A constructive commutative quantum Lovász Local Lemma, and beyond

Toby S. Cubitt*[1] and Martin Schwarz†[2]

[1] *Departamento de Análisis Matemático, Facultad de CC Matemáticas, Universidad Complutense de Madrid, Plaza de Ciencias 3, Ciudad Universitaria, 28040 Madrid, Spain*
[2] *Faculty of Physics, University of Vienna Boltzmanngasse 7, A-1090 Vienna, Austria*

6 December 2011

Abstract

The recently proven Quantum Lovász Local Lemma generalises the well-known Lovász Local Lemma. It states that, if a collection of subspace constraints are "weakly dependent", there necessarily exists a state satisfying all constraints. It implies e.g. that certain instances of the k–QSAT quantum satisfiability problem are necessarily satisfiable, or that many-body systems with "not too many" interactions are always frustration-free.

However, the QLLL only asserts existence; it says nothing about how to *find* the state. Inspired by Moser's breakthrough classical results, we present a constructive version of the QLLL in the setting of commuting constraints, proving that a simple quantum algorithm converges efficiently to the required state. In fact, we provide two different proofs, one using a novel quantum coupling argument, the other a more explicit combinatorial analysis. Both proofs are independent of the QLLL. So these results also provide independent, constructive proofs of the commutative QLLL itself, but strengthen it significantly by giving an efficient algorithm for finding the state whose existence is asserted by the QLLL. We give an application of the constructive commutative QLLL to convergence of CP maps.

We also extend these results to the non-commutative setting. However, our proof of the general constructive QLLL relies on a conjecture which we are only able to prove in special cases.

*tcubitt@mat.ucm.es
†m.schwarz@univie.ac.at

Contents

1 Introduction and Background

The Lovász Local Lemma (LLL), proven by Erdös and Lovász [1975], is a well-known and widely-used result in probability theory. It states that if individual events are "not too" dependent on each other and occur with "not too high" a probability, then there is a positive probability that none of them occur. This is a non-trivial extension of the trivial fact that, if the individual events were completely independent, and if none of them occurred with certainty, then there would be a positive probability that none of them occur. The LLL gives tight bounds on just how dependent the events are allowed to be and how high the probability of the events can be. In its most general form, it states:

Theorem 1 (Erdös and Lovász [1975]) *Let $A_1, A_2, \ldots, A_m$ be events in a probability space, and let $\Gamma(A_i)$ denote the set of events that are* not *independent of A_i, excluding A_i itself. If there exist values $0 \leq x_1, x_2, \ldots, x_m \leq 1$ such that*

$$\forall i : \Pr(A_i) \leq x_i \cdot \prod_{A_j \in \Gamma(A_i)} (1 - x_j), \tag{1}$$

then the probability that none *of the events occur is at least $\prod_i (1 - x_i)$. (In particular, it is positive.)*

Applications of the LLL abound [Alon and Spencer, 2008]. It is frequently invoked in order to prove existence of some mathematical object via the probabilistic method. For example, it can be used to prove existence of solutions to boolean satisfiability problems. The k–SAT problem asks whether there exists an assignment of truth values to a set of boolean variables that satisfies a boolean expression in conjunctive-normal form (CNF), where each clause in the CNF formula involves at most k variables. Letting A_i be the event that the i^{th} clause is violated when the assignment is chosen at random, the LLL implies:

Corollary 2 (Symmetric Lovász Local Lemma) *If each variable in a k–CNF formula appears in at most $2^k/(e \cdot k)$ clauses, then there exists a satisfying assignment for the formula.*

Indeed, this amounts to a symmetric, uniform version of the full Lovász Local Lemma.

Recently, by replacing events with subspaces, and probabilities with relative dimensions, Ambainis, Kempe, and Sattath [2009] succeeded in generalising the LLL to the quantum setting. The relative dimension $R(X)$ of a subspace X in a vector space V is the ratio $R(X) = \dim(X)/\dim(V)$. Two subspaces X_i, X_j are said to be *R-independent* if $R(X_i \cap X_j) = R(X_i)R(X_j)$. In its most general form, the *Quantum Lovász Local Lemma* (QLLL) states:

Theorem 3 (Ambainis, Kempe, and Sattath [2009])
Let $X_1, X_2, \ldots, X_m$ be subspaces, and let $\Gamma(X_i)$ denote the set of subspaces that are *not* R-independent of X_i, excluding X_i itself. If there exist values $0 \leq x_1, x_2, \ldots, x_m \leq 1$ such that the relative dimensions $R(X_i)$ satisfy

$$R(X_i) \geq 1 - x_i \cdot \prod_{X_j \in \Gamma(X_i)} (1 - x_j), \tag{2}$$

then $R(\bigcap_i X_i) \geq \prod_i (1 - x_i)$. In particular, the intersection $\bigcap_i X_i$ has positive dimension.

(Note that, because of the properties of R-independence under orthogonal complement, the QLLL has to be stated "the other way around" to the LLL, so that the subspaces correspond to events that one *does* want to occur. See Ambainis, Kempe, and Sattath [2009, Lemma 11], and the discussion thereafter.)

Viewed from one perspective, the QLLL has little to do with quantum physics; rather, it is a mathematical generalisation of the standard LLL to a geometrical setting. However, viewed from another perspective, the quantum version is closely related to current topics of physics research.

For example, just as the LLL can be applied to k–SAT problems, the QLLL can be applied to its quantum generalisation, k–QSAT [Bravyi, 2006]. Boolean variables become qubits, clauses in a CNF formula become projectors Π_i that act non-trivially on k qubits, and the k–QSAT problem asks whether there is a state $|\psi\rangle$ of the qubits satisfying $\forall i : \Pi_i |\psi\rangle = 0$. The QLLL implies:

Corollary 4 (Ambainis, Kempe, and Sattath [2009])
Let $\Pi_1, \Pi_2, \ldots, \Pi_m$ be a k–QSAT instance where all projectors have rank at most r. If each qubit occurs in at most $2^k/(e \cdot r \cdot k)$ projectors, then there exists a satisfying state for the k–QSAT instance.

An equivalent way of expressing the k–QSAT problem is to ask whether the Hamiltonian $H = \frac{1}{m}\sum_i \Pi_i$ has a zero-energy ground state. Since the Π_i are positive-semidefinite, this is equivalent to asking whether the ground state of the overall Hamiltonian is simultaneously the ground state of all the individual local terms; i.e. we are asking whether or not the ground state is *frustrated*. Replacing the projectors Π_i with positive-semidefinite local Hamiltonian terms h_i whose support (coimage) is Π_i does not affect whether the ground state is frustrated. So the k–QSAT problem amounts to asking whether the ground state of an interacting many-body

system is frustrated or not. The QLLL implies that many-body systems in which each particle interacts with "not too many" others are *never* frustrated.

The LLL and its quantum counterpart assert existence, e.g. of a satisfying assignment. But they give no indication as to how to *find* this assignment. In a breakthrough result, Moser [2009] gave a beautiful proof of a constructive version of the classical LLL. This not only gives an independent proof of the LLL, but does so by providing an efficient algorithm for *finding* the point in probability space whose existence is asserted by the LLL (e.g. the satisfying assignment to a k–SAT problem).

Moser [2009] originally proved this for the symmetric LLL, achieving the tight asymptotic scaling but not quite achieving the tight constant in the bounds of Corollary 2. With Tardos, he quickly generalised his proof to cover the general LLL and match the tight constants of Theorem 1 [Moser and Tardos, 2010]. The proof imposes a very slight restriction, in requiring that the events in the LLL be determined by different subsets of underlying, mutually-independent, random variables:

Theorem 5 (Moser and Tardos [2010]) *Let $p_1, p_2, \ldots, p_n$ be mutually-independent random variables, and let $A_1, A_2, \ldots, A_m$ be events determined by these variables. If there exist values $0 \leq x_1, x_2, \ldots, x_m \leq 1$ such that*

$$\forall i : \Pr(A_i) \leq x_i \cdot \prod_{A_j \in \Gamma(A)} (1 - x_j), \tag{3}$$

then there exists an assignment of values to the variables p_i such that none *of the events A_i occur. Moreover, there is a randomised algorithm (Algorithm 1) that finds this assignment in expected time*

$$O\Big(n + \sum_{i=1}^{m} \frac{x_i}{1 - x_i} \cdot |A_i|\Big), \tag{4}$$

where $|A_i|$ is the number of variables involved in determining event A_i.

The algorithm is almost the simplest randomised algorithm one could imagine (Algorithm 1). We are looking for an assignment such that none of the events occur, so we say that an event is *violated* by an assignment if it occurs for that assignment. The algorithm maintains a register $v = v_1 v_2 \ldots v_n$ of assignments to the variables p_i. At each step, it checks whether any event A_j is violated by the current assignment, and if so replaces the assignments v_i to the variables involved in that event by values chosen uniformly at random. It repeats this procedure until no events are violated. A priori it is not obvious that this process will ever terminate; Theorem 5 proves that it in fact terminates in linear expected time!

Algorithm 1 Classical Solver

```
function solve_lll((A_1, A_2, ..., A_m))
    for all p_i do
        v_i ← a random evaluation of p_i
    end
    while ∃A_j violated by v do
        pick a violated event A_j;
        for all p_i ∈ A_j do
            v_i ← a random evaluation of p_i;
        end
    end while
    return v;
end function
```

2 Results

In this paper we prove a constructive version of the commutative case of the Quantum Lovász Local Lemma. As in the constructive version of the (classical) LLL, we have to impose a slight restriction, in requiring that the subspaces respect an underlying tensor product structure. We can, without loss of generality, take the underlying structure to be the state space of a set of qudits, analogous to the underlying random variables in Theorem 5. Each subspace in the QLLL is then defined on some subset of the qudits. (More precisely, it is the extension of such a subspace to the full Hilbert space of all the qudits.)

It will be convenient in the constructive QLLL to represent subspaces by projectors. We define the relative dimension of a projector Π to simply be the relative dimension of the subspace X onto which it projects: $R(\Pi) := \operatorname{rank}(\Pi)/\dim(V) = \dim(X)/\dim(V) = R(X)$. With the restriction to an underlying tensor-product space, two subspaces are R-independent iff their corresponding projectors act non-trivially on at least one qudit in common; we say that the projectors *intersect* in this case. Conversely, two projectors that act non-trivially on disjoint subsets of qudits are said to be *disjoint*. We can simplify the notation by letting $[i]$ denote the subset of qudits on which projector Π_i acts non-trivially; then Π_i and Π_j intersect iff $[i] \cap [j] \neq \varnothing$. We write $\Gamma(\Pi_i)$ for the set of projectors that intersect with Π_i, *excluding* Π_i itself, and $\Gamma^+(\Pi_i) = \Gamma(\Pi_i) \cup \{\Pi_i\}$ for the set that *includes* Π_i.

Definition 6 (Lovász conditions) *Let $\Pi_1, \Pi_2, \ldots, \Pi_m$ be projectors that act on subsets of n qudits. We say that the set of projectors $\{\Pi_i\}$ satisfies the* Lovász conditions *if there exist values $0 \leq x_1, x_2, \ldots, x_m \leq 1$ such that*

$$R(\Pi_i) \leq x_i \cdot \prod_{\Pi_j \in \Gamma(\Pi_i)} (1 - x_j). \tag{5}$$

We prove the following constructive version of the QLLL (Theorem 3) in the case of commuting projectors:

Theorem 7 (Constructive Commutative QLLL) *Let $\Pi_1, \Pi_2, \ldots, \Pi_m$ be* mutually commuting *projectors acting on subsets of n qudits. If $\{\Pi_i\}$ satisfy the Lovász conditions, then there exists a joint state ρ of the qudits such that $\forall i : \operatorname{Tr}[\Pi_i \rho] = 0$.*

Moreover, there is a quantum algorithm that converges to a state ρ' such that $\operatorname{Tr}[\Pi_i \rho'] \leq \varepsilon$ (or, equivalently, such that $\operatorname{Tr}[P_0 \rho] \geq 1 - \varepsilon$ when P_0 is the projector onto the subspace $\operatorname{span}\{|\psi\rangle : \forall i \, \Pi_i |\psi\rangle = 0\}$) in time

$$O\left(n + \frac{m}{\varepsilon} \sum_{i=1}^{m} \frac{x_i}{1 - x_i} \cdot \big|[i]\big|\right), \tag{6}$$

where $|[i]|$ is the number of qudits on which the projector Π_i acts non-trivially.

(Note that, because we restrict to subspaces with a tensor product structure, there is no longer any need to state the constructive QLLL "the other way around" as there was in Theorem 3.) The algorithm is a generalisation of Algorithm 1 to the quantum setting, and is described in Section 3.

There are two significant challenges in generalising Moser's constructive LLL to the quantum setting. First, the state we are trying to construct may be highly entangled, whereas the algorithm only has access to measurements of the local projectors Π_i. Another way of expressing this in terms of the Hamiltonian H defining a k–QSAT instance is that, in the classical setting, we know in advance in which basis the overall Hamiltonian H is diagonal—the computational basis—and the local projectors Π_i are local in this same basis. In the quantum setting, the basis which diagonalises the overall Hamiltonian is only defined globally, and can be highly entangled.

The projectors Π_i will not in general be local in this basis, but will act non-trivially on the entire system. Were we to naïvely apply the Moser and Tardos [2010] algorithm in this diagonal basis, every time we measured a projector Π_i to be "violated" (i.e. we obtained the Π_i outcome upon performing the $\{\Pi_i, \mathbb{1} - \Pi_i\}$ measurement), we would have to discard the entire state and start from scratch. Thus a naïve application of the Moser and Tardos [2010] algorithm to the quantum case reduces to picking a random state, checking if it satisfies a constraint, and, if not, discarding the entire state and trying again. This certainly cannot find the correct state efficiently.

Note that this challenge remains just as problematic even if the projectors Π_i commute. In the commutative case of the QLLL, we may know a priori that there exists a basis which diagonalizes all projectors simultaneously, but this basis is still only defined globally, the ground state can still be highly entangled, and the projectors can still be non-local in the diagonal basis. (Stabiliser states are a simple example of commuting Hamiltonians with highly entangled ground states [Nielsen and Chuang, 2000].) Indeed, this and related questions in the commutative setting have recently been gaining increasing attention in the context of Hamiltonian complexity [Aharonov and Eldar, 2011, Bravyi and Vyalyi, 2003, Schuch, 2011].

The second challenge comes from non-commutativity: quantum states are disturbed by measurement. The classical algorithm is free to check which k–SAT clauses are currently satisfied, without affecting the current variable assignment. But quantum mechanically, even if we measure a k–QSAT projector Π_i to be "satisfied" (i.e. we obtain the desired $\mathbb{1} - \Pi_i$ outcome upon performing the $\{\Pi_i, \mathbb{1} - \Pi_i\}$ measurement), the measurement can disturb the state so as to increase the probability of measing another Π_j to be violated.

Here, we address and give a complete solution to the first of these two challenges: we prove a constructive version of the commutative QLLL (i.e. the case in which all the Π_i commute). A priori, it is not at all clear that Moser's proof extends even to the commutative quantum case, for the reasons discussed above. Nonetheless, by extending the proof techniques of Moser [2009] and Moser and Tardos [2010] in a more subtle way, we *are* able to prove a constructive version of the commutative QLLL. Moreover, the quantum algorithm involved in Theorem 7 is just the natural quantum generalisation of Moser and Tardos [2010], and almost the simplest imaginable (see Algorithm 2). It also coincides with the natural dissipative state preparation algorithm studied in Verstraete et al. [2009].

Whilst the algorithm is straightforward, its analysis is not. We provide *two* different proofs of the constructive commutative QLLL, using two very different approaches. The first proof, described in the main text, generalises the probabilistic approach of Moser and Tardos [2010]. The key step in the proof is the replacement of the classical coupling argument used by Moser and Tardos [2010], with a *quantum coupling argument*, which uses a coupling by entanglement. To our knowledge, this is the first example of a quantum coupling argument, used as a proof technique to establish convergence of a quantum stochastic process, which may be of independent interest. Even though the entanglement is not used directly as a resource by the algorithm, it is the unique properties of entanglement that allow us to prove via the quantum coupling that the algorithm can find the correct global basis even though it has access only to local measurements.

Coupling arguments are long established as a very powerful proof technique in probability theory, often providing the simplest or even the only proof of many results [Lindvall, 2002, Thorisson, 2000]. Our quantum generalisation of the coupling method is no exception, providing an elegant and concise proof of the constructive commutative QLLL of Theorem 7. But coupling arguments often seem a little like "black magic". In our second proof, desribed in Section 4, we replace the coupling argument with a combinatorial proof. Whilst (as is often the case) the combinatorial argument is significantly more involved than the coupling argument, it is nonetheless more explicit. It demonstrates how the algorithm can be understood as a quantum stochastic process produced by iterated measurement, and leads to interesting new results on

such iterated measurement processes.

To generalise these results to the general non-commutative QLLL requires that we address the second challenge: non-commutativity of quantum measurement, and the concomitant measurement-disturbance issue. The difficulty here is that "satisfied" measurements now play a significant role, and can sometimes make things worse instead of better for later measurements. Even the order in which the "satisfied" outcomes occur is now significant. We give a simple non-commutative counter-example which already violates the crucial bounds that we prove in the commuting case by both coupling and combinatorial arguments. This suggests that sharper techniques will be required to address this second challenge, and extend our results to the general, non-commutative setting.

Nonetheless, although we are not able to give a complete proof of a constructive QLLL in the non-commutative setting, we *can* prove it if we assume a technical conjecture. We prove the conjecture in certain simple cases, and it is also supported in more general cases by numerical evidence (though the numerics we have done are limited). Furthermore, it is likely that the conjecture *must* hold if the natural quantum generalisation of Moser's classical algorithm is to work. If our conjecture is false, either there is no efficient constructive version of the general QLLL, or an entirely different approach is needed in the non-commutative setting.

The paper is organised as follows. In Section 3 we briefly review the classical proof of Moser and Tardos [2010], then prove the constructive commutative Quantum Lovász Local Lemma using a novel quantum coupling argument. Section 4 contains an alternative, combinatorial proof of this result. In Section 5 we apply the results of the previous section to bound the convergence time of certain classes of CP maps. In Section 6 we generalise the results to the full non-commuting setting, though we have to assume a technical conjecture which we are currently unable to prove except in some simple special cases. In Section 7 we conclude with an outline of applications of our results to physics and quantum algorithms, some possible generalisations in the commutative setting, and a discussion of open problems and potential directions in the non-commutative setting.

3 A Constructive Proof of the Commutative Quantum Lovász Local Lemma

In order to build a constructive proof of the QLLL, we start by adapting the elegant argument of Moser and Tardos [2010] so that it applies to the commutative quantum case. Our first step is to generalise the Moser-Tardos algorithm in the obvious way. The quantum algorithm acts on an *assignment register* of n qudits, holding a state (or *assignment*) $|\alpha\rangle$. (We abuse notation slightly by using $|\alpha\rangle$ to denote both the register itself and the state of the register, even though that state need not be pure.) If the measurement $\{\Pi_i, \mathbb{1} - \Pi_i\}$ is performed on an assignment and yields the outcome Π_i, then we say that projector Π_i was *violated*. We denote the set of qudits on which a projector Π_i acts non-trivially by $[i]$. In another abuse of notation, the i^{th} qudit of the assignment register will be denoted $|\alpha_i\rangle$, even though the reduced state of that qudit need not be pure. The algorithm will also require a uniform random source $P = (\{|0\rangle, |1\rangle, \ldots, |d-1\rangle\}, \Pr(|i\rangle) = 1/d)$ that emits qudits in a random computational basis state. We use the notation $|\alpha_i\rangle \leftarrow P$ to indicate replacing the contents of the quantum register $|\alpha_i\rangle$ with a fresh sample from the random source P.

Our task is, firstly, to show that Algorithm 2 converges to a state satisfying the requirements of Theorem 7, thereby proving that such a state exists, and, secondly, to prove that this convergence occurs in polynomial time.

