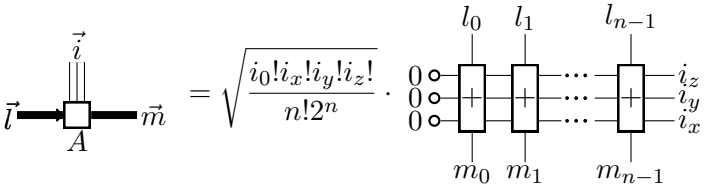
(58)

the remaining contraction is fast, dominated by the contraction between F and u or o which takes time $\mathrm{O}(n^6)$. If ψ is given as many copies of the quantum state, then $\widetilde{\rho}$ can be obtained by tomography on the λ and q-subspace, as described in the main text. If ψ is given classically as an MPS, then we show later around Eq. (69) how time evolution can be performed efficiently.

Note that time evolution can also be performed if U is defined through an invariant Hamiltonian $H = A(h)$ via $U = e^{itH}$. In this case, we calculate $\widetilde{h}$ as in Eq. (54), exponentiate the individual $\widetilde{h}^\lambda$, and plug $e^{it\widetilde{h}}$ into Eq. (57) instead of u and F.

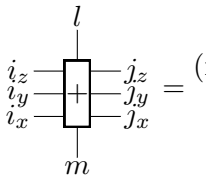
C Tensor-network method for calculation of the entries of F

In this Appendix, we show how to efficiently calculate the explicit matrix F defined in Eq. (55). We cannot directly efficiently contract the tensor network on the right-hand side since the two indices shared between A and S have bond dimension 2^n. In order to make contraction efficient, we will write A and S as MPOs and then contract the tensor network horizontally from qubit to qubit. Let us start by writing A as an MPO,



(59)

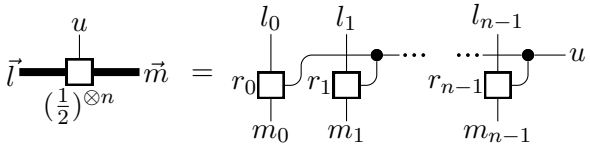
The global prefactor on the right-hand side was not included in the definition of A in the main text, but is necessary for the normalization in Eq. (90). The arrows over the indices on the left-hand side indicate that they correspond to collections of indices on the right-hand side. We have also used a triple-line for the X-index $\vec{i}$, to indicate that it is labelled by a triple (i_x, i_y, i_z). The bond dimension of all three horizontal indices is n, and the tensor with the + label is defined as



$$= (\mathrm{id}_2)_{lm}\delta_{i_x,j_x}\delta_{i_y,j_y}\delta_{i_z,j_z} + (\sigma_x)_{lm}\delta_{i_x,j_x+1}\delta_{i_y,j_y}\delta_{i_z,j_z} + (\sigma_y)_{lm}\delta_{i_x,j_x}\delta_{i_y,j_y+1}\delta_{i_z,j_z} + (\sigma_z)_{lm}\delta_{i_x,j_x}\delta_{i_y,j_y}\delta_{i_z,j_z+1} \,. \tag{60}$$

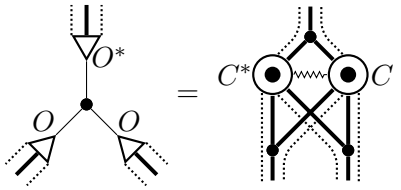
Note that the value of $\delta_{a,b}$ is understood to be zero unless $0 \leq a, b < n$. We can see that the value of the bottom, middle, and top horizontal index in Eq. (59) is the count of Xs, Ys, and Zs up to this point, respectively. To achieve this, the value of the bottom horizontal indices of a + tensor increases by 1 from left to right if the two vertical indices are in an X configuration, and analogously for the middle and top horizontal indices.

In order to write S as an MPO, we make use of the fact that the commutant of the representation R of S_n is spanned by the representation $(\frac{1}{2})^{\otimes n}$ of $SU(2)$. Thus, the isomorphism S is the same for both those representations, just that the internal and multiplicity indices are exchanged. Now, the representation $(\frac{1}{2})^{\otimes n}$ is a tensor product of on-site representations, and thus an MPO,



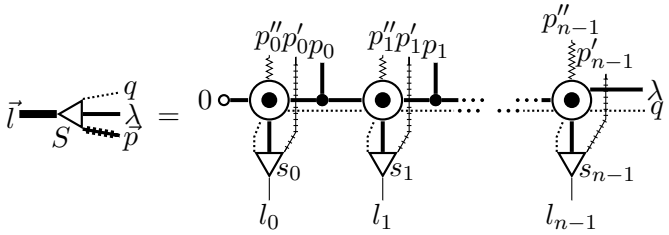
(61)

Note that u labels an element of $SU(2)$, so this is a tensor network with a continuous bond dimension, which might not be directly suitable for practical computation but still makes sense formally. In our case all the r_i are the spin-$\frac{1}{2}$ representation, but we will discuss the following few steps for arbitrary representations. In order to get the isomorphism S for the overall representation (for us $(\frac{1}{2})^{\otimes n}$), we start by block-diagonalizing the individual r_i. We end up with many copies of the isomorphism O for the group algebra of $SU(2)$. Those copies of O need to be pulled through all the δ-tensors. When doing this, the δ-tensor splits up into two copies of a tensor C,



