Anti-symmetrization is cheap and useful

Adam Bene Watts[*] John Bostanci[†] Matthias C. Caro[‡]
Daniel Grier[§] Sihan Liu[¶] Gaurav Mahajan[||]

Abstract

We consider the computational task of anti-symmetrization of an unknown mixed quantum state: Given iid copies of a rank-r qudit state $\rho = \sum_{i=1}^{r} \lambda_i |\varphi_i\rangle\langle\varphi_i|$, prepare a copy of the state proportional to $\Pi^r_{\text{anti}}(|\varphi_1\rangle \otimes \ldots \otimes |\varphi_r\rangle)$, with Π^r_{anti} the anti-symmetric subspace projector. We first give a simple algorithm for anti-symmetrization that uses $\widetilde{O}(r/\lambda_{\min})$ copies of ρ, where $\lambda_{\min}$ is the minimum non-zero eigenvalue of ρ. This algorithm works even in the streaming setting with small working memory, and it uses only simple quantum operations like controlled-SWAPs and constantly many layers of single-qubit gates. As an application of this streaming anti-symmetrization procedure, we give a reduction from gapped mixed-state one-way state generators to pure-state one-way state generators. Then, using tools from Schur-Weyl duality, we give and analyze a second (non-streaming) anti-symmetrization procedure that uses $O(r/\lambda_{\min})$ copies of ρ, which then serves as a building block in new algorithms for weak Schur sampling and unitary Schur sampling, outperforming existing approaches in some regimes of number of copies n and local dimension d, as well as in an algorithm for optimal purity amplification for qudit states.

1 Introduction

Mixedness is a fundamental property in quantum computation and information. In quantum learning theory, algorithms for tomography of mixed quantum states [OW16; Haa+17] have, until recently [Pel+25], taken on a form quite different from that of tomography algorithms for pure states [BM99]. In quantum cryptography, various mixed-state versions of cryptographic primitives, like one-way state generators, seem to have fundamentally different hardness than their pure-state analogs [BCN25; Beh+24; BJ25]. At the same time, at least conceptually, mixed quantum objects can almost always be purified into corresponding pure objects, for instance via purification for quantum states, Stinespring's dilation theorem for quantum channels, or Naimark's dilation theorem for positive operator-valued measurements. Informally, this idea has taken on the moniker "the church of the larger Hilbert space"[1]. The quest to understand the nature of mixedness in quantum computation has recently led to the discovery of the random purification

[*]University of Calgary. adam.benewatts1@ucalgary.ca
[†]UC Berkeley & Simons Institute. cbostanc@berkeley.edu
[‡]University of Warwick. matthias.caro@warwick.ac.uk
[§]University of California San Diego. dgrier@ucsd.edu
[¶]University of California San Diego. sil046@ucsd.edu
[||] Yale University and Toulouse School of Economics. gaurav.mahajan@yale.edu
[1]This expression is attributed to John Smolin in [GL00].

channel [TWZ26; AG26; Che+26b], which combines the church of the larger Hilbert space with the church of the symmetric subspace [Har13]: many copies of a mixed state can be mapped to many copies of a *random* purification of that mixed state.

While the symmetric subspace has many interesting properties, its sibling, the anti-symmetric subspace, has received less attention and is sometimes overlooked. In this work, we present a different perspective on purifying mixed states, namely a sense in which quantum states can be canonically mapped into an *anti-symmetric* pure state. Switching from producing a symmetric state to an anti-symmetric state has certain (admittedly modest) advantages over the random purification channel, most notably that, when the number of anti-symmetrized registers reaches the rank of the mixed state, our algorithm deterministically produces a *unique* pure state—the unique state in the anti-symmetric subspace of r copies of an r-dimensional space. The pure output state being unique and the same across all rank-r states with the same support is what allows us to circumvent no-go results about universal purification [Liu+26].

Our procedure also has a few tangible algorithmic benefits relative to works on random purification. For example, it can be implemented efficiently using simple tools like controlled-SWAP operations in quadratic time in the number of copies, as opposed to requiring complicated representation-theoretic transforms like the Schur transform. Additionally, one of our anti-symmetrization algorithms can be naturally implemented in the "streaming" model, where copies of the unknown mixed state are fed into the algorithm one copy at a time.

1.1 Results and discussion

Anti-symmetrization algorithms. Our work revolves around the following anti-symmetrization problem: Given iid copies of an unknown rank-r mixed qudit state ρ with spectral decomposition $\rho = \sum_{i=1}^{r} \lambda_i |\varphi_i\rangle\langle\varphi_i|$, prepare a copy of the pure state proportional to $\Pi^r_{\text{anti}}(|\varphi_1\rangle \otimes \ldots \otimes |\varphi_r\rangle)$. Here, Π^r_{anti} is the projector onto the anti-symmetric subspace of $(\mathbb{C}^d)^{\otimes r}$. We refer to the pure state $\propto \Pi^r_{\text{anti}}(|\varphi_1\rangle \otimes \ldots \otimes |\varphi_r\rangle)$ as the anti-symmetric state over the support of ρ.

We present two algorithms for solving this anti-symmetrization problem. Both are based on a simple "growth" measurement—provided the first k registers of our state are already in the anti-symmetric subspace, we measure the anti-symmetric subspace projector on the first $k+1$ registers—that allows us to iteratively increase the number of registers in the anti-symmetric subspace one register at a time. Clearly, if this measurement succeeds (accepts), the resulting state is then in the anti-symmetric subspace over $k+1$ registers. To treat the reject case, we exploit a simple but crucial fact: even when the anti-symmetric subspace projector measurement on $k+1$ registers rejects (fails), we retain an anti-symmetric state in the initial k registers. This allows us, in the failure case, to simply ignore the $(k+1)$st register and carry on growing the anti-symmetric state using fresh copies. The following theorem provides an informal statement of the guarantees for our anti-symmetric growth measurement.

Theorem (Growing anti-symmetric states, informally)**.** *There is a measurement on $k+1$ registers that, if the first k registers are in the anti-symmetric subspace, projects onto the support of Π^{k+1}_{anti} when it succeeds, and when it fails, outputs a state in the support of Π^{k}_{anti} on the first k registers. Furthermore, such a measurement can be implemented using two controlled-SWAP operations with a $\lceil \log(k+1) \rceil$-qubit control register, a single multi-controlled-Z on $\lceil \log(k+1) \rceil$ qubits, and two layers of single-qubit gates on $\lceil \log(k+1) \rceil$ qubits.*

As an aside, we show the above algorithm can be implemented in constant depth (i.e., in the class QNC^0_f), possibly making it more practical for near-term quantum computers.

We use the growth measurement as the core building block in our two algorithms for anti-symmetrization. The first of our two algorithms uses a small quantum memory of $O(r\log(d))$ qubits, by simply discarding copies of the unknown state on which the growth measurement failed. This, however, is wasteful from a sample complexity perspective. Our second algorithm uses a larger quantum memory but saves a factor of $\log(r)$ in the sample complexity, by re-using copies on which the growth measurement failed in earlier rounds to build the largest-possible anti-symmetric state from the available copies. We first state the guarantee achieved by our low-memory procedure.

Theorem (Anti-symmetrization without copy re-use, informally)**.** *There is a "streaming-like" algorithm that, given copies of an unknown rank-r state ρ with smallest non-zero eigenvalue $\lambda_{\min}$, uses at most $O(r\log(d))$ qubits of working memory, requests a total of $n = O(r\log(r)/\lambda_{\min})$ copies of ρ, and prepares the unique anti-symmetric state over the support of ρ with success probability* 0.99*.*

The algorithm is "streaming-like" in the sense that it only ever acts on one fresh copy of ρ at a time. It keeps its working memory low by discarding registers that fail the growth measurement. The analysis of this algorithm straightforwardly reduces to bounding the success probability of a coupon collector. However, it is in fact possible to "re-use" registers that failed the growth measurement in earlier rounds. This leads to our second algorithm for anti-symmetrization.

Theorem (Anti-symmetrization with copy re-use, informally)**.** *There is an algorithm that, given copies of an unknown rank-r state ρ with smallest non-zero eigenvalue $\lambda_{\min}$, requests $n = O(r/\lambda_{\min})$ copies ρ and prepares the unique anti-symmetric state over the support of ρ with success probability* 0.99*.*

This second anti-symmetrization algorithm achieves its improved sample complexity by keeping the copies on which the growth measurement failed in a big pool of states and sometimes, instead of processing a fresh copy, re-applying the growth measurement on a register from that pool. The analysis of the algorithm's success probability is more involved and requires tools from the representation theory of the symmetric group (specifically, Pieri's rule). It also more clearly reveals the underlying advantage of the anti-symmetrization with re-use vs the one that discards failed copies: The former builds the largest representation-theoretically possible anti-symmetric state from the available n copies, no matter the value of n; the latter does so only for sufficiently large n. This feature is crucial when applying our anti-symmetrization with copy re-use to develop Schur sampling routines.

From anti-symmetrization to Schur sampling and purity amplification. We apply anti-symmetrization with copy re-use to give alternative algorithms for weak and unitary Schur sampling as well as for purity amplification of qudit states. To describe Schur sampling, recall from Schur-Weyl duality that $\left(\mathbb{C}^d\right)^{\otimes n} \simeq \bigoplus_{\lambda\vdash_d n} V_\lambda \otimes W_\lambda$, where each W_λ is an irreducible module of GL_d and each V_λ is an irreducible module of S_n. Here, the index λ runs over partitions of n into (at most) d parts. Fixing a Schur transform $U_{\text{schur}} : \left(\mathbb{C}^d\right)^{\otimes n} \mapsto \bigoplus_{\lambda\vdash_d n} V_\lambda \otimes W_\lambda$, we call $U^\dagger_{\text{schur}}(V_\lambda \otimes W_\lambda)$ the λ-isotypic component, and we denote the projector onto it by Π_λ. Then, δ-approximate weak Schur sampling is the following task: Given an input state $\rho \in \mathcal{L}\left(\left(\mathbb{C}^d\right)^{\otimes n}\right)$, sample a partition λ from a probability distribution $(p_\lambda)_{\lambda\vdash_d n}$ such that $\sum_{\lambda\vdash n} |p_\lambda - \mathrm{Tr}[\Pi_\lambda\rho]| \leq \delta$. In approximate unitary

Schur sampling, in addition to sampling a partition λ as in approximate weak Schur sampling, we require the algorithm to also output a state that approximately recovers the reduced state on the W_λ module (see Section 4.3 for a detailed formulation). We use our anti-symmetrization with copy re-use to show:

Corollary (Weak and unitary Schur sampling from anti-symmetrization, informally)**.** *Using anti-symmetrization, we can perform weak and unitary Schur sampling to accuracy δ with $\widetilde{O}(n^2 d \log(1/\delta))$ controlled-SWAP operations and T-count $\widetilde{O}(n^2 d \log^2(1/\delta))$.*

Prior approaches to constant-accuracy weak and unitary Schur sampling for general d, either using the full Schur transform [Har05; BCH06; BCH07; KS18; Kro19; Bur+25] or aiming directly at weak/unitary Schur sampling via Clebsch-Gordan transforms [CM23; CMT24], have gate complexities no better than $\widetilde{O}(n^a d^b)$ with integers $a \geq 1$ and $b \geq 0$ satisfying $a + b \geq 4$. Our procedure achieves $a = 2$ and $b = 1$, so $a + b = 3$, thus improving the overall complexity in some regimes of n and d. For $d = 2$, [Bra+25]'s unitary Schur sampling algorithm using $\widetilde{O}(n)$ random SWAP tests outperforms ours.

Motivated by the need to combat noise in quantum systems, Li et al. [Li+25] introduced the task of purity amplification: Given n iid copies of a mixed qudit state ρ, output a state ρ' that is close to achieving the maximum overlap with the top eigenvector of ρ. We develop an anti-symmetrization-based procedure that serves as a simplification of the optimal quantum purity amplification protocol of [Li+25] while producing the same output state.

Corollary (Purity amplification from anti-symmetrization, informally)**.** *Using anti-symmetrization, we can achieve optimal purity amplification to accuracy δ with $\widetilde{O}(n^2 d \log(1/\delta))$ controlled-SWAP operations and T-count $\widetilde{O}(n^2 d \log^2(1/\delta))$.*

Prior work achieved quantum purity amplification with an overall gate complexity polynomial in n and $\log(d)$, using generalized quantum phase estimation [Li+25; Li+26]. While our dependence on the local dimension d is worse, if d is taken to be fixed, then we achieve an improved gate complexity with an arguably simpler implementation. We note that, as our purity amplification procedure builds on our anti-symmetrization with copy re-use, it does not apply to the streaming version of the task [Chi+25; Gri+25].

Anti-symmetrization for one-way state generators. Our second application is in quantum cryptography. In their original work, [MY24] defined a one-way state generator (OWSG) to be a keyed family of states ϕ_k such that it is hard to learn the key k from copies of ϕ_k but computationally easy to verify ϕ_k given k. If one requires ϕ_k to be pure, there are at least two natural verification algorithms: one may either perform repeated SWAP tests between copies of ϕ_k and the unknown state, or one may uncompute the state preparation circuit for ϕ_k on the unknown state and measure in the computational basis, checking for an all-zero outcome. When ϕ_k is not pure, these strategies do not immediately determine whether an unknown state is equal to ϕ_k. However, there may be other verification strategies for low-rank ϕ_k. Prior work left open the question of whether pure-state output OWSGs were a different kind of hardness assumption than mixed-state output OWSGs, even when the rank of those mixed states is low. We present a simple reduction from low-rank, gapped mixed-state output OWSGs to pure-state output OWSGs.

Corollary. *If there is a low-rank one-way state generator where every ϕ_k has a minimum eigenvalue that is inverse-polynomial in the security parameter λ, then pure-state one-way state generators exist.*

This reduction is achieved by applying either of our two anti-symmetrization algorithms to a low-rank, gapped mixed-state output OWSG, exploiting that, for each k, there is a unique pure state obtained as the output of anti-symmetrization. Furthermore, since our anti-symmetrization algorithms are efficient, computational one-wayness carries over to the anti-symmetrized state. We also show that there is likely no black-box reduction (that only uses copies of the state) from mixed-state OWSGs to pure-state OWSGs beyond $\mathrm{poly}(\lambda)$-rank mixed states by extending the oracle separation of [BCN25] to $\lambda^{\log^{1+\epsilon}(\lambda)}$-rank OWSGs.