Before adapting the Moser and Tardos [2010] proof to the quantum case, let us review the high-level structure of their argument. Note that, as they proceed, Algorithms 1 and 2 keep a log (Lines 6 and 9, respectively) of which events or projectors were violated. For each entry in the log, Moser and Tardos [2010] imagine constructing a "witness tree" from all the log data up

Algorithm 2 Quantum Solver

```
1: function solve_qlll((Π_1, Π_2, ..., Π_m))
2:     for all i do
3:         |α_i⟩ ← P
4:     end
5:     loop
6:         pick a projector Π_i uniformly at random;
7:         measure the projector Π_i;
8:         if the projector was violated then
9:             append Π_i to the execution log;
10:            for all j ∈ [i] do
11:                |α_j⟩ ← P
12:            end
13:        end if
14:    end loop
15: end function
```

to that point. (We will describe the precise procedure for constructing the trees later, but the details are not important for the overall structure of the argument.) Each violation adds one entry to the log, and each log entry is associated with a different witness tree, so the number of distinct witness trees that can be constructed from—or *occur in*—the log gives the number of violations seen by the algorithm.

Therefore, in order to compute the expected number of violations seen by the algorithm, we would like to compute the probability that a particular witness tree can occur in the algorithm's log. Of course, it is not at all clear how to compute this probability directly. So Moser and Tardos [2010] use a coupling argument to relate the behaviour of the random process generated by the algorithm to a much simpler random process, whose behaviour is more easily analysed. They call this simple random process "τ-check", because it executes a random process on a witness tree τ, and outputs either "pass" or "fail" at the end. (Again, we will describe the τ-check procedure later, but the details are not important for now.)

Viewed separately, we have two different random processes: the algorithm, whose behaviour is difficult to analyse, and τ-check, whose behaviour is more easily computed. The coupling is established by proving that, if these two random processes are fed the *same* random source, then their behaviour becomes correlated in such a way that if a witness tree τ occurs in the algorithm's log, then τ-check will always output "pass", i.e.

$$\Pr(\tau\text{-check passes} \mid \tau \text{ occurs in the log}) = 1. \tag{7}$$

A simple application of Bayes' theorem then implies that

$$\Pr(\tau \text{ occurs in the log}) = \frac{\Pr(\tau \text{ occurs in the log} \mid \tau\text{-check passes})}{\Pr(\tau\text{-check passes} \mid \tau \text{ occurs in the log})} \cdot \Pr(\tau\text{-check passes}) \tag{8a}$$

$$\leq \Pr(\tau\text{-check passes}), \tag{8b}$$

(using Eq. (7) and the fact that the numerator, being a probability, is upper-bounded by 1). Thus we can bound the probability of a particular witness tree occurring in the algorithm's log (which is the same as the probability of a particular violation occurring) by the probability that τ-check passes. In order to bound the latter, Moser and Tardos [2010] relate the τ-check procedure to a Galton-Watson branching process,[1] which can be analysed straightforwardly.

[1] The Galton-Watson process is a standard random process in probability theory, originally derived in order to model extinction of aristocratic surnames in family trees, something of great concern to the Victorians in 19$^{\text{th}}$ century England.

3.1 Witness Trees

Imagine constructing a tree of violated projectors from the data in the log, at any point during the algorithm, exactly as in Moser and Tardos [2010, Section 2]. Denote projectors appearing in the log by $\Pi^{(i)}$, for $i = 0 \dots l$ (where l is the number of entries in the log). To construct a tree starting from entry n in the log, we work backwards from projector $\Pi^{(n)}$. First, create a root node labelled by $\Pi^{(n)}$. For each preceding projector $\Pi^{(i<n)}$ in the log, create a new vertex, labelling it by $\Pi^{(i)}$, and attach it below an existing vertex labelled by a projector which intersects with $\Pi^{(i)}$. If there is more than one such vertex, attach it below the one furthest from the root (breaking ties arbitrarily). If there is no such vertex, simply discard the new vertex.

Note two key observations about the trees constructed by this procedure. Firstly, projectors located at the same level in the tree can never intersect. Secondly, if a vertex labelled by $\Pi^{(i)}$ is located at a higher level of the tree (closer to the root) than a vertex labelled by $\Pi^{(j)}$, and if projectors $\Pi^{(i)}$ and $\Pi^{(j)}$ intersect, then $\Pi^{(i)}$ must have occurred later[1] in the log than $\Pi^{(j)}$, i.e. $i > j$. Together, these properties imply that the tree encodes an ordering of the violated projectors with respect to the partial order defined by projector intersections.

A tree is called *proper* if distinct children of the same vertex always receive distinct labels, and we say that a tree τ *occurs* in the log if τ can be generated by constructing a tree starting from some entry of the log. We now show that trees occurring in the log of Algorithm 2 still satisfy Lemma 2.1 from Moser and Tardos [2010], reinterpreting it for the quantum case in the obvious way:

Lemma 8 *Let τ be a fixed tree and L the (random) log produced by Algorithm 2.*

(i). If τ occurs in L, then τ is proper.

(ii). The probability that τ occurs in L is at most $\prod_{v\in\tau} \Pr[\Pi(v)]$,

where $\Pi(v)$ is the projector labelling vertex v, and $\Pr[\Pi(v)]$ is the probability of measuring $\Pi(v)$ on the maximally mixed state.

Proof Part (i) follows immediately from the above observations about the way the trees are constructed. (Indeed, there is nothing quantum about part (i), so the statement and proof are identical to the corresponding part of Lemma 2.1 in Moser and Tardos [2010].)

To prove part (ii), we introduce a new technique, a *quantum coupling argument*, to establish a coupling between Algorithm 2 and Algorithm 3. Algorithm 3 is a straightforward quantum version of the classical τ-check procedure. In the classical case the two processes are coupled by a joint random source. To establish the commutative quantum case we replace the joint random source of qudits of Algorithm 2 and Algorithm 3 by a source of maximally entangled qudit pairs. One half of the pair is provided to the each of the two coupled quantum processes. Since tracing out one half of each pair results in the maximally mixed state on the other half, the coupling is undetectable by the process acting on the remaining subsystem: the marginal distributions of the two coupled random processes are identical to the original, separate random processes Algorithm 2 and Algorithm 3. Nevertheless, entanglement across the subsystem boundary can be exploited to bound joint event probabilities, just as correlations are exploited by a classical coupling argument. More formally, we proceed by proving the following lemmas, which together imply part (ii).

Lemma 9 $\Pr(\tau\text{-check passes}) = \prod_{v\in\tau} \Pr[\Pi(v)]$, *where $\Pr[\Pi(v)]$ is the probability of measuring $\Pi(v)$ on the maximally mixed state.*

Proof This is trivial, since τ-check independently measures each projector $\Pi(v)$ on the maximally mixed state drawn from the uniform random sources Q_i. (Indeed, this is true independent of the order in which τ-check visits the vertices.) □

[1] Remember that the tree is constructed by reading the log *backwards*.

Algorithm 3 Quantum τ-check

```
1: function τ-check(τ)
2:     sort nodes of τ in reverse breadth-first order;
3:     for all v ∈ τ do
4:         for all x ∈ [i] do
5:             |β_i⟩ ← Q_i
6:         end
7:         measure Π_i^T on |β_[i]⟩;
8:         if the measurement was satisfied (gave outcome 𝟙 − Π_i) then
9:             return "failed"
10:        end if
11:    end
12:    return "passed"
13: end function
```

The next lemma establishes the quantum generalisation of Eq. (7) in the classical argument.

Lemma 10 *Couple the random sources P_i of Algorithm 2 and Q_i of Algorithm 3, so that the nth qudit from P_i is maximally entangled with the nth qudit from Q_i. Then, if $\{\Pi_i\}$ commute,* $\Pr(\tau\text{-check passes}|\tau \text{ occurs}) = 1$.

Proof First, notice that if τ occurs in Algorithm 2's execution log, then each label $\Pi(v)$ in τ corresponds to a unique violation $\Pi(v)$ in the log. Furthermore, the partial ordering with respect to projector intersections of that section of the log that generates τ, is precisely the partial order defined by τ itself. Therefore, by visiting the vertices in reverse-breadth-first order, τ-check will measure the projectors $\Pi(v)^T$ in the same order as Algorithm 2 measured $\Pi(v)$, up to reorderings of projectors that act on disjoint sets of qudits. Thus, when τ-check draws random qudits on which order to measure $\Pi(v)^T$, it draws exactly the qudits from Q_i that started off maximally entangled with those on which Algorithm 2 measured $\Pi(v)$.[1]

It is helpful to picture each random source P_i, Q_i as a semi-infinite stack of qudits (each being one half of a maximally entangled pair between P_i and Q_i). When a fresh qudit is drawn, the random source takes it from the bottom of it stack. The nth qudit in the P_i stack is then in one-to-one correspondence with the nth qudit in the Q_i stack, with which it is maximally entangled. Also, when Algorithm 2 replaces the qudits on which it has just measured a violation Π_i with fresh ones from the P_i stacks (Line 11), we can equivalently think of this as appending the fresh qudits to the register $|\alpha\rangle$, and redefining all projectors that act on qudits $[i]$ to now act on these new qudits instead. For a given state of register $|\alpha\rangle$, let $|\beta\rangle$ denote the corresponding qudits from the Q_i stacks. (As with $|\alpha\rangle$ this is a slight abuse of notation, as the reduced state of the qudits in $|\beta\rangle$ will not necessarily be pure.)

In this picture, let P_t denote the projector on the appropriate qudits on $|\alpha\rangle$ corresponding to the tth measurement outcome of Algorithm 2, which could be either a violation Π_i or a satisfied measurement $\mathbb{1} - \Pi_i$. (Since we are now appending qudits from P_i to the register $|\alpha\rangle$ whenever Algorithm 2 measures a violation, and simply redefining which qudits of $|\alpha\rangle$ future measurements Π_i act on instead of tracing out the old ones, the set of qudits on which the projector P_t acts depends on the violations $P_{i<t}$ that occurred prior to P_t.)

Let $|\Omega\rangle = \otimes|\omega\rangle = \left(\sum_i |i\rangle|i\rangle/d\right)^{\otimes m}$ denote the maximally entangled state between $|\alpha\rangle$ and $|\beta\rangle$. Assume Algorithm 2 measures a violation $P_t = \Pi(v)$ (acting on the appropriate qudits of

[1] So far, this is identical to the argument in [Moser and Tardos, 2010, Lemma 2.1]. However, between drawing qudits from P_i and measuring $\Pi(v)$ on them, Algorithm 2 may have performed arbitrarily many "satisfied" measurements.

the growing $|\alpha\rangle$ register) in the tth measurement. The (unnormalised) state of $|\alpha, \beta\rangle$ after this measurement is then given by

$$|\alpha, \beta\rangle = \left[\Pi(v) P_{t-1} \cdots P_2 P_1 \otimes \mathbb{1}\right] |\Omega\rangle \tag{9}$$

What is probability that τ-check "fails" when it performs the corresponding $\Pi(v)^T$ measurement, given that τ occurs in the log? We know from above that when τ-check draws qudits from $Q_{[v]}$ to measure $\Pi(v)^T$, it obtains precisely those qudits from $|\beta\rangle$ that correspond to the qudits in $|\alpha\rangle$ on which $\Pi(v)$ acts. Now, τ-check outputs "fail" if it measures $\mathbb{1} - \Pi(v)^T$, so the probability of this occurring is given by

$$\begin{aligned}
&\mathrm{Tr}\left[\mathbb{1} \otimes (\mathbb{1} - \Pi(v)^T) |\alpha, \beta\rangle\langle\alpha, \beta|\right] && \text{(10a)}\\
&= \langle\alpha, \beta| \, \mathbb{1} \otimes (\mathbb{1} - \Pi(v)^T) \, |\alpha, \beta\rangle && \text{(10b)}\\
&\propto \langle\Omega| \left[P_1 P_2 \cdots P_{t-1} \Pi(v) P_{t-1} \cdots P_2 P_1\right] \otimes (\mathbb{1} - \Pi(v)^T) \, |\Omega\rangle && \text{(10c)}\\
&= \langle\Omega| \left[P_1 P_2 \cdots P_{t-1}\right] \otimes \left[(\mathbb{1} - \Pi(v)^T) P_1^T P_2^T \cdots P_{t-1}^T \Pi(v)^T\right] |\Omega\rangle && \text{(10d)}\\
&= \langle\Omega| \left[P_1 P_2 \cdots P_{t-1}\right] \otimes \left[P_1 P_2^T \cdots P_{t-1} (\mathbb{1} - \Pi(v)^T) \Pi(v)^T\right] |\Omega\rangle && \text{(10e)}\\
&= 0, && \text{(10f)}
\end{aligned}$$

where Eq. (10c) is only a proportionality because we have not normalised the state, and in Eq. (10d) we have used the property of the maximally entangled state $A \otimes \mathbb{1} |\omega\rangle = \mathbb{1} \otimes A^T |\omega\rangle$, which holds for any operator A. In Eq. (10e), we have used the fact that all the $\{P_i\}$ commute with $\Pi(v)$, which follows immediately from the fact that $\{\Pi_i\}$ commute, by assumption.

This holds for any violation $\Pi(v)$, and any sequence of measurement outcomes $P_{i<t}$ prior to $P_t = \Pi(v)$. Therefore, given that τ occurs in Algorithm 2's log, the probability of τ-check "failing" on any of its measurements is zero, which proves the lemma. □

If we trace out τ-check, then the states drawn from P_i are maximally mixed, as required in Algorithm 2. Similarly, if we trace out the algorithm, then the states drawn from Q_i by τ-check are maximally mixed as required in Algorithm 3, so Lemma 9 still holds when the random sources are entangled. Thus, as in the analogous classical proof of Moser and Tardos [2010], Lemmas 9 and 10 together prove part (ii). □

3.2 Expected number of violations

Having shown in Lemma 8 that a quantum version of Lemma 2.1 from Moser and Tardos [2010] holds, the remainder of the argument in Moser and Tardos [2010] goes through unchanged. We repeat the argument here for completeness.

In order to bound the probability of a tree τ occurring in the log given in part (ii) of Lemma 8, Moser and Tardos [2010] relate it to the probability of the following Galton-Watson branching process generating a proper witness tree τ_a whose root vertex is labelled by some fixed projector Π_a. In the first iteration, the process produces a root vertex labelled by Π_a. In subsequent iterations, the process considers each vertex added in the previous iteration independently. For vertex v labelled by Π_i, it considers each projector Π_j in the set $\Gamma^+(\Pi_i)$ of projectors that intersect with Π_i (including Π_i itself). Independently for each such $\Pi_j \in \Gamma^+(\Pi_i)$, it chooses at random whether to add a child vertex labelled by Π_j below v with probability x_j, or whether to skip Π_j with probability $1 - x_j$. Thus if all the Π_j are skipped, the branch dies out at v. The process continues until all branches die out. (Depending on the probabilities, the process could of course continue indefinitely.)

Lemma 3.1 of Moser and Tardos [2010] computes the probability that this process produces a given tree τ_a using standard techniques for Galton-Watson branching processes. We restate

the lemma here for convenience. For notational simplicity, let

$$x'_i \coloneqq x_i \cdot \prod_{\Pi_j \in \Gamma(\Pi_i)} (1 - x_j). \tag{11}$$

Lemma 11 (Moser and Tardos [2010]) *Let τ_a be a given proper witness tree whose root vertex is labelled by Π_a. The probability that the Galton-Watson process described above produces the tree τ_a is*

$$\Pr(\tau_a) = \frac{1 - x_a}{x_a} \prod_{\Pi_i \in \tau_a} x'_i, \tag{12}$$

(where the product is over all vertex labels in τ_a, including repetitions).

Using this, we can bound the expected number of violations seen by Algorithm 2. Note that this bound is *independent* of the number of iterations in Algorithm 2. The expected number of violations is constant even if we run Algorithm 2 forever.

Theorem 12 *For any set of mutually-commuting $\{\Pi_i\}$ satisfying the Lovász conditions of Definition 6, the expected number of violations seen by Algorithm 2 is bounded by*

$$\mathbb{E}(\text{total number of violations}) \leq \sum_{i=1}^{m} \frac{x_i}{1 - x_i}. \tag{13}$$

Proof Let N_a be the number of times a given projector Π_a appears in Algorithm 2's log, which is the same as the number of times Π_a is measured to be violated. Let $\mathcal{T}_a$ denote all proper witness trees whose root vertex is labelled by Π_a. Then, from Lemma 8, we have

$$\mathbb{E}(N_a) = \sum_{\tau \in \mathcal{T}_a} \Pr(\tau \text{ appears in the log}) \leq \sum_{\tau \in \mathcal{T}_a} \prod_{v \in \tau} \Pr[\Pi(v)]. \tag{14}$$

Now, the probability $\Pr[\Pi_i]$ of a projector being violated on a random state is just given by its relative dimension $\Pr(\Pi_i) = R(\Pi_i)$. Since by assumption the projectors satisfy the Lovász conditions (Definition 6), the relative dimension satisfies $R(\Pi_i) \leq x'_i$. Thus

$$\mathbb{E}(N_a) \leq \sum_{\tau \in \mathcal{T}_a} \prod_{v \in \tau} \Pr[\Pi(v)] \leq \sum_{\tau \in \mathcal{T}_a} \prod_{\Pi_i \in \tau} R(\Pi_i) \leq \sum_{\tau \in \mathcal{T}_a} \prod_{\Pi_i \in \tau} x'_i \leq \sum_{\tau \in \mathcal{T}_a} \frac{x_a}{1 - x_a} \Pr(\tau), \tag{15}$$

the final inequality following from Lemma 11, $\Pr(\tau_a)$ being the probability of the Galton-Watson process of Lemma 11 generating tree τ_a. Since that process either produces a proper witness tree in $\mathcal{T}_a$ with root vertex labelled by Π_a, or continues indefinitely, we have $\sum_{\tau \in \mathcal{T}_a} \Pr(\tau) \leq 1$, thus

$$\mathbb{E}(N_a) \leq \frac{x_a}{1 - x_a} \sum_{\tau \in \mathcal{T}_a} \Pr(\tau_a) \leq \frac{x_a}{1 - x_a}, \tag{16}$$

and the theorem follows from summing over all projectors. □

3.3 An efficient quantum algorithm

We are now in a position to prove Theorem 7: a constructive version of the commutative Quantum Lovász Local Lemma. From the bound on the expected number of violations seen by Algorithm 2 already proven in the previous section, it is easy to prove the existence part of the commutative Quantum Lovász Local Lemma (i.e. that there exists a state satisfying all the constraints).

Proof (of existence part of Theorem 7) The theorem asserts existence of a state ρ such that $\forall i : \mathrm{Tr}[\Pi_i \rho] = 0$, when $\{\Pi_i\}$ is a set of projectors satisfying the Lovász conditions. Assume for contradiction that no such state exists. Then we have

$$\delta \coloneqq \min_{\rho} \frac{1}{m} \sum_i \mathrm{Tr}[\Pi_i \rho] > 0. \tag{17}$$

But $\frac{1}{m} \sum_i \mathrm{Tr}[\Pi \rho]$ is just the probability of a randomly chosen projector being violated by state ρ, so δ is a lower bound on the probability of seeing a violation in one iteration of Algorithm 2. Therefore, the expected number of violations after running the algorithm for t iterations is lower-bounded by $t\delta$. For sufficiently large t, this leads to a contradiction with Theorem 12. □

However, this argument is insufficient to prove the second part of Theorem 7: that there exists an *efficient* algorithm for approximating this state. To prove this, we construct a modified version of Algorithm 2, described in Algorithm 4, which essentially runs Algorithm 2 for a random number of iterations.

Algorithm 4 Quantum converger

```
function converge_qlll((Π_1, Π_2, ..., Π_m))
    Pick an integer 0 ≤ τ ≤ t uniformly at random;
    Initialise the assignment register as in Algorithm 2;
    i ← 0;
    loop t times
        if i < τ then
            Apply one iteration of Algorithm 2's main loop;
            i ← i + 1;
        end if
    end loop
    return assignment register;
end function
```

The following general theorem shows that the time required for Algorithm 4 to converge to the desired state is directly related to the expected number of violations seen by Algorithm 2. (Note that this theorem does *not* require commuting projectors.)