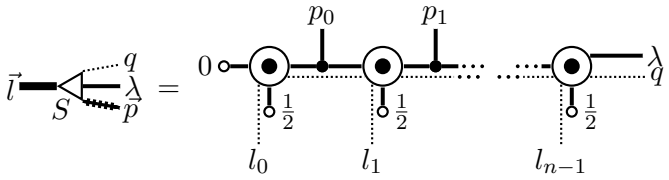
(62)

C exists since the group algebra together with the δ-tensor forms a Hopf algebra. When writing the bi-algebra axiom in the block-diagonal basis, one can see that the left-hand side for a fixed configuration of irrep indices is a projector P between all three left internal indices to all three right internal indices. So C can be defined as an isometry such that $CC^\dagger = P$. The index connecting the two copies of C has a bond dimension which depends on the three irreps, and will be referred to as *fusion index*. The entries of C are known as the *Clebsch-Gordon coefficients*. We can now use this equation to pull the isomorphism O through all the δ-tensors in Eq. (61). As a result, we obtain S as



(63)

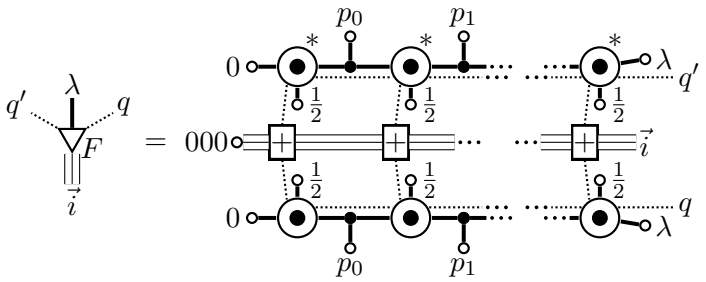
For our present situation where the group is $SU(2)$, the Clebsch-Gordon coefficients are well-known and can be computed efficiently up to p bits of precision (i.e., up to additive error exponentially small in p) in time poly (n, p) by the Racah formula [53]. Note that the fusion index is always trivial, that is, it has dimension either 0 or 1. All local representations r_i are equal to spin-$\frac{1}{2}$ such that all the local s_i can be chosen equal to the identity. With those simplifications, we get,



(64)

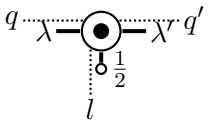
The tensor product of the spin-$\frac{c}{2}$ representation with the spin-$\frac{1}{2}$ representation is $\frac{c}{2} \otimes \frac{1}{2} = \frac{c-1}{2} \oplus \frac{c+1}{2}$. Thus, any two consecutive irreps p_i, p_{i+1} in the sequence $p_0, \ldots, p_{n-1} = \lambda$ must differ by $\pm\frac{1}{2}$, otherwise the MPO above yields 0. Such a sequence exists if and only if $(2\lambda) \mod 2 = n \mod 2$ and $\lambda \leq \frac{n}{2}$. In fact, one can directly see the correspondence between such sequences $\vec{p}$ and the Young tableaux for the Young diagram with two rows of lengths $(n/2 + \lambda, n/2 - \lambda)$: Starting at an empty diagram, we fill its fields consecutively with the numbers $0, \ldots, n-1$. In the ith step, the number i is appended to the first row if $p_i = p_{i-1} + \frac{1}{2}$, and added to the second row if $p_i = p_{i-1} - \frac{1}{2}$. Note that a representation of the Schur-Weyl basis equivalent to this MPO has also been used in Ref. [54], and was originally presented in Ref. [27].

Let us now plug the derived MPO representations in Eq. (64) and Eq. (59) into Eq. (55). For the vector v, we choose a fixed basis vector corresponding to a valid sequence $\vec{p} = (p_0, \ldots, p_{n-1})$ depending on λ,



(65)

The maximal possible value of the p_i is $O(n)$, and thus the bond dimension of the horizontal q-indices is $O(n)$ as well. We can already see that contracting the tensor network from the left to the right has polynomial runtime. The contraction becomes even faster by the following considerations. For a fixed $\lambda' = \lambda \pm \frac{1}{2}$, consider the Clebsch-Gordon tensor,



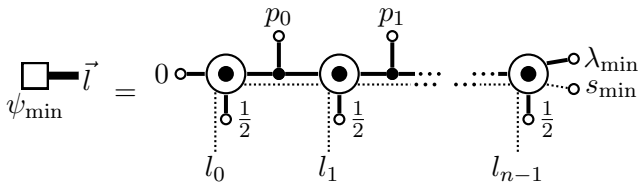
(66)

as collection of matrices $M_{qq'}$ for different values of l. Those matrices are simultaneously constant-width block-diagonal. The + tensor is also band-diagonal, and the bond dimension of the vertical indices is $O(1)$. Since there are 5 horizontal indices each of bond dimension $O(n)$ and all tensors are band diagonal, adding one column of tensors to the contraction takes time $O(n^5)$. Since there are n qubits/contraction steps, as well as $O(n)$ different values of λ, the total runtime for the contraction is $O(n^7)$.