1.2 Open questions

Optimal bounds for (streaming) anti-symmetrization. We achieved streaming anti-symmetrization from $O(r\log(r)/\lambda_{\min})$ copies. Without the streaming requirement we showed that $O(r/\lambda_{\min})$ copies suffice. This raises two natural questions: First, what is the optimal complexity of anti-symmetrization? Among protocols that only act through permutation operations, arguments from Schur-Weyl duality suggest that $\Theta(\sum_{i=1}^{r}\lambda_i^{-1})$ copies are necessary and sufficient. But among general protocols, only $\Omega(\lambda_{\min}^{-1})$ copies may be needed. Is there a protocol matching this bound? Answering this question will also help clarify whether a reduction from gapped low-rank mixed-state output OWSGs to their pure-state counterpart via anti-symmetrization fundamentally requires the gappedness assumption. Second, can the optimal complexity of anti-symmetrization be achieved by a streaming algorithm that uses a small (say, polynomial in r and $\log(d)$) working memory, or are there trade-offs between the available quantum memory and the achievable sample complexity?

Anti-symmetrization for support learning and testing. The random purification channel of [TWZ26] and its variants have proven useful in developing algorithms for learning mixed states via existing pure-state procedures, see e.g. [Pel+25; MB26; Che+26a; Che+26b; Gir+26]. However, in the context of property testing, using the random purification channel is more challenging because any single mixed state admits a whole (local-unitary) orbit of possible purifications, from which it draws one at random. Thus, even if the initial testing problem is about certifying a single mixed target hypothesis state, under random purification it becomes a task of certifying a target orbit of pure states. Anti-symmetrization with its unique pure target state may help here. Note that anti-symmetrization applied to copies of an unknown mixed state as studied in this work intentionally loses much of the spectral information (except the number of non-zero eigenvalues), but it retains information about the state's support. It therefore may be used to reduce tasks of learning or testing properties of the support of a mixed state to corresponding pure anti-symmetric state problems.

Anti-symmetrization for quantum chemistry. In quantum chemistry, pure states in the anti-symmetric subspace are often referred to as *Slater determinants*, which describe certain non-interacting fermionic systems. They appear as central objects in many mean-field theories, such as Hartree-Fock theory and density functional theory [CF23], are important to our understanding of chemical processes [Cha24], and serve as a starting point for simulating fermionic interactions [Fom+24]. Whereas standard algorithms for preparing a Slater determinant start from a classical description, which comes either from a theoretical prediction or an experimental setup, in a world with quantum computers, quantum systems may instead be specified implicitly through copies of quantum

states. Does our streaming anti-symmetrization, which can prepare a pure Slater determinant from such a quantum description without having to learn a full description of the state, be used for applications in quantum chemistry?

Note added. While we were preparing our manuscript, the closely related, independent work [Bra+26] by Shrigyan Brahmachari, David Jakab, Henry D. Pfister, and Iman Marvian appeared on the arXiv. An earlier work that involved many of the same authors [Bra+25] was a major influence on Section 4 of our paper. The main goal of [Bra+26] is unitary Schur sampling, which that work achieves via the same growth routine as our Algorithm 1 to iteratively extract the largest anti-symmetric subsystem. Thus, the relationship between Schur sampling and anti-symmetrization is common to our works, both conceptually and technically. While we implement unitary Schur sampling and purity amplification for qudits with $\widetilde{O}(n^2 d)$ controlled-SWAP operations, their implementation uses $\widetilde{O}(\min\{n^4, nd^3\})$ pairwise SWAP tests. Our proofs were completed before [Bra+26] appeared on the arXiv, but (as we acknowledge below) they were developed with the help of AI and we cannot rule out the possibility that this AI was influenced by an unpublished draft of their work.

Acknowledgments and AI disclosure

We thank Jack Spilecki for helpful discussions about representation theory and applications of the random purification channel.

AI made a substantive contribution to this work. The human authors wrote an early draft of this work, including algorithms for anti-symmetrization and symmetrization that involved substantially more representation theory. When asked to review said draft, ChatGPT-6 Pro proposed simplifications to the earlier anti-symmetrization algorithm that the authors had developed, which led to the current presentation of Algorithms 1 and 2. Claude Opus 5.5 and Fable 5.1 were used to aid with technical aspects of representation theory in developing Section 4 and to help improve presentation of the mathematical arguments. All final proofs were written and independently verified by the authors.

2 Preliminaries

We consider a system of n qudits, described by the Hilbert space $\left(\mathbb{C}^d\right)^{\otimes n}$. We let S_n denote the symmetric group on n elements, and for any permutation $\sigma \in S_n$ we let P_σ denote the natural action of σ on this space, defined by

$$P_\sigma \left(|\psi_1\rangle \otimes |\psi_2\rangle \otimes ... \otimes |\psi_n\rangle\right) = |\psi_{\sigma^{-1}(1)}\rangle \otimes |\psi_{\sigma^{-1}(2)}\rangle \otimes ... \otimes |\psi_{\sigma^{-1}(n)}\rangle. \tag{1}$$

When a permutation τ acts on fewer than n elements, we define P_τ to act on the first registers of the space. So, for example, for a permutation $\tau \in S_{n-1}$,

$$P_\tau \left(|\psi_1\rangle \otimes |\psi_2\rangle \otimes ... \otimes |\psi_{n-1}\rangle \otimes |\psi_n\rangle\right) = |\psi_{\tau^{-1}(1)}\rangle \otimes |\psi_{\tau^{-1}(2)}\rangle \otimes ... \otimes |\psi_{\tau^{-1}(n-1)}\rangle \otimes |\psi_n\rangle. \tag{2}$$

The anti-symmetric subspace projector on the first k qudits of this space is denoted by Π^k_{anti} and defined as

$$\Pi^k_{\text{anti}} := \frac{1}{k!} \sum_{\sigma \in S_k} \text{sgn}(\sigma) P_\sigma. \tag{3}$$

Routine calculation shows that this is indeed a self-adjoint projector.

We now make a standard but useful observation: the anti-symmetric subspace projector on $k+1$ registers can be written in terms of the anti-symmetric subspace projector on k registers multiplied by a sum of transpositions. We formalize this in the following lemma.

Lemma 1. *We have*

$$\left(1 - \sum_{i=1}^{k} P_{(i,k+1)}\right) \Pi^k_{\text{anti}} = \Pi^k_{\text{anti}} \left(1 - \sum_{i=1}^{k} P_{(i,k+1)}\right) = (k+1)\Pi^{k+1}_{\text{anti}}\,.$$

Proof. Multiplying out terms gives that

$$\begin{aligned}
\left(1 - \sum_{i=1}^{k} P_{(i,k+1)}\right) \Pi^k_{\text{anti}} &= \frac{1}{k!} \left(1 - \sum_{i=1}^{k} P_{(i,k+1)}\right) \sum_{\tau \in S_k} \text{sgn}(\tau) P_\tau \\
&= \frac{1}{k!} \left(\sum_{\sigma \in S_{k+1}} \text{sgn}(\sigma) P_\sigma \right) \\
&= (k+1)\Pi^{k+1}_{\text{anti}}\,,
\end{aligned}$$

where the equality on the second line follows by noting that any element in S_{k+1} can be written uniquely as an element in S_k left multiplied by either a transposition touching $k+1$ or the identity. (To verify this, simply write a permutation $\sigma \in S_{k+1}$ as $\sigma = (k+1, \sigma(k+1))^2 \circ \sigma$ and then note $(k+1, \sigma(k+1)) \circ \sigma \in S_k$.) A similar proof yields the other equality claimed in the lemma, only now multiplying by transpositions on the right. □

Motivated by Lemma 1, we define the operator

$$A^{[k],k+1} := \frac{1}{k+1} \left(1 - \sum_{i=1}^{k} P_{(i,k+1)}\right). \tag{4}$$

We also note, as a simple corollary of Lemma 1, that anti-symmetric subspace projectors commute.

Corollary 2. *We have*

$$\Pi^{k+1}_{\text{anti}} \Pi^k_{\text{anti}} = \Pi^k_{\text{anti}} \Pi^{k+1}_{\text{anti}}\,. \tag{5}$$

Proof. Lemma 1 gives

$$\Pi^k_{\text{anti}} \Pi^{k+1}_{\text{anti}} = \Pi^k_{\text{anti}} A^{[k],k+1} \Pi^k_{\text{anti}} = \Pi^{k+1}_{\text{anti}} \Pi^k_{\text{anti}} \tag{6}$$

□

We extend the notation developed so far to act on arbitrary subsets of n qudits. For any $S \subseteq [n]$ let Π^S_{anti} be the anti-symmetric subspace projector acting on those qudits. Similarly, for every $S \subseteq [n]$ and $j \in [n] \backslash S$ define

$$A^{S,j} = \frac{1}{|S|+1} \left(1 - \sum_{s \in S} P_{(s,j)} \right). \tag{7}$$

Note that for any such S and j we have

$$A^{S,j} \Pi^S_{\text{anti}} = \Pi^S_{\text{anti}} A^{S,j} = \Pi^{S \cup \{j\}}_{\text{anti}} \tag{8}$$

and

$$\Pi^S_{\text{anti}} \, \Pi^{S \cup \{j\}}_{\text{anti}} = \Pi^{S \cup \{j\}}_{\text{anti}} \Pi^S_{\text{anti}}. \tag{9}$$

3 Two algorithms for anti-symmetrization

This section gives a formal description and analysis of two algorithms for preparing the anti-symmetric state over the support of an unknown mixed state from access to a stream of copies. It begins with Algorithm 1, which describes an efficient circuit that implements the two-outcome "growth" measurement

$$\left\{ A^{[k],k+1}, \left(1 - A^{[k],k+1}\right)^{1/2} \right\} \tag{10}$$

on the subspace of all states $|\psi\rangle$ such that $\Pi^k_{\text{anti}} |\psi\rangle = |\psi\rangle$. We call the first outcome of this measurement the success outcome, and the second the failure outcome. When this measurement is applied to an anti-symmetric state on the first k registers and succeeds, Lemma 1 implies that it produces an anti-symmetric state on $k+1$ registers. When it fails, it still produces an anti-symmetric state on the first k registers because of Corollary 2.

Algorithm 2 repeatedly applies the measurement described in Algorithm 1 to the anti-symmetric state grown so far and a fresh copy taken from a stream of iid mixed-state copies to produce an anti-symmetric state over the support of the unknown state. When the growth measurement succeeds, the anti-symmetric state grows by one register. When it fails, the mixed-state register is discarded. In either case, the process repeats.

Algorithm 3 is a more sample-efficient but less time-efficient version of Algorithm 2 in which, rather than being discarded, mixed states involved in a failure are saved and retested later. The representation-theoretic properties of this algorithm are discussed in Section 4.

We now give a formal description of each of these algorithms and their analysis. For ease of presentation, we write Algorithm 1 as processing a pure input state, but it can clearly also be invoked on mixed states, and the guarantees extend accordingly by linearity.

Algorithm 1. Anti-symmetrization growth measurement

Input: A state $|\psi\rangle \in \left(\mathbb{C}^d\right)^{\otimes k+1}$ satisfying $\Pi^k_{\text{anti}} |\psi\rangle = |\psi\rangle$.

Output: Either: *Success*, and a state $|\phi\rangle \in \left(\mathbb{C}^d\right)^{\otimes k+1}$ satisfying $\Pi_{\text{anti}}^{k+1} |\phi\rangle = |\phi\rangle$; or *Failure*, and a state $|\psi'\rangle \in \left(\mathbb{C}^d\right)^{\otimes k+1}$ satisfying $\Pi_{\text{anti}}^{k} |\psi'\rangle = |\psi'\rangle$.

1. Let W be any unitary preparing the uniform superposition on a $(k+1)$-dimensional ancilla register: $W|0\rangle = \frac{1}{\sqrt{k+1}} \sum_{i=0}^{k} |i\rangle$. Prepare this state, and label it the `ancilla` register. Call the register holding $|\psi\rangle$ the `active` register.

2. For each $1 \leq i \leq k$, apply the (negated) swap operation $-P_{(i,k+1)}$ to the state $|\psi\rangle$ controlled on the ancilla being in state $|i\rangle$. That is, apply the unitary

$$U_A = |0\rangle\langle 0| \otimes I - \sum_{i=1}^{k} |i\rangle\langle i| \otimes P_{(i,k+1)} \tag{11}$$

   to the `ancilla` and `active` registers.

3. Apply $W^\dagger$.

4. Apply the two-outcome projective measurement

$$\{|0\rangle\langle 0|, 1 - |0\rangle\langle 0|\} \tag{12}$$

   to the `ancilla` register.

5. If outcome $|0\rangle\langle 0|$ is observed, report *success* and return the state in the `active` register.

6. If outcome $1 - |0\rangle\langle 0|$ is observed, report *failure*, apply the operator $U_A(W \otimes I)$ to the active and ancilla registers, then return the state in the `active` register.

Theorem 3. *Algorithm 1 is correct, i.e., it outputs a state with the claimed anti-symmetry properties in both the success and failure cases. Furthermore, the algorithm outputs success with probability* $\langle\psi| \Pi_{\text{anti}}^{k+1} |\psi\rangle$.

Proof. We analyze the *success* case first. This can be understood as an example of the linear combination of unitaries technique. After the $|0\rangle\langle 0|$ outcome is observed in step 5 of the algorithm, the (unnormalized) state in the `ancilla` and `active` registers is:

$$\frac{1}{\sqrt{k+1}} \left(|0\rangle\langle 0| \otimes I\right) \left(W^\dagger \otimes I\right) U_A \left(\sum_{i=0}^{k} |i\rangle \otimes |\psi\rangle\right) \tag{13}$$

$$= \frac{1}{\sqrt{k+1}} \left(|0\rangle\langle 0| \otimes I\right) \left(W^\dagger \otimes I\right) \left(|0\rangle |\psi\rangle - \sum_{i=1}^{k} |i\rangle P_{(i,k+1)} |\psi\rangle\right) \tag{14}$$

$$= \frac{1}{k+1} \left(|0\rangle \left(\sum_{i=0}^{k} \langle i|\right) \otimes I\right) \left(|0\rangle |\psi\rangle - \sum_{i=1}^{k} |i\rangle P_{(i,k+1)} |\psi\rangle\right) \tag{15}$$

$$= \frac{1}{k+1} |0\rangle \otimes \left(1 - \sum_{i=1}^{k} P_{(i,k+1)}\right) |\psi\rangle \tag{16}$$

$$= |0\rangle \otimes A^{[k],k+1} |\psi\rangle, \tag{17}$$

where we recall the definition of $A^{[k],k+1}$ given in Equation (4). By assumption of the algorithm, we have $|\psi\rangle = \Pi^k_{\text{anti}} |\psi\rangle$ and therefore

$$A^{[k],k+1} |\psi\rangle = A^{[k],k+1} \Pi^k_{\text{anti}} |\psi\rangle \tag{18}$$

$$= \Pi^{k+1}_{\text{anti}} |\psi\rangle \ , \tag{19}$$

which both gives the probability of the algorithm outputting *success*, and proves correctness of the algorithm in this case.