Theorem 13 *For an arbitrary set of m projectors $\{\Pi_i\}$, let E be the expected number of violations seen when Algorithm 2 is run with those projectors. If E is finite, then the state ρ produced by running Algorithm 4 for time $t = mE/\varepsilon$ satisfies $\forall i : \mathrm{Tr}[\Pi_i \rho] \leq \varepsilon$.*

Proof Let ρ_τ denote the final state of Algorithm 4's assignment register given that it picked the value τ. Note that this is the same as the state of Algorithm 2's assignment register after the τ^{th} iteration. Since τ is chosen at random, and we don't learn its value, the final state of Algorithm 4's assignment register is described by the density matrix

$$\rho = \frac{1}{t} \sum_{\tau=0}^{t} \rho_\tau. \tag{18}$$

Since ρ_τ is the same as the state of Algorithm 2's assignment register after τ iterations, the quantity $\frac{1}{m} \sum_i \mathrm{Tr}[\Pi_i \rho_\tau]$ is equal to the probability of seeing a violation in the $\tau + 1^{\text{th}}$ iteration of Algorithm 2. So the expected number of violations when Algorithm 2 runs for $t+1$ iterations is given by

$$E = \sum_{\tau=0}^{t} \Pr(\text{violation in } \tau^{\text{th}} \text{ iteration}) = \sum_{\tau=0}^{t} \left(\frac{1}{m} \sum_i \mathrm{Tr}[\Pi_i \rho_\tau] \right). \tag{19}$$

This together with Eq. (18) implies

$$\frac{1}{m}\sum_i \mathrm{Tr}[\Pi_i\rho] = \frac{1}{m}\sum_i \mathrm{Tr}\left[\Pi_i \cdot \frac{1}{t}\sum_{\tau=0}^{t}\rho_\tau\right] = \frac{1}{t}\sum_{\tau=0}^{t}\left(\frac{1}{m}\sum_i \mathrm{Tr}[\Pi_i\rho_\tau]\right) = \frac{E}{t}. \tag{20}$$

Since $\mathrm{Tr}[\Pi_i\rho]$ are positive, each term must be bounded by

$$\mathrm{Tr}[\Pi_i\rho] \leq \frac{mE}{t}, \tag{21}$$

thus running for time $t = mE/\varepsilon$ gives the desired bound $\mathrm{Tr}[\Pi_i\rho] \leq \varepsilon$. □

For commuting $\{\Pi_i\}$, we have a bound $E \leq \sum_i x_i/(1-x_i)$ from Theorem 12. Thus Theorems 12 and 13 together prove an efficient algorithm in the commutative case. Accounting for all parameters in the run-time analysis, note that Algorithm 2 requires $O(n)$ time to generate the initial assignment, and $O(|[i]|)$ time to resample $|[i]|$ qudits for each iteration in which projector Π_i was violated. Thus, if we run Algorithm 4 for $t = m/\varepsilon \sum_i x_i/(1-x_i)$ iterations, taking a total time of

$$O\left(n + \frac{m}{\varepsilon}\sum_{i=1}^{m}\frac{x_i}{1-x_i}\big|[i]\big|\right), \tag{22}$$

then Theorem 13 guarantees that the resulting state ρ will satisfy $\forall i : \mathrm{Tr}[\Pi_i\rho] \leq \varepsilon$.

To see that this is equivalent to the condition $\mathrm{Tr}[P_0\rho] \geq 1-\varepsilon$, where P_0 is the projector onto the subspace $\mathrm{span}\{|\psi\rangle : \forall i\, \Pi_i|\psi\rangle = 0\}$, note that for commuting projectors $P_0 = \mathbb{1} - \prod_{i=1}^m \Pi_i$. Thus

$$\mathrm{Tr}[P_0\rho] = \mathrm{Tr}\left[\left(\mathbb{1} - \prod_{i=1}^{m}\Pi_i\right)\rho\right] = 1 - \mathrm{Tr}\left[\prod_{i=1}^{m}\Pi_i\rho\Pi_i\right] \geq 1 - \mathrm{Tr}[\Pi_i\rho] \geq 1-\varepsilon, \tag{23}$$

where the second equality relies of the fact that the $\{\Pi_i\}$ mutually commute.

Thus the state ρ fulfils all the requirements of Theorem 7, thereby proving the constructive commutative Quantum Lovász Local Lemma.

3.4 An exact efficient quantum algorithm

The algorithm of the previous section constructs in polyomial-time a state that is ε-close to the one whose existence is asserted by the QLLL. This is the natural behaviour to demand of a constructive algorithm in the quantum setting. As we will see in Section 6, it generalises straightforwardly to the non-commutative case.

However, Moser [2009] and Moser and Tardos [2010] express the classical result slightly differently. They prove that their algorithm constructs the desired state *exactly*, in polynomial *expected* time.[1] It is not reasonable to demand this in the general quantum setting. But in the case of commuting projectors, this is a valid and equivalent way to formulate the constructive commutative QLLL. In this section, we use the results of the previous sections to give an algorithm that constructs the desired state exactly, with polynomial expected run-time.

The subspace of states fulfilling the QLLL requirement $\forall i : \Pi_i|\psi\rangle = 0$ can be characterised as the intersection of the supports (coimages) of the projectors $\mathbb{1} - \Pi_i$. Let S_i be the support of $P_i = \mathbb{1} - \Pi_i$. Then the desired subspace S_0 is $S = \bigcap_i^m S_i$, and P_0 is exactly the projector onto S_0. Given any state ρ with non-zero overlap with P_0, i.e. $\mathrm{Tr}[P_0\rho] > 0$, we can construct a state in P_0 by projecting onto P_0: $P_0\rho P_0/\mathrm{Tr}(P_0\rho)$. For commuting projectors Π_i, $P_0 = \prod_{i=1}^m(\mathbb{1} - \Pi_i)$. This motivates Algorithm 5, a slight modification of Algorithm 2 where we fix an order in which to apply the projectors. We immediately observe the following:

[1] Moser and Tardos [2010] also prove a deterministic version.

Lemma 14 *Let $P_0 \rho P_0$ be the* unnormalised *projection of ρ onto the subspace of states satisfying the QLLL requirement $\forall i : \Pi_i |\psi\rangle = 0$. Consider a run of Algorithm 5, starting from an initial state ρ with $\mathrm{Tr}[P_0\rho] > 0$. If no measurement has been violated throughout the entire run, then the assignment register contains the state $T\rho T$, where $T = P_m \cdot P_{m-1} \cdots P_2 \cdot P_1$.*

Algorithm 5 Exact Commutative QLLL solver

```
1:  procedure solve_qlll((Π_1, Π_2, ..., Π_m))
2:      for all x ∈ X do
3:          |α_i⟩ ← P
4:      end
5:      c ← 0
6:      loop until c = m or for a maximum of (m + 1)(pm' + 1) iterations
7:          Pick the next projector Π_i according to some fixed order;
8:          Measure Π_i;
9:          if the projector was violated then
10:             c ← 0;
11:             Append Π_i to the execution log;
12:             for all x ∈ [i] do
13:                 |α_i⟩ ← P;
14:             end
15:         else
16:             c ← c + 1;
17:         end if
18:     end loop
19:     if c = m then
20:         Return "success"
21:     else
22:         Return "failure"
23:     end if
24: end procedure
```

Quantum mechanics does not allow us to apply projector $T = P_m \cdot P_{m-1} \cdots P_2 \cdot P_1$ deterministically: some measurements will be violated with positive probability. But we know from Theorem 12, which applies equally well to Algorithm 5, that the expected number of violations is constant no matter how many iterations are performed in Algorithm 5. We can use this to upper-bound the expected number of iterations required to produce one contiguous run of all m projectors.

Lemma 15 *For any integer $p > 1$, the probability that Algorithm 5 returns "success" is at least $1 - 1/p$. In this case, the assignment register is in state $T\rho T / \mathrm{Tr}[T\rho T]$, where T is the desired sequence of projections.*

Proof From Theorem 12, we know that the expected number of violations is bounded from above by $m' = \sum_i x_i/(1 - x_i)$. Let M be the random variable counting the total number of violations. By Markov's inequality, $\Pr[M \leq pm'] \geq 1 - 1/p$. Thus, running the algorithm for $(m + 1)(pm' + 1)$ iterations, we will see with probability $1 - 1/p$ at most pm' undesired projections.

These pm' projections partition the execution log of length $(m + 1)(pm' + 1)$ into at most $pm' + 1$ stretches of desired projections. Thus at least one run of length m consisting *exclusively* of the desired projections must occur within any run of the algorithm with probability $1 - 1/p$. The termination condition of Algorithm 5 ensures that it halts after the first occurence of such a run. □

Combining Lemmas 14 and 15, we see that the algorithm must perform $O(pmm')$ iterations of which at most pm' lead to a resampling of $p\sum_{i=1}^{m}\frac{x_i}{1-x_i}\cdot\big|[i]\big|$ qudits, thus leading to an overall effort of

$$O\left(n+pm\sum_{i=1}^{m}\frac{x_i}{1-x_i}\cdot\big|[i]\big|\right), \tag{24}$$

where n reflects the cost for sampling the initial n random qudits. Note that we can easily recover Theorem 7 from this, by defining $p=1/\varepsilon$.

Assume we repeatedly run Algorithm 5 until it suceeds. By fixing $p>1$ to be some constant, we know from Lemma 15 that in each iteration Algorithm 5 succeeds with probability $1-1/p$, after which it stops, or fails with probability $1/p$ leading to another iteration. Thus this is a process of repeated Bernoulli trials until the first success with a geometric distribution $\Pr(X=k)=(1/p)(1-1/p)^{k-1}$, and expected termination time $\mathbb{E}(X)=p$. Thus, this process has an overall expected run-time of

$$O\left(n+m\sum_{i=1}^{m}\frac{x_i}{1-x_i}\cdot\big|[i]\big|\right), \tag{25}$$

and terminates only when it has applied a contiguous run of all m mutually-commuting projectors, which constructs the desired state exactly.

4 A Combinatorial Proof

In this section, we give an alternative proof of the key Theorem 12, which bounded the expected number of violations seen by Algorithm 2. In Section 3, we proved Theorem 12 using a quantum coupling argument. Here, we replace the coupling argument with a combinatorial one. The resulting proof is substantially more involved than the proof based on coupling, which attests to the power of even the simple quantum coupling argument used in Section 3. However, like many such proofs, the quantum coupling argument may look a little like "black magic". The alternative proof described here uses straightforward linear algebra and a slight generalisation of witness trees to directed acyclic graphs (DAG) to arrive at the same bound, providing different insight into how the QLLL algorithm works.

Moser and Tardos [2010] prove the constructive LLL by bounding the probability of observing a given sequence of violated events in Algorithm 1. The quantum generalisation of this type of algorithm is a quantum stochastic process generated by iterated measurement. Here, we prove a number of results concerning iterated quantum measurement processes, which may be of independent interest. We then re-prove Theorem 12 using these results.

4.1 A Simple Iterated Measurement Process

As a warm up, let us start with the following quantum stochastic process, which will serve as a building block for later, more complex processes.

Lemma 16 *Let $\{\Pi_i\}$ be a finite set of commuting projectors on a $\mathbb{C}^d$. Consider the quantum stochastic process produced by repeatedly picking one of the projectors uniformly at random, and performing the two-outcome measurement $\{\Pi_i, \mathbb{1}-\Pi_i\}$, starting from the maximally mixed state $\mathbb{1}/d$, until a measurement gives the Π outcome, whereupon the process halts.*

Let ρ_a be the final state of the system given that the process halted on outcome Π_a, and p_a the probability that this occurs. Then the unnormalised density operator $X_a=p_a\rho_a$ corresponding to this outcome satisfies the operator inequality

$$X_a = p_a\rho_a \leq \Pi_a/d, \tag{26}$$

hence

$$p_a \leq \mathrm{Tr}[\Pi_a/d]. \tag{27}$$

Proof Let m be the total number of projectors, so that in each iteration there is a probability $1/m$ of picking a given projector. The generalised measurement describing the different possible outcomes in a given iteration is described by a collection of trace non-increasing CP maps $\{\mathcal{M}_k\}$, such that $\sum_k \mathcal{M}_k$ is trace-preserving. To obtain the outcome Π_a in a given iteration, we must pick the projector Π_a in that iteration and also obtain the outcome Π_a in the resulting $\{\Pi_i, \mathbb{1} - \Pi_i\}$ measurement. This corresponds to the measurement element $\frac{1}{m}\mathcal{M}_a(\rho) = \frac{1}{m}\Pi_a \rho \Pi_a$.

In order to have reached that iteration, we must have obtained one of the $\mathbb{1} - \Pi$ outcomes in all previous iterations, otherwise the process would already have halted. The measurement element corresponding to the process *not* halting at a particular iteration is therefore given by $\mathcal{M}_{\text{cont}}(\rho) = \sum_i \frac{1}{m}(\mathbb{1} - \Pi_i)\rho(\mathbb{1} - \Pi_i)$. Thus the unnormalised density operator produced by the process halting with outcome Π_a in the $t + 1^{\text{th}}$ iteration is described by

$$\frac{1}{m}\mathcal{M}_a \circ \underbrace{\mathcal{M}_{\text{cont}} \circ \cdots \circ \mathcal{M}_{\text{cont}}}_{t}(\rho) = \frac{1}{m}\mathcal{M}_a \circ \mathcal{M}_{\text{cont}}^t(\mathbb{1}/d). \tag{28}$$

The process can halt after any number of iterations, so the overall unnormalised density operator corresponding to the process eventually halting on outcome Π_a is

$$\begin{aligned}
X_a &= \frac{1}{m}\sum_{t=0}^{\infty} \sum_{i_0,\ldots,i_t} \frac{1}{m^t}\Pi_a(\mathbb{1} - \Pi_{i_t}) \cdots (\mathbb{1} - \Pi_{i_0})\frac{\mathbb{1}}{d}(\mathbb{1} - \Pi_{i_0}) \cdots (\mathbb{1} - \Pi_{i_t})\Pi_a && \text{(29a)}\\
&= \frac{1}{m}\sum_{t=0}^{\infty} \sum_{i_0,\ldots,i_t \neq a} \frac{1}{m^t}\Pi_a(\mathbb{1} - \Pi_{i_t}) \cdots (\mathbb{1} - \Pi_{i_0})\frac{\mathbb{1}}{d}(\mathbb{1} - \Pi_{i_0}) \cdots (\mathbb{1} - \Pi_{i_t})\Pi_a && \text{(29b)}\\
&\leq \frac{1}{m}\sum_{t=0}^{\infty} \sum_{i_0,\ldots,i_t \neq a} \frac{1}{m^t}\Pi_a \frac{\mathbb{1}}{d}\Pi_a && \text{(29c)}\\
&= \frac{1}{m}\sum_{t=0}^{\infty} \left(\frac{m-1}{m}\right)^t \Pi_a \frac{\mathbb{1}}{d}\Pi_a && \text{(29d)}\\
&= \Pi_a/d && \text{(29e)}
\end{aligned}$$

as claimed. The equality in Eq. (29b) comes from commuting Π_a all the way through the product of $(\mathbb{1} - \Pi_i)$'s, killing all terms in the sum containing any $(1 - \Pi_a)$ factor. Thus the sum is only over the remaining $(m-1)^t$ terms. The inequality follows from the fact that, since all the $(\mathbb{1} - \Pi_i)$'s commute,

$$(\mathbb{1} - \Pi_{i_t}) \cdots (\mathbb{1} - \Pi_{i_0})\frac{\mathbb{1}}{d}(\mathbb{1} - \Pi_{i_0}) \cdots (\mathbb{1} - \Pi_{i_t}) \leq \frac{\mathbb{1}}{d}. \tag{30}$$

The final part of the lemma follows immediately from the fact that the probability of the process halting on outcome Π_a is given by $\text{Tr}[X_a]$. □

4.2 Resample DAGs

We now turn to a more complicated quantum stochastic process that produces an infinite sequence of measurement outcomes. For this we require some basic definitions.

Definition 17 (Resample DAG) *Let $\{\Pi_i\}$ be a finite set of projectors acting on a tensor product space $\bigotimes_i \mathcal{H}_i$, and let $\Pi_{a_1}, \Pi_{a_2}, \ldots, \Pi_{a_l}$ be a sequence of projectors chosen from this set. The* resample DAG $\mathfrak{G}(\Pi_{a_1}, \Pi_{a_2}, \ldots, \Pi_{a_l})$ *of such a sequence is the directed acyclic graph (DAG) with vertices labelled by sequence elements, and a directed edge from Π_{a_i} to Π_{a_j} iff the two projectors intersect and Π_{a_i} occurs* before *Π_{a_j}. (Thus the resample DAG encodes the partial ordering of the sequence with respect to projector intersections.)*

(Note that this is the same as the *"resample graph"* defined by Kolipaka and Szegedy [2011] for the classical setting.) $\mathfrak{G}(\Pi_{a_1}, \Pi_{a_2}, \ldots, \Pi_{a_l})$ can be constructed from the sequence as follows. Work *backwards* through the sequence, starting with the final projector Π_{a_l}. For each projector Π_{a_i}, find the set L_i of all vertices labelled by projectors that intersect with Π_{a_i} (which may be empty). Add a new vertex, labelling it by Π_{a_i}, and create directed edges from each element of L_i to the new vertex.

Definition 18 (DAG Probability) *Let $\{\Pi_i\}$ be a finite set of projectors acting on a tensor product space $\bigotimes_i \mathcal{H}_i$, and let $\mathfrak{G}$ be a resample DAG over these projectors. The* DAG probability *with respect to $\mathfrak{G}$ of the sequence $\Pi_{a_1}, \Pi_{a_2}, \ldots, \Pi_{a_l}$, denoted $p_{\mathfrak{G}}(\Pi_{a_1}, \Pi_{a_2}, \ldots, \Pi_{a_l})$, is the probability that the following process generates the sequence. Starting with the resample DAG $\mathfrak{G}$ and an empty initial sequence, pick one of the leaf vertices uniformly at random and append its label to the end of the sequence, removing the vertex from the DAG. Repeat this procedure until no vertices are left.*

4.3 A More Complicated Iterated Measurement Process

We are now in a position to study a more complicated quantum stochastic process, on which our constructive QLLL algorithm will ultimately be based.

Lemma 19 *Let $\{\Pi_i\}$ be a finite set of commuting projectors on an n-qudit Hilbert space $\bigotimes^n \mathcal{H}$ with total dimension d. Consider the quantum stochastic process which starts from the maximally mixed state $\mathbb{1}/d$, runs the process of Lemma 16 until it halts on some measurement outcome Π_a, then applies the trace-preserving CP map*

$$\mathcal{E}_a(\rho) = \mathrm{Tr}_{[a]}(\rho) \otimes \frac{\mathbb{1}_{[a]}}{d_{[a]}}, \tag{31}$$

which reinitialises the qudits measured by Π_a to the maximally mixed state, and repeats these two steps indefinitely. Here, $[a]$ denotes the subset of qudits on which projector Π_a acts non-trivially, and $d_{[a]}$ denotes the Hilbert space dimension of that subset.

Let $X_{a_1,a_2,\ldots,a_t}$ denote the unnormalised density operator corresponding to obtaining an outcome in which the first t iterations of the Lemma 16 sub-process halted in turn on the sequence of measurement outcomes $\Pi_{a_1}, \Pi_{a_2}, \ldots, \Pi_{a_t}$. Then $X_{a_1,a_2,\ldots,a_t}$ satisfies the operator inequality

$$X_{a_1,a_2,\ldots,a_t} \leq p_{\mathfrak{G}}(\Pi_{a_1}, \Pi_{a_2}, \ldots, \Pi_{a_t}) \prod_{i=1}^{t} \mathrm{Tr}[\Pi_{a_i}/d] \cdot \frac{\mathbb{1}}{d}, \tag{32}$$

and the probability of this occurring is bounded by

$$\Pr(\Pi_{a_1}, \Pi_{a_2}, \ldots, \Pi_{a_t}) \leq p_{\mathfrak{G}}(\Pi_{a_1}, \Pi_{a_2}, \ldots, \Pi_{a_t}) \prod_{i=1}^{t} \mathrm{Tr}[\Pi_{a_i}/d], \tag{33}$$

where the resample DAG $\mathfrak{G} = \mathfrak{G}(\Pi_{a_1}, \Pi_{a_2}, \ldots, \Pi_{a_t})$.