D Time evolution and ground state using MPS

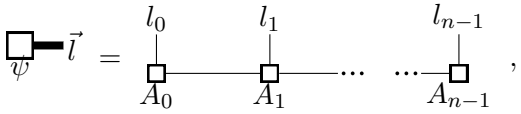
In this appendix, discuss the two algorithms from Theorems 9 and 10, which are classically end to end using MPS as input or output, in more detail. Let us start by Theorem 9 giving an MPS description of

the ground states of an invariant Hamiltonian by combining Eq. (56) and Eq. (64). A basis of ground states is labelled by valid sequences $p_0, \ldots, p_{n-1} = \lambda_{\min}$. For every such sequence, the ground state is given as an MPS of bond dimension $\mathrm{O}(n)$,



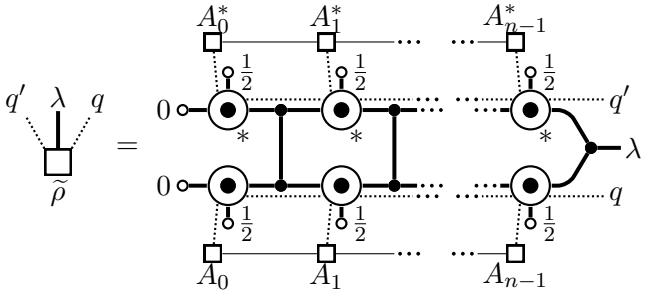
(67)

Next, let us discuss Theorem 10, which calculates the time-evolved expectation value giving an initial state ψ as an MPS,



(68)

of bond dimension χ. We first note that for u representing an invariant unitary $U = A(u)$, $U|\psi\rangle$ is again an MPS of bond dimension $\chi\,\mathrm{O}(n^3)$ due to the MPO representation of A in Eq. (59). An invariant observable $O = A(o)$ can be written as an MPO as well, so the expectation value $\langle\psi|UOU^\dagger|\psi\rangle$ can be evaluated as a product of MPS and MPO from left to right in polynomial time. However, the most efficient way to calculate that expectation value is to build on the method presented around Eq. (58). The reduced density matrix $\widetilde{\rho}$ in Eq. (58) can be computed efficiently using the MPO representation of S in Eq. (64),



(69)

As usual, this tensor network can be contracted from the left to the right. There are five independent horizontal indices, two of them of bond dimension χ, and three of bond dimension $\mathrm{O}(n)$. As argued above, the Clebsch-Gordon coefficients are band-diagonal. This is even true if the irrep indices are not fixed, since they can only change by $\pm\frac{1}{2}$. The runtime of adding the ith column of tensors to the contraction is thus dominated by contracting the MPS tensors A_i, which takes time $\mathrm{O}(\chi^w n^3)$. Since there are n steps, the total runtime is $\mathrm{O}(\chi^w n^4)$.

E Alternative combinatorial method for calculation of the entries of F

In Appendix C, we have presented a tensor-network method for calculating the matrix entries $F^{\boldsymbol{i},\lambda}_{q_\lambda,q'_\lambda}$ used in the main text to efficiently simulate permutation symmetric Hamiltonians. In this section, we present an alternative method that calculated these matrix elements in polynomial runtime $\mathrm{O}(n^{10})$ using combinatorics. Even though the runtime scaling for this method is slower, it may be more approachable and insightful for people without tensor-network background.

We also like to remark that in this method, the matrix entries are calculated individually, each taking a runtime of $\mathrm{O}(n^4)$. This might help if the given Hamiltonian is sparse in the symmetrized Pauli basis. In this case it suffices to calculate only the matrix elements for the non-zero Pauli monomials. For example, consider a family of permutation invariant Hamiltonians that is k-local for a constant k. In this case, we only need the matrix elements for a constant number of Pauli monomials and thus in this case (since the Schur basis has dimension $\mathrm{O}(n^3)$) the relevant matrix elements can also be calculated

in $\mathrm{O}(n^7)$. In contrast, restricting to a sparse subset of symmetrized Pauli monomials does not directly lead to an improvment of the $\mathrm{O}(n^7)$ runtime of the tensor-network based algorithm.

The combinatorial formula for the matrix elements that we will derive in this appendix is as follows.

Lemma 13. *Recalling that λ is given by a Young diagram, we choose $p_{\lambda 0}$ to be the standard Young tableaux for that diagram, with numbers increasing first in the column direction and then in row direction, as shown in Eq. (42). Then the tensor components $F^{\boldsymbol{i},\lambda}_{q_\lambda,q'_\lambda}$ discussed in the main text for the completely symmetrized Pauli representation are given by:*

$$F^{\boldsymbol{i},\lambda}_{q_\lambda,q'_\lambda} = \sum_{\substack{f_{11},f_{xx},f_{yy},f_{zz},\\ g_{010},g_{111},g_{0x1},g_{1x0},\\ g_{0y1},g_{1y0},g_{0z0},g_{1z1}\\ /Eq.\ (71)}} \frac{1}{\sqrt{\binom{n-2\lambda_1}{q_\lambda}\binom{n-2\lambda_1}{q'_\lambda}}} i^{2f_{xx}+2f_{yy}+2f_{zz}+2g_{1z1}-g_{0y1}+g_{1y0}} \cdot \frac{\lambda_1!(n-2\lambda_1)!}{f_{11}!f_{xx}!f_{yy}!f_{zz}!g_{010}!g_{111}!g_{0x1}!g_{1x0}!g_{0y1}!g_{1y0}!g_{0z0}!g_{1z1}!}\,, \tag{70}$$

where the sum is over a set of 12 non-negative integers fulfilling the constraints

$$\begin{aligned} g_{010}+g_{0z0}+g_{0x1}+g_{0y1} &= n-2\lambda_1-q_\lambda,\\ g_{010}+g_{0z0}+g_{1x0}+g_{1y0} &= n-2\lambda_1-q'_\lambda,\\ g_{111}+g_{1z1}+g_{1x0}+g_{1y0} &= q_\lambda,\\ g_{111}+g_{1z1}+g_{0x1}+g_{0y1} &= q'_\lambda,\\ 2f_{11}+g_{010}+g_{111} &= i_1,\\ 2f_{xx}+g_{0x1}+g_{1x0} &= i_x,\\ 2f_{yy}+g_{0y1}+g_{1y0} &= i_y,\\ 2f_{zz}+g_{0z0}+g_{1z1} &= i_z \end{aligned} \tag{71}$$

and λ_1 is the length of the second row of λ.