In the *failure* case, the overall operator applied to the `active` and `ancilla` registers is

$$U_A (W \otimes I) \left((I - |0\rangle\langle 0|) \otimes I \right) (W^\dagger \otimes I) U_A = I - U_A \left(\left(W |0\rangle\langle 0| W^\dagger \right) \otimes I \right) U_A. \tag{20}$$

Building on the analysis of the *success* case gives

$$\frac{1}{\sqrt{k+1}} U_A \left(\left(W |0\rangle\langle 0| W^\dagger \right) \otimes I \right) U_A \left(\sum_{i=0}^{k} |i\rangle \otimes |\psi\rangle \right) \tag{21}$$

$$= U_A \left(W |0\rangle \otimes \Pi^{k+1}_{\text{anti}} |\psi\rangle \right) \tag{22}$$

$$= \frac{1}{\sqrt{k+1}} \left(|0\rangle \otimes I - \sum_{i=1}^{k} |i\rangle \otimes P_{(i,k+1)} \right) \Pi^{k+1}_{\text{anti}} |\psi\rangle \tag{23}$$

$$= W |0\rangle \otimes \Pi^{k+1}_{\text{anti}} |\psi\rangle \tag{24}$$

from which it follows that the state in the `ancilla` and `active` registers after applying all operations in the *failure* branch of the algorithm is

$$\left(I - U_A \left(\left(W |0\rangle\langle 0| W^\dagger \right) \otimes I \right) U_A \right) (W |0\rangle \otimes |\psi\rangle) = W |0\rangle \otimes (I - \Pi^{k+1}_{\text{anti}}) |\psi\rangle \ , \tag{25}$$

and the state returned is

$$|\psi'\rangle = (I - \Pi^{k+1}_{\text{anti}}) |\psi\rangle \tag{26}$$

$$= (I - \Pi^{k+1}_{\text{anti}}) \Pi^k_{\text{anti}} |\psi\rangle \tag{27}$$

$$= \Pi^k_{\text{anti}} (I - \Pi^{k+1}_{\text{anti}}) |\psi\rangle = \Pi^k_{\text{anti}} |\psi'\rangle \ , \tag{28}$$

where we have used the assumption on $|\psi\rangle$ on the second line, and Corollary 2 on the third. □

We briefly describe a low-depth implementation of Algorithm 1 given large fanout gates:

Lemma 4. *Algorithm 1 can be implemented in* QNC^0_f*, that is, polynomial-size, constant-depth circuits with unbounded fanout gates.*

Proof. We describe constant-depth implementations for each of the first three steps of the algorithm over qubits, though the construction would be similar for other local dimensions:

Step 1 *&* 3*:* It will be easiest if we prepare the states in their "unary" encoding, with $|0\rangle$ represented as $|0^k\rangle$ and $|i\rangle$ represented by $|0, \ldots, 1, \ldots 0\rangle$ where the 1 is in the ith position. To do this, we can first create the W-state on $k+1$ qubits [GMW26], flip the $(k+1)$st qubit if the first k are in the all-zero state (with one generalized Toffoli gate), and then drop the last qubit.

Step 2: The key barrier to parallelizing this step is that all the (controlled) $P_{(i,k+1)}$ terms touch the $(k+1)$st qudit of $|\psi\rangle$. To circumvent this issue, we will use a variant of a technique from [HS05], which parallelizes mutually commuting operations. While the $P_{(i,k+1)}$ transpositions do not commute, the controlled versions do. Below is the sequence of operations in QNC^0_f we use to implement this scheme. It uses, at its heart, the fact that a SWAP gate can be built from three CNOT gates. We track how the classical basis states evolve under each operation. For simplicity, first assume the local dimension is $d = 2$. The registers below are (in order): k-bit ancilla register; first k bits of $|\psi\rangle$; the $(k+1)$st bit of $|\psi\rangle$ appended with $(k-1)$ zero bits; k zero bits; 1 zero bit. We have (when $i \neq 0$):

$$
\begin{aligned}
& |i\rangle\,|x_1,\ldots,x_i,\ldots,x_k\rangle\,|y,0,\ldots,0\rangle\,|0,\ldots,0,\ldots,0\rangle\,|0\rangle \\
\xrightarrow{\text{Fanout}}\ & |i\rangle\,|x_1,\ldots,x_i,\ldots,x_k\rangle\,|y,\ldots,y,\ldots y\rangle\,|0,\ldots,0,\ldots,0\rangle\,|0\rangle \\
\xrightarrow{\text{Toffolis}}\ & |i\rangle\,|x_1,\ldots,y\oplus x_i,\ldots,x_k\rangle\,|y,\ldots,y,\ldots y\rangle\,|0,\ldots,y\oplus x_i,\ldots,0\rangle\,|0\rangle \\
\xrightarrow{\text{Parity}}\ & |i\rangle\,|x_1,\ldots,y\oplus x_i,\ldots,x_k\rangle\,|y,\ldots,y,\ldots y\rangle\,|0,\ldots,y\oplus x_i,\ldots,0\rangle\,|y\oplus x_i\rangle \\
\xrightarrow{\text{Fanout}}\ & |i\rangle\,|x_1,\ldots,y\oplus x_i,\ldots,x_k\rangle\,|x_i,\ldots,x_i,\ldots x_i\rangle\,|0,\ldots,y\oplus x_i,\ldots,0\rangle\,|y\oplus x_i\rangle \\
\xrightarrow{\text{Parity}}\ & |i\rangle\,|x_1,\ldots,y\oplus x_i,\ldots,x_k\rangle\,|x_i,\ldots,x_i,\ldots x_i\rangle\,|0,\ldots,y\oplus x_i,\ldots,0\rangle\,|0\rangle \\
\xrightarrow{\text{Toffolis}}\ & |i\rangle\,|x_1,\ldots,y,\ldots,x_k\rangle\,|x_i,\ldots,x_i,\ldots x_i\rangle\,|0,\ldots,0,\ldots,0\rangle\,|0\rangle \\
\xrightarrow{\text{Fanout}}\ & |i\rangle\,|x_1,\ldots,y,\ldots,x_k\rangle\,|x_i,\ldots,0,\ldots 0\rangle\,|0,\ldots,0,\ldots,0\rangle\,|0\rangle
\end{aligned}
$$

When $i = 0$, the circuit construction above does nothing as desired. To obtain the minus signs, we only need to apply a layer of Z gates to our ancilla register since at most one location will be a 1. To generalize to multiple qubits (i.e., $d = 2^m$), we simply run the entire circuit above in parallel for each bit. We use a Fanout gate to parallelize the controls even further; that is, the ancilla state $|i\rangle$ becomes $|0^m,\ldots,1^m,\ldots 0^m\rangle$ where the 1's are in the ith block of n bits. □

Next, we describe and analyze our first anti-symmetrization algorithm, which repeatedly applies Algorithm 1 to prepare an anti-symmetric state while discarding failed copies.

Algorithm 2. Anti-symmetrization without copy re-use

Input: n copies of some unknown mixed state $\rho \in \mathcal{L}(\mathbb{C}^d)$.
Output: An anti-symmetric state on k qudits for some $1 \leq k \leq d$.

1. Label three registers: `anti-symmetric`, `input`, and `discard`.
2. Set $k = 1$ and add one copy of ρ to the `anti-symmetric` register, and all others to the `input` register.
3. While the `input` register is non-empty and $k < d$:
   (a) Take the state from the `anti-symmetric` register and the first qudit in the `input` register, then apply Algorithm 1 to them.

(b) If the run of Algorithm 1 from (a) returns *success*, increase k by 1, and add the state output by the test to the `anti-symmetric` register.

(c) If the run of Algorithm 1 from (a) returns *failure*, add the first k qudits of the state returned to the `antisymmetric` register and the last qudit to the `discard` register.

4. Return the state in the `anti-symmetric` register.

Theorem 5. *Algorithm 2 is correct, that is, it always returns an anti-symmetric state. Furthermore if the input state ρ has rank r and minimum non-zero eigenvalue $\lambda_{\min}$ then Algorithm 2 returns the unique anti-symmetric state over the support of ρ with probability at least $1 - r\left(1 - \frac{\lambda_{\min}}{r}\right)^n \geq 1 - r\exp(-n\lambda_{\min}/r)$.*

Explicitly, if

$$\rho = \sum_{i=1}^{r} \lambda_i \left|\varphi_i\right\rangle\!\left\langle\varphi_i\right| \tag{29}$$

then Algorithm 2 returns a state proportional to

$$\Pi^r_{\text{anti}} \left(\left|\varphi_1\right\rangle \otimes \left|\varphi_2\right\rangle \otimes ... \otimes \left|\varphi_r\right\rangle\right) \tag{30}$$

with probability as bounded above.

Proof. Correctness of Algorithm 2 is immediate from correctness of Algorithm 1. It remains to bound the probability that this algorithm outputs the anti-symmetric state over the support of ρ.

Rather than analyzing the algorithm over mixed states directly, we interpret each ρ as a probabilistic mixture of orthogonal pure (eigen)states $\{\left|\varphi_1\right\rangle, \left|\varphi_2\right\rangle, ..., \left|\varphi_r\right\rangle\}$ and analyze the algorithm on each pure state in the resulting mixture. As the algorithm only acts on these states with controlled-SWAP operations and since Algorithm 1 always outputs an anti-symmetric state, any state stored in the `anti-symmetric` register during the run of the algorithm must be equivalent to a mixture of states produced by applying the anti-symmetric subspace projector applied to some subset of states from $\{\left|\varphi_1\right\rangle, \left|\varphi_2\right\rangle, ..., \left|\varphi_r\right\rangle\}$. Consider the case where this subset has size k, and describe one arbitrary state in this mixture as

$$\left|\Phi^S\right\rangle \propto \Pi^k_{\text{anti}} \left(\bigotimes_{s\in S} \left|\varphi_s\right\rangle\right), \tag{31}$$

where $S \subseteq [r]$, $|S| = k$. Now straightforward calculation gives

$$\left\langle\Phi^{(S)}\right| \left\langle\varphi_j\right| \Pi^{k+1}_{\text{anti}} \left|\Phi^{(S)}\right\rangle \left|\varphi_j\right\rangle = 0 \tag{32}$$

if $j \in S$, and

$$\left\langle\Phi^{(S)}\right| \left\langle\varphi_j\right| \Pi^{k+1}_{\text{anti}} \left|\Phi^{(S)}\right\rangle \left|\varphi_j\right\rangle = \left\langle\Phi^{(S)}\right| \left\langle\varphi_j\right| A^{[k],k+1} \Pi^k_{\text{anti}} \left|\Phi^{(S)}\right\rangle \left|\varphi_j\right\rangle = \frac{1}{k+1} \tag{33}$$

if $j \notin S$. A fresh state drawn from the `input` register is in the latter $j \notin S$ case with probability

$$\sum_{j\notin S} \lambda_j \geq (r-k)\lambda_{\min}. \tag{34}$$

So we conclude that every time a state is removed from the `input` register, the next measurement (i.e., the next run of Algorithm 1) increases the size of the state in the `anti-symmetric` register with probability at least

$$\frac{\lambda_{\min}(r-k)}{(k+1)} \geq \frac{\lambda_{\min}(r-k)}{r}, \tag{35}$$

since the size of the state in the anti-symmetric register is strictly less than r while that register is incomplete. The latter probability is exactly the probability of incrementing the number of collected coupons if there are r coupons overall and each coupon appears with probability $\lambda_{\min}/r$. So, standard union bound arguments give that the probability of not collecting all coupons after n rounds is upper bounded by

$$\sum_{i\in[r]} \Pr[\text{Coupon } i \text{ is not collected}] = r\left(1-\frac{\lambda_{\min}}{r}\right)^n \leq re^{-n\lambda_{\min}/r}. \tag{36}$$

This is the claimed bound on the probability of Algorithm 2 returning the desired anti-symmetric state. □

Remark 6. The proof of Theorem 5 also shows the following: If we are promised in advance that the unknown state ρ has rank r, then we can change the WHILE loop in Step 3 of Algorithm 2 to be over $k < r$. If ρ has rank r, then the algorithm can never increment the counter k beyond r.

In Algorithm 2, states moved to the `discard` register are never used again. Thus, they can be traced out without affecting the algorithm's performance. This makes it well suited for a "streaming" model in which new states are added to the `input` register one at a time, and only states in the `anti-symmetric` register are stored.

However, it is possible to improve the sample complexity of Algorithm 2 by abandoning this streaming-like behavior. The following key observation underlies this improvement: operators applied to states in the *Failure* mode case of Algorithm 1 across different rounds do not commute with one another, so it is possible that a state which is initially rejected by the algorithm can later be used to grow the anti-symmetric state (after a different state is rejected). Algorithm 3 describes this process formally. This algorithm also plays a key role in the quantum purification techniques described in Section 4. Because this algorithm is used in these two different contexts, we allow its input to be an arbitrary state; the subsequent analysis in Theorem 7 then describes the algorithm's performance in the special case that its input is $\rho^{\otimes n}$.

Algorithm 3. Anti-symmetrization with copy re-use

Input: An unknown state $\rho \in \mathcal{L}\left(\left(\mathbb{C}^d\right)^{\otimes n}\right)$ and a confidence parameter ϵ.
Output: An anti-symmetric state on k qudits for some $1 \leq k \leq d$.

1. Label two registers (sets of qudits): `anti-symmetric` and `candidates`.
2. Set $k = 1$ and the first qudit of the input state to the `anti-symmetric` register, and all other qudits to the `candidates` register.
3. Repeat $\lceil nd \log(d/\epsilon) \rceil$ times:
   (a) Take the state from the `anti-symmetric` register and a randomly selected qudit from the `candidates` register, then apply Algorithm 1 to them.
   (b) If the run of Algorithm 1 from (a) returns *success*, increase k by 1, and add the returned state to the `anti-symmetric` register. If $k = d$ or `candidates` is empty, go to step 4.
   (c) If the run of Algorithm 1 from (a) returns *failure*, add the first k qudits of the returned state to the `antisymmetric` register and the last qudit of the returned state to the `candidates` register.
4. Return the state in the `anti-symmetric` register.