Before proving this lemma, we first establish some basic identities concerning compositions of the various maps involved in Lemmas 16 and 19.

Lemma 20 *Let the CP maps $\mathcal{M}_a$, $\mathcal{M}_{\mathrm{cont}}$ and $\mathcal{E}_a$ be defined as in Lemmas 16 and 19:*

$$\mathcal{M}_a(\rho) = \Pi_a \rho \Pi_a, \tag{34}$$

$$\mathcal{M}_{\mathrm{cont}}(\rho) = \sum_i \frac{1}{m} (\mathbb{1} - \Pi_i) \rho (\mathbb{1} - \Pi_i), \tag{35}$$

$$\mathcal{E}_a(\rho) = \mathrm{Tr}_{[a]}(\rho) \otimes \frac{\mathbb{1}_{[a]}}{d_{[a]}}. \tag{36}$$

In addition, for disjoint Π_a, Π_b*, define the CP maps*

$$\mathcal{M}_\Sigma = \sum_{t=0}^{\infty} \mathcal{M}_{\text{cont}}^t, \tag{37}$$

$$\mathcal{M}_{a,b}(\rho) = (\Pi_a \otimes \Pi_b)\rho(\Pi_a \otimes \Pi_b), \tag{38}$$

$$\mathcal{E}_{a,b}(\rho) = \mathcal{E}_a \circ \mathcal{E}_b(\rho) = \text{Tr}_{[a]\cup[b]}(\rho) \otimes \frac{\mathbb{1}_{[a]\cup[b]}}{d_{[a]\cup[b]}}. \tag{39}$$

Recall that all projectors Π_i *in these definitions are assumed to commute. The following relations hold:*

(i). $\frac{1}{m}\mathcal{M}_{a_1,\ldots,a_k} \circ \mathcal{M}_\Sigma(\mathbb{1}/d) \leq \frac{1}{k}\mathcal{M}_{a_1,\ldots,a_k}(\mathbb{1}/d)$ *for disjoint* Π_{a_i},

(ii). $\mathcal{E}_a \circ \mathcal{M}_a(\mathbb{1}/d) = \text{Tr}[\Pi_a/d]\ \frac{\mathbb{1}}{d}$.

(iii). $\mathcal{M}_a = \mathcal{M}_a \circ \mathcal{M}_a$,

(iv). $\mathcal{M}_a \circ \mathcal{M}_b = \mathcal{M}_b \circ \mathcal{M}_a = \mathcal{M}_{a,b}$ *for disjoint* Π_a, Π_b,

(v). $\mathcal{E}_a \circ \mathcal{E}_b = \mathcal{E}_b \circ \mathcal{E}_a = \mathcal{E}_{a,b}$ *for disjoint* Π_a, Π_b,

(vi). $\mathcal{M}_a \circ \mathcal{E}_b = \mathcal{E}_b \circ \mathcal{M}_a$ *for disjoint* Π_a, Π_b,

(vii). $\mathcal{M}_a \circ \mathcal{M}_\Sigma = \mathcal{M}_\Sigma \circ \mathcal{M}_a$,

(When we write $\mathcal{M} \leq \mathcal{N}$ for CP maps $\mathcal{M}, \mathcal{N}$, we mean that $\mathcal{M}(X) \leq \mathcal{N}(X)$ for any positive operator X.)

Proof To prove Part (i), we start by applying almost exactly the same argument as in the proof of Lemma 16, replacing Π_a with $\bigotimes_{i=1}^k \Pi_{a_i}$ and noting that this tensor product of projectors kills k terms from the sum in Eq. (29b), rather than just the one in the proof of Lemma 16. Thus

$$\begin{aligned} \mathcal{M}_{a_1,\ldots,a_k} \circ \mathcal{M}_{\text{cont}}^t(\mathbb{1}/d) &\leq \frac{1}{m}\sum_{t=0}^{\infty} \sum_{i_0,\ldots,i_t \neq a} \frac{1}{m^t} \left(\bigotimes_{i=1}^k \Pi_{a_i}\right) \frac{\mathbb{1}}{d} \left(\bigotimes_{i=1}^k \Pi_{a_i}\right) \\ &= \left(\frac{m-k}{m}\right)^t \mathcal{M}_{a_1,\ldots,a_k}(\mathbb{1}/d) \end{aligned} \tag{40}$$

and

$$\frac{1}{m}\mathcal{M}_{a_1,\ldots,a_k} \circ \mathcal{M}_\Sigma(\mathbb{1}/d) = \frac{1}{m}\sum_{t=0}^{\infty} \mathcal{M}_{a_1,\ldots,a_k} \circ \mathcal{M}_{\text{cont}}^t(\mathbb{1}/d) \tag{41a}$$

$$\leq \frac{1}{m}\sum_{t=0}^{\infty} \left(\frac{m-k}{m}\right)^t \mathcal{M}_{a_1,\ldots,a_k}(\mathbb{1}/d) \tag{41b}$$

$$= \frac{1}{k}\mathcal{M}_{a_1,\ldots,a_k}(\mathbb{1}/d). \tag{41c}$$

Part (ii) follows immediately from the definitions of $\mathcal{E}_a$ and $\mathcal{M}_a$. Part (iii) is immediate from $\Pi_a^2 = \Pi_a$. Parts (iv) and (v) follow trivially from the fact that two disjoint projectors by definition only act non-trivially on different subsystems. Similarly, in Part (vi), since Π_a and Π_b are disjoint, the partial trace in $\mathcal{E}_b$ is over a subsystem on which the projector Π_a in $\mathcal{M}_a$ acts trivially. Finally, Part (vii) follows from the fact that all the Π_i commute. □

We are now in a position to prove Lemma 19.

Proof (of Lemma 19) The unnormalised density operator $X_{\Pi_{a_1},\Pi_{a_2},\ldots,\Pi_{a_t}}$ corresponding to the process of Lemma 19 producing the sequence $\Pi_{a_1},\Pi_{a_2},\ldots,\Pi_{a_t}$ is given by

$$X_{\Pi_{a_1},\Pi_{a_2},\ldots,\Pi_{a_t}} = \mathcal{E}_{a_t}\circ\tfrac{1}{m}\mathcal{M}_{a_t}\circ\mathcal{M}_\Sigma\circ\mathcal{E}_{a_{t-1}}\circ\tfrac{1}{m}\mathcal{M}_{a_{t-1}}\circ\mathcal{M}_\Sigma\circ\cdots\circ\mathcal{E}_{a_1}\circ\tfrac{1}{m}\mathcal{M}_{a_1}\circ\mathcal{M}_\Sigma(\mathbb{1}/d). \quad (42)$$

We start by using Part (iii) of Lemma 20 to duplicate every $\mathcal{M}_{a_i}$ which can be commuted all the way through the expression to right using Parts (iv), (vi) and (vii). This will produce a string of $\mathcal{M}_{a_i}$'s of the form $\mathcal{M}_{a_i}\circ\mathcal{M}_{a_j}\circ\cdots\circ\mathcal{M}_{a_1}$ next to the inner-most $\mathcal{M}_{a_1}$. Since each $\mathcal{M}_{a_j}$ is preceded by a $\mathcal{E}_{a_j}$ in Eq. (42), and $\mathcal{M}_{a_i}$ can only be commuted through $\mathcal{E}_{a_j}$ if Π_{a_i} and Π_{a_j} are disjoint, all $\mathcal{M}_{a_i}$'s in this string necessarily have disjoint Π_{a_i}, so we can combine them into a single $\mathcal{M}_{a_i,a_j,\ldots,a_1}$ using Part (iv) of Lemma 20.

At this point, the expression has the form

$$X_{\Pi_{a_1},\Pi_{a_2},\ldots,\Pi_{a_t}} = \mathcal{E}_{a_t}\circ\tfrac{1}{m}\mathcal{M}_{a_t}\circ\mathcal{M}_\Sigma\circ\mathcal{E}_{a_{t-1}}\circ\tfrac{1}{m}\mathcal{M}_{a_{t-1}}\circ\mathcal{M}_\Sigma\circ\cdots\circ\mathcal{E}_{a_1}\circ\tfrac{1}{m}\mathcal{M}_{a_i,a_j,\ldots,a_1}\circ\mathcal{M}_\Sigma(\mathbb{1}/d). \quad (43)$$

We can now use Part (i) of Lemma 20 to eliminate the inner-most $\mathcal{M}_\Sigma$:

$$\begin{aligned}&\mathcal{E}_{a_t}\circ\tfrac{1}{m}\mathcal{M}_{a_t}\circ\mathcal{M}_\Sigma\circ\mathcal{E}_{a_{t-1}}\circ\tfrac{1}{m}\mathcal{M}_{a_{t-1}}\circ\mathcal{M}_\Sigma\circ\cdots\circ\mathcal{E}_{a_1}\circ\tfrac{1}{m}\mathcal{M}_{a_i,a_j,\ldots,a_1}\circ\mathcal{M}_\Sigma(\mathbb{1}/d)\\ &\quad\le \mathcal{E}_{a_t}\circ\tfrac{1}{m}\mathcal{M}_{a_t}\circ\mathcal{M}_\Sigma\circ\mathcal{E}_{a_{t-1}}\circ\tfrac{1}{m}\mathcal{M}_{a_{t-1}}\circ\mathcal{M}_\Sigma\circ\cdots\circ\mathcal{E}_{a_1}\circ\tfrac{1}{k}\mathcal{M}_{a_i,a_j,\ldots,a_1}(\mathbb{1}/d), \quad (44)\end{aligned}$$

where k is the number of Π_{a_i}'s in the $\mathcal{M}_{a_i,a_j,\ldots,a_1}$.

Now we reverse the above steps. First we separate out $\mathcal{M}_{a_i,a_j,\ldots,a_1}$ again into $\mathcal{M}_{a_i}\circ\mathcal{M}_{a_j}\circ\cdots\circ\mathcal{M}_{a_1}$ using Part (iv) of Lemma 20. Then we use Parts (iv), (vi) and (vii) to move all the separated $\mathcal{M}_{a_i}$ (except $\mathcal{M}_{a_1}$) back through the expression to the left, until they can be recombined using Part (iii) with the $\mathcal{M}_{a_i}$ that they were originally duplicated from. We are left with

$$X_{\Pi_{a_1},\Pi_{a_2},\ldots,\Pi_{a_t}} \le \mathcal{E}_{a_t}\circ\tfrac{1}{m}\mathcal{M}_{a_t}\circ\mathcal{M}_\Sigma\circ\mathcal{E}_{a_{t-1}}\circ\tfrac{1}{m}\mathcal{M}_{a_{t-1}}\circ\mathcal{M}_\Sigma\circ\cdots\circ\mathcal{E}_{a_1}\circ\tfrac{1}{k}\mathcal{M}_{a_1}(\mathbb{1}/d). \quad (45)$$

We now use Part (ii) of Lemma 20 to arrive at

$$X_{\Pi_{a_1},\Pi_{a_2},\ldots,\Pi_{a_t}} \le \frac{1}{k}\operatorname{Tr}[\Pi_{a_i}/d]\cdot\mathcal{E}_{a_t}\circ\tfrac{1}{m}\mathcal{M}_{a_t}\circ\mathcal{M}_\Sigma\circ\mathcal{E}_{a_{t-1}}\circ\tfrac{1}{m}\mathcal{M}_{a_{t-1}}\circ\mathcal{M}_\Sigma\circ\cdots\circ\mathcal{E}_{a_2}\circ\tfrac{1}{m}\mathcal{M}_{a_2}\circ\mathcal{M}_\Sigma(\mathbb{1}/d). \quad (46)$$

Comparing with Eq. (42), we see that, in this way, we have eliminated the inner-most $\mathcal{E}_{a_1}\circ\frac{1}{m}\mathcal{M}_{a_1}\circ\mathcal{M}_\Sigma$ terms from the expression, picking up a scalar factor of $1/k$ in the process which depends on how many $\mathcal{M}_{a_i}$ could be commuted through far enough to reach the inner-most $\mathcal{M}_{a_1}$.

The sequence of CP maps on the right hand side of Eq. (46) now has exactly the same form as the original Eq. (42), but shortened by omiting the a_1 terms. Thus we have obtained a recursive operator inequality for $X_{\Pi_{a_1},\Pi_{a_2},\ldots,\Pi_{a_t}}$:

$$X_{\Pi_{a_1},\Pi_{a_2},\ldots,\Pi_{a_t}} \le \frac{1}{k}\operatorname{Tr}[\Pi_{a_i}/d]\cdot X_{\Pi_{a_2},\ldots,\Pi_{a_t}}. \quad (47)$$

Applying this inequality recursively, we finally arrive at

$$X_{\Pi_{a_1},\Pi_{a_2},\ldots,\Pi_{a_t}} \le \frac{1}{\mathcal{P}}\cdot\prod_{i=1}^{t}\operatorname{Tr}[\Pi_i\rho]\cdot\frac{\mathbb{1}}{d}, \quad (48)$$

with a yet-to-be-determined combinatorial coefficient $1/\mathcal{P}$ arising from the $1/k$ coefficients introduced by Part (i) of Lemma 20.

It remains to determine the combinatorial coefficient $1/\mathcal{P}$. This is a product of all the $1/k$ coefficients contributed by using Part (i) of Lemma 20 to eliminate an $\mathcal{M}_\Sigma$. Now, each $\mathcal{M}_\Sigma$ is eliminated by a map of the form $\mathcal{M}_{a_\alpha,a_\beta,\dots}$ (cf. Eq. (44)). Let $S = \{\Pi_{a_i}, \Pi_{a_j}, \dots\}$ be the set of elements from the original sequence $\Pi_{a_1}, \Pi_{a_2}, \dots, \Pi_{a_t}$ that are involved in the map $\mathcal{M}_{a_\alpha,a_\beta,\dots}$ that eliminates the $\mathcal{M}_\Sigma$ in question. Then the elimination of $\mathcal{M}_\Sigma$ by $\mathcal{M}_{a_\alpha,a_\beta,\dots}$ contributes a factor of $1/|S|$. Thus, to understand the combinatorial coefficient $1/\mathcal{P}$, it suffices to understand all the sets S. It will be useful in what follows to consider both the subset S of elements from the original sequence, and the set of vertices $V(S)$ from the resample DAG $\mathfrak{G} = \mathfrak{G}(\Pi_{a_1}, \Pi_{a_2}, \dots, \Pi_{a_t})$ that are labelled by the elements of S.

Each $\mathcal{M}_\Sigma$ appears immediately to the right of a particular $\mathcal{M}_{a_i}$ in the original expression of Eq. (42), which in turn corresponds to a particular element Π_{a_i} of the original sequence. Thus each $\mathcal{M}_\Sigma$ can be identified with a unique element Π_{a_i} of the sequence. Consider a particular $\mathcal{M}_\Sigma$, and let Π_{a_s} be the element of the original sequence with which it is identified. Now, the map $\mathcal{M}_{a_\alpha,a_\beta,\dots}$ that eliminates this $\mathcal{M}_\Sigma$ is made up of all those $\mathcal{M}_{a_{i>s}}$ lying to the *left* of $\mathcal{M}_{a_s}$ in the original expression Eq. (42) which could be commuted past all the intervening maps using Parts (iv), (vi) and (vii) of Lemma 20. Note that $\mathcal{M}_{a_i}$ can only be moved past a $\mathcal{E}_{a_j}$ if Π_{a_i} is disjoint from Π_{a_j}. Thus we can determine the set S corresponding to the $\mathcal{M}_{a_\alpha,a_\beta,\dots}$ that eliminates a given $\mathcal{M}_\Sigma$ just by looking at the original sequence alone.

S can be found by the following procedure, which simply mirrors on the level of the sequence the steps involved in eliminating the $\mathcal{M}_\Sigma$. (Cf. the reduction of Eq. (42) to Eq. (46).) Start with the final element of the sequence,[1] and move it as far as possible towards the beginning of the sequence, subject to the rule that it can only be moved past an element if it is disjoint from that element. Repeat this procedure once[2] for each element $\Pi_{i>s}$ that appears later in the sequence than Π_{a_s}, starting from the final element Π_{a_t} working backwards through $\Pi_{a_{i>s}}$ until Π_{a_s}. S is then the set of all elements that have been moved to the left of Π_{a_s}, including Π_{a_s} itself (but not including any elements $\Pi_{a_{i<s}}$, which were never moved).

How do the elements of S relate to vertices in the resample DAG $\mathfrak{G}$? If any descendant of Π_{a_i} appears in between Π_{a_s} and Π_{a_i} in the sequence, then Π_{a_i} cannot be in S. To see this, note that by Definition 17 all ancestors of an element must appear later in the sequence than that element.[3] Therefore, if Π_{a_j} is a descendant of Π_{a_i}, then there is at least one child (possibly Π_{a_j} itself) of Π_{a_i} in between Π_{a_s} and Π_{a_i}. But by Definition 17 Π_{a_i} intersects with all its children, so it cannot be moved past this child.

Thus no element of S has any descendant appearing between it and Π_{a_s}. But, as we've already noted, any descendant of an element must appear earlier in the sequence than that element. Therefore, no element of S has any descendants appearing *anywhere* after Π_{a_s} in the sequence. Conversely, if none of the descendents of an element Π_{a_i} appear after Π_{a_s} in the sequence, then Π_{a_i} is in S. This is because the only thing that could stop Π_{a_i} from being moved past Π_{a_s} is a projector with which it intersects appearing between it and Π_{a_s}, but by Definition 17 any such projector will be a descendant of Π_{a_i}.

S is therefore the set of all elements appearing after Π_{a_s} in the sequence which have no descendants amongst the elements $\Pi_{a_{i>s}}$. In other words, if we consider the resample DAG $\mathfrak{G}_s(\Pi_{a_s}, \Pi_{a_{s+1}}, \dots, \Pi_{a_t})$ for the subsequence $\Pi_{a_s}, \Pi_{a_{s+1}}, \dots, \Pi_{a_t}$, then the elements of S are the leaf vertices of $\mathfrak{G}_s$. Note that $\mathfrak{G}_s$ can be obtained from the full resample DAG $\mathfrak{G}(\Pi_{a_s}, \Pi_{a_{s+1}}, \dots, \Pi_{a_t})$ by removing the vertices labelled by $\Pi_{a_1}, \dots, \Pi_{a_{s-1}}$.

Recall that the elimination of the $\mathcal{M}_\Sigma$ corresponding to Π_{a_s} contributes a factor $1/|S|$ to

[1] Recall that the maps $\mathcal{M}_{a_i}$ in the composition of Eq. (42) are in reverse order to the corresponding elements Π_{a_i} in the sequence.

[2] Once we have finished moving a particular element, we are not allowed to move it again later, even if subsequent moves have "unblocked" it.

[3] Recall from Definition 17 that the resample DAG is constructed by working *backwards* through the sequence.

the combinatorial coefficient $1/\mathcal{P}$, and we have seen that S is the set of all leaves remaining in the resample DAG after vertices labelled by elements preceding Π_{a_s} have been removed. So $1/|S|$ is just the probability of removing Π_{a_s} when a leaf is picked uniformly at random and removed, given that vertices labelled by elements preceding Π_{a_s} have already been removed from the resample DAG. There is one such factor $1/|S|$ contributed by each element Π_{a_s} of the sequence, so the overall combinatorial coefficient $1/\mathcal{P}$ in Eq. (48), which is the product of all these $1/|S|$ factors, is precisely the DAG probability of Definition 18. This completes the proof of Eq. (32).

The probability bound of Eq. (33) follows immediately from Eq. (32) and the fact that the probability of the obtaining the sequence corresponding to the unnormalised density operator $X_{a_1,a_2,\ldots,a_t}$ is given by $\mathrm{Tr}[X_{a_1,a_2,\ldots,a_t}]$. □

Lemma 19 immediately implies a simple bound on the probability of the measurement outcomes from the Lemma 19 process forming a given resample DAG.