Proof. Following Appendix A, we can project onto the space with an S_n irrep λ and a fixed multiplicity label $p_{\lambda 0}$ using the Young symmetrizer in Eq. (41). Acting with the Young symmetrizer on a computational basis state yields a superposition of basis states with the same number of 0s and 1s. Let us write $\lambda = (\lambda_0, \lambda_1)$ for the lengths of the first and second row of λ. Then we see that applying the Young symmetrizer yields 0 unless the number of 1s is between λ_1 and λ_0. This is because the row symmetrizer does not change the number of 1s, and the antisymmetrizer on λ_1 length-2 columns yields 0 if any columns are 00 or 11. Thus, the irrep basis states can be obtained by applying the Young symmetrizer to states with $\lambda_1 + q_\lambda$ ones, where $0 \le q_\lambda \le n - 2\lambda_1$. Specifically, we can use

$$|\lambda, p_{\lambda 0}, q_\lambda\rangle = \Pi_{\lambda:p_{\lambda 0}} |x_{q_\lambda}\rangle\,, \tag{72}$$

with

$$|x_{q_\lambda}\rangle \coloneqq |01\rangle^{\otimes\lambda_1} \otimes |0\rangle^{\otimes n-2\lambda_1-q_\lambda} \otimes |1\rangle^{\otimes q_\lambda}\,. \tag{73}$$

Let us first evaluate

$$\sum_{r\in \mathrm{Row}(p_{\lambda 0})} R(r)\,|x_{q_\lambda}\rangle = \Pi_{r\to c}\left(\left|\Sigma^{q_\lambda}_{\lambda_0}\right\rangle \otimes |1\rangle^{\otimes\lambda_1}\right), \tag{74}$$

where $|\Sigma^y_x\rangle$ denotes the equal-weight superposition of all computation basis states on x qubits with $x-y$ zeros and y ones, which (up to normalization) is also known as *Dicke state* on x qubits [55, 56]. $\Pi_{r\to c}$ denotes the permutation of qubits needed to obtain the "column-standard" Young tableau $p_{\lambda 0}$ from an analogous "row-standard" Young tableau where the numbers first increase in the row direction and then in column direction. In other words, if we think of the qubits being associated to the tiles of

the Young diagram λ, then the qubits in the first row are in state $\left|\Sigma_{\lambda_0}^{q_\lambda}\right\rangle$, and the qubits in the second row are in state $|1\rangle^{\otimes\lambda_1}$.

Next, for a two-row standard Young tableau $p_{\lambda 0}$, we have

$$\sum_{c\in\mathrm{Col}(p_{\lambda 0})} \mathrm{sgn}(c)R(c) = (\mathrm{id}_2 - \tau)^{\otimes\lambda_1} \otimes \mathrm{id}_2^{\otimes n-2\lambda_1} = (|\Psi\rangle\langle\Psi|)^{\otimes\lambda_1} \otimes \mathrm{id}_2^{\otimes n-2\lambda_1} , \tag{75}$$

where $|\Psi\rangle$ is the 2-qubit singlet state

$$|\Psi\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) , \tag{76}$$

and τ denotes the swap operator acting on two qubits. The qubits in the second row of λ in the state of Eq. (74) are fixed to $|1\rangle$, so applying $|\Psi\rangle\langle\Psi|$ to each of the first λ_1 columns has the same effect as applying $|\Psi\rangle\langle\Psi|(|0\rangle\langle 0|\otimes\mathrm{id}_2)$. Applying $|0\rangle\langle 0|$ to the first λ_1 qubits of $\left|\Sigma_{\lambda_0}^{q_\lambda}\right\rangle$ yields $|0\rangle^{\otimes\lambda_1}\otimes\left|\Sigma_{\lambda_0-\lambda_1}^{q_\lambda}\right\rangle$. Thus, we find:

$$|\lambda, p_{\lambda 0}, q_\lambda\rangle = \Pi_{\lambda:p_{\lambda 0}}|x_{q_\lambda}\rangle = |\Psi\rangle^{\otimes\lambda_1}\otimes\left|\Sigma_{n-2\lambda_1}^{q_\lambda}\right\rangle . \tag{77}$$

Now, we are ready to evaluate

$$\begin{aligned} F_{q_\lambda,q'_\lambda}^{\boldsymbol{i},\lambda} &\coloneqq \langle\lambda, q_\lambda, p_{\lambda 0}| A_{\boldsymbol{i}} |\lambda, q'_\lambda, p_{\lambda 0}\rangle \\ &= (\langle\Psi|^{\otimes\lambda_1}\otimes\left\langle\Sigma_{n-2\lambda_1}^{q_\lambda}\right|)\left(\sum_{p_{\boldsymbol{i}}\in P_{\boldsymbol{i}}} p_{\boldsymbol{i}}\right)\left(|\Psi\rangle^{\otimes\lambda_1}\otimes\left|\Sigma_{n-2\lambda_1}^{q'_\lambda}\right\rangle\right) \\ &= \frac{1}{\sqrt{\binom{n-2\lambda_1}{q_\lambda}\binom{n-2\lambda_1}{q'_\lambda}}}\left(\sum_{s\in S_{n-2\lambda_1}^{q_\lambda}}\langle\Psi|^{\otimes\lambda_1}\otimes\langle s|\right)\left(\sum_{p_{\boldsymbol{i}}\in P_{\boldsymbol{i}}} p_{\boldsymbol{i}}\right)\left(\sum_{s'\in S_{n-2\lambda_1}^{q'_\lambda}}|\Psi\rangle^{\otimes\lambda_1}\otimes|s'\rangle\right) , \end{aligned} \tag{78}$$