Theorem 7. *Algorithm 3 is correct, that is, it always returns an anti-symmetric state. Furthermore if the input state is of the form $\rho^{\otimes n}$, where ρ has rank r and minimum non-zero eigenvalue $\lambda_{\min}$, then the algorithm returns the unique anti-symmetric state over the support of ρ with probability at least $1 - (\epsilon + \exp(-\Omega(n\lambda_{\min})))$ provided $n \geq Cr/\lambda_{min}$ for a sufficiently large constant $C > 0$.*
Explicitly, if

$$\rho = \sum_{i=1}^{r} \lambda_i \left|\varphi_i\right\rangle\!\left\langle\varphi_i\right| \tag{37}$$

then Algorithm 3 returns a state proportional to

$$\Pi^r_{\text{anti}} \left(\left|\varphi_1\right\rangle \otimes \left|\varphi_2\right\rangle \otimes \ldots \otimes \left|\varphi_r\right\rangle\right) \tag{38}$$

with probability as bounded above.

Proof. Correctness follows immediately from correctness of Algorithm 1. The analysis of the state returned by the algorithm is most cleanly expressed using techniques from Schur-Weyl duality, and so is deferred to Section 4. □

Remark 8. The proof of Theorem 7 also shows the following: If we are promised in advance that the unknown state ρ has rank r, then it suffices to take $\lceil nr \log(r/\epsilon) \rceil$ repetitions in Step 3 of Algorithm 3.

4 Connections to Schur-Weyl duality

In this section we discuss the anti-symmetrization algorithm developed in Section 3 from the perspective of Schur-Weyl duality. This perspective gives two new insights: an analysis of the sample-efficient anti-symmetrization algorithm (Algorithm 3) and an extension of that algorithm to give a new algorithm for optimal quantum purity amplification [Li+25]. The latter algorithm gives a qudit generalization of the swap-test-based optimal quantum purity amplification protocol for qubits developed in [Bra+25], answering an open question from that work. We begin this section with a review of some necessary concepts from Schur-Weyl duality, then present the analysis of Algorithm 3 and its extension.

4.1 Schur-Weyl background

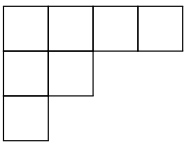
This subsection reviews some facts from the theory of Schur-Weyl duality. We review only what is required for the proofs in this paper, and so many foundational details of the theory are omitted. We refer the reader to [Wri16, Chapter 2] for a more complete introduction to the topic and its relationship with quantum information.

The key insight we require is that $\left(\mathbb{C}^d\right)^{\otimes n}$ is isomorphic to a direct sum of tensor products of irreducible modules of the symmetric group S_n and the general linear group GL_d. We can write this correspondence as

$$\left(\mathbb{C}^d\right)^{\otimes n} \simeq \bigoplus_{\lambda \vdash_d n} V_\lambda \otimes W_\lambda \,, \tag{39}$$

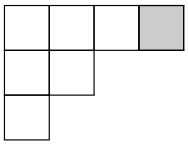
where each W_λ is an irreducible module of GL_d and each V_λ is an irreducible module of S_n. It is worth noting that the isomorphism mapping from $\left(\mathbb{C}^d\right)^{\otimes n}$ to $\bigoplus_{\lambda \vdash_d n} V_\lambda \otimes W_\lambda$, which we commonly refer to as the full Schur transform, is in general not unique. In particular, to define a concrete Schur transform, one needs to specify a proper basis for both W_λ and V_λ, e.g., the Gelfand-Tsetlin basis for W_λ and the Young-Yamanouchi basis for V_λ. The particular choice of basis will not be important for our application. Nonetheless, we will arbitrarily fix a basis and denote the corresponding Schur transform unitary operator by $U_{\mathrm{schur}} : \left(\mathbb{C}^d\right)^{\otimes n} \mapsto \bigoplus_{\lambda \vdash_d n} V_\lambda \otimes W_\lambda$ for ease of discussion.

The irreducible modules V_λ and W_λ are indexed by *partitions* of n into at most d parts, denoted $\lambda \vdash_d n$. Each such λ is a non-increasing sequence of at most d positive integers that sum to n. We represent these partitions using Young diagrams: sequences of boxes with lengths of the rows corresponding to the integers in the partition. For example,

$\lambda = (4, 2, 1) =$  (40)

is a partition of 7 into three parts. We use the notation λ_i for the length of the i-th row of the Young diagram, and λ'_i for the length of the i-th column.

For any given box (or cell) of a Young tableau, its content is defined to be its column index minus its row index. For example, the shaded box in the diagram



(41)

has content $4 - 1 = 3$. For a given box $\square$ we denote its content by $c(\square)$. The content sum of a Young diagram, denoted $c(\lambda)$ is defined to be the sum of the contents of all its boxes. For our running example:

$$c(\lambda) = \sum_{\square \in \lambda} c(\square) = 0 - 1 - 2 + 1 + 0 + 2 + 3 = 3. \tag{42}$$

We say a vector $|\psi\rangle \in \left(\mathbb{C}^d\right)^{\otimes n}$ is in the λ-isotypic component of $\left(\mathbb{C}^d\right)^{\otimes n}$ if $U_{\text{schur}} |\psi\rangle$ lies in a $V_\lambda \otimes W_\lambda$ module [2]. We also denote the projector onto this isotypic component by Π_λ.

For any $\sigma \in S_n$, the permutation operator P_σ can be block diagonalized into an operator that acts trivially on the W_λ modules under the Schur transform, i.e.,

$$U_{\text{schur}} P_\sigma U_{\text{schur}}^\dagger = \bigoplus_{\lambda \vdash n} D_\lambda(\sigma) \otimes I_{W_\lambda}, \tag{43}$$

where $D_\lambda(\sigma)$ is the representation of σ under the irrep V_λ, and I_{W_λ} is the identity under the irrep W_λ.

This also means it is sometimes possible to characterize the action of some collection of permutations on a state just by knowing the isotypic component the state lies in. We give one example next.

Lemma 9. *Let $|\psi\rangle$ be a state in the λ-isotypic component of $\left(\mathbb{C}^d\right)^{\otimes n}$ and define*

$$T([n]) = \sum_{i,j \in [n],\, i<j} P_{(i,j)}.$$

Then we have

$$T([n]) |\psi\rangle = c(\lambda) |\psi\rangle . \tag{44}$$

Proof. This is, or at least is equivalent to, a standard result in the representation theory of the symmetric group [OV96; Juc74; Mur81]. We briefly sketch the main ideas here. The starting point is to observe that the sum of transpositions $\sum_{i,j \in [n], i<j} (i,j)$ is central in the group algebra of the symmetric group, meaning it commutes with every other element in the algebra. Applying Schur's lemma then gives that for any vector $|x\rangle$ that lies in some irreducible representation of type λ it holds $T([n]) |x\rangle = c(\lambda) |x\rangle$ for some constant $c(\lambda)$. For a vector $|\psi\rangle$ within the λ-isotypic component, we can decompose $|\psi\rangle$ into its projections $\{|x_i\rangle\}$ within each irreducible representations of type λ within the isotypic component. We thus immediately have that $T([n]) |\psi\rangle = \sum_i T([n]) |x_i\rangle = c(\lambda) \sum_i |x_i\rangle = c(\lambda) |\psi\rangle$. This concludes the proof of Lemma 9. □

[2] As we have said, U_{schur} is not unique. However, a standard fact from representation theory is that the isotypic component itself is unique regardless of the choice of U_{schur}.

Another important concept in our work is that of *restrictions* of isotypic components. Consider an n-qudit state $|\psi\rangle$ in a λ-isotypic component. Assume further that a subset of k qudits of that state are in a μ-isotypic component, meaning that if we project those qudits onto the μ-isotypic component, it leaves the state $|\psi\rangle$ unchanged (or, equivalently, if we trace out all but those k qudits, the resulting state is supported on the μ-isotypic component). The Littlewood-Richardson rule then describes the possible isotypic components which the complementary subset of $n-k$ qudits of $|\psi\rangle$ can be supported on. We will be interested in a special case of this rule, called Pieri's rule, which considers the same situation when the k qudits are supported on the $\mu = (1,1,1,..,1)$-isotypic subspace. We state this next.

Lemma 10 (Pieri's Rule, e.g., see Theorem 2.7.1 of [GR20])**.** *Let $|\psi\rangle$ be an n-qudit state in a λ-isotypic component. Assume that a subset of k qudits of $|\psi\rangle$ are in the $\mu = (1,1,...,1)$-isotypic component. Then the $n-k$ complementary qubits of $|\psi\rangle$ can only have support overlapping ν-isotypic components of partitions ν that can be obtained by removing exactly one box from k distinct rows of λ.*

As an illustrative example, assume a 6-qudit state $|\psi\rangle$ is in the λ-isotypic component with

$$\lambda = (2,2,2) = \ydiagram{2,2,2}. \tag{45}$$

If the first two qudits of $|\psi\rangle$ are in the $(1,1)$-isotypic component, then the mixed state $\mathrm{Tr}_{12}\left[|\psi\rangle\langle\psi|\right]$ may be supported on the $(2,1,1)$-isotypic component, since it can be obtained by deleting one box from the second row of λ and one from the third, as indicated:

$$\ydiagram{2,2,2} \;\rightarrow_{\text{allowed deletion}}\; \ydiagram{2,1,1}. \tag{46}$$

However, this mixed state may not be supported on the $(2,2)$-isotypic component (despite it having the correct number of overall boxes) because this transition would require deleting two boxes from the same row:

$$\ydiagram{2,2,2} \;\rightarrow_{\text{forbidden deletion}}\; \ydiagram{2,2}. \tag{47}$$

There is a close relation between Pieri's rule and the anti-symmetric subspace, coming from the following fact.

Lemma 11. *For any state $|\psi\rangle$ on n qudits and for any subset of qudits S, the following are equivalent:*

(i) $\Pi^S_{\text{anti}}|\psi\rangle = |\psi\rangle$.

(ii) The qudits of $|\psi\rangle$ in S are in the $(1,1,...,1)$-isotypic component.

Proof. This follows quickly from some standard results in representation theory. For completeness, we sketch the argument here, although some background is omitted. In particular, the isotypic subspace of type $(1,1,\cdots,1)$ is of form $\mathrm{Span}\{\ U^{\otimes n}P_\pi\Pi^S_{\text{anti}}|w\rangle\ \forall \pi \in S_n, U \in GL_d\}$, where $|w\rangle \in (\mathbb{C}^d)^{\otimes n}$ is some state that makes the subspace non-vanishing. It is then not hard to see that this is exactly the anti-symmetric subspace. □

4.2 Analysis of Algorithm 3

We now use the machinery developed in the previous section to analyze the sample-efficient anti-symmetrization algorithm given in Algorithm 3. The key tool used in this analysis is the following lemma, which we will use to lower bound the probability of a measurement successfully growing the size of the `anti-symmetric` register during step 3 of the algorithm.

Lemma 12. *Let $|\psi\rangle$ be an n-qudit state in the λ-isotypic component and recall that λ'_1 is the length of the first column of λ. Assume there is a subset of qudits $S \subseteq [n]$ with $|S| = k$ and*

$$\Pi^S_{\text{anti}} |\psi\rangle = |\psi\rangle . \tag{48}$$

Then define the operator

$$B^S = \frac{1}{n-k} \left(\sum_{j \in [n] \setminus S} A^{S,j} \right) . \tag{49}$$

We have

$$\langle \psi | B^S |\psi\rangle = 0 \tag{50}$$

whenever $k = \lambda'_1$ and

$$\langle \psi | B^S |\psi\rangle \geq \frac{\lambda'_1 - k}{(n-k)(k+1)} \tag{51}$$

otherwise.

Proof. We first extend the notation of Lemma 9 and, for any subset $R \subseteq [n]$ define

$$T(R) = \sum_{i,j \in R,\ i<j} P_{(i,j)}. \tag{52}$$

Then we rewrite the operator B^S as

$$\begin{aligned} B^S &= \frac{1}{n-k} \left(\sum_{j \in [n] \setminus S} A^{S,j} \right) && (53)\\ &= \frac{1}{(n-k)(k+1)} \left(\sum_{j \in [n] \setminus S} \left(I - \sum_{i \in S} P_{(i,j)} \right) \right) && (54)\\ &= \left(\frac{1}{k+1} \right) I - \frac{1}{(n-k)(k+1)} \sum_{i \in S} \sum_{j \in [n] \setminus S} P_{(i,j)} && (55)\\ &= \left(\frac{1}{k+1} \right) I - \frac{1}{(n-k)(k+1)} \left(T([n]) - T(S) - T([n] \setminus S) \right) . && (56) \end{aligned}$$

Next, by Lemmas 10 and 11 we can write

$$|\psi\rangle = \sum_{\nu \in \mathcal{V}} \alpha_\nu |\psi_\nu\rangle \tag{57}$$

where we have

$$\left(\Pi^S_{anti} \otimes \Pi^{S^c}_\nu\right) |\psi_\nu\rangle = |\psi_\nu\rangle \tag{58}$$

where $\Pi^{S^c}_\nu$ denotes the projector onto the ν isotypical component applied to the qudits in S^c and the sum runs over a set $\mathcal{V}$ of all partitions which can be obtained from λ by deleting exactly one box from k different rows. We also fix $|\psi_\nu\rangle$ to be properly normalized, so $\sum_\nu \alpha_\nu^2 = 1$. Combining Equations (56) and (57) and using Lemma 9 gives

$$\langle\psi| B^S |\psi\rangle = \frac{1}{k+1} - \frac{1}{(n-k)(k+1)} \left(c(\lambda) - c(1,1,...,1) - \sum_{\nu\in\mathcal{V}} \alpha_\nu^2 c(\nu) \right) \tag{59}$$

$$= \frac{1}{k+1} - \frac{1}{(n-k)(k+1)} \left(c(\lambda) + \frac{k(k-1)}{2} - \sum_{\nu\in\mathcal{V}} \alpha_\nu^2 c(\nu) \right) \tag{60}$$

$$= \frac{1}{k+1} - \frac{1}{(n-k)(k+1)} \sum_{\nu\in\mathcal{V}} \alpha_\nu^2 \left(\frac{k(k-1)}{2} + c(\lambda) - c(\nu) \right), \tag{61}$$

where explicitly computed the content of the $(1,1,...,1)$ partition on the second line above, and rearranged terms on the third.