Corollary 21 *Let G be a fixed resample DAG with vertices $v \in V(G)$ labelled by Π_v, and let $\Pi_{a_1}, \Pi_{a_2}, \ldots, \Pi_{a_t}$ be the first t measurement outcomes produced by the Lemma 19 process. Then*

$$\Pr\left[\mathfrak{G}(\Pi_{a_1}, \Pi_{a_2}, \ldots, \Pi_{a_t}) = G\right] \leq \prod_{v \in V(G)} \mathrm{Tr}[\Pi_v \rho]. \tag{49}$$

Proof To stand any chance of producing the DAG G, we must have $\{\Pi_{a_1}, \ldots, \Pi_{a_t}\} = \{\Pi_v\}_{v \in V(G)}$. The probability of producing a given resample DAG G is obtained by summing the probabilities of all distinct sequences that generate that DAG. Let

$$\mathcal{S} = \{s = \Pi_{a_{\sigma(1)}}, \Pi_{a_{\sigma(2)}}, \ldots, \Pi_{a_{\sigma(t)}} \mid \mathcal{G}(s) = G\} \tag{50}$$

be the set of all sequences with resample DAG G. Note that this is just the set of all permutations σ of $\{\Pi_v\}$ consistent with the partial order encoded by G.

Now, every sequence $s \in \mathcal{S}$ can be generated with probability $p_G(s)$ by running the process of Definition 18 on G; conversely, every sequence generated by running that process on G is in $\mathcal{S}$. Since the process always generates *some* sequence, we have

$$\sum_{s \in \mathcal{S}} p_G(s) = 1. \tag{51}$$

Thus, summing the probabilities from Eq. (33) of Lemma 19, we have

$$\begin{aligned} \Pr\left[\mathfrak{G}(\Pi_{a_1}, \Pi_{a_2}, \ldots, \Pi_{a_t}) = G\right] &= \sum_{s:\mathfrak{G}(s)=G} \Pr(s) \leq \sum_{s \in \mathcal{S}} p_G(s) \cdot \prod_{i=1}^{t} \mathrm{Tr}[\Pi_{a_i} \rho] \\ &= \prod_{v \in G} \mathrm{Tr}[\Pi_v \rho] \cdot \sum_{s \in \mathcal{S}} p_G(s) = \prod_{v \in G} \mathrm{Tr}[\Pi_v \rho]. \end{aligned} \tag{52}$$

4.4 Partial Resample DAGs

Later on, we will need to consider a variant of the resample DAG, constructed almost exactly as in Definition 17, but with one key difference. Instead of including all projectors from the sequence in the DAG, we only include those that are reachable from the vertex labelled by the final projector in the sequence; i.e. any projector that cannot be attached to an existing vertex when constructing the DAG (cf. Definition 17) is simply discarded.

Definition 22 (Partial Resample DAG) *Let $\{\Pi_i\}$ be a finite set of projectors acting on a tensor product space $\bigotimes_i \mathcal{H}_i$, and let $\Pi_{a_1}, \Pi_{a_2}, \ldots, \Pi_{a_l}$ be a sequence of projectors chosen from this set. The* partial resample DAG *is the subgraph of the resample DAG $\mathfrak{G}(\Pi_{a_1}, \Pi_{a_2}, \ldots, \Pi_{a_l})$ which consisting of the subset of vertices that are reachable from the vertex labelled by Π_{a_l}, and all edges that begin and end at vertices in this subset.*

The *partial resample DAG* $\mathfrak{g}(\Pi_{a_1}, \Pi_{a_2}, \ldots, \Pi_{a_l})$ can be constructed from the sequence as follows. Start with the final projector Π_{a_l} in the sequence. Create a root vertex and label it with Π_{a_l}. Then, working *backwards* through the sequence, for each projector $\Pi_{a_{i<l}}$ find the set L_i of all vertices labelled by projectors that intersect with Π_{a_i}. If L_i is empty, skip this projector. Otherwise, create a new vertex, labelling it by Π_{a_i}, and attach it to the DAG by creating directed edges from each element of L_i to the new vertex.

Definition 23 (Relevant Subsequence) *Let $\Pi_{a_1}, \Pi_{a_2}, \ldots, \Pi_{a_t}$ be a sequence of projectors, and $\mathfrak{g}(\Pi_{a_i}, \Pi_{a_2}, \ldots, \Pi_{a_t})$ the corresponding partial resample DAG. The* relevant subsequence *of $\Pi_{a_1}, \Pi_{a_2}, \ldots, \Pi_{a_t}$ is the subsequence obtained by discarding from the sequence all elements that get discareded when $\mathfrak{g}(\Pi_{a_i}, \Pi_{a_2}, \ldots, \Pi_{a_t})$ is constructed according to Definition 24. Note that two sequences with the same relevant subsequence necessarily generate the same partial resample DAG.*

Definition 24 (Partial DAG Probability) *Let $\{\Pi_i\}$ be a finite set of projectors acting on a tensor product space $\bigotimes_i \mathcal{H}_i$, and let $\mathfrak{g}$ be a partial resample DAG over these projectors. The* partial DAG probability *with respect to $\mathfrak{g}$ of the sequence $\Pi_{a_1}, \Pi_{a_2}, \ldots, \Pi_{a_l}$, denoted $p_{\mathfrak{g}}(\Pi_{a_1}, \Pi_{a_2}, \ldots, \Pi_{a_l})$, is the probability that the following process generates the sequence. Starting with the partial resample DAG $\mathfrak{g}$ and an empty initial sequence, pick one of the leaf vertices uniformly at random and append its label to the end of the sequence, removing the vertex from the DAG. Repeat this procedure until no vertices are left.*

The analogous result to Corollary 21 also holds for partial resample DAGs.

Lemma 25 *Let g be a fixed partial resample DAG with vertices $v \in V(g)$ labelled by projectors Π_v, and let $\Pi_{a_1}, \Pi_{a_2}, \ldots, \Pi_{a_t}$ be the first length-t relevant subsequence of outcomes from the Lemma 19 process. Then*

$$\Pr[\mathfrak{g}(\Pi_{a_1}, \Pi_{a_2}, \ldots, \Pi_{a_t}) = g] \leq \prod_{v \in V(g)} \mathrm{Tr}[\Pi_v \rho]. \tag{53}$$

Proof The proof is similar to the proofs of Lemma 19 and Corollary 21, but some of the outcomes generated by the Lemma 19 process might not be irrelevant because they get discarded when constructing a partial resample DAG. We will first bound the probability of a given sequence $\Pi_{a_1}, \Pi_{a_2}, \ldots, \Pi_{a_t}$ forming the first length-t relevant subsequence. We can then sum over all relevant subsequences that generate the specified partial resample DAG g to bound the overall probability of producing g.

Previously, the map $\mathcal{M}_\Sigma$ allowed for any number of $\mathbb{1} - \Pi$ measurement outcomes in Lemma 19. We must now also allow for any number of irrelevant outcomes. A projector is discarded if and only if it does not intersect with any projector that occurs *later* in the sequence.[1] So, between any two elements $\Pi_{a_{i-1}}$ and Π_{a_i} of the sequence $\Pi_{a_1}, \Pi_{a_2}, \ldots, \Pi_{a_t}$, we can have any number of outcomes Π_x such that Π_x is disjoint from all $\Pi_{a_{j>i}}$, as well as any number of $\mathbb{1} - \Pi$ outcomes.

In other words, if $\mathcal{X}_i$ is the set of irrelevant outcomes that will be discarded from the DAG if they occur between elements $\Pi_{a_{i-1}}$ and Π_{a_i} in the sequence, then the generalised measurement element corresponding to a single iteration of the Lemma 16 sub-process *not* producing a relevant outcome is given by

$$\begin{aligned} \mathcal{M}^{(i)}_{\mathrm{cont}}(\sigma) &= \sum_j \frac{1}{m}(\mathbb{1} - \Pi_j)\sigma(\mathbb{1} - \Pi_j) + \sum_{\Pi_a \in \mathcal{X}_i} \mathcal{E}_a \circ \tfrac{1}{m}\mathcal{M}_a(\sigma) \\ &= \frac{1}{m} \sum_{\Pi_a \notin \mathcal{X}_i} (\mathbb{1} - \Pi_a)\sigma(\mathbb{1} - \Pi_a) + \frac{1}{m} \sum_{\Pi_a \in \mathcal{X}_i} \mathcal{T}_a(\sigma) \end{aligned} \tag{54}$$

[1] Once again, recall that the partial resample DAG is constructed *backwards*.

where the trace-preserving CP map

$$\begin{aligned}\mathcal{T}_a(\sigma) &= (\mathbb{1}-\Pi_a)\sigma(\mathbb{1}-\Pi_a) + \mathcal{E}_a \circ \mathcal{M}_a \\ &= (\mathbb{1}-\Pi_a)\sigma(\mathbb{1}-\Pi_a) + \rho_{[a]} \otimes \mathrm{Tr}_{[a]}(\Pi_a \sigma \Pi_a).\end{aligned} \tag{55}$$

The map corresponding to any number of irrelevant outcomes occurring is then given by

$$\mathcal{M}_\Sigma^{(i)}(\sigma) = \sum_{t=0}^{\infty} (\mathcal{M}_{\mathrm{cont}}^{(i)})^t(\sigma). \tag{56}$$

We now prove a version of Part (i) of Lemma 20 for the map $\mathcal{M}_\Sigma^{(i)}$.

Lemma 26 *Let $\Pi_{a_\alpha}, \Pi_{a_\beta}, \ldots$ be disjoint from each other* and *from all $\Pi_x \in \mathcal{X}_i$, and let $\mathrm{Tr}_{[\mathcal{X}_i]}$ denote the partial trace over all qudits on which some $\Pi_x \in \mathcal{X}_i$ acts non-trivially. Then*

$$\mathrm{Tr}_{[\mathcal{X}_i]}\left[\tfrac{1}{m}\mathcal{M}_{a_\alpha,a_\beta,\ldots} \circ (\mathcal{M}_\Sigma^{(i)})^t(\mathbb{1}/d)\right] \leq \frac{1}{k}\,\mathrm{Tr}_{[\mathcal{X}_i]}[\mathcal{M}_{a_\alpha,a_\beta,\ldots}(\mathbb{1}/d)]. \tag{57}$$

Proof First, note that, since $\Pi_{a_1}, \ldots, \Pi_{a_k}$ are disjoint from each other and from Π_x, we have

$$\begin{aligned}\mathrm{Tr}_{[x]}\left[\mathcal{M}_{a_\alpha,a_\beta,\ldots} \circ \mathcal{T}_x(\sigma)\right] &= \mathrm{Tr}_{[x]}\left[\mathcal{T}_x \circ \mathcal{M}_{a_\alpha,a_\beta,\ldots}(\sigma)\right] \\ &= \mathrm{Tr}_{[x]}\left[\mathcal{M}_{a_\alpha,a_\beta,\ldots}(\sigma)\right],\end{aligned} \tag{58}$$

since $\mathcal{T}_x$ is trace-preserving and only acts non-trivially on $[X]$.

Now,

$$\begin{aligned}&\mathrm{Tr}_{[\mathcal{X}_i]}\left[\mathcal{M}_{a_\alpha,a_\beta,\ldots} \circ (\mathcal{M}_{\mathrm{cont}}^{(i)})^t(\mathbb{1}/d)\right] \\ &\qquad = \mathrm{Tr}_{[\mathcal{X}_i]}\Bigg[\mathcal{M}_{a_\alpha,a_\beta,\ldots}\Bigg(\frac{1}{m}\sum_{\Pi_a \notin \mathcal{X}_i} (\mathbb{1}-\Pi_a)(\mathcal{M}_{\mathrm{cont}}^{(i)})^{t-1}(\mathbb{1}/d)(\mathbb{1}-\Pi_a)\Bigg) \\ &\qquad\qquad + \frac{1}{m}\sum_{\Pi_a \in \mathcal{X}_i} \mathcal{M}_{a_\alpha,a_\beta,\ldots} \circ \mathcal{T}_a \circ (\mathcal{M}_{\mathrm{cont}}^{(i)})^{t-1}(\mathbb{1}/d)\Bigg].\end{aligned} \tag{59}$$

Since $\Pi_{a_\alpha}, \Pi_{a_\beta}, \ldots$ are disjoint from all $\Pi_a \in \mathcal{X}_i$, all the terms involving projectors that intersect with $\Pi_{a_\alpha}, \Pi_{a_\beta}, \ldots$ occur in the first sum. The first sum can therefore be treated exactly as in Eq. (40) from Lemma 20, giving

$$\begin{aligned}&\mathrm{Tr}_{[\mathcal{X}_i]}\left[\mathcal{M}_{a_\alpha,a_\beta,\ldots}\left(\frac{1}{m}\sum_{\Pi_a \notin \mathcal{X}_i} (\mathbb{1}-\Pi_a)(\mathcal{M}_{\mathrm{cont}}^{(i)})^{t-1}(\mathbb{1}/d)(\mathbb{1}-\Pi_a)\right)\right] \\ &\qquad\qquad \leq \mathrm{Tr}_{[\mathcal{X}_i]}\left[\frac{m-|\mathcal{X}_i|-k}{m}\,\mathcal{M}_{a_\alpha,a_\beta,\ldots} \circ (\mathcal{M}_{\mathrm{cont}}^{(i)})^{t-1}(\mathbb{1}/d)\right].\end{aligned} \tag{60}$$

Using Eq. (58), the second sum simplifies to

$$\begin{aligned}&\mathrm{Tr}_{[\mathcal{X}_i]}\left[\frac{1}{m}\sum_{\Pi_a \in \mathcal{X}_i} \mathcal{M}_{a_\alpha,a_\beta,\ldots} \circ \mathcal{T}_a \circ (\mathcal{M}_{\mathrm{cont}}^{(i)})^{t-1}(\mathbb{1}/d)\right] \\ &\qquad = \mathrm{Tr}_{[\mathcal{X}_i]}\left[\frac{1}{m}\sum_{\Pi_a \in \mathcal{X}_i} \mathcal{M}_{a_\alpha,a_\beta,\ldots} \circ (\mathcal{M}_{\mathrm{cont}}^{(i)})^{t-1}(\mathbb{1}/d)\right] &\text{(61a)} \\ &\qquad = \mathrm{Tr}_{[\mathcal{X}_i]}\left[\frac{|\mathcal{X}_i|}{m}\mathcal{M}_{a_\alpha,a_\beta,\ldots} \circ (\mathcal{M}_{\mathrm{cont}}^{(i)})^{t-1}(\mathbb{1}/d)\right]. &\text{(61b)}\end{aligned}$$

Putting these together, we obtain

$$\mathrm{Tr}_{[\mathcal{X}_i]}\left[\mathcal{M}_{a_\alpha,a_\beta,\ldots}\circ(\mathcal{M}_{\mathrm{cont}}^{(i)})^t(\mathbb{1}/d)\right]\leq\mathrm{Tr}_{[\mathcal{X}_i]}\left[\frac{m-k}{m}\mathcal{M}_{a_\alpha,a_\beta,\ldots}\circ(\mathcal{M}_{\mathrm{cont}}^{(i)})^{t-1}(\mathbb{1}/d)\right]. \tag{62}$$

By induction on t (the base case $t=0$ is trivial) and summing over t, we obtain the identity in the lemma. □

Proof (of Lemma 25, continued)
The probability of obtaining $\Pi_{a_1},\Pi_{a_2},\ldots,\Pi_{a_t}$ for the first t relevant outcomes is given by

$$\begin{aligned}\Pr(\Pi_{a_1},\Pi_{a_2},\ldots,\Pi_{a_t})&\\ =\mathrm{Tr}\Big[\mathcal{E}_{a_t}\circ\tfrac{1}{m}\mathcal{M}_{a_t}\circ\mathcal{M}_\Sigma^{(t)}\circ\mathcal{E}_{a_{t-1}}\circ\tfrac{1}{m}\mathcal{M}_{a_{t-1}}\circ\mathcal{M}_\Sigma^{(t-1)}\circ\cdots&\\ \cdots\circ\mathcal{E}_{a_1}\circ\tfrac{1}{m}\mathcal{M}_{a_1}\circ\mathcal{M}_\Sigma^{(1)}(\rho)\Big].&\end{aligned} \tag{63}$$

(Compare with Eq. (42) from Lemma 19. The $\mathcal{M}_\Sigma^{(i)}$ now also account for outcomes that are discarded when constructing the partial resample DAG.)

We eliminate the $\mathcal{M}_\Sigma^{(i)}$ from this expression in much the same way as in Lemma 19, using Lemma 26 instead of Lemma 20, Part (i). Since a projector is discarded when constructing the DAG if and only if it is disjoint from *all* projectors that occur *later* in the sequence, they do not affect which $\mathcal{M}_{a_i}$'s can be commuted all the way to the right, so iterating the same sequence of steps again leads to a recursive operator inequality of the form

$$X_{\Pi_{a_1},\Pi_{a_2},\ldots,\Pi_{a_t}}\leq\frac{1}{k}\,\mathrm{Tr}[\Pi_{a_i}/d]\cdot X_{\Pi_{a_2},\ldots,\Pi_{a_t}}. \tag{64}$$

The only difference as compared to the inequality in Eq. (47) lies in the combinatorial factor $1/k$, which now only counts relevant measurement outcomes.

The combinatorial factors arising from Lemma 26 are otherwise identical to those of Lemma 20, Part (i), leading to an overall factor given this time by the *partial* DAG probability $p_{\mathfrak{g}}(\Pi_{a_1},\Pi_{a_2},\ldots,\Pi_{a_t})$ with $\mathfrak{g}=\mathfrak{g}(\Pi_{a_1},\Pi_{a_2},\ldots,\Pi_{a_t})$. Thus we finally end up with the following bound on the probability of obtaining $\Pi_{a_1},\Pi_{a_2},\ldots,\Pi_{a_t}$ as the first t outcomes that are not discarded from the partial resample DAG:

$$\Pr(\Pi_{a_1},\Pi_{a_2},\ldots,\Pi_{a_t})\leq p_{\mathfrak{g}}(\Pi_{a_1},\Pi_{a_2},\ldots,\Pi_{a_t})\cdot\prod_{i=1}^{t}\mathrm{Tr}[\Pi_{a_i}\rho]. \tag{65}$$

As in Corollary 21, when we sum over all relevant subsequences that generate the specified partial resample DAG g, the combinatorial coefficients p_g sum to 1 and we arrive at the bound claimed in the lemma. □

4.5 Expected number of violations

We are now in a position to re-prove the bound of Theorem 12 on the expected number of violations seen by Algorithm 2. Recall that Algorithm 2 keeps a log of violated measurements $\Pi_{a_1},\Pi_{a_2},\ldots,\Pi_{a_t}$, whose only purpose is to aid in the analysis. We say that a given partial resample DAG g *occurs* in Algorithm 2's log if it can be constructed starting from some entry of the log, i.e. if for some l $\mathfrak{g}(\Pi_{a_1},\ldots,\Pi_{a_l})=g$.

The process implemented by Algorithm 2 is just the iterated measurement process defined in Lemma 19. So Lemma 25 immediately implies the following result:

Corollary 27 *The probability that the partial resample DAG g occurs in the log is at most $\prod_{v \in V(g)} R(\Pi_v)$, where $V(g)$ is the vertex set of g, Π_v is the projector labelling vertex v, and the relative dimension $R(\Pi_v)$ is equivalent to the probability of measuring Π_v on the maximally mixed state.*

(Though proved by very different means, Corollary 27 is essentially a quantum version of Lemma 2.1 from Moser and Tardos [2010], though that Lemma is not quite a special case of Corollary 27 as it concerns "witness trees" rather than partial resample DAGs.)

In order to bound the expected number of violations seen by Algorithm 2, we again follow Moser and Tardos [2010] in relating Corollary 27 to a Galton-Watson branching process. Notice that the bounds in Corollaries 21 and 27 are determined solely by the set of vertex labels appearing in the DAG; the structure of the DAG plays no role. Let $\tau(g)$ be a (labelled) spanning tree for the partial resample DAG g. Because it is the spanning tree for an resample DAG, all the children of a vertex in $\tau(g)$ carry distinct labels (cf. Definition 22). Labelled trees with this property will be called *proper*, irrespective of whether they are spanning trees for some DAG; all spanning trees are proper, but not all proper trees are spanning trees.

We will once again relate the bound in Corollary 27 to the probability that the Galton-Watson process of Lemma 11 generates a proper tree τ_a whose root vertex is labelled by some fixed projector Π_a, yielding an alternative proof of Theorem 12.