where we used S_y^x to denote the set of bitstrings of length y with exactly x ones. This is a sum over (more than) exponentially many terms. Similarly to the previous Appendix, it can be evaluated efficiently by realizing that many summands have equal value. Thus, we instead sum over the different possible values multiplied with the number of summands with that value, which can be counted using combinatorics. Each summand is an overlap of two product states with a product operator in between. More precisely, we have a product of first λ_1 two-qubit overlaps, and then $n-2\lambda_1$ single-qubit overlaps.

For each summand in Eq. (78), let us denote by L_{ab} with $a, b \in \{1, x, y, z\}$ the subset of two-qubit pairs:

$$L_{ab} \coloneqq \{(2l, 2l+1) : (p_{\boldsymbol{i}})_{2l} = \sigma_a, (p_{\boldsymbol{i}})_{2l+1} = \sigma_b, 0 \le l < \lambda_1\} , \tag{79}$$

and let us write $f_{ab} = |L_{ab}|$ for the number of elements in those subsets. The according overlap

$$\langle\Psi|(\sigma_a\otimes\sigma_b)|\Psi\rangle \tag{80}$$

is 0 if $a \neq b$, so we only need to consider subsets where $a = b$. The number of summands for given numbers f_{aa} is the number of decompositions of the first λ_1 qubit pairs into the four subsets L_{aa} with $a \in \{1, x, y, z\}$, which equals

$$\frac{\lambda_1!}{f_{11}!f_{xx}!f_{yy}!f_{zz}!} . \tag{81}$$

The value which the overlap on the first λ_1 qubit pairs contributes to each summand only depends on the numbers f_{aa}. The overlap in Eq. (80) is given by 1 if $a = b = 1$, and -1 if $a = b$ otherwise. Thus, the overall contribution to each summand is

$$(-1)^{f_{xx}+f_{yy}+f_{zz}} . \tag{82}$$

Next, let us consider the $n-2\lambda_1$ single-qubit overlaps. For each summand in Eq. (78), let us denote by K_{iaj} for $i,j \in \{0,1\}$ and $a \in \{1,x,y,z\}$ the subset of the last $n-2\lambda_1$ qubits

$$K_{iaj} \coloneqq \{l : (p_{\boldsymbol{i}})_{2\lambda_1+l} = \sigma_a, s_l = i, s'_l = j, 0 \leq l < n-2\lambda_1\} \,, \tag{83}$$

and let us write $g_{iaj} = |K_{iaj}|$ for the number of elements in those subsets. The according overlap

$$\langle i| \sigma_a |j\rangle \tag{84}$$

is only non-zero if $i = j$ for $a \in \{1,z\}$ and $i \neq j$ for $a \in \{x,y\}$, so we can restrict to summands where only those 8 subsets are non-empty. The number of summands for given numbers g_{iaj} is the number of decompositions of the set of the last $n-2\lambda_1$ qubits into the 8 subsets K_{iaj}, and is thus given by

$$\frac{(n-2\lambda_1)!}{g_{010}!g_{111}!g_{0x1}!g_{1x0}!g_{0y1}!g_{1y0}!g_{0z0}!g_{1z1}!} \,. \tag{85}$$

The contribution of the overlap on the last $n-2\lambda_1$ qubits to each summand only depends on the numbers g_{iaj}. The single-qubit overlap in Eq. (84) evaluates to 1 for g_{010}, g_{111}, g_{0X1}, g_{1X0} and g_{0z0}, -1 for g_{1z1}, i for g_{0y1}, and $-i$ for g_{1y0}. Thus the overall contribution to each summand is

$$(-1)^{g_{1z1}}(-i)^{g_{0y1}}(i)^{g_{1y0}} \,. \tag{86}$$

Overall, the number of summands for given f_{aa} and g_{iaj} is the product of Eq. (81) and Eq. (85), and the value of each summand is given by the product of Eq. (82) and Eq. (86). Plugging this into Eq. (78) yields Eq. (70). The constraints in Eq. (71) are explained as follows. The first four constraints are due to the fact that the number of zeros and ones in s and s' is determined by q_λ and q'_λ, respectively. The last four constraints correspond to the fact that the number of Pauli operators σ_1, σ_x, σ_y, σ_z in $p_{\boldsymbol{i}}$ is given by i_1, i_x, i_y, and i_z, respectively. □

In a similar fashion to the previous Appendix, we can easily evaluate the runtime this method achieves in calculating all of the matrix elements. Note that each component is a sum over four independent variables due to the constraints, yielding a runtime of $\mathrm{O}(n^4)$. Taking into account the $\mathrm{O}(n^6)$ tensor components of F yields the final runtime of $\mathrm{O}(n^{10})$. Once again, it seems likely that the $\mathrm{O}(n^4)$ runtime for a single tensor component can be reduced to a smaller exponent. We will leave this open to further investigation.

F Ground state energy via structure coefficients

In this appendix, we will give an alternative algorithm for finding the ground state energy only. The runtime of this algorithm is slower than for the algorithm presented in the main text. Yet it might help the reader develop further intuition for why the computation is tractable in polynomial time. It is also simpler in that it does not require any knowledge of the Schur transform at all, but only determining the structure coefficients of the algebra X.