Next, we analyze the expression above when $k = \lambda'_1$. In this case $\mathcal{V}$ contains a single irrep ν obtained from λ by deleting one box from each row of λ. The difference $c(\lambda) - c(\nu)$ is equal to the content of each of the deleted boxes, so we have

$$c(\lambda) - c(\nu) = \sum_{i\in[\lambda'_1]} (\lambda_i - i) = n - \frac{k(k+1)}{2}, \tag{62}$$

and then Equation (61) simplifies to

$$\langle\psi| B^S |\psi\rangle = \frac{1}{k+1} - \frac{1}{(n-k)(k+1)} \left(\frac{k(k-1)}{2} + n - \frac{k(k+1)}{2} \right) \tag{63}$$

$$= \frac{1}{k+1} - \frac{1}{k+1} = 0. \tag{64}$$

Now, if $k < \lambda'_1$, then $\mathcal{V}$ contains all irreps obtained by deleting boxes from k different rows of λ. Let ν be one such irrep, and let $R \subset [\lambda'_1]$ label the rows from which boxes are deleted. Then

$$c(\lambda) - c(\nu) = \sum_{i\in R} (\lambda_i - i) \tag{65}$$

$$\leq \left(\sum_{i\in R} \lambda_i \right) - \frac{k(k+1)}{2} \tag{66}$$

$$\leq (n - (\lambda'_1 - k)) - \frac{k(k+1)}{2}, \tag{67}$$

since each of the $\lambda'_1 - k$ rows not in R contains at least one box. Plugging this in to Equation (61) for each $\nu \in \mathcal{V}$ gives

$$\langle\psi| B^S |\psi\rangle \geq \frac{1}{k+1} - \frac{1}{(n-k)(k+1)} \left(\frac{k(k-1)}{2} + n - \frac{k(k+1)}{2} - (\lambda'_1 - k) \right) \tag{68}$$

$$= \frac{\lambda'_1 - k}{(n-k)(k+1)}, \tag{69}$$

as desired. □

To relate Lemma 12 and Algorithm 3 we will view the input $\rho^{\otimes n}$ to the algorithm as a probabilistic mixture of pure states, and then analyze the evolution of these pure states under Algorithm 3 depending on the λ-isotypic component they are contained in. The last step is a simple corollary of Lemma 12, stated next.

Corollary 13. *Let $|\psi\rangle$ be a state in the λ-isotypic component of $(\mathbb{C}^d)^{\otimes n}$. If Algorithm 3 is applied to $|\psi\rangle$, it returns an anti-symmetric state on $k = \lambda'_1$ qudits with probability at least $1 - \epsilon$.*

Note that, by convexity, if a mixed state ρ is supported on the λ-isotypic component, then the same conclusion as in Corollary 13 holds.

Proof. Let $|\tilde{\psi}\rangle$ describe the state stored in the `anti-symmetric` and `candidate` registers at some arbitrary intermediate stage of the algorithm, and let $S \subseteq [n]$ denote the qudits of $|\tilde{\psi}\rangle$ in the `anti-symmetric` register. Algorithm 1 only acts on $|\psi\rangle$ by representations of the symmetric group algebra, which are block diagonal with respect to the isotypic components, so $|\tilde{\psi}\rangle$ is also in the λ-isotypic component. By construction, step 3(a) of Algorithm 3 returns success with probability

$$\langle\tilde{\psi}| \frac{1}{n-k} \sum_{j \in [n]\setminus S} A^{S,j} |\tilde{\psi}\rangle = \langle\tilde{\psi}| B^S |\tilde{\psi}\rangle . \tag{70}$$

But then Lemma 12 gives that each iteration of step 3 grows the size of the `anti-symmetric` register from k to $k+1$ with probability

$$\frac{\lambda'_1 - k}{(n-k)(k+1)} \geq \frac{\lambda'_1 - k}{n\lambda'_1}. \tag{71}$$

Relating this to the probability of a coupon collector with λ'_1 coupons and a probability $(n\lambda'_1)^{-1}$ of picking up any coupon (as in the proof of Theorem 5) shows that after t repetitions, the probability that the `anti-symmetric` register does not contain λ'_1 qudits is upper bounded by

$$\lambda'_1 \left(1 - \frac{1}{n\lambda'_1}\right)^t \leq \lambda'_1 e^{-t/n\lambda'_1}. \tag{72}$$

Setting $t = \lceil nd(\log(d) + \log(1/\epsilon))\rceil$ and noting $d \geq \lambda'_1$ proves the result. □

We now have all the tools required to complete the proof of Theorem 7 and the analysis of Algorithm 3. We present this next.

Proof (Theorem 7, continued). First note that Algorithm 3 acts on the input state only through representations of the symmetric group algebra and so never mixes isotypic components. Corollary 13 and a union bound then give: when Algorithm 3 is applied to any pure state that has overlap $1 - \delta$ with λ-isotypic components satisfying $\lambda'_1 = r$, it returns an anti-symmetric state on r qudits with probability at least $1 - \delta - \epsilon$. This observation extends immediately to mixed states. But now standard arguments (for example, Remark 2.25 of [OW21] plus a geometric sum argument and concentration inequality) give that if a mixed state ρ has rank r and minimum non-zero eigenvalue $\lambda_{\min}$, sampling from the Schur-Weyl distribution of $\rho^{\otimes n}$ yields a partition λ for which $\lambda'_1 = r$ with probability $1 - \exp(-\Omega(n\lambda_{\min}))$ provided $n \geq Cr/\lambda_{\min}$ for a sufficiently large constant $C > 0$. The result follows. □

4.3 Algorithms for Schur sampling and purity amplification

We begin with some observations about Algorithm 3. For any partition $\lambda \vdash n$, Corollary 13 above shows that if this algorithm is run on a pure state $|\psi\rangle$ in the λ-isotypic component of $\left(\mathbb{C}^d\right)^{\otimes n}$, it returns an anti-symmetric state on λ'_1 qudits with probability $1-\epsilon$. In this case, Pieri (Lemma 10) then tells us the system in the `candidates` register must be in the ν-isotypic component, where $\nu \vdash (n-\lambda'_1)$ is the unique partition obtained by deleting one box from each row of λ. If we then run Algorithm 3 again on the `candidates` register, it will output an anti-symmetric state on $\nu'_1 = \lambda'_2$ qudits. Continuing in this way we can learn the lengths $\lambda'_1, \lambda'_2, ...\lambda'_{\lambda_1}$ of each column in λ.

If this same algorithm is run on a state $|\psi\rangle$ that is in a superposition of different isotypic components, the algorithm will output some $\lambda'_1, ..., \lambda'_{\lambda_1}$ with probability corresponding to the overlap of $|\psi\rangle$ with the corresponding λ-isotypic component (and some error depending on ϵ). This is exactly the task of *weak Schur sampling*. We present this algorithm next.

Algorithm 4. Weak Schur Sampling

Input: An unknown state $\rho \in \mathcal{L}\left(\left(\mathbb{C}^d\right)^{\otimes n}\right)$ and an accuracy parameter δ.
Output: A partition λ drawn from a probability distribution p_λ satisfying

$$\sum_{\lambda \vdash n} |p_\lambda - \mathrm{Tr}[\Pi_\lambda \rho]| \leq \delta\,. \tag{73}$$

1. Label a register `unsampled` and add ρ to it. Initialize a parameter $j = 1$.
2. While `unsampled` is not empty:
   (a) Take the state from the `unsampled` register and apply Algorithm 3 to it with parameter $\epsilon = \delta/2n$. Add the state in the `candidates` register of this algorithm to the `unsampled` register.
   (b) Set λ'_j equal to the length of the anti-symmetric state returned by Algorithm 3. Increment j.
3. Return the partition λ with column lengths $\lambda'_1, \lambda'_2, ..., \lambda'_{j-1}$.

Theorem 14. *Algorithm 4 is correct, that is, it returns a partition λ drawn from a probability distribution of the promised form. It requires at most $2n\lceil nd\log(2nd/\delta)\rceil$ controlled-SWAP operations and has T-count $O\left(n^2 d\log(d)\log^2(nd/\delta)\right)$.*

Proof. First consider the case where the algorithm is run on a pure state $\rho = |\psi\rangle\langle\psi|$ in the λ-isotypic component of $\left(\mathbb{C}^d\right)^{\otimes n}$ for some arbitrary $\lambda \vdash n$. Let ρ_i be the state stored in the `unsampled` register after step 2 of the algorithm has repeated i times (so $\rho_0 = \rho$). For every positive integer $i \leq \lambda_1$ let $\nu(i)$ be the partition obtained by deleting the i leftmost columns of λ. Assume for induction that ρ_i is in the $\nu(i)$-isotypic component. Then Pieri (Lemma 10) gives that ρ_{i+1} is in the $\nu(i+1)$-isotypic component provided the $(i+1)$-th repetition of step 2 of Algorithm 4 returns an anti-symmetric state on $\nu(i)'_1$ qudits. Corollary 13 gives that this happens with probability $1-\epsilon = 1-\delta/2n$. Then

a union bound argument gives that Algorithm 4 returns the correct partition λ with probability at least $1 - \lambda_1(\delta/2n) \geq 1 - \delta/2$.

Now when the input is an arbitrary mixed state ρ, as Algorithm 4 does not mix isotypic components, the analysis above gives that with probability at least $1 - \delta/2$ the algorithm returns a partition λ drawn exactly from the distribution

$$q_\lambda = \mathrm{Tr}[\Pi_\lambda \rho]. \tag{74}$$

Letting r_λ be the distribution output by the algorithm a $\delta/2$-fraction of the time which includes any time the algorithm does not output from the q_λ distribution, we have

$$p_\lambda = (1 - \delta/2)q_\lambda + (\delta/2)r_\lambda\,, \tag{75}$$

and so we see that, for any r_λ, we have

$$\|p_\lambda - q_\lambda\|_1 \leq \delta, \tag{76}$$

as desired.

To bound the number of controlled-SWAP operations we note each application of Algorithm 3 to build an antisymmetric state on λ_i' qudits requires at most $\lceil nd \log(d/\epsilon)\rceil = \lceil nd \log(2nd/\delta)\rceil$ applications of Algorithm 1, each of which requires at most $2\lambda_i'$ controlled-SWAP gates. So the total number of controlled-SWAP operations is upper bounded by

$$\lceil nd \log(2nd/\delta)\rceil \sum_{i=1}^{\lambda_1} 2\lambda_i' = 2n\lceil nd \log(2nd/\delta)\rceil \tag{77}$$

To convert to a T-count note each controlled-SWAP operation requires $O(\log(d))$ T gates. The each application of Algorithm 1 also requires $\mathcal{O}(\log(k)) \leq \mathcal{O}(\log(d))$ single qubit rotations, each of which which can be implemented up to error η using $\mathcal{O}(\log(\eta^{-1}))$ T gates. Setting

$$\eta^{-1} = \lceil nd \log(2nd/\delta)\rceil/\delta \tag{78}$$

to keep total error bounded gives the result.

□

In Algorithm 4 all quantum registers are discarded and the only output is a classical description of the partition λ. If the algorithm is altered so that quantum information is retained and included in the output, this modified algorithm performs the task of *Unitary Schur Sampling*. The name refers to the fact that the algorithm destroys any superposition the input state has in each irreducible module of S_n (by projecting onto a fixed vector within that module) but leaves superpositions within the irreducible modules of GL_d untouched. Informally: the algorithm destroys permutation information about the input state while preserving unitary information.

Algorithm 5. Unitary Schur Sampling

Input: A state $\rho^{\otimes n}$ for unknown $\rho \in \mathcal{L}\left(\mathbb{C}^d\right)^{\otimes n}$ and an accuracy parameter δ.
Output: A state σ satisfying

$$\|\sigma - \sum_{\lambda \vdash n} \rho^{[\lambda]} \otimes |\lambda\rangle\langle\lambda|\|_1 \leq \delta \tag{79}$$

where $|\lambda\rangle$ is a classical description of the partition λ and $\rho^{[\lambda]}$ is the unnormalized mixed state defined (in terms of the V_λ and W_λ modules, see Equation (39)) by

$$U_{\text{schur}} \rho^{[\lambda]} U_{\text{schur}}^\dagger = |0_\lambda\rangle\langle 0_\lambda| \otimes \text{Tr}_{V_\lambda}[U_{\text{schur}} \Pi_\lambda \rho^{\otimes n} \Pi_\lambda U_{\text{schur}}^\dagger], \tag{80}$$

where $|0_\lambda\rangle$ is some reference state [3] in the V_λ module fixed by the algorithm.

1. Label registers `unsampled` and `sampled`. Add the state $\rho^{\otimes n}$ to the `unsampled` register and initialize a parameter $j = 1$.
2. While `unsampled` is not empty:
   (a) Take the state in the `unsampled` register and apply Algorithm 3 to it with parameter $\epsilon = \delta/2n$. Add the state in the `antisymmetric` register of this algorithm to the `sampled` register and the state in the `candidates` register of this algorithm to the `unsampled` register.
   (b) Set λ'_j equal to the length of the anti-symmetric state returned by Algorithm 3. Increment j.
3. Return the state in the `sampled` register along with $|\lambda\rangle$ where λ is the partition with column lengths $\lambda'_1, \lambda'_2, ..., \lambda'_{j-1}$.

Theorem 15. *Algorithm 5 is correct, that is, it returns a state σ with the promised form. It requires at most $2n\lceil nd \log(2nd/\delta)\rceil$ controlled-SWAP operations and has T-count $O\left(n^2 d \log(d) \log^2(nd/\delta)\right)$.*

Proof. Because our input is block diagonal over isotypical components, i.e.

$$\sum_\lambda \Pi_\lambda \rho^{\otimes n} \Pi_\lambda = \rho^{\otimes n} \tag{81}$$

it suffices to analyze the algorithm over each isotypical component. For one such λ, let G_λ be the event that each application of Algorithm 3 works as intended, i.e. that the i-th call to the algorithm returns an antisymmetric state on λ'_i qudits for all $i \in [\lambda_1]$. By the same analysis as in Theorem 14 we have

$$\Pr[G_\lambda] \geq 1 - \delta/2 \tag{82}$$

for all λ.

Let σ_λ be the (normalized) state in the `sampled` register conditioned on event G_λ when the algorithm returns partition label λ. The same arguments as in Theorem 14 shows that σ_λ is in the

[3] The exact value of $|0_\lambda\rangle$ depends also on the choice of the basis of U_{schur}, which is not important for our application.