Proof (of Theorem 12) Let N_a be the number of times a given projector Π_a appears in Algorithm 2's log, which is the same as the number of times Π_a was measured to be violated. Let $\mathcal{G}_a$ denote the set of all partial resample DAGs whose root vertex is labelled by Π_a, and let $\mathcal{T}_a$ denote the set of all proper trees with root vertex labelled by Π_a. Then, from Corollary 27, we have

$$\begin{aligned} \mathbb{E}(N_a) &= \sum_{g_a \in \mathcal{G}_a} \Pr(g_a \text{ occurs in the log}) \leq \sum_{g_a \in \mathcal{G}_a} \prod_{v \in V(g_a)} R(\Pi_v) \\ &= \sum_{g_a \in \mathcal{G}_a} \prod_{v \in V(\tau(g_a))} R(\Pi_v) \leq \sum_{\tau_a \in \mathcal{T}_a} \prod_{v \in V(\tau_a)} R(\Pi_v). \end{aligned} \tag{66}$$

The final relation holds because distinct resample DAGs have distinct spanning trees. It is an inequality for two reasons: a DAG can have multiple spanning trees, leading to double-counting, and the set of proper trees is a strict superset of the set of spanning trees.

Now, by the assumption of the QLLL (Theorem 7), the relative dimension satisfies $R(\Pi_i) \leq x'_i$. Thus

$$\mathbb{E}(N_a) \leq \sum_{\tau_a \in \mathcal{T}_a} \prod_{v \in V(\tau_a)} R(\Pi_v) \leq \sum_{\tau_a \in \mathcal{T}_a} \prod_{v \in V(\tau_a)} x'_v \leq \sum_{\tau_a \in \mathcal{T}_a} \frac{x_a}{1 - x_a} \Pr(\tau_a), \tag{67}$$

the final inequality following from Lemma 11, $\Pr(\tau_a)$ being the probability of the Galton-Watson process generating tree τ_a. Since that process either produces a proper tree in $\mathcal{T}_a$, or continues indefinitely, we have $\sum_{\tau_a \in \mathcal{T}_a} \Pr(\tau_a) \leq 1$, thus

$$\mathbb{E}(N_a) \leq \frac{x_a}{1 - x_a} \sum_{\tau_a \in \mathcal{T}_a} \Pr(\tau_a) \leq \frac{x_a}{1 - x_a}, \tag{68}$$

and the theorem follows from summing over all projectors. □

Using this alternate proof of Theorem 12, the rest of the proof of the constructive commutative QLLL (Theorem 7) goes through as before using the arguments of Sections 3.3 and 3.4.

5 Application: Bounding Convergence Times of CP Maps

A (time-homogeneous) *quantum Markov process* is generated by iterating a fixed completely positive trace-preserving (CP) map $\mathcal{E}$, so that t steps of the process are described by the composition

$$\mathcal{E}^t = \underbrace{\mathcal{E} \circ \mathcal{E} \circ \cdots \circ \mathcal{E}}_{t}. \tag{69}$$

As in the case of (classical) Markov chains, convergence times of quantum stochastic processes are an important field of study, with applications to quantum dynamics [Aharonov et al., 2001, Cubitt et al., 2009, Sanz et al., 2009], quantum information theory [Sanz et al., 2009], and quantum algorithms [Schwarz et al., 2011, Temme et al., 2011, Verstraete et al., 2009].

The results of Section 3 prove that Algorithm 4 converges in polynomial time to a state satisfying the requirements of the commutating QLLL, or equivalently the ground state subspace of the associated Hamiltonian. This holds promise as a new technique for proving fast convergence of quantum Markov processes to their steady state. However, strictly speaking Algorithm 4 does not implement a quantum Markov process. Although each iteration of Algorithm 2, on which it is based, does implement a fixed CP map $\mathcal{E}$, the averaging trick used in Algorithm 4 means that the latter algorithm does not simply iterate this map. (The overall evolution of Algorithm 4 for a given total time t can of course be described by some CP map, but this map is not of the form $\mathcal{E}^t$.) Algorithm 5 is further still from being a quantum Markov process.

In this section, we instead focus on the quantum Markov process defined by Algorithm 2, and prove that if the set of projectors defining the process satisfy the Lovász conditions of Theorem 7, then this process converges quickly to the steady-state subspace (in time polynomial in the overlap with the steady-state subspace, and polynomial in the spectral gap of the associated Hamiltonian).

Proving fast convergence is slightly more involved than in the case of Algorithm 4, as we cannot use the averaging trick of Section 3.3, but instead must combine the fact that the overlap with the steady-state subspace is monotonic under the CP map, with Poissonian accumulation of the probability of measuring a violation, leading to a contradiction with Theorem 12 were the convergence rate too slow.

Theorem 28 *For a given set of commuting projectors $\Pi_1, \Pi_2, \ldots, \Pi_m$, let $\mathcal{E}$ be the CP map defined by*[1]

$$\mathcal{E}(\rho) = \frac{1}{m} \sum_i (\mathbb{1} - \Pi_i)\rho(\mathbb{1} - \Pi_i) + \mathrm{Tr}_{/[i]}[\Pi_i \rho] \otimes \frac{\mathbb{1}_{[i]}}{d^k}. \tag{70}$$

If the projectors satisfy the Lovász conditions of Theorem 7, then the Markov process produced by iterating this map converges in $O(m/\delta^2 \varepsilon)$ iterations to a state ρ with fidelity $\mathrm{Tr}[P_0 \rho] \geq 1 - \varepsilon$ with the steady-state subspace of $\mathcal{E}$ (where P_0 is the projector onto that subspace, and δ is the spectral gap of $H = \frac{1}{m} \sum_i \Pi_i$).

Note, that according to the definition above the spectral gap $\delta = 1/m$ in the case of m commuting projectors considered here.

Proof Note that Eq. (70) is precisely the map implemented by one iteration of Algorithm 2. If the projectors satisfy the Lovász conditions, then we know from Theorem 3 that P_0 is non-zero. Let $\rho_t = \mathcal{E}^t(\rho_0)$ be the state of Algorithm 2's assignment register at time t. We can decompose ρ_t in terms of P_0:

$$\rho_t = P_0 \rho_t P_0 + P_0 \rho_t (\mathbb{1} - P_0) + (\mathbb{1} - P_0)\rho_t P_0 + (\mathbb{1} - P_0)\rho_t(\mathbb{1} - P_0). \tag{71}$$

[1] This is the same as the map shown in Verstraete et al. [2009] to converge to the ground state of any frustration-free Hamiltonian, but with no control on the convergence rate.

Note that $P_0\rho_t P_0$ is an (unnormalised) state supported on P_0, the cross-terms are traceless, and $(\mathbb{1}-P_0)\rho_t(\mathbb{1}-P_0)$ is an (unnormalised) state supported on $\mathbb{1}-P_0$.

Since $P_0\rho_t P_0$ is supported on the subspace of states that do not violate any projector, it is invariant under $\mathcal{E}$, thus

$$\begin{aligned}\operatorname{Tr}[P_0\mathcal{E}(\rho_t)] &= \operatorname{Tr}[P_0\rho_t] + \operatorname{Tr}[P_0\mathcal{E}(P_0\rho_t(\mathbb{1}-P_0))] + \operatorname{Tr}[P_0\mathcal{E}((\mathbb{1}-P_0)\rho_t P_0)] \\ &\quad + \operatorname{Tr}[P_0\mathcal{E}((\mathbb{1}-P_0)\rho_t(\mathbb{1}-P_0))] \qquad (72a)\\ &\geq \operatorname{Tr}[P_0\rho_t], \qquad (72b)\end{aligned}$$

where the inequality follows from dropping non-negative terms. $\operatorname{Tr}[P_0\rho_t]$ is therefore monotonically non-decreasing under $\mathcal{E}$.

Let $\alpha \coloneqq \operatorname{Tr}[P_0\rho_t]$. The probability of a randomly chosen projector being violated by ρ_t is given by $\frac{1}{m}\sum_i \operatorname{Tr}[\Pi_i\rho_t]$. Since $\Pi_i P_0 = 0$ by definition, we have

$$\begin{aligned}\frac{1}{m}\sum_i \operatorname{Tr}[\Pi_i\rho_t] &= \frac{1}{m}\sum_i \Big(\operatorname{Tr}[\Pi_i P_0\rho_t P_0] + \operatorname{Tr}[\Pi_i P_0\rho_t(\mathbb{1}-P_0)] \qquad (73a)\\ &\qquad + \operatorname{Tr}[\Pi_i(\mathbb{1}-P_0)\rho_t P_0] + \operatorname{Tr}[\Pi_i(\mathbb{1}-P_0)\rho_t(\mathbb{1}-P_0)]\Big) \\ &= \frac{1}{m}\sum_i \operatorname{Tr}[\Pi_i(\mathbb{1}-P_0)\rho_t(\mathbb{1}-P_0)] \qquad (73b)\\ &\leq \operatorname{Tr}[(\mathbb{1}-P_0)\rho_t] \qquad (73c)\\ &= 1-\alpha, \qquad (73d)\end{aligned}$$

where the inequality follows from the fact that $\Pi_i \leq \mathbb{1}$.

Recall that the spectral gap δ is defined as

$$\delta = \min_{\rho\,:\,P_0\rho=0} \frac{1}{m}\sum_i \operatorname{Tr}[\Pi_i\rho]. \qquad (74)$$

Note that $\delta > 0$, since $\delta = 0$ would imply that $\operatorname{Tr}[\Pi_i\psi] = 0$ for all ψ, Π_i which is impossible except in the trivial case of all $\Pi_i = 0$. Then

$$\begin{aligned}\frac{1}{m}\sum_i \operatorname{Tr}[\Pi_i\rho_t] &= \frac{1}{m}\sum_i \Big(\operatorname{Tr}[\Pi_i P_0\rho_t] + \operatorname{Tr}[\Pi_i P_0\rho_t(\mathbb{1}-P_0)] \qquad (75a)\\ &\qquad + \operatorname{Tr}[\Pi_i(\mathbb{1}-P_0)\rho_t P_0] + \operatorname{Tr}[\Pi_i(\mathbb{1}-P_0)\rho_t(\mathbb{1}-P_0)]\Big) \\ &= \operatorname{Tr}[(\mathbb{1}-P_0)\rho_t]\frac{1}{m}\sum_i \operatorname{Tr}\left[\Pi_i\frac{(\mathbb{1}-P_0)\rho_t(\mathbb{1}-P_0)}{\operatorname{Tr}[(\mathbb{1}-P_0)\rho_t]}\right] \qquad (75b)\\ &\geq (1-\alpha)\,\delta. \qquad (75c)\end{aligned}$$

Since $\alpha = \operatorname{Tr}[P_0\rho_t]$ is non-decreasing with t, this also holds for all $\rho_{\tau<t}$.

So, the probability of a randomly chosen projector being violated by ρ_t is upper-bounded by $1-\alpha$, and the probability of a randomly chosen projector being violated by $\rho_{\tau<t}$ is lower-bounded by $(1-\alpha)\delta$. Imagine that we iterate the process for t iterations. Given that M violations occur, what is the probability that one of the violations occurs in the final (t^{th}) iteration? From Eqs. (73) and (75), the worst-case distribution of violations is given by

$$\begin{aligned}\Pr(\text{violation in } t^{\text{th}} \text{ iteration}) &\propto 1-\alpha, \qquad (76a)\\ \Pr(\text{violation in } \tau < t^{\text{th}} \text{ iteration}) &\propto (1-\alpha)\,\delta. \qquad (76b)\end{aligned}$$

The probability of a violation in the final iteration is then equivalent to the probability of picking the t^{th} element, when we pick M from t elements according to the distribution of

Eq. (76). By the union bound, this is at most $M \times \Pr(\text{pick } t^{\text{th}} \text{ element})$. Normalising the distribution of Eq. (76), we have

$$\alpha = 1 - \frac{1}{t\delta + 1}, \tag{77}$$

so

$$\Pr(\text{violation in } t^{\text{th}} \text{ iteration} \mid M \text{ violations}) \leq \frac{M}{t\delta + 1}. \tag{78}$$

We have a bound on the expected number of violations from Theorem 12. Using this together with Eq. (78), we have

$$\Pr(\text{violation in } t^{\text{th}} \text{ iteration}) \leq \sum_M \frac{M}{t\delta + 1} \Pr(M \text{ violations}) = \frac{\mathbb{E}(M)}{t\delta + 1} \leq \frac{m}{t\delta + 1}. \tag{79}$$

From Eq. (75), we know that $\Pr(\text{violation in } t^{\text{th}} \text{ iteration}) \geq (1 - \alpha)\, \delta$. Together with Eq. (79), this implies

$$\mathrm{Tr}[P_0 \rho_t] =: \alpha \geq 1 - \frac{m}{t\delta(\delta + 1)} = 1 - O\left(\frac{m}{t\delta^2}\right). \tag{80}$$

Therefore, in order to achieve an overlap $\mathrm{Tr}[P_0 \rho_t] \geq 1 - \varepsilon$ with the steady-state subspace, we require $t = O(m/\delta^2 \varepsilon)$ iterations. □

Since the map $\mathcal{E}$ of Theorem 28 implements exactly the dissipative state engineering map of Verstraete et al. [2009], Theorem 28 implies the following result about the convergence of this map.

Corollary 29 *Consider the many-body Hamiltonian $H = \sum_{i=1}^m h_i$ on n particles, where each local term h_i acts on some subset of the particles, and all the h_i mutually commute. Let Π_i be the projector onto the orthogonal complement of the lowest-energy eigenspace of h_i.*

If the projectors satisfy the Lovász conditions of Theorem 7, then the CP map $\mathcal{E}$ defined in Theorem 28 will converge in time $O(m/\delta^2 \varepsilon)$ to a state with fidelity $1 - \varepsilon$ with the ground state subspace of H, where δ is the spectral gap of the Hamiltonian $H_\Pi = \frac{1}{m} \sum_i \Pi_i$ in which the local terms are replaced by projectors.

Note that the convergence time is *independent* of the number of particles and of their local Hilbert space dimension.

6 The Non-Commutative Case

In the previous sections, we have overcome the first challenge to proving a constructive QLLL: that of efficiently constructing the desired entangled state using only local measurements. To prove a constructive version of the general QLLL of Ambainis et al. [2009], we must overcome the second challenge: non-commutativity of the projectors, and the concomitant measurement-disturbance problem.

In the non-commutative setting, there is a new parameter that can play a role in the run-time of the QLLL algorithm: the spectral gap of the Hamiltonian associated with a QLLL instance.

Definition 30 (QLLL spectral gap) *The* spectral gap δ *of a set of projectors $\{\Pi_i\}$ is the spectral gap of the associated Hamiltonian*[1]

$$H = \frac{1}{m} \sum_i \Pi_i, \tag{81}$$

[1] Note our choice of normalisation, which follows that of Ambainis, Kempe, and Sattath [2009], but differs from some papers in the literature by a factor of m.

i.e. the difference between the smallest and second-smallest eigenvalues (ignoring degeneracies).

If the projectors satisfy the Lovász conditions (Definition 6), then the minimum eigenvalue of H is 0 and the spectral gap can be expressed as

$$\delta = \min_{\rho \,:\, P_0\rho=0} \frac{1}{m} \sum_i \mathrm{Tr}[\Pi_i \rho], \tag{82}$$

where P_0 is the projector on the subspace of states fulfilling the QLLL requirements, i.e. the projector onto $\mathrm{span}\{|\psi\rangle : \forall i\, \Pi_i |\psi\rangle = 0\}$.

Note that Eq. (82) can be interpreted as the probability of measuring a randomly chosen projector to be violated, minimised over all states in the orthogonal complement of the subspace of states that do not violate any projector.

6.1 Witness trees

The proof of the key Lemma 8 for the commutative case in Section 3.1, leading to the bound in Theorem 12 on the expected number of violations seen by Algorithm 2, made crucial use of commutativity of the $\{\Pi_i\}$. (Commutativity is also necessary for the alternative combinatorial proof given in Section 4.) In principle, Lemma 8 and hence Theorem 12 could still hold in the non-commutative case, without necessarily implying the same run-time as in the commutative case. We will see in Section 6.3 that, depending on the type of convergence we demand, the spectral gap enters naturally in the required run-time in the non-commutative setting, through using a bound on the expected number of violations to prove a bound the rate of convergence to the desired state.

However, there are strong indications that the bound in Lemma 8 part (ii) on the probability of a witness tree τ occurring in Algorithm 2's log no longer holds for non-commutative projectors. The proof of Lemma 8 still goes through in the commutative setting if, instead of constructing a witness tree by discarding any projectors that do not intersect with one already in the tree, we instead include these projectors, thereby constructing a DAG instead of a tree. This still encodes the partial ordering of the violations with respect to projector intersections, so the coupling argument in Section 3.1 still works, giving the analogous bound on the probability of these DAGs occurring in the log (still in the commutative case). Indeed, we proved exactly this result for resample DAGs in Corollary 21, by more involved linear algebra and combinatorial techniques.

But the following counter-example shows that this bound on the probability of a given DAG occurring in the log *no longer holds* if the projectors do not commute.

Example 31 Consider a system of two qubits. Let $\Pi_1 = |0\rangle\langle 0| \otimes \mathbb{1}$, $\Pi_2 = \mathbb{1} \otimes |0\rangle\langle 0|$, and $\Pi_3 = |\psi\rangle\langle\psi| + |01\rangle\langle 01| + |10\rangle\langle 10|$, where $|\psi\rangle = \sqrt{a}\,|00\rangle + \sqrt{b}\,|11\rangle$. Let τ be the graph consisting of two (unconnected) vertices labelled by Π_1 and Π_2.

If Corollary 21 or (the generalisation of) Lemma 8 were true in the non-commutative case, the probability of τ occurring in Algorithm 2's log would be bounded by

$$\Pr(\tau) \leq \mathrm{Tr}(\Pi_1/d)\,\mathrm{Tr}(\Pi_2/d) = 1/4. \tag{83}$$

There are two ways τ can occur: either the sequence Π_1, Π_2 occured at the beginning of the log, or the same violations occurred in the other order Π_2, Π_1. The probability of these two sequences is the same, by symmetry. Direct analytical calculation gives

$$\Pr(\tau) = 2\Pr(\Pi_1, \Pi_2) = \begin{cases} \frac{1}{9} + \frac{7a}{24(1+a)} + \frac{b(11+12a)}{144(1+a)^2} & \text{if } a < 1 \\ \frac{1}{9} & \text{if } a = 1, \end{cases} \tag{84}$$

which violates the bound in Eq. (83) for $a > \frac{3}{20}(\sqrt{41} - 1)$, with the probability tending to $\frac{37}{144}$ as $a \to 1$ (a violation of $\frac{1}{144}$).

6.1.1 A Conjecture

Nevertheless, we conjecture that a similar bound does still hold in general, weakened by a polynomial factor.

Conjecture 1 *Let $\{\Pi_i\}$ be an arbitrary set of projectors, let τ be a fixed witness tree with vertices labelled by these projectors, and L the log produced by running Algorithm 2 with these projectors. We conjecture that the probability that τ occurs in L is at most*

$$\Pr(\tau) \leq \text{poly}(|\tau|) \prod_{v \in \tau} \Pr[\Pi(v)], \tag{85}$$

where $|\tau|$ is the size of τ (total number of vertices).

(We implicitly also allow here a polynomial dependence on the other parameters n, m, d, x_i and $1/\delta$. Only the dependence on τ plays a role in the subsequent proof; any dependence on the other parameters carries straight through to the final run-time.)

In Section 6.2, we will see that this conjecture relates the probability of a tree occurring in the general quantum case to the expectation value of a functional over the *total progeny* of the Galton-Watson process described in Section 3.2. In the classical and commutative cases, we saw that this conjecture held for the trivial constant functional $\text{poly}(|\tau|) = 1$.

For the simplest case of trees containing a single vertex, we can prove Conjecture 1 even *without* the polynomial factor (i.e. with $\text{poly}(|\tau|) = 1$). The single-vertex case is not completely trivial, as an arbitrary number of satisfied measurements can occur before Algorithm 2 sees its first violation. In fact, we will prove a slightly stronger *operator* bound, which immediately implies the conjecture for single-vertex trees. We already proved this result for commuting projectors in Lemma 16. The following proposition generalises that result to arbitrary sets of projectors.