F.1 Algorithm for general symmetry groups

Let us start by phrasing the alternative ground-state energy algorithm for general symmetry groups and representations.

Lemma 14 (Finding the ground state energy of symmetric Hamiltonians)**.** *Consider a subalgebra X of dimension N, and assume that the structure constants of X in some preferred basis are known. Let $H \in A(X)$ be a Hamiltonian given in the preferred basis as in Eq. (4). Then the ground state energy of H can be found in time $\mathrm{O}(N^\omega)$.*

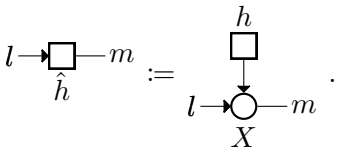
Proof. Consider the operator with indices:

$$\hat{h}_k^j := \sum_i h_i X_k^{i,j} \ , \tag{87}$$

which is nothing but the regular representation of h for the algebra X. Then we have that their ground state energies are equal:

$$\mathrm{GSE}(H) = \mathrm{GSE}(\hat{h}) \ , \tag{88}$$

or, in tensor-network notation (c.f. Eq. (43)),



(89)

This is because the regular representation is faithful, and the ground state energy of an operator is the same in any faithful representation. Since X has dimension N, the ground state energy of $\hat{h}$ can be found in time $\mathrm{O}(N^\omega)$. □

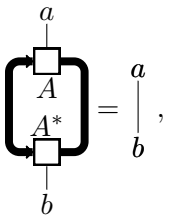
An advantage of this algorithm is that the only necessary information are the structure constants of X; no knowledge of the irreps of X is needed. However, due to this we have poor scaling with the number of irreps n_λ, as the direct sum structure of X is not necessarily known. Another disadvantage of this approach is that it only gives the ground state energy, rather than the ground state itself (in a representation that is not the regular representation).

F.2 Algorithm for qubit permutation invariance

Let us now apply the algorithm in Lemma 14 to the case of qubit permutation invariance.

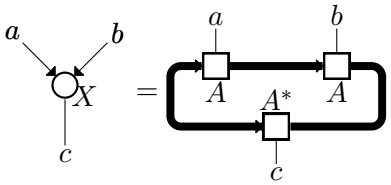
Corollary 15. *The ground state energy of a permutation-symmetric Hamiltonian on n qubits, given as h_i in the basis of symmetrized Pauli monomials above, can be computed in time $\mathrm{O}\left(n^{3\omega}\right)$ via Lemma 14.*

Proof. All that is needed for applying Lemma 14 are the structure constants of the algebra X, which are computed in the following. Let us start by showning how these structure coefficients are computed in general, given only the representation A. If we normalize A such that it defines an orthonormal set of operators,



(90)

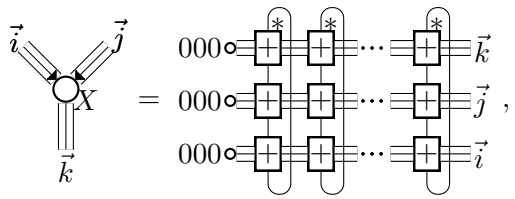
we can calculate X via



(91)

Here, we use the fact that by definition of X, A is a faithful definition of X. Note that the $*$ denotes complex conjugation of the tensor. We cannot directly efficiently contract the tensor network on the right-hand side of Eq. (91), since the thick indices have a bond dimension of 2^n. This problem can

be solved by realizing that A can be written as an MPO, as we have seen in Eq. (59). Plugging this MPO representation into Eq. (91), we obtain



(92)

where we have vertically shrinked the + tensors and the distance between the horizontal indices, and $*$ denotes complex conjugation. We see that now the tensor network can be contracted efficiently by contracting in the horizontal instead of the vertical direction. Indeed, the overall horizontal bond dimension is n^9, so contraction can be performed in polynomial runtime. To further speed up computation, we note that the values of each pair of horizontal indices on the left and right of a + tensor differ by at most 1. Thus, the tensor + is constant-width band-diagonal in its horizontal indices, independent of the value of its vertical indices. Thus, the matrix corresponding to a column of three + tensors is band-diagonal as well. Applying a band-diagonal matrix to a vector of bond dimension d takes time $\mathrm{O}(d)$. Thus, contracting the above tensor network takes n steps of runtime $\mathrm{O}(n^9)$, so runtime $\mathrm{O}(n^{10})$ in total. □

F.3 Alternative combinatorial method for computing the structure coefficients of X

In this Appendix we give an alternative combinatorial algorithm for calculating structure constants of the algebra X, similar to the algorithm used in Appendix E. Again, the algorithm is slower than the tensor-network method in Appendix F.2, namely $\mathrm{O}(n^{15})$. However, it might have the advantage of being more accessible to readers not familiar with tensor network.