λ-isotypic component with probability at least $1 - \delta/2$, that is,

$$\mathrm{Tr}[\Pi_\lambda \sigma_\lambda] \geq 1 - \delta/2. \tag{83}$$

Correctness of Algorithm 3 also gives that

$$\mathrm{Tr}\left[\left(\Pi_{\mathrm{anti}}^{\lambda'_1} \otimes \Pi_{\mathrm{anti}}^{\lambda'_2} \otimes ... \otimes \Pi_{\mathrm{anti}}^{\lambda'_{\lambda_1}}\right) \sigma_\lambda\right] = 1. \tag{84}$$

For any $S \subseteq [n]$ let

$$A(S) = \frac{1}{|S|!} \sum_{\substack{\sigma \in S_n \\ \sigma(i)=i \, \forall \, i \notin S}} \mathrm{sign}(\sigma)\sigma \tag{85}$$

be the element of the symmetric group algebra corresponding to an antisymmetrizer on the elements of S and let

$$A(\lambda'_1, \lambda'_2, ..., \lambda'_{\lambda_1}) = \prod_{i=1}^{\lambda_1} A\left(\left\{\sum_{j=1}^{i-1} \lambda'_j + 1, \sum_{j=1}^{i-1} \lambda'_j + 2, ..., \sum_{j=1}^{i} \lambda'_j\right\}\right) \tag{86}$$

so $\Pi_{\mathrm{anti}}^{\lambda'_1} \otimes \Pi_{\mathrm{anti}}^{\lambda'_2} \otimes ... \otimes \Pi_{\mathrm{anti}}^{\lambda'_{\lambda_1}}$ is exactly the natural representation of $A(\lambda'_1, \lambda'_2, ..., \lambda'_{\lambda_1})$. Standard representation theory techniques (see Lemma 21 in Appendix A for a proof) give that there is a unique state $|0_\lambda\rangle \in V_\lambda$ satisfying

$$D_\lambda\left(A(\lambda'_1, \lambda'_2, ..., \lambda'_{\lambda_1})\right)|0_\lambda\rangle = |0_\lambda\rangle \, . \tag{87}$$

For example, this state is exactly the column ordered Young-Yamanouchi vector of [Li+25]. This shows that σ_λ must overlap the V_λ module on exactly the state $|0_\lambda\rangle\langle 0_\lambda|$, that is, we must have

$$\Pi_\lambda \sigma_\lambda \Pi_\lambda = |0_\lambda\rangle\langle 0_\lambda| \otimes \sigma_\lambda^W \tag{88}$$

for some state $\sigma_\lambda^W \in \mathcal{L}(W_\lambda)$. Algorithm 3 only ever acts on the state ρ with representations of the symmetric group algebra, which act as the identity on the W modules, so we also have

$$\sigma_\lambda^W \propto \sum_r \mathrm{Tr}_{V_\lambda}[(K_r \otimes I) U_{\mathrm{schur}} \Pi_\lambda \rho^{\otimes n} \Pi_\lambda U_{\mathrm{schur}}^\dagger (K_r \otimes I)^\dagger] \tag{89}$$

$$= \mathrm{Tr}_{V_\lambda}[\Pi_\lambda \rho^{\otimes n} \Pi_\lambda]. \tag{90}$$

where $\{K_r\}$ are Kraus operators describing the action of the algorithm on the V_λ module. We conclude

$$\sigma = \left(1 - \frac{\delta}{2}\right)\left(\sum_{\lambda \vdash n} |0_\lambda\rangle\langle 0_\lambda| \otimes \mathrm{Tr}_{V_\lambda}[\Pi_\lambda \rho^{\otimes n} \Pi_\lambda] \otimes |\lambda\rangle\langle\lambda|\right) + \frac{\delta}{2}(\sigma') \tag{91}$$

$$= \left(1 - \frac{\delta}{2}\right)\left(\sum_{\lambda \vdash n} \rho^{[\lambda]} \otimes |\lambda\rangle\langle\lambda|\right) + \frac{\delta}{2}(\sigma') \tag{92}$$

for some (normalized) state σ' and correctness of the algorithm follows.

The required number of controlled-SWAPs and T-count of the algorithm follows immediately from the analysis of Algorithm 4. □

We end this section with a discussion of *quantum purity amplification*. Given an input $\rho^{\otimes n} \in \left(\mathcal{L}(\mathbb{C}^d)^{\otimes n}\right)$, the goal of purity amplification (in its simplest form) is to output a state $\rho' \in \mathcal{L}(\mathbb{C}^d)$ that is concentrated on the top eigenvector of ρ, that is, to return a state maximizing

$$\langle \varphi_1 | \rho' | \varphi_1 \rangle \,, \tag{93}$$

where $|\varphi_1\rangle$ is the largest eigenvector of ρ. This task was formalized in [Li+25], which also gave an optimal protocol for it based on measuring the information in the V_λ modules of the state $\rho^{\otimes n}$ and then applying a measurement-dependent correction to the resulting state. The follow-up work [Bra+25] gave an alternate protocol for optimal quantum purity amplification of qubit input states based on random SWAP tests. We show that a simple modification of Algorithm 5 extends the ideas of [Bra+25] to give an optimal protocol for quantum purity amplification of arbitrary qudit input states. Before presenting the formal algorithm, we give an intuitive explanation of why preparing antisymmetric states can be a helpful subroutine for purity amplification.

Consider Algorithm 5 acting on an input state $\rho^{\otimes n}$. Label the eigenvectors of ρ in the order of non-increasing eigenvalue by $|\varphi_1\rangle, |\varphi_2\rangle, ..., |\varphi_r\rangle$. Any k-qudit antisymmetric state produced by the algorithm must overlap equally with k different eigenvectors of ρ. Preparing such a state and removing it from the system then has the effect of "pumping out" asymmetry from the system, leaving the remaining system more concentrated on the most common eigenvector $|\varphi_1\rangle$. Following this intuition suggests that the final antisymmetric state prepared by Algorithm 5 will be the one most concentrated on the top eigenvector of ρ, and returning a random qudit from this state (or the state itself, in the case that it is a one-qudit state already) will provide a solution to the task of purity amplification. We now formalize this process, and show (by reducing to the analysis of [Li+25]) that this intuition is correct.

Algorithm 6. Quantum Purity Amplification

Input: An state $\rho^{\otimes n}$ for unknown $\rho \in \mathcal{L}\left(\mathbb{C}^d\right)$ and an accuracy parameter δ.
Output: A state ρ' satisfying

$$\|\rho' - \rho_{\text{opt}}\|_1 \leq \delta \,, \tag{94}$$

where ρ_{opt} is the state returned by the optimal quantum purity amplification protocol of [Li+25].

1. Label registers `unsampled` and `purified`. Add the state $\rho^{\otimes n}$ to the `unsampled` register.
2. While `unsampled` is not empty:
   (a) Remove the state in the `unsampled` register and apply Algorithm 3 to it with parameter $\epsilon = \delta/2n$. **Replace** the state in the `purified` register with the state in the `antisymmetric` register of this algorithm, and add the state in the `candidates` register of this algorithm to the `unsampled` register.
3. Return the last qudit from the `purified` register.

Theorem 16. *Algorithm 6 is correct, that is, it returns a state ρ' with the desired properties. It requires at most $2n\lceil nd \log(2nd/\delta)\rceil$ controlled-SWAP operations and has T-count $O\left(n^2 d \log(d) \log^2(nd/\delta)\right)$.*

Proof. We briefly adopt the notation of [Li+25] in order to compare. The operational description of their protocol shows that it is equivalent to the following three steps: (i) measure the Young Diagram (λ) of the input along with the Young-Yamanouchi label (s) of the permutation register; (ii) apply a transition operator $\Pi^{[\lambda]}_{\lambda^\diamond,s}$ to map this state to the column ordered subspace ($\lambda^\diamond$); (iii) return the register corresponding to the bottom box of the rightmost column of $\lambda^\diamond$, tracing out the rest. In our notation, the state prepared after steps (i) and (ii) of this protocol is exactly

$$\sum_{\lambda\vdash n}\mathrm{Tr}[\Pi_\lambda\rho^{\otimes n}]\frac{\left(\Pi^{\lambda'_1}_{\text{anti}}\otimes\Pi^{\lambda'_2}_{\text{anti}}\otimes\ldots\otimes\Pi^{\lambda'_{\lambda_1}}_{\text{anti}}\right)\Pi_\lambda\rho^{\otimes n}\Pi_\lambda\left(\Pi^{\lambda'_1}_{\text{anti}}\otimes\Pi^{\lambda'_2}_{\text{anti}}\otimes\ldots\otimes\Pi^{\lambda'_{\lambda_1}}_{\text{anti}}\right)}{\mathrm{Tr}\left[\left(\Pi^{\lambda'_1}_{\text{anti}}\otimes\Pi^{\lambda'_2}_{\text{anti}}\otimes\ldots\otimes\Pi^{\lambda'_{\lambda_1}}_{\text{anti}}\right)\Pi_\lambda\rho^{\otimes n}\Pi_\lambda\left(\Pi^{\lambda'_1}_{\text{anti}}\otimes\Pi^{\lambda'_2}_{\text{anti}}\otimes\ldots\otimes\Pi^{\lambda'_{\lambda_1}}_{\text{anti}}\right)\right]}\otimes|\lambda\rangle\langle\lambda|\ . \tag{95}$$

(See Appendix A, and in particular Lemma 22 for a justification of this fact.) But the analysis of Algorithm 5 shows that it returns a state at most δ far in trace distance from the state above. Algorithm 6 returns the last qudit of the state prepared by Algorithm 5, and since trace distance is non-increasing under partial traces, the result follows.

The number of controlled-SWAPs and T-count is immediate from the analysis of Algorithm 4. □

5 Mixed versus pure one-way state generators via anti-symmetrization

This section presents an illustrative example application of anti-symmetrization to quantum cryptography. To preface our application, we begin by describing one-way state generators. One-way state generators were created to be a quantum analog to one-way functions, and roughly are states that can be efficiently prepared from a key k, can be efficiently verified, given copies of the state and key, and for whom the key k cannot be efficiently learned.

In their original definition [MY24], one-way state generators were assumed to have a *computationally* efficient verification algorithm, and indeed if the output of a one-way state generator is pure, it always has a canonical, computationally efficient verification (namely, prepare the state $|\psi_k\rangle$ and perform a SWAP test). However, when the rank of the one-way state is allowed to be greater than one, the resulting states might no longer admit computationally efficient verification (although they will always admit an inefficient verification). This led to the definition of *inefficiently verifiable* one-way state generators [BJ25], which were shown to be equivalent to EFI pairs [BCQ22]. In a series of works [BCN25; Beh+24; GZ25], it was shown that these two notions (efficiently and inefficiently verifiable) of one-way state generators can be separated relative to a state and unitary oracle. However, it is an open question whether efficiently verifiable mixed-state one-way state generators are equivalent to pure-state one-way state generators.

Using the anti-symmetrization algorithm, we can show: when the rank of a one-way state generator is promised to be polynomial in the security parameter λ, there is a reduction from mixed one-way state generators to pure state output one-way state generators. Our result requires a slight modification to the definition of one-way state generators, namely adding the assumption that there is a non-negligible lower bound on the smallest non-zero eigenvalue . We formally define gapped one-way state generators below.

Definition 17 (Gapped rank-r one-way state generators)**.** A one-way state generator, described by algorithms $(\mathsf{KeyGen}, \mathsf{StateGen}, \mathsf{Ver})$, is an *$\epsilon$-gapped rank-$r$ one-way state generator* if for all k, $\rho_k \leftarrow \mathsf{StateGen}(k)$ is rank-r and the smallest non-zero eigenvalue of ρ_k is at least ϵ.

We prove the following corollary of our anti-symmetrization result.

Corollary 18. *Pure-state output one-way state generators exist if and only if there exist polynomials r, t such that $1/t$-gapped rank-r one-way state generators exist.*

Proof. In one direction, pure-state output one-way state generators are already 1-gapped rank-1 one-way state generators, so their existence immediately implies the existence of mixed state output one-way state generators.

In the other direction, say that $1/t$-gapped rank-r one-way state generators exist, and for key k, say that the one-way state generator outputs a state ρ_k that has spectral decomposition

$$\rho_k = \sum_i \lambda_i \left|\psi_{k,i}\right\rangle\!\left\langle\psi_{k,i}\right| .$$

Then we claim that the pure-state output one-way state generator that maps the key k to the state

$$\left|\psi_k\right\rangle \propto \sum_{\sigma \in S_r} \mathrm{sign}(\sigma) P_\sigma \left(\left|\psi_{k,1}\right\rangle \otimes \ldots \otimes \left|\psi_{k,r}\right\rangle\right)$$

exists. First, we note that our anti-symmetrization procedure in (Algorithm 2) prepares, except with negligible failure probability, the pure state $\left|\psi_k\right\rangle$ from polynomially many copies of ρ_k in $\mathrm{poly}(r, t, \log(d))$ time, since ρ_k is assumed to have rank r and its minimum eigenvalue is assumed to be at least $1/t$. It is also clear that an algorithm that could break the one-wayness of this one-way state generator can break the one-wayness of the original one-way state generator by first applying the anti-symmetrization algorithm (on polynomially many copies of ρ_k) and running the algorithm for breaking the one-wayness of the pure-state one-way state generator. □

Finally, we note that this reduction is "almost-tight" among black-box reductions between mixed-state and pure-state one-way state generators, in the sense that when r becomes sufficiently super-polynomial (growing faster than $\lambda^{\log(\lambda)}$), there is a quantum state separation between rank r one-way state generators and efficiently verifiable one-way state generators, using the work of [BCN25; Beh+24] and [BJ25], which can be lifted to a unitary oracle separation by using [GZ25].

Claim 19. *For every function $r(\lambda) \leq 2^\lambda$ that satisfies $\log(r) = \omega(\log^2(\lambda))$, there exists a quantum state oracle (or unitary oracle) relative to which $(1/r)$-gapped rank-r one-way state generators exist, but efficiently verifiable one-way state generators (and thus in particular pure-state output one-way state generators) do not exist.*

Proof. Fix a local dimension $d(\lambda) = 2^{\sqrt{\log(r(\lambda))}}$. For r as in the claim, this is super-polynomial. The oracle is (for every λ) to take a Haar random state in d dimensions, as well as a $\mathsf{UnitaryPSPACE}$ oracle. From [BCN25], efficiently verifiable one-way state generators do not exist relative to this oracle. Additionally, from the work of [AGL24; CCS25], there exists a construction of 1PRS relative to this oracle that have dimension d, and such that no adversary receiving $t(\lambda)$ copies of the state can distinguish between the construction and a maximally mixed state on d-dimensions with probability greater than $O(t^2/d) = \mathrm{negl}(\lambda)$. Recall that the construction of one-way state

generators from [BJ25] is to fix a negligible $\eta = \eta(\lambda)$ (which determines the soundness of the construction) and map a key of $\log(1/\eta)$ bits to the same number of copies of either the 1PRS or the maximally mixed state on d dimensions. Taking $\eta = 1/d$ (which is negligible), we have that the total dimension of the one-way state generator is $d^{\log(d)} = r$, and an adversary can only invert the one-way state generator with negligible probability in λ.