Proposition 32 *Let $\{\Pi_i\}$ be a set of m projectors on $\mathbb{C}^d$. Consider the following iterated measurement process, starting from the maximally mixed state. In each step, a projector Π_i is chosen independently uniformly at random, and the two-outcome measurement $\{\Pi_i, \mathbb{1} - \Pi_i\}$ is performed. This is repeated until a Π_i outcome is obtained, at which point the process halts.*

Let ρ_a denote the final state of the system given that it halted on outcome Π_a, and p_a the probability that this occurs. Then the (unnormalised) density matrix corresponding to the process halting on outcome Π_a satisfies

$$X_a = p_a \rho_a \leq \Pi_a / d, \tag{86a}$$

hence

$$p_a = \text{Tr}(X_a) \leq \text{Tr}(\Pi_a)/d. \tag{86b}$$

This result is quite striking. It tells us that we if start from the maximally mixed state and perform binary, two-outcome projective measurements at random until we obtain the first "0" outcome, we can effectively ignore all intermediate measurements withboutcome "1", and replace this process with that of simply measuring the final outcome on the maximally mixed state directly.

Proof We will in fact prove a slightly more general result. Let A^{-1} denote the Moore-Penrose pseudoinverse of a matrix A, and A^* denote entry-wise complex conjugation. The operation $vec(A)$ or $|A\rangle$ denotes vectorisation of the matrix A, i.e. the treatment of matrix $A \in \mathcal{M}_d(\mathbb{C})$ as a vector in $\mathbb{C}^{d^2}$. By a slight abuse of notation, we sometimes write $|X\rangle \leq |Y\rangle$ to mean $X \leq Y$.

We can express the unnormalised density matrix X_a as

$$X_a = \frac{1}{m}\Pi_a \sum_{t=0}^{\infty} \left(\sum_{i_1,\ldots,i_t} \frac{1}{m}(\mathbb{1} - \Pi_{i_t}) \cdots \left(\sum_{i_1} \frac{1}{m}(\mathbb{1} - \Pi_{i_1}) \frac{\mathbb{1}}{d} (\mathbb{1} - \Pi_{i_1}) \right) \cdots (\mathbb{1} - \Pi_{i_t}) \right) \Pi_a, \tag{87}$$

or, vectorising,

$$|X_a\rangle = \frac{1}{m}[\Pi_a^* \otimes \Pi_a] \sum_{t=0}^{\infty} \left(\frac{1}{m} \sum_i (\mathbb{1} - \Pi_i^*) \otimes (\mathbb{1} - \Pi_i) \right)^t \frac{1}{d} |\mathbb{1}\rangle . \tag{88}$$

But this is a Neumann series $\sum_{t=0}^{\infty} T^t = (\mathbb{1} - T)^{-1}$. Thus the infinite sum can be written as

$$|X_a\rangle = \frac{1}{m}[\Pi_j^* \otimes \Pi_j] \left(\mathbb{1} - \frac{1}{m} \sum_i (\mathbb{1} - \Pi_i^*) \otimes (\mathbb{1} - \Pi_i) \right)^{-1} \frac{1}{d} |\mathbb{1}\rangle \tag{89}$$

$$= (\Pi_j^* \otimes \Pi_j) \left(\sum_i (\mathbb{1} \otimes \Pi_i) + (\Pi_i^* \otimes \mathbb{1}) - (\Pi_i^* \otimes \Pi_i) \right)^{-1} \frac{1}{d} |\mathbb{1}\rangle . \tag{90}$$

For a more compact notation, define $Q = \sum_i \Pi_i$, $B_i = \Pi_i^* \otimes \Pi_i$, $B = \sum_i B_i$, and $A = \sum_i (\mathbb{1} \otimes \Pi_i + \Pi_i^* \otimes \mathbb{1}) = Q^* \otimes \mathbb{1} + \mathbb{1} \otimes Q$. Then the claim of the Proposition is equivalent to

$$B_i (A - B)^{-1} |\mathbb{1}\rangle \leq B_i |\mathbb{1}\rangle . \tag{91}$$

We will prove a slightly more general result. Let $P = \Pi_{i_1} \Pi_{i_2} \cdots \Pi_{i_k}$ be a product of k *commuting* projectors chosen from $\{\Pi_i\}$, and define $B_P = P^* \otimes P$. We claim that

$$B_P (A - B)^{-1} |\mathbb{1}\rangle \leq \left(\frac{1}{2} + \frac{1}{2k} \right) B_P |\mathbb{1}\rangle , \tag{92}$$

the Proposition being equivalent to the $k = 1$ case of this. To show this, we need the following lemmas, which we prove below.

Lemma 33 *$PQ^{-1}P \leq \frac{1}{k} P$.*

Lemma 34 *$A^{-1} |\mathbb{1}\rangle = \frac{1}{2} |Q^{-1}\rangle + |Q_\perp\rangle$, where $Q_\perp$ is an operator whose support (coimage) lies in the kernel of Q.*

Lemma 35 *$B_P A^{-1} |\mathbb{1}\rangle \leq \frac{1}{2k} B_P |\mathbb{1}\rangle$, and in particular $BA^{-1} |\mathbb{1}\rangle \leq \frac{1}{2} B |\mathbb{1}\rangle$.*

Lemma 36 *$B_P A^{-1} B |\mathbb{1}\rangle = \frac{1}{2} B_P |\mathbb{1}\rangle$, and in particular $BA^{-1}B |\mathbb{1}\rangle = \frac{1}{2} B |\mathbb{1}\rangle$.*

Lemma 37 *Let $t \geq 1$ be an integer. Then $(BA^{-1})^t |\mathbb{1}\rangle \leq \frac{1}{2^t} B |\mathbb{1}\rangle$.*

Eq. (92) now follows from

$$B_P (A - B)^{-1} |\mathbb{1}\rangle = B_P (A(\mathbb{1} - A^{-1}B))^{-1} |\mathbb{1}\rangle = B_P (\mathbb{1} - A^{-1}B)^{-1} A^{-1} |\mathbb{1}\rangle \tag{93a}$$

$$= B_P \sum_{t=0}^{\infty} (A^{-1}B)^t A^{-1} |\mathbb{1}\rangle = B_P A^{-1} \sum_{t=0}^{\infty} (BA^{-1})^t |\mathbb{1}\rangle \tag{93b}$$

$$= B_P A^{-1} |\mathbb{1}\rangle + B_P A^{-1} \sum_{t=1}^{\infty} (BA^{-1})^t |\mathbb{1}\rangle \tag{93c}$$

$$\leq B_P A^{-1} |\mathbb{1}\rangle + B_P A^{-1} \sum_{t=1}^{\infty} \frac{1}{2^t} B |\mathbb{1}\rangle \tag{93d}$$

$$= B_P A^{-1} |\mathbb{1}\rangle + B_P A^{-1} B |\mathbb{1}\rangle \leq \frac{1}{2k} B_P |\mathbb{1}\rangle + \frac{1}{2} B_P |\mathbb{1}\rangle \tag{93e}$$

$$= \left(\frac{1}{2} + \frac{1}{2k} \right) B_P |\mathbb{1}\rangle , \tag{93f}$$

where the first equality in Eq. (93a) follows as B is in the support of A, Eq. (93d) from Lemma 37, and the inequality in Eq. (93e) from Lemma 35 and Lemma 36. □

We now prove the five lemmas used above.

Proof (of Lemma 33) First note that

$$\sum_i \Pi_i \geq k\, \Pi_{i_1} \Pi_{i_2} \cdots \Pi_{i_k} \tag{94}$$

because

$$\sum_i \Pi_i - k\, \Pi_{i_1} \Pi_{i_2} \cdots \Pi_{i_k} = \Pi_{i_1} + \Pi_{i_2} + \Pi_{i_k} + \sum_{j \notin \{i_1,\ldots,i_k\}} \Pi_j - k \Pi_{i_1} \Pi_{i_2} \cdots \Pi_{i_k} \tag{95}$$

$$\geq k \Pi_{i_1} \Pi_{i_2} \cdots \Pi_{i_k} + \sum_{j \notin \{i_1,\ldots,i_k\}} \Pi_j - k \Pi_{i_1} \Pi_{i_2} \cdots \Pi_{i_k} \tag{96}$$

$$= \sum_{j \notin \{i_1,\ldots,i_k\}} \Pi_j \ \geq 0, \tag{97}$$

where Eq. (96) follows from lower bounding each term Π_{i_n} by $\Pi_{i_n} P$, which is true for any projector. (Here, $P = \prod_{n=1}^k \Pi_{i_n}$ is a projector since the Π_{i_n} commute by assumption.) But this implies

$$\Big(\sum_i \Pi_i\Big)^{-1} \leq \frac{1}{k} \Pi_{i_1} \Pi_{i_2} \cdots \Pi_{i_k} = \frac{1}{k} P, \tag{98}$$

and the lemma follows by left- and right-multiplying by P. □

Proof (of Lemma 34) We consider the equation $A^{-1} |\mathbb{1}\rangle = |\sigma\rangle$ and solve for the unknown $|\sigma\rangle$. By first multiplying with A from the left, we see that $AA^{-1} |\mathbb{1}\rangle = A |\sigma\rangle$, where AA^{-1} is the projector onto the support of A. Furthermore, note that $AA^{-1} = (QQ^{-1})^* \otimes QQ^{-1}$, thus $AA^{-1} |\mathbb{1}\rangle = |QQ^{-1}\rangle$. Therefore, written in unvectorised form, we have to solve the matrix equation $Q\sigma + \sigma Q = QQ^{-1}$. Clearly the restriction of σ to the support of Q can only be $\frac{1}{2}Q^{-1}$, but we can add an arbitrary operator in the kernel of Q. □

Proof (of Lemma 35) By Lemma 34, $B_P A^{-1} |\mathbb{1}\rangle = \frac{1}{2} B_P |Q^{-1}\rangle =: |\sigma\rangle$, using the fact that P is in the support of Q. Thus, unvectorised, we have $\frac{1}{2} PQ^{-1}P = \sigma$. But by Lemma 33, $\frac{1}{2} PQ^{-1}P \leq \frac{1}{2k} P$, thus $|\sigma\rangle \leq \frac{1}{2k} |P\rangle = \frac{1}{2k} B_P |\mathbb{1}\rangle$. The second part of the lemma follows simply by summing the $k = 1$ case over i. □

Proof (of Lemma 36) First, note that $B |\mathbb{1}\rangle = |Q\rangle$, thus $B_P A^{-1} B |\mathbb{1}\rangle = B_P A^{-1} |Q\rangle$. Defining $A^{-1} |Q\rangle =: |\sigma\rangle$, we can multiply with A from the left and solve $AA^{-1} |Q\rangle = A |\sigma\rangle$ for $|\sigma\rangle$. Now note that $AA^{-1} |Q\rangle = [(QQ^{-1})^* \otimes QQ^{-1}] |Q\rangle = |Q\rangle$. Unvectorising, we have $Q\sigma + \sigma Q = Q$, and clearly $\sigma = \frac{1}{2}\mathbb{1} + Q_\perp$, where we can add an arbitrary operator $Q_\perp$ in the kernel of Q. The lemma follows easily when we recall that P is in the support of Q. □

Proof (of Lemma 37) First we bound

$$(BA^{-1})^t |\mathbb{1}\rangle = (BA^{-1})^{t-1} BA^{-1} |\mathbb{1}\rangle \leq \frac{1}{2} (BA^{-1})^{t-1} B |\mathbb{1}\rangle \tag{99}$$

using Lemma 35. To show $(BA^{-1})^{t-1} B |\mathbb{1}\rangle = \frac{1}{2^{t-1}} B |\mathbb{1}\rangle$ by induction, it suffices to show

$$BA^{-1} B |\mathbb{1}\rangle = \frac{1}{2} B |\mathbb{1}\rangle, \tag{100}$$

which is then iterated $t - 1$ times. But this is already established by Lemma 36. □

6.1.2 A Weaker Conjecture

Conjecture 1 weakens the bound we proved in the commutative case by a polynomial factor. In fact, we will see that we can even cope with an *exponential* factor, and still prove a constructive QLLL with polynomial run-time, albeit at the expense of having to strengthen the Lovász conditions in the non-commutative case. (We recover the original Lovász conditions if the projectors commute.)

Conjecture 2 *Let $\{\Pi_i\}$ be an arbitrary set of projectors, let τ be a fixed tree with vertices labelled by these projectors, and L the log produced by running Algorithm 2 with these projectors. We conjecture that the probability that τ occurs in L is at most*

$$\Pr(\tau) \leq \frac{1}{(m\delta)^{|\tau|}} \prod_{v \in \tau} \Pr[\Pi(v)], \tag{101}$$

where $|\tau|$ is the size of τ (total number of vertices).

The motivation for this conjecture comes from the following result, which once again proves the conjecture for the simplest case of trees containing a single vertex.

Proposition 38 *Let $\{\Pi_i\}$ be a set of m projectors on $\mathbb{C}^d$, and define δ to be the spectral gap of the associated Hamiltonian*

$$H = \frac{1}{m} \sum_i \Pi_i. \tag{102}$$

Consider the following iterated measurement process, starting from the maximally mixed state. In each step, a projector Π_i is chosen independently uniformly at random, and the two-outcome measurement $\{\Pi_i, \mathbb{1} - \Pi_i\}$ is performed. This is repeated until a Π_i outcome is obtained, at which point the process halts.

Let ρ_a denote the final state of the system given that it halted on outcome Π_a, and p_a the probability that this occurs. Then the (unnormalised) density matrix corresponding to the process halting on outcome Π_a satisfies

$$X_a = p_a \rho_a \leq \frac{1}{m\delta} \Pi_a / d, \tag{103a}$$

hence

$$p_a = \operatorname{Tr}(X_a) \leq \frac{1}{m\delta} \frac{\operatorname{Tr}(\Pi_a)}{d}. \tag{103b}$$

For commuting projectors, we always have $\delta \geq 1/m$ (and equality holds unless *every* state outside the kernel of H violates more than one projector), and we recover Lemma 16. For non-commuting projectors, δ can range between 0 and 1, but we would expect the more difficult cases to be when $\delta < 1/m$, in which case Proposition 38 is a weaker statement than that already proven in Proposition 32.

Nonetheless, Proposition 38 is substantially easier to prove than Proposition 32, which could suggest that Conjecture 2 is the "correct" non-commutative generalisation of the bound in Lemma 8, and Proposition 32 is a red-herring that only applies to the very special case of single-vertex trees.

Proof Define for the linear map $\mathcal{M}(\rho) \coloneqq \frac{1}{m} \sum_i (\mathbb{1} - \Pi_i) \rho (\mathbb{1} - \Pi_i)$, and let $\mathcal{M}^t$ denote the t-fold composition of $\mathcal{M}$. Then the unnormalised density matrix X_a is given by

$$X_a = \frac{1}{m} \Pi_a \sum_{t=0}^{\infty} \left(\sum_{i_1, \ldots, i_t} \frac{1}{m} (\mathbb{1} - \Pi_{i_t}) \cdots \left(\sum_{i_1} \frac{1}{m} (\mathbb{1} - \Pi_{i_1}) \frac{\mathbb{1}}{d} (\mathbb{1} - \Pi_{i_1}) \right) \cdots (\mathbb{1} - \Pi_{i_t}) \right) \Pi_a \tag{104a}$$

$$= \frac{1}{m} \Pi_a \left(\sum_{t=0}^{\infty} \mathcal{M}^t(\mathbb{1}/d) \right) \Pi_a. \tag{104b}$$

Let G be the projector onto the ground state subspace (kernel) of H, and $E = G^\perp$ be the projector onto the subspace of excited states (the support, or coimage, of H). Note that $\forall i : \Pi_i G = G\Pi_i = 0$ and $\Pi_i E = E\Pi_i = \Pi_i$. For any scalars α, β,

$$\mathcal{M}(\alpha G + \beta E) = \frac{1}{m}\sum_{i=1}^{m}(\mathbb{1} - \Pi_i)(\alpha G + \beta E)(\mathbb{1} - \Pi_i) = \alpha G + \beta\frac{1}{m}\sum_{i=1}^{m}(E - \Pi_i) \tag{105a}$$

$$= \alpha G + \beta E - H \leq \alpha G + \beta(1-\delta)E. \tag{105b}$$

Applying this operator inequality t times starting from $\alpha = \beta = 1$ gives

$$\mathcal{M}^t(\mathbb{1}) \leq G + (1-\delta)^t E. \tag{106}$$

Using this in Eq. (104b) leads to

$$X_a \leq \frac{1}{m}\sum_{t=0}^{\infty}\frac{1}{d}\Pi_a\left(G + (1-\delta)^t E\right)\Pi_a = \frac{1}{m}\sum_{t=0}^{\infty}\frac{1}{d}(1-\delta)^t\Pi_a = \frac{1}{m\delta}\Pi_a/d, \tag{107}$$

which proves the proposition. □

6.2 Expected number of violations

From the conjectures, we can derive bounds on the expected number of violations seen by Algorithm 2.

Definition 39 (ϵ-strengthened Lovász conditions) *Let $\Pi_1, \Pi_2, \ldots, \Pi_m$ be projectors that act on arbitrary subsets of n qudits. We say that the set of projectors $\{\Pi_i\}$ satisfies the* ϵ-strengthened Lovász conditions *if there exist values $0 \leq x_1, x_2, \ldots, x_m \leq 1$ such that*

$$R(\Pi_i) \leq (1-\epsilon)x_i \cdot \prod_{\Pi_j \in \Gamma(\Pi_i)}(1 - x_j). \tag{108}$$

Theorem 40 *Let $\{\Pi_i\}$ be a set of m projectors satisfying the ϵ-strengthened Lovász conditions of Definition 39 for some $\epsilon > 0$. If Conjecture 1 is true, then there exists a constant C (which depends on ϵ) such that the expected number of violations seen by Algorithm 2 is bounded by*

$$\mathbb{E}(\text{total number of violations}) \leq C\sum_{i=1}^{m}\frac{x_i}{1-x_i}. \tag{109}$$

Proof The proof is very similar to that of Theorem 12 in Section 3.2 for the commutative case. Let N_a be the number of times a given projector Π_a appears in Algorithm 2's log, which is the same as the number of times Π_a is measured to be violated. Let $\mathcal{T}_a$ denote all proper witness trees whose root vertex is labelled by Π_a. Then, using Conjecture 1, we have

$$\mathbb{E}(N_a) = \sum_{\tau \in \mathcal{T}_a}\Pr(\tau \text{ appears in the log}) \leq \sum_{\tau \in \mathcal{T}_a}\text{poly}(|\tau|)\prod_{v \in \tau}\Pr[\Pi(v)]. \tag{110}$$

The probability $\Pr[\Pi_i]$ of a projector being violated on a random state is just given by its relative dimension $\Pr(\Pi_i) = R(\Pi_i)$. Since by assumption the projectors satisfy the Lovász conditions of Definition 39, we have

$$\mathbb{E}(N_a) \leq \sum_{\tau \in \mathcal{T}_a}\text{poly}(|\tau|)\prod_{\Pi_i \in \tau}R(\Pi_i) \leq \sum_{\tau \in \mathcal{T}_a}\text{poly}(|\tau|)\prod_{\Pi_i \in \tau}(1-\epsilon)x_i' \tag{111a}$$

$$\leq \frac{x_a}{1-x_a}\sum_{\tau \in \mathcal{T}_a}\text{poly}(|\tau|)(1-\epsilon)^{|\tau|}\Pr(\tau), \tag{111b}$$

where $\Pr(\tau_a)$ is the probability that the Galton-Watson process defined in Lemma 11 generates tree τ_a. The inequality in Eq. (111b) follows from Lemma 11.