Lemma 16. *The structure coefficients $X_{\boldsymbol{k}}^{\boldsymbol{i},\boldsymbol{j}}$ of the completely symmetrized Pauli representation are given by:*

$$X_{\boldsymbol{k}}^{\boldsymbol{i},\boldsymbol{j}} = \sum_{\substack{\{f_{ab}\}_{a,b\in\{1,x,y,z\}} \\ /Eq.(94),Eq.(95)}} \frac{k_1!}{f_{11}!f_{xx}!f_{yy}!f_{zz}!}\frac{k_x!}{f_{1x}!f_{x1}!f_{yz}!f_{zy}!}\frac{k_y!}{f_{1y}!f_{y1}!f_{xz}!f_{zx}!}\frac{k_z!}{f_{1z}!f_{z1}!f_{xy}!f_{yx}!}(i)^{f_{xy}+f_{yz}+f_{zx}}(-i)^{f_{yx}+f_{xz}+f_{zy}} \tag{93}$$

where the variables in the sum are non-negative integers subject to the constraints

$$\sum_{a\in\{1,x,y,z\}} f_{ab} = j_b\,, \qquad \sum_{b\in\{1,x,y,z\}} f_{ab} = i_a, \tag{94}$$

and

$$\begin{aligned} f_{11}+f_{xx}+f_{yy}+f_{zz} &=: k_1, \\ f_{1x}+f_{x1}+f_{yz}+f_{zy} &= k_x, \\ f_{1y}+f_{y1}+f_{xz}+f_{zx} &= k_y, \\ f_{1z}+f_{z1}+f_{xy}+f_{yx} &= k_z. \end{aligned} \tag{95}$$

Proof. For calculating the structure constants X, we first note that

$$A_{\boldsymbol{j}} = \frac{1}{i_1!i_x!i_y!i_z!}\sum_{\pi\in\mathrm{S}_n} R(\pi)\left(\sigma_1^{\otimes i_1}\otimes\sigma_x^{\otimes i_x}\otimes\sigma_y^{\otimes i_y}\otimes\sigma_z^{\otimes i_z}\right)R^{-1}(\pi) = \sum_{p_{\boldsymbol{i}}\in P_{\boldsymbol{i}}} p_{\boldsymbol{i}}\,, \tag{96}$$

where $P_{\boldsymbol{i}}$ is the set of Pauli words with i_a times σ_a for $a\in\{1,x,y,z\}$. Now we evaluate the product:

$$A_{\boldsymbol{i}}\cdot A_{\boldsymbol{j}} = \sum_{p_{\boldsymbol{i}}\in P_{\boldsymbol{i}},p_{\boldsymbol{j}}\in P_{\boldsymbol{j}}} p_{\boldsymbol{i}}p_{\boldsymbol{j}} = \sum_{\boldsymbol{k},p_{\boldsymbol{k}}\in P_{\boldsymbol{k}}}\ \sum_{\substack{p_{\boldsymbol{i}}\in P_{\boldsymbol{i}},p_{\boldsymbol{j}}\in P_{\boldsymbol{j}}: \\ p_{\boldsymbol{i}}p_{\boldsymbol{j}}=\alpha_{p_{\boldsymbol{i}},p_{\boldsymbol{j}},p_{\boldsymbol{k}}}\cdot p_{\boldsymbol{k}}}} \alpha_{p_{\boldsymbol{i}},p_{\boldsymbol{j}},p_{\boldsymbol{k}}}\cdot p_{\boldsymbol{k}}. \tag{97}$$

This is a sum over products of exponentionally many Pauli words. The idea to evaluate this is that many of the summands have equal value, so it suffices to sum over a few different values multiplied by the number of summands with that value.

For every summand, define the subsets of qubits L_{ab} for $a, b \in \{1, x, y, z\}$,

$$L_{ab} \coloneqq \{l : (p_{\boldsymbol{i}})_l = \sigma_a, (p_{\boldsymbol{j}})_l = \sigma_b, 0 \leq l < n\}, \tag{98}$$

and let

$$f_{ab} \coloneqq |L_{ab}| \tag{99}$$

be the numbers of elements in those subsets. Since every Pauli operator i_l at a qubit l is paired with some other Pauli operator j_l, f_{ab} fulfill the constraints in Eq. (94). The multiplication algebra of Pauli operators directly implies Eq. (95).

Let us now count how many Pauli words there are in the sum for a fixed set of numbers f_{ab} and a fixed resulting Pauli word $p_{\boldsymbol{k}}$. Every triple $p_{\boldsymbol{i}}$, $p_{\boldsymbol{j}}$, $p_{\boldsymbol{k}}$ corresponds to a decomposition of each $\boldsymbol{k}_c$-element set of qubits $\{l : (p_{\boldsymbol{k}})_l = \sigma_c\}$ for $c \in \{1, x, y, z\}$ into four subsets L_{ab} for the four different combinations $a, b \in \{1, x, y, z\}$ with $\sigma_a \sigma_b \propto \sigma_c$ under the Pauli algebra. For each c, the number of decompositions into the corresponding four subsets is given by

$$\frac{\boldsymbol{k}_c!}{\prod_{a,b:\sigma_a\sigma_b\propto\sigma_c} f_{ab}!} \,. \tag{100}$$

In total, the number of decompositions into four subsets for different c is given by

$$\frac{k_1!}{f_{11}!f_{xx}!f_{yy}!f_{zz}!}\frac{k_x!}{f_{1x}!f_{x1}!f_{yz}!f_{zy}!}\frac{k_y!}{f_{1y}!f_{y1}!f_{xz}!f_{zx}!}\frac{k_z!}{f_{1z}!f_{z1}!f_{xy}!f_{yx}!}. \tag{101}$$