To prove a bound on the minimum non-zero eigenvalue of this one-way state generator, we have to inspect the construction of [AGL24]. The construction of EFI pairs implied by [AGL24] involves either outputting a maximally mixed state, or outputting $\mathbb{E}_k[Z^k |\psi\rangle\langle\psi| Z^k]$, where the Z operators act on the first m qubits of the system (where m can be chosen to be anything $\omega(\log(\lambda))$). Writing the Haar random state as $|\psi\rangle = \sum_x |x\rangle |\psi_x\rangle$, this state is exactly $\sum_x |x\rangle\langle x| \otimes |\psi_x\rangle\langle\psi_x|$. This matrix, then, is block-diagonal, and the minimum eigenvalue is exactly the minimum over strings $x \in \{0,1\}^m$ of $\mathrm{Tr}[(|x\rangle\langle x| \otimes \mathrm{id}) |\psi\rangle\langle\psi|]$. For every fixed x, Levy's lemma [Mec19] tells us that the probability that this is less than 2^{-m-1} is at most $\exp(-2^{n-m}/24)$. Taking a union bound over all $x \in \{0,1\}^m$, when m is, say $\log(d)/2$ (which satisfies the required condition on m), tells us that with overwhelming probability the minimum eigenvalue of this state is greater than $1/d$. On the other hand, the maximally mixed state over d-dimensions has a minimum non-zero eigenvalue of $1/d$.

Thus the minimum eigenvalue of both states in the EFI pair is greater than $1/d$ with overwhelming probability. Taking the tensor product of $\log(d)$ of them yields a state whose rank is at most $d^{\log(d)} = r$, and whose minimum eigenvalue is at least $1/r$, as desired. □

We note that it remains an open question whether one can get pure-state one-way state generators from very high rank mixed-state one-way state generators, using their efficient verification algorithm, or if the two notions might actually be separate. We point out that finding an oracle relative to which one-way puzzles do not exist, but efficiently verifiable one-way state generators exist (extending the result of [Man26]) would separate pure-state one-way state generators from efficiently verifiable mixed-state one-way state generators.

References

[AG26] Prabhanjan Ananth and Eli Goldin. *Less is More: On Copy Complexity in Quantum Cryptography*. 2026. arXiv: 2510.04992 [quant-ph]. URL: https://arxiv.org/abs/2510.04992 (page 2).

[AGL24] Prabhanjan Ananth, Aditya Gulati, and Yao-Ting Lin. *Cryptography in the Common Haar State Model: Feasibility Results and Separations*. Cryptology ePrint Archive, Paper 2024/1043. 2024. URL: https://eprint.iacr.org/2024/1043 (pages 27, 28).

[BCH06] Dave Bacon, Isaac L. Chuang, and Aram W. Harrow. "Efficient Quantum Circuits for Schur and Clebsch-Gordan Transforms". In: *Physical Review Letters* 97.17 (Oct. 2006). ISSN: 1079-7114. DOI: 10.1103/physrevlett.97.170502. URL: http://dx.doi.org/10.1103/PhysRevLett.97.170502 (page 4).

[BCH07] Dave Bacon, Isaac L. Chuang, and Aram W. Harrow. "The quantum Schur and Clebsch-Gordan transforms: I. efficient qudit circuits". In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '07. New Orleans, Louisiana: Society for Industrial and Applied Mathematics, 2007, pp. 1235–1244. ISBN: 9780898716245. URL: https://dl.acm.org/doi/10.5555/1283383.1283516 (page 4).

[BJ25] Rishabh Batra and Rahul Jain. *Commitments are equivalent to statistically-verifiable one-way state generators*. 2025. arXiv: 2404.03220 [quant-ph]. URL: https://arxiv.org/abs/2404.03220 (pages 1, 26–28).

[Beh+24] Amit Behera, Giulio Malavolta, Tomoyuki Morimae, Tamer Mour, and Takashi Yamakawa. *A New World in the Depths of Microcrypt: Separating OWSGs and Quantum Money from QEFID*. Cryptology ePrint Archive, Paper 2024/1567. 2024. URL: https://eprint.iacr.org/2024/1567 (pages 1, 26, 27).

[BCN25] John Bostanci, Boyang Chen, and Barak Nehoran. "Oracle separation between quantum commitments and quantum one-wayness". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2025, pp. 3–22. DOI: 10.1007/978-3-031-91098-2_1 (pages 1, 5, 26, 27).

[Bra+25] Shrigyan Brahmachari, Austin Hulse, Henry D. Pfister, and Iman Marvian. *Optimal Qubit Purification and Unitary Schur Sampling via Random SWAP Tests*. 2025. arXiv: 2508.05046 [quant-ph]. URL: https://arxiv.org/abs/2508.05046 (pages 4, 6, 15, 25).

[Bra+26] Shrigyan Brahmachari, David Jakab, Henry D. Pfister, and Iman Marvian. *Unitary Schur Sampling of Qudits via Random SWAP Tests: Hunt for Antisymmetry*. 2026. arXiv: 2610.02103 [quant-ph]. URL: https://arxiv.org/abs/2610.02103 (page 6).

[BCQ22] Zvika Brakerski, Ran Canetti, and Luowen Qian. *On the computational hardness needed for quantum cryptography*. 2022. arXiv: 2209.04101 [quant-ph]. URL: https://arxiv.org/abs/2209.04101 (page 26).

[BM99] Dagmar Bruß and Chiara Macchiavello. "Optimal state estimation for d-dimensional quantum systems". In: *Physics Letters A* 253.5–6 (Mar. 1999), pp. 249–251. ISSN: 0375-9601. DOI: 10.1016/s0375-9601(99)00099-7. URL: http://dx.doi.org/10.1016/S0375-9601(99)00099-7 (page 1).

[Bur+25] Adam Burchardt, Jiani Fei, Dmitry Grinko, Martin Larocca, Maris Ozols, Sydney Timmerman, and Vladyslav Visnevskyi. *High-dimensional quantum Schur transforms*. 2025. arXiv: 2509.22640 [quant-ph]. URL: https://arxiv.org/abs/2509.22640 (page 4).

[CF23] Eric Cancès and Gero Friesecke. *Density functional theory*. Springer, 2023. DOI: 10.1007/978-3-031-22340-2 (page 5).

[CM23] Enrique Cervero and Laura Mančinska. *Weak Schur sampling with logarithmic quantum memory*. 2023. arXiv: 2309.11947 [quant-ph]. URL: https://arxiv.org/abs/2309.11947 (page 4).

[CMT24] Enrique Cervero-Martín, Laura Mančinska, and Elias Theil. *A memory and gate efficient algorithm for unitary mixed Schur sampling*. 2024. arXiv: 2410.15793 [quant-ph]. URL: https://arxiv.org/abs/2410.15793 (page 4).

[Cha24] Garnet Kin-Lic Chan. *Quantum chemistry, classical heuristics, and quantum advantage*. 2024. arXiv: 2407.11235 [quant-ph]. URL: https://arxiv.org/abs/2407.11235 (page 5).

[CCS25] Boyang Chen, Andrea Coladangelo, and Or Sattath. *The power of a single Haar random state: constructing and separating quantum pseudorandomness*. 2025. arXiv: 2404.03295 [quant-ph]. URL: https://arxiv.org/abs/2404.03295 (page 27).

[Che+26a] Kean Chen, Filippo Girardi, Aadil Oufkir, Nengkun Yu, and Zhicheng Zhang. *Quantum channel tomography: optimal bounds and a Heisenberg-to-classical phase transition*. 2026. arXiv: 2604.17369 [quant-ph]. URL: https://arxiv.org/abs/2604.17369 (page 5).

[Che+26b] Senrui Chen, Marco Fanizza, Filippo Girardi, Ludovico Lami, Francesco Anna Mele, Michael Walter, and Freek Witteveen. *Optimal tomography of bosonic and fermionic Gaussian states*. 2026. arXiv: 2607.11847 [quant-ph]. URL: https://arxiv.org/abs/2607.11847 (pages 2, 5).

[Chi+25] Andrew M. Childs, Honghao Fu, Debbie Leung, Zhi Li, Maris Ozols, and Vedang Vyas. "Streaming quantum state purification". In: *Quantum* 9 (Jan. 2025), p. 1603. ISSN: 2521-327X. DOI: 10.22331/q-2025-01-21-1603. URL: https://doi.org/10.22331/q-2025-01-21-1603 (page 4).

[Fom+24] Stepan Fomichev, Kasra Hejazi, Modjtaba Shokrian Zini, Matthew Kiser, Joana Fraxanet, Pablo Antonio Moreno Casares, Alain Delgado, Joonsuk Huh, Arne-Christian Voigt, Jonathan E. Mueller, and Juan Miguel Arrazola. "Initial State Preparation for Quantum Chemistry on Quantum Computers". In: *PRX Quantum* 5.4 (Dec. 2024). ISSN: 2691-3399. DOI: 10.1103/prxquantum.5.040339. URL: http://dx.doi.org/10.1103/PRXQuantum.5.040339 (page 5).

[Gir+26] Filippo Girardi, Francesco Anna Mele, Haimeng Zhao, Marco Fanizza, and Ludovico Lami. *Random Stinespring superchannel: converting channel queries into dilation isometry queries*. 2026. arXiv: 2512.20599 [quant-ph]. URL: https://arxiv.org/abs/2512.20599 (page 5).

[GZ25] Eli Goldin and Mark Zhandry. *Translating Between the Common Haar Random State Model and the Unitary Model*. Cryptology ePrint Archive, Paper 2025/489. 2025. URL: https://eprint.iacr.org/2025/489 (pages 26, 27).

[GL00] Daniel Gottesman and Hoi-Kwong Lo. "From quantum cheating to quantum security". In: *Physics Today* 53.11 (2000), pp. 22–27. DOI: 10.1063/1.1333282 (page 1).

[Gri+25] Daniel Grier, Debbie Leung, Zhi Li, Hakop Pashayan, and Luke Schaeffer. *Streaming quantum state purification for general mixed states*. 2025. arXiv: 2503.22644 [quant-ph]. URL: https://arxiv.org/abs/2503.22644 (page 4).

[GMW26] Daniel Grier, Jackson Morris, and Kewen Wu. QAC^0 *Contains* TC^0 *(with Many Copies of the Input)*. 2026. arXiv: 2601.03243 [cs.CC]. URL: https://arxiv.org/abs/2601.03243 (page 10).

[GR20] Darij Grinberg and Victor Reiner. *Hopf Algebras in Combinatorics*. 2020. arXiv: 1409.8356 [math.CO]. URL: https://arxiv.org/abs/1409.8356 (page 17).

[Haa+17] Jeongwan Haah, Aram W. Harrow, Zhengfeng Ji, Xiaodi Wu, and Nengkun Yu. "Sample-optimal tomography of quantum states". In: *IEEE Transactions on Information Theory* (2017), pp. 1–1. ISSN: 1557-9654. DOI: 10.1109/tit.2017.2719044. URL: http://dx.doi.org/10.1109/TIT.2017.2719044 (page 1).

[Har05] Aram W. Harrow. *Applications of coherent classical communication and the Schur transform to quantum information theory*. 2005. arXiv: quant-ph/0512255 [quant-ph]. URL: https://arxiv.org/abs/quant-ph/0512255 (page 4).

[Har13] Aram W. Harrow. *The Church of the Symmetric Subspace*. 2013. arXiv: 1308.6595 [quant-ph]. URL: https://arxiv.org/abs/1308.6595 (page 2).

[HS05] Peter Hoyer and Robert Spalek. "Quantum Fan-out is Powerful". In: *Theory of Computing* 1.1 (2005), pp. 81–103. ISSN: 1557-2862. DOI: 10.4086/toc.2005.v001a005. URL: http://dx.doi.org/10.4086/toc.2005.v001a005 (page 11).

[Jam06] Gordon Douglas James. *The representation theory of the symmetric groups*. Springer, 2006. DOI: 10.1007/BFb0067708 (page 35).

[Juc74] A-AA Jucys. "Symmetric polynomials and the center of the symmetric group ring". In: *Reports on Mathematical Physics* 5.1 (1974), pp. 107–112. DOI: 10.1016/0034-4877(74)90019-6 (page 16).

[KS18] William M. Kirby and Frederick W. Strauch. "A practical quantum algorithm for the schur transform". In: *Quantum Info. Comput.* 18.9–10 (Aug. 2018), pp. 721–742. ISSN: 1533-7146. DOI: 10.26421/QIC18.9-10-1 (page 4).

[Kro19] Hari Krovi. "An efficient high dimensional quantum Schur transform". In: *Quantum* 3 (Feb. 2019), p. 122. ISSN: 2521-327X. DOI: 10.22331/q-2019-02-14-122. URL: https://doi.org/10.22331/q-2019-02-14-122 (page 4).

[Li+25] Zhaoyi Li, Honghao Fu, Takuya Isogawa, Caio Silva, and Isaac Chuang. *Optimal Quantum Purity Amplification*. 2025. arXiv: 2409.18167 [quant-ph]. URL: https://arxiv.org/abs/2409.18167 (pages 4, 15, 24–26, 35, 36).

[Li+26] Zhaoyi Li, Elias Theil, Aram W. Harrow, and Isaac Chuang. *Quantum Purity Amplification for Arbitrary Eigenstates and Multiple Outputs*. 2026. arXiv: 2605.21570 [quant-ph]. URL: https://arxiv.org/abs/2605.21570 (page 4).

[Liu+26] Zhenhuan Liu, Zhenyu Du, Jens Eisert, Zhenyu Cai, and Zi-Wen Liu. *No Universal Purification in Quantum Mechanics*. 2026. arXiv: 2509.21111 [quant-ph]. URL: https://arxiv.org/abs/2509.21111 (page 2).

[Man26] Atul Mantri. *EFI Pairs Without One-Way Puzzles: Oracle Separations from Communication Complexity*. 2026. arXiv: 2609.11901 [quant-ph]. URL: https://arxiv.org/abs/2609.11901 (page 28).

[Mec19] Elizabeth S. Meckes. *The Random Matrix Theory of the Classical Compact Groups*. Cambridge Tracts in Mathematics. Cambridge University Press, 2019. DOI: 10.1017/9781108303453 (page 28).

[MB26] Antonio Anna Mele and Lennart Bittel. *Optimal learning of quantum channels in diamond distance*. 2026. arXiv: 2512.10214 [quant-ph]. URL: https://arxiv.org/abs/2512.10214 (page 5).

[MY24] Tomoyuki Morimae and Takashi Yamakawa. *One-Wayness in Quantum Cryptography*. 2024. arXiv: 2210.03394 [quant-ph]. URL: https://arxiv.org/abs/2210.03394 (pages 4, 26).

[Mur81] G Eugene Murphy. "A new construction of Young's seminormal representation of the symmetric groups". In: *Journal of Algebra* 69.2 (1981), pp. 287–297. DOI: 10.1016/0021-8693(81)90205-2 (page 16).

[OW16] Ryan O'Donnell and John Wright. "Efficient quantum tomography". In: *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*. STOC '16. Cambridge, MA, USA: Association for Computing Machinery, 2016, pp. 899–912. ISBN: 9781450341325. DOI: 10.1145/2897518.2897544. URL: https://doi.org/10.1145/2897518.2897544 (page 1).

[OW21] Ryan O'Donnell and John Wright. "Quantum spectrum testing". In: *Communications in Mathematical Physics* 387.1 (2021), pp. 1–75. DOI: 10.1007/s00220-021-04180-1 (page 20).

[OV96] Andrei Okounkov and Anatoly Vershik. "A new approach to representation theory of symmetric groups". In: *Selecta Mathematica New Series* 2.4 (1996), pp. 581–606. DOI: 10.1007/BF02433451 (page 16).

[Pel+25] Angelos Pelecanos, Jack Spilecki, Ewin Tang, and John Wright. *Mixed state tomography reduces to pure state tomography*. 2025. arXiv: 2511.15806 [quant-ph]. URL: https://arxiv.org/abs/2511.15806 (pages 1, 5).

[TWZ26] Ewin Tang, John Wright, and Mark Zhandry. *Conjugate queries can help*. 2026. arXiv: 2510.07622 [quant-ph]. URL: https://arxiv.org/abs/2510.07622 (pages 2, 5).

[Wri16] John Wright. "How to learn a quantum state". PhD thesis. Carnegie Mellon University, 2016. URL: http://reports-archive.adm.cs.cmu.edu/anon/2016/CMU-CS-16-108.pdf (page 15).

A Additional representation-theoretic details and an alternate perspective on Algorithms 5 and 6

This appendix contains some supplementary proofs and technical discussion which may support a reader's understanding of our paper, but which is not required to follow its main narrative.

We remind the reader of the following notation:

- V_λ : The irreducible S_n module (Specht module) indexed by λ.
- W_λ : The GL_d module indexed by λ.
- D_λ : The irreducible representation $D_\lambda : \mathbb{C}[S_n] \to \mathcal{L}(V_\lambda)$ mapping elements of the symmetric group algebra to their action on the V_λ module.
- Π^λ : The projector onto the λ-isotypic component of $\left(\mathbb{C}^d\right)^{\otimes n}$.
- $A(S)$: The element of $\mathbb{C}[S_n]$ which projects onto the anti-symmetric subspace on the indices indicated by $S \subseteq [n]$.
- $A(x_1, x_2, ..., x_t)$: The element of $\mathbb{C}[S_n]$ which simultaneously projects onto the anti-symmetric subspaces on the first x_1 elements of $[n]$, the subsequent x_2 elements, and so on.

We will also develop some additional notation to discuss restrictions of Specht modules. For any n and subset $R \subseteq [n]$, let $S_R \times S_{R^c} \subseteq S_n$ be the subgroup of the symmetric group S_n containing permutations which do not mix the elements in R and those in R^c. Additionally, let $\mathrm{Res}^{S_n}_{S_R, S_{R^c}}$ be the functor which maps the Specht module V_λ to the module with the same vector space and representation, but where the group is restricted to $S_R \times S_{R^c}$. This notation lets us restate Pieri's rule, now without any reference to states in $\left(\mathbb{C}^d\right)^{\otimes n}$.

Lemma 20 (Pieri's Rule[4])**.** *For any n, partition $\lambda \vdash n$, and $R \subseteq [n]$ there is an isomorphism between modules:*

$$Res^{S_n}_{S_R, S_{R^c}} V_\lambda \cong \bigoplus_{\mu \vdash |R|, \nu \vdash |R^c|} c^\lambda_{\mu,\nu} \left(V_\mu \otimes V_\nu\right). \tag{96}$$

Equivalently, there is a $(S_R \times S_{R^c})$-equivariant homomorphism mapping vector spaces

$$\phi^\lambda_{R,R^c} : V_\lambda \to \bigoplus_{\mu \vdash |R|, \nu \vdash |R^c|} \left(V_\mu \otimes V_\nu\right)^{\oplus c^\lambda_{\mu,\nu}}. \tag{97}$$

If $\mu = (1, 1, ..., 1)$ is a one column representation then we have $c^\lambda_{\mu,\nu} = 1$ whenever ν can be obtained from λ by deleting exactly one box from $|R|$ distinct rows of λ and $c^\lambda_{\mu,\nu} = 0$ otherwise.

[4] Technically only the second half of this statement (concerning $c^\lambda_{\mu,\nu}$ when μ is a single column partition) is Pieri's rule, while the first half sets up the more general Littlewood-Richardson rule.

We are now in a position to prove a (standard) technical statement which is used without proof in the main document.

Lemma 21. *For every partition λ, there is a unique element $|0_\lambda\rangle \in V_\lambda$ with*

$$D_\lambda(A(\lambda'_1, \lambda'_2, ..., \lambda'_{\lambda_1}))\,|0_\lambda\rangle = |0_\lambda\rangle\,. \tag{98}$$

Equivalently, for each partition λ, the matrix $D_\lambda(A(\lambda'_1, \lambda'_2, ..., \lambda'_{\lambda_1}))$ is a rank one projector.

Proof. We prove the result by induction on the size of λ_1, i.e. the number of columns of λ. The base case is immediate, since any one column irrep corresponds to the sign representation, and in particular is one dimensional.

Next we consider an arbitrary partition λ. We take $R = [\lambda'_1] \subseteq [n]$ and consider the restriction $\mathrm{Res}^{S_n}_{S_R, S_{R^c}}$ applied to V_λ, with ϕ^λ_{R,R^c} the corresponding homomorphism. By equivariance, we have

$$\phi^\lambda_{R,R^c}\left(D_\lambda(A(\lambda'_1, \lambda'_2, ..., \lambda'_{\lambda_1}))\right) = \bigoplus_{\mu \vdash |R|,\, \nu \vdash |R^c|} \left(D_\mu(A(\lambda'_1)) \otimes D_\nu(A(\lambda'_2, \lambda'_3, ..., \lambda'_{\lambda_1}))\right)^{c^\lambda_{\mu,\nu}} \phi^\lambda_{R,R^c}. \tag{99}$$

Because $A(\lambda'_1)$ is the projector onto the anti-symmetric subspace, $D_\mu(A(\lambda'_1))$ is nonzero only when μ is the single partition $(1, 1, ..., 1)$. Note this column has length $|R| = \lambda'_1$. Fix this value for μ. But then Pieri gives that $c^\lambda_{\mu,\nu}$ is nonzero only when ν is the unique partition which can be obtained by deleting one box from each row of λ. Fix this value for ν. Then we have

$$\phi^\lambda_{R,R^c}\left(D_\lambda(A(\lambda'_1, \lambda'_2, ..., \lambda'_{\lambda_1}))\right) = \left(D_\mu(A(\lambda'_1)) \otimes D_\nu(A(\lambda'_2, \lambda'_3, ..., \lambda'_{\lambda_1}))\right) \phi^\lambda_{R,R^c}. \tag{100}$$

But $D_\mu(A(\lambda'_1))$ is the identity on a one-dimensional space, and $D_\nu(A(\lambda'_2, \lambda'_3, ..., \lambda'_{\lambda_1}))$ is a projector onto a one dimensional space by our induction hypothesis. Since ϕ^λ_{R,R^c} is a homomorphism, we are done. □

We now comment briefly on how Lemma 21 can be used to understand the output of Algorithms 5 and 6, providing an alternate perspective on those algorithms. First observe that, for any partition λ,

$$U_{\text{schur}}\left(\Pi^{\lambda'_1}_{\text{anti}} \otimes \Pi^{\lambda'_2}_{\text{anti}} \otimes ... \otimes \Pi^{\lambda'_{\lambda_1}}_{\text{anti}}\right)\Pi^\lambda U^\dagger_{\text{schur}} = D_\lambda(A(\lambda'_1, \lambda'_2, \lambda'_3, ..., \lambda'_{\lambda_1})) \otimes I_{W_\lambda} \tag{101}$$

and we have just shown that $D_\lambda(A(\lambda'_1, \lambda'_2, \lambda'_3, ..., \lambda'_{\lambda_1}))$ is a projector onto a one dimensional space. Then we have, for any $\sigma \in S_n$

$$\mathrm{Im}\left(\left(\Pi^{\lambda'_1}_{\text{anti}} \otimes \Pi^{\lambda'_2}_{\text{anti}} \otimes ... \otimes \Pi^{\lambda'_{\lambda_1}}_{\text{anti}}\right)\Pi^\lambda\right) \supseteq \mathrm{Im}\left(\left(\Pi^{\lambda'_1}_{\text{anti}} \otimes \Pi^{\lambda'_2}_{\text{anti}} \otimes ... \otimes \Pi^{\lambda'_{\lambda_1}}_{\text{anti}}\right)P_\sigma\,\Pi^\lambda\right) \tag{102}$$

with equality whenever

$$\left(\Pi^{\lambda'_1}_{\text{anti}} \otimes \Pi^{\lambda'_2}_{\text{anti}} \otimes ... \otimes \Pi^{\lambda'_{\lambda_1}}_{\text{anti}}\right)P_\sigma\,\Pi^\lambda \neq 0. \tag{103}$$

A similar equation holds when P_σ is replaced by the natural representation of any polynomial in $\mathbb{C}[S_n]$.

Now consider the output of Algorithm 5 applied to some arbitrary input state $\rho^{\otimes n} \in \mathcal{L}\left((\mathbb{C}^d)^{\otimes n}\right)$. Conditioned on its returning partition label λ, and ignoring the $\delta/2$ probability of error, we can write the output of the algorithm (up to normalization) as

$$\sum_r \Pi^\lambda \left(\Pi^{\lambda'_1}_{\text{anti}} \otimes \Pi^{\lambda'_2}_{\text{anti}} \otimes ... \otimes \Pi^{\lambda'_{\lambda_1}}_{\text{anti}}\right) K_r \rho^{\otimes n} K_r^\dagger \left(\Pi^{\lambda'_1}_{\text{anti}} \otimes \Pi^{\lambda'_2}_{\text{anti}} \otimes ... \otimes \Pi^{\lambda'_{\lambda_1}}_{\text{anti}}\right) \Pi^\lambda \tag{104}$$

where the Π^λ and $\left(\Pi^{\lambda'_1}_{\text{anti}} \otimes \Pi^{\lambda'_2}_{\text{anti}} \otimes ... \otimes \Pi^{\lambda'_{\lambda_1}}_{\text{anti}}\right)$ projectors express the fact the output state is a column-anti-symmetric state in the λ isotypic component, and the K_r are Kraus operators describing the full sequences of anti-symmetrization measurements (i.e. applications of Algorithm 1) and subsequent rearrangements of qudits which occurred during the algorithm. The sum over r expresses the fact there are many possible records of intermediate measurement outcomes which can lead to Algorithm 5 outputting partition label λ, and the algorithm deliberately forgets those. But critically, each K_r is just a representation of some polynomial in $\mathbb{C}(S_n)$, so

$$\text{Im}\left(\Pi^\lambda \left(\Pi^{\lambda'_1}_{\text{anti}} \otimes \Pi^{\lambda'_2}_{\text{anti}} \otimes ... \otimes \Pi^{\lambda'_{\lambda_1}}_{\text{anti}}\right) K_r\right) \tag{105}$$

is the same, independent of K_r. This gives an alternate explanation of why no record of the sequence of anti-symmetrization measurements other than the lengths of the final anti-symmetric states produced appears in the permutation information (i.e. the V_λ module of the Schur-transformed state) of the state produced by Algorithm 5.

We can use the same idea to compare Algorithm 6 to the original purity amplification algorithm given in [Li+25]. The first two steps of that algorithm measure the isotypic component and YY basis information of an input $\rho^{\otimes n} \in \left(\mathbb{C}^d\right)^{\otimes n}$, then apply a measurement-dependent correction to rotate to the column-ordered YY basis state, which (following their notation) we denote $|\lambda^\diamond\rangle$. Conditioned on the measuring the λ-isotypic component we can write the post-measurement state produced by this protocol as

$$\sum_{|T\rangle \in YY(V_\lambda)} U^\dagger_{\text{schur}}(|\lambda^\diamond\rangle\langle T| \otimes I_{W_\lambda}) U_{\text{schur}} \Pi^\lambda \rho^{\otimes n} \Pi^\lambda U^\dagger_{\text{schur}}(|T\rangle\langle\lambda^\diamond| \otimes I_{W_\lambda}) U_{\text{schur}} \tag{106}$$

where the sum runs over the set of all YY basis vectors for V_λ, which we denote $YY(V_\lambda)$ and which are described by Young Tableau. We now make a basic, but critical, observation about the column ordered YY basis element $|\lambda^\diamond\rangle$

Lemma 22. *For any partition $\lambda \vdash n$ let $|\lambda^\diamond\rangle$ be the column-ordered YY basis element. Then we have*

$$D_\lambda(A(\lambda'_1, \lambda'_2, ..., \lambda'_{\lambda_1}))\,|\lambda^\diamond\rangle = |\lambda^\diamond\rangle \tag{107}$$

Proof. By Young's orthogonal form (see, for example [Jam06, Section 3.4]) for any YY basis vector indexed by Young Tableau T we have

$$D_\lambda((i, i+1))\,|T\rangle = -\,|T\rangle \tag{108}$$

whenever symbols i and $i+1$ appear in the same column of T. The result follows. □

Combining Lemmas 21 and 22 gives that $|\lambda^\diamond\rangle = |0_\lambda\rangle$, which is the technical fact used in the proof of Theorem 16. It also gives, for any $|T\rangle \in YY(V_\lambda)$

$$U_{\text{schur}}^\dagger(|\lambda^\diamond\rangle\langle T| \otimes I_{W_\lambda})U_{\text{schur}}\Pi^\lambda = \left(\Pi_{\text{anti}}^{\lambda'_1} \otimes \Pi_{\text{anti}}^{\lambda'_2} \otimes ... \otimes \Pi_{\text{anti}}^{\lambda'_{\lambda_1}}\right) U_{\text{schur}}^\dagger(|\lambda^\diamond\rangle\langle T| \otimes I_{W_\lambda})U_{\text{schur}}\Pi^\lambda \tag{109}$$

and

$$\text{Im}(U_{\text{schur}}^\dagger(|\lambda^\diamond\rangle\langle T| \otimes I_{W_\lambda})U_{\text{schur}}\Pi^\lambda) = \text{Im}\left(\left(\Pi_{\text{anti}}^{\lambda'_1} \otimes \Pi_{\text{anti}}^{\lambda'_2} \otimes ... \otimes \Pi_{\text{anti}}^{\lambda'_{\lambda_1}}\right)\Pi^\lambda\right). \tag{110}$$

So we see the image of the first two steps of [Li+25] is the same as the image of Algorithm 5, giving an intuitive justification for Algorithm 6.