Finally, the prefactor $\alpha_{p_{\boldsymbol{i}},p_{\boldsymbol{j}},p_{\boldsymbol{k}}}$ in Eq. (97) only depends on the f_{ab}. Using the Pauli algebra,

$$\begin{aligned} &\sigma_x\sigma_y = i\sigma_z \quad \sigma_y\sigma_z = i\sigma_x \quad \sigma_z\sigma_x = i\sigma_y \\ &\sigma_y\sigma_x = -i\sigma_z \quad \sigma_z\sigma_y = -i\sigma_x \quad \sigma_x\sigma_z = -i\sigma_y \,, \end{aligned} \tag{102}$$

it is given by

$$\alpha_{p_{\boldsymbol{i}},p_{\boldsymbol{j}},p_{\boldsymbol{k}}} = (i)^{f_{xy}+f_{yz}+f_{zx}}(-i)^{f_{yx}+f_{xz}+f_{zy}} \,. \tag{103}$$

Using Eq. (101) and Eq. (103) in Eq. (97) directly yields Eq. (93). □

Let us quickly discuss the complexity of the computation of $X_{\boldsymbol{k}}^{\boldsymbol{i},\boldsymbol{j}}$. In the summation of Eq. (93), we sum over 16 variables within a range of the order n, so if we naively evaluate the sum, we already obtain a polynomial runtime $\mathrm{O}(n^{16})$. However, due to the constraint Eq. (94), we can reduce the summation to only 9 variables f_{ab} with $a, b \in \{x, y, z\}$. Eq. (95) poses another three independent constraints, reducing the summation to 6 variables. Thus, an individual entry $X_{\boldsymbol{k}}^{\boldsymbol{i},\boldsymbol{j}}$ can be calculated in $\mathrm{O}(n^6)$ runtime, whereas all $\mathrm{O}(n^9)$ coefficients together take runtime $\mathrm{O}(n^{15})$.

G Permutation invariance for qudits

In this appendix, we sketch the generalization of the proposed algorithms to permutation invariant systems with n d-dimensional qudits instead of qubits. In this case, the dimension of the symmetrized basis is $\dim(X) = \binom{n+d^2-1}{d^2-1} = \mathrm{O}\left(n^{d^2-1}\right)$. This can be derived by utilizing a stars and bars counting argument, representing the number of ways to place $d^2 - 1$ bars among n stars. Let us now determine the numbers n_λ and n_q that determine the runtimes of our algorithms, and can also be used to rederive $\dim(X) = \mathrm{O}(n_\lambda n_q^2)$. The irreducible representations of $SU(d)$ are identified with Young diagrams with $d-1$ rows, which can be labelled by a sequence $\mathbf{j} = \{j_i\}_{0<i<d-1}$ with $j_i \geq j_{i+1}$ where j_i is the number of boxes in the ith row. The representation acting on a single qudit is the d-dimensional fundamental

irrep, corresponding to a Young diagram with only one box, which we denote by $\mathbf{d}$. The overall representation is $R = \mathbf{d}^{\otimes n}$ In order to decompose it into irreps, we only need to know the tensor product of an arbitrary irrep $\mathbf{j}$ with the fundamental irrep $\mathbf{d}$. It is given by a direct sum of at most $d+1$ irreps with at most $x+1$ boxes, if x is the number of boxes of $\mathbf{j}$. Specifically, the direct sum consists of irreps which are obtained by adding a box to any row, as well as an irrep obtained by removing a box from every row (as long as these operation still yield valid Young diagrams). So we see that for n qubits, the irreps in the decomposition of R are Young diagrams with at most n boxes. The number of these Young diagrams is $\mathrm{O}(n^{d-1})$. The maximum dimension of an irrep is $\mathrm{O}(n^{\frac{d(d-1)}{2}})$, as can be seen from the asymptotics of the formula

$$|\lambda| = \left[\prod_{1\leq i<j\leq d} (\lambda_j - \lambda_i + j - i)\right] / \left[\prod_{m=1}^{d} m!\right] \tag{104}$$

given in Ref. [27], with $\lambda_k \approx n(k/d)$.

Let us now look at how the computation of the matrix elements F via Eq. (24) carries over to the case of qudits. Analogous to Eq. (23), the symmetrized Pauli basis for qudits can be represented as an MPO of bond dimension $\mathrm{O}\left(n^{d^2-1}\right)$. This MPO is still band-diagonal. Next, the tensors in the MPO representation in Eq. (23) are the Clebsch-Gordan coefficients $C^{(\lambda',q'_\lambda)}_{(\lambda,q_\lambda),(\mathbf{d},q_\mathbf{d})}$, which can be found in Section 18.2.6. of Ref. [57]. For an appropriate choice of basis, C band diagonal. That is, for a fixed λ and q_λ, there is a constant number of λ' and q'_λ with $C^{(\lambda',q'_\lambda)}_{(\lambda,q_\lambda),(\mathbf{d},q_\mathbf{d})} \neq 0$. For the λ part, this follows from the fact that tensor product with the fundamental representation yields at most $d+1$ irrep factors as discussed above. Overall, the total bond dimension in Eq. (24) is $\mathrm{O}(n_q^2 n^{d^2-1}) = \mathrm{O}(n^{2d^2-d-1})$. Since the overall matrix (consisting of twice C and the tensor $+$) is band diagonal, a single vector-matrix multiplication in the contracting of Eq. (24) from left to right takes time $\mathrm{O}(n^{2d^2-d-1})$. Since there are $\mathrm{O}(n_\lambda) = \mathrm{O}(n^{d-1})$ involved irreps, and the contraction consists of n vector-matrix multiplications, the total runtime is $\mathrm{O}(n^{2d^2-1})$. So we see that we do get a polynomial runtime also for qudits, but the exponents explode quickly with increasing local dimension, for example we have $\mathrm{O}(n^{17})$ for qutrits.