The summand in Eq. (111b) is just the expectation of a functional $f(x) = \mathrm{poly}(x)(1-\epsilon)^x$ over the *total progeny* $|\tau|$ of the Galton-Watson process. For our purposes, it sufficies to observe that this functional is always bounded. Let $C := \max_{x \geq 0} f(x)$. Then

$$\mathbb{E}(N_a) \leq \frac{x_a}{1-x_a} \sum_{\tau \in \mathcal{T}_a} C \Pr(\tau) \leq C \frac{x_a}{1-x_a}, \tag{112}$$

the final inequality coming from the fact that the Galton-Watson process either produces a tree in $\mathcal{T}_a$, or continues indefinitely. The theorem follows from summing over all projectors. □

In fact, to prove Theorem 40, all we require is that the expectation of the functional $\mathrm{poly}(|\tau|)$ of the total progeny τ of the Galton-Watson multi-type branching process be bounded. It is therefore likely that the $1-\epsilon$ factor can be removed by a more careful analysis of the distribution of total progeny for the branching process.

Theorem 41 *Let $\{\Pi_i\}$ be a set of m projectors with spectral gap δ, which satisfy the ϵ-strengthened Lovász conditions for $\epsilon = 1 - m\delta$. If Conjecture 2 is true, then the expected number of violations seen by Algorithm 2 is bounded by*

$$\mathbb{E}(\text{total number of violations}) \leq \sum_{i=1}^{m} \frac{x_i}{1-x_i}. \tag{113}$$

Proof As in Theorem 40,

$$\mathbb{E}(N_a) = \sum_{\tau \in \mathcal{T}_a} \Pr(\tau \text{ appears in the log}) \leq \sum_{\tau \in \mathcal{T}_a} \frac{1}{(m\delta)^{|\tau|}} \prod_{v \in \tau} \Pr[\Pi(v)], \tag{114}$$

this time using Conjecture 2. Thus

$$\mathbb{E}(N_a) \leq \frac{x_a}{1-x_a} \sum_{\tau \in \mathcal{T}_a} \frac{(1-\epsilon)^{|\tau|}}{(m\delta)^{|\tau|}} \Pr(\tau) = \frac{x_a}{1-x_a} \sum_{\tau \in \mathcal{T}_a} \Pr(\tau) \leq \frac{x_a}{1-x_a}, \tag{115}$$

and the theorem follows by summing over all projectors. □

6.3 Converging to a solution

In the non-commutative setting, there is an ambiguity in what it means for an algorithm to construct a state satisfying the requirements of the QLLL of Theorem 3. The non-constructive QLLL asserts the existence of a state that does not violate any of the projectors. We could demand that the algorithm converges to a state whose probability of violating any projector is at most ε; we will call this "weak convergence", or "convergence in energy". Alternatively, we could demand that the algorithm converges to a state that is ε-close (in some suitable distance measure, say fidelity) to the subspace of states that do not violate any projector; we will call this "strong convergence", or "convergence in fidelity". In the classical and in the commutative cases, these are equivalent, and there is a single, unambiguous definition of convergence. However, non-commutativity of the projectors means that these two notions of convergence are in general no longer equivalent in the quantum case.

Definition 42 (Strong convergence, or convergence in fidelity) *We say that an algorithm converges in the strong sense to a state fulfilling the QLLL if, given any $\varepsilon > 0$, there exists a t such that the state ρ produced by running the algorithm for time t satisfies $\mathrm{Tr}[P_0 \rho] \geq 1 - \varepsilon$, where P_0 is the projector onto the subspace of states defined by $\mathrm{span}\{|\psi\rangle : \forall i \, \Pi_i |\psi\rangle = 0\}$; i.e. the state ρ is ε-close in fidelity to the subspace of states fulfilling the QLLL requirements.*

Definition 43 (Weak convergence, or convergence in energy) *We say that an algorithm converges in the weak sense to a state fulfilling the QLLL if, given any $\varepsilon > 0$, there exists a t such that the state ρ produced by running the algorithm for time t satisfies $\forall i\ \mathrm{Tr}[\Pi_i \rho] \leq \varepsilon$.*

As our terminology suggests, strong convergence implies weak convergence, but in general the converse does *not* hold.

Lemma 44 *Strong convergence is strictly stronger than weak convergence.*

Proof It is trivial to see that strong convergence implies weak convergence: if ρ is such that $\mathrm{Tr}[P_0 \rho] \geq 1 - \varepsilon/m$, then

$$\sum_i \mathrm{Tr}[\Pi_i \rho] = \sum_i \mathrm{Tr}[\Pi_i (\mathbb{1} - P_0)\rho] \leq \sum_i \mathrm{Tr}[(\mathbb{1} - P_0)\rho] \leq \varepsilon \tag{116}$$

(the first equality following from the fact that $\Pi_i P_0 = 0$ by definition, the middle inequality from the operator inequality $\Pi_i \leq \mathbb{1}$).

To see that weak convergence is strictly weaker, consider the example of two almost-identical projectors acting on the same set of qudits: $\Pi_0 = |\psi\rangle\langle\psi|$ and $\Pi_1 = |\psi'\rangle\langle\psi'|$, with $|\langle\psi|\psi'\rangle|^2 = 1 - \delta$. Let

$$|\psi^\perp\rangle = \frac{(\mathbb{1} - |\psi\rangle\langle\psi|)\,|\psi'\rangle}{\sqrt{1-\delta}} \tag{117}$$

be the state in the span of $\{|\psi\rangle, |\psi'\rangle\}$ orthogonal to $|\psi\rangle$. For $\rho = |\psi^\perp\rangle\langle\psi^\perp|$, we have

$$\mathrm{Tr}[\Pi_0 \rho] = |\langle\psi|\psi^\perp\rangle|^2 = 0, \tag{118}$$

$$\mathrm{Tr}[\Pi_1 \rho] = |\langle\psi'|\psi^\perp\rangle|^2 = |\langle\psi'|(\mathbb{1} - |\psi\rangle\langle\psi|)|\psi'\rangle|^2 = \frac{\delta^2}{1-\delta}, \tag{119}$$

whereas $\mathrm{Tr}[P_0 \rho] = 0$. Therefore, letting δ tend to 0, an algorithm which converges to the state ρ has converged to arbitrarily high precision in the weak sense, yet it is as far as possible from convergence in the strong sense. □

However, weak convergence *does* imply strong convergence if we are prepared to pay a price in the convergence time. That price is a dependence on the spectral gap of Definition 30.

Lemma 45 *If an algorithm converges weakly in time $O(f(1/\varepsilon))$, then it converges strongly in time $O(f(1/m\delta\varepsilon))$.*

Proof If we run the algorithm for time $O(f(1/m\delta\varepsilon))$ then, by Definition 43 of weak convergence, the state ρ produced by the algorithm must satisfy $\forall i : \mathrm{Tr}[\Pi_i \rho] \leq m\delta\varepsilon$. Decomposing the state ρ as

$$\rho = P_0\rho P_0 + P_0\rho(\mathbb{1} - P_0) + (\mathbb{1} - P_0)\rho P_0 + (\mathbb{1} - P_0)\rho(\mathbb{1} - P_0), \tag{120}$$

and noting that $\Pi_i P_0 = 0$ by definition, we have

$$\begin{aligned}
\delta\varepsilon &\geq \frac{1}{m}\sum_i \mathrm{Tr}[\Pi_i \rho] \\
&= \frac{1}{m}\sum_i \Big(\mathrm{Tr}[\Pi_i P_0 \rho P_0] + \mathrm{Tr}[\Pi_i P_0 \rho(\mathbb{1} - P_0)] \qquad (121a) \\
&\qquad\qquad + \mathrm{Tr}[\Pi_i(\mathbb{1} - P_0)\rho P_0] + \mathrm{Tr}[\Pi_i(\mathbb{1} - P_0)\rho(\mathbb{1} - P_0)]\Big) \\
&= \frac{1}{m}\sum_i \mathrm{Tr}[\Pi_i(\mathbb{1} - P_0)\rho(\mathbb{1} - P_0)]. \qquad (121b)
\end{aligned}$$

But, by Definition 30 of the spectral gap δ,

$$\frac{1}{m}\sum_i \frac{\mathrm{Tr}[\Pi_i(\mathbb{1}-P_0)\rho(\mathbb{1}-P_0)]}{\mathrm{Tr}[(\mathbb{1}-P_0)\rho]} \geq \delta, \tag{122}$$

thus from Eqs. (121) and (122) we obtain $\mathrm{Tr}[(\mathbb{1}-P_0)\rho] \leq \varepsilon$, or

$$\mathrm{Tr}[P_0\rho] \geq 1-\varepsilon, \tag{123}$$

proving strong convergence in time $O(f(1/m\delta\varepsilon))$ as claimed. □

Which is the "correct" notion of convergence depends on what one is trying to achieve. From a physics perspective, an algorithm for finding a state fulfilling the Lovász conditions can be interpreted as an algorithm for finding the ground state of the Hamiltonian

$$H = \frac{1}{m}\sum_i \Pi_i. \tag{124}$$

If we are interested in cooling a system to the ground state, producing a state with low energy is sufficient and weak convergence is the appropriate notion, since the parameter ε in Definition 43 upper-bounds the energy of the state with respect to the Hamilton H. On the other hand, if we are interested in ground state properties, we want the algorithm to produce a state close in fidelity to the true ground state, requiring strong convergence.

If the spectral gap scales inverse-polynomially, then the run-time remains polynomial even for strong convergence. For some important cases of the QLLL, the gap does indeed scale inverse-polynomially, and the run-time is provably polynomial even for strong convergence. (For example, in the commutative case the gap is constant and we have already seen that an efficient algorithm always exists.) On the other hand, there are certainly cases for which the spectral gap is exponentially small, and in that case the run-time we require to ensure strong convergence will be exponentially large. However, an inverse-linear dependence on the gap may be the best one could hope to achieve, at least for algorithms that work by measuring the projectors, as it takes expected time $1/\delta$ to see even *one* violation if the system is in the lowesr excited state (a state necessarily *orthogonal* to the ground-state subspace).

From a complexity theoretic perspective, weak convergence is the more natural notion, as it corresponds to the canonical FQMA-complete problem of low-energy state preparation [Janzing et al., 2003]. (FQMA is the quantum analogue of FNP, the functional version of NP; whereas an NP or QMA problem asks whether there *exists* an input producing a "yes" answer, an FNP or FQMA problem requires such an input to be prepared.) Similarly, the k–QSAT problem, as defined and shown to be QMA_0-complete by Bravyi [2006], promises that a given k-local Hamiltonian either has a ground state (minimum eigenvalue eigenstate) with *exactly* zero energy, or has a ground state with energy larger than α (where α is inverse polynomial in the problem size). The k–QSAT problem is then to decide whether a zero-energy state exists. The corresponding FQMA problem is to produce a state that witnesses the existence of a zero-energy state. Setting $\varepsilon \leq \alpha$ and running an algorithm until it weakly converges to precision ε produces precisely such a witness.

On the other hand, formulating strong convergence as a well-defined complexity-theoretic concept requires a little care. For all the standard complexity classes, the projectors defining a problem instance must be specified classically. Thus if the problem size is to be well-defined, the projectors can only be specified to finite precision. However, there exist instances which are arbitrarily close to each other (the matrix elements of the two sets of projectors are arbitrarily close), yet a state satisfying the strong convergence requirements for one instance is maximally far from satisfying those conditions for the other instance. (The two-projector example considered in Lemma 44 provides an instance of this. By making δ arbitrarily small, the example can

be made arbitrarily close to a QLLL problem consisting of two identical projectors Π_0. Yet the state $|\psi^\perp\rangle$, which satisfies the strong convergence requirements when both projectors are Π_0, is as far as possible for satisfying those requirements for the instance given in Lemma 44. In order for any algorithm to determine that $|\psi^\perp\rangle$ is *not* a valid solution, the projectors would have to be specified to arbitrarily high precision.)

To make the strong convergence case into a well-defined complexity-theoretic problem, we must either formulate it as a *weak-membership* problem [Grötschel et al., 1988], with a parameter δ specifying the precision to which the answer must be given, or equivalently we can formulate it as a promise problem, with a promise on the spectral gap (cf. Cubitt et al. [2009]). Either way, this means that *the problem instance itself* sets a lower-bound on the spectral gap. But Lemma 45 shows that an efficient algorithm for weak convergence already implies an efficient algorithm for the complexity-theoretic formulation of strong convergence in this case. So in the most reasonable complexity-theoretic formulation, strong convergence and weak convergence are equivalent.

6.4 An efficient quantum algorithm

In Section 3.3, we used the bound from Theorem 12 together with Theorem 13 to prove that Algorithm 4 converges efficiently to the desired state. But Theorem 13 applies to any set of projectors; it does not depend on commutativity. Therefore, the analogous non-commutative bounds in Theorems 40 and 41 (which depend on Conjectures 1 and 2) together with Theorem 13 imply that Algorithm 4 converges efficiently (in the weak sense) in the non-commutative case. Lemma 45 extends this to strong convergence, proving the following constructive results in the general, non-commutative case of the Quantum Lovász Local Lemma. (The existence parts of the following theorems are true independent of the conjectures, by Theorem 3.)

Theorem 46 (Constructive QLLL) *Let $\{\Pi_i\}$ be a set of m projectors projectors acting on subsets of n qudits, with spectral gap δ (Definition 30). If $\{\Pi_i\}$ satisfy the ϵ-strengthened Lovász conditions (Definition 39) for any $\epsilon > 0$, then there exists a joint state ρ of the qudits such that $\forall i : \mathrm{Tr}[\Pi_i \rho] = 0$.*

Moreover, if Conjecture 1 holds, there is a quantum algorithm that constructs a state ρ' such that $\forall i\ \mathrm{Tr}[\Pi_i \rho'] \leq \varepsilon$, in time

$$O\left(n + \frac{m}{\varepsilon} \sum_{i=1}^{m} \frac{x_i}{1 - x_i} \cdot \big|[i]\big|\right) \tag{125}$$

where $|[i]|$ is the number of qudits on which the projector Π_i acts non-trivially, thus ρ' has probability at most ε of violating any of the constraints defined by the Π_i.

The same algorithm constructs a state ρ'' such that $\mathrm{Tr}[P_0 \rho''] \geq 1 - \varepsilon$, where P_0 is the projector onto the subspace $\mathrm{span}\{|\psi\rangle : \forall i\, \Pi_i |\psi\rangle = 0\}$, in time

$$O\left(n + \frac{1}{\delta\varepsilon} \sum_{i=1}^{m} \frac{x_i}{1 - x_i} \cdot \big|[i]\big|\right), \tag{126}$$

thus ρ'' is ε-close in fidelity to the subspace of states satisfying $\forall i\, \Pi_i |\psi\rangle = 0$.

It may well be possible to remove the $1 - \epsilon$ strengthening of the Lovász conditions, by a more refined analysis of the branching process in Theorem 40. (See discussion after the proof of that theorem.)

Theorem 47 (Constructive QLLL, strengthened Lovász conditions) *Let $\{\Pi_i\}$ be a set of m projectors projectors acting on subsets of n qudits, with spectral gap δ (Definition 30). If $\{\Pi_i\}$ satisfy the ϵ-strengthened Lovász conditions (Definition 39) for $\epsilon = 1 - m\delta$, then there exists a joint state ρ of the qudits such that $\forall i\ \mathrm{Tr}[\Pi_i \rho] = 0$.*

Moreover, if Conjecture 2 holds, there is a quantum algorithm that constructs a state ρ' such that $\mathrm{Tr}[\Pi_i \rho'] \leq \varepsilon$, in time

$$O\left(n + \frac{m}{\varepsilon} \sum_{i=1}^{m} \frac{x_i}{1 - x_i} \cdot \left|[i]\right|\right) \tag{127}$$

where $|[i]|$ is the number of qudits on which the projector Π_i acts non-trivially, thus ρ' has probability at most ε of violating any of the constraints defined by the Π_i.

The same algorithm constructs a state ρ'' such that $\mathrm{Tr}[P_0 \rho''] \geq 1 - \varepsilon$, where P_0 is the projector onto the subspace $\mathrm{span}\{|\psi\rangle : \forall i \, \Pi_i |\psi\rangle = 0\}$, in time

$$O\left(n + \frac{1}{\delta\varepsilon} \sum_{i=1}^{m} \frac{x_i}{1 - x_i} \cdot \left|[i]\right|\right), \tag{128}$$

thus ρ'' is ε-close in fidelity to the subspace of states satisfying $\forall i \, \Pi_i |\psi\rangle = 0$.

Note that all the arguments of Section 5 go through for the non-commutative case, so that Theorems 46 and 47 lead directly to CP map convergence results for more general classes of maps. This generalises Theorem 28 and Corollary 29 to non-commutative local Hamiltonians, and their associated dissipative state engineering CP maps.

7 Conclusions

We have proven that a simple quantum algorithm (Algorithm 4) efficiently constructs a state satisfying the requirements of the Quantum Lovász Local Lemma (Theorem 3) of Ambainis et al. [2009] in the setting of commuting constraints. Not only does this give an efficient algorithm for constructing the quantum state whose existence is guaranteed by the commutative QLLL. In fact, since we do not assume the QLLL in order to prove that the algorithm finds the state, this also gives an independent, constructive proof of the commutative QLLL itself.

Until Beck [1991] provided the first algorithm, it was not a priori clear whether the combinatorial objects whose existence was guaranteed by the classical LLL could also be constructed efficiently. In the quantum case, it was perhaps even less obvious whether the states guaranteed to exist by the QLLL could be prepared efficiently, as those states can have a highly complex entanglement structure. Our result gives a new quantum algorithm for efficiently constructing these complex states in the commutative case. Or, in physics terms, our result provides a new method of cooling to the ground state of certain many-body Hamiltonians with commuting local terms.

In the non-commutative setting, we can only prove the constructive QLLL modulo a technical conjecture that the probability of witness trees is at most polynomially weaker than the bound we have proven in the commuting case (Conjecture 1). If we impose stronger Lovász conditions (which reduce to the usual conditions in the commutative case), we can prove a constructive QLLL using a weaker conjecture (Conjecture 2), which weakens the commutative bound by an exponential factor in the non-commutative setting. We have given a simple counter-example showing that the bound from the commutative case does not hold in general. The violation of the commuting bound is directly attributable to the measurement-disturbance effect of non-commuting quantum measurements, which means that even "satisfied" outcomes can disturb the state in an undesirable way.

We can prove both our conjectures in the simplest case of single-vertex trees. This is already non-trivial, as it shows that the "satisfied" measurements that can cause so much trouble in the non-commutative case can effectively be ignored up to the first violation. However, as our counter-example demonstrates, the case of multiple violations is substantially different, and proving the conjectures in general is likely to require new ideas and techniques. Our conjectures are supported by numerical evidence for some small multi-vertex trees, though the numerics

we have done are very limited. However, it is very likely that the conjectures *must* hold if the natural quantum generalisation of Moser's algorithm (Algorithm 2) is to work. Since the probability of witness trees occurring in the algorithm's log is directly related to the expected number of violations seen by the algorithm (Theorems 12, 40 and 41), if the witness tree probability does not shrink fast enough with the size of the tree, the expected number of violations will be unbounded. Thus disproving our conjectures would strongly suggest that an entirely different approach to the one pioneered by Moser in the classical setting is required in the non-commutative quantum setting. (Or that there is no efficient constructive version of the general QLLL.)

In the classical case, Moser and Tardos [2010] impose a slight restriction on the events that feature in the LLL, requiring that they be determined by an underlying set of random variables. In the quantum case, we imposed the analogous restriction, requiring that the subspaces in the QLLL are defined on an underlying set of qudits. Kolipaka and Szegedy [2011] have generalised the original Moser and Tardos [2010] results, and removed this restriction in the classical case. Their results also prove a constructive algorithm right up to the *Shearer bound*, the tightest possible version of the Lovász conditions. Finally, they also generalise an earlier result of Haeupler et al. [2010], showing that the Moser algorithm is efficient in the number of variables even if the number of events is super-polynomial. It seems likely that these results can be generalised to the quantum setting, at least in the commutative case. In particular, this would remove the assumption of an underlying tensor product structure, proving a constructive version of the QLLL for general subspaces, matching the formulation of Ambainis et al. [2009].

Although not expressed in this way in the published versions of the papers, Moser pointed out that his constructive proof [Moser, 2009] of the symmetric Lovász Local Lemma (Corollary 2) can be formulated as an elegant compression argument [Moser, 2010, Tao, 2009]: the sequence of random bits used by the algorithm can be recovered perfectly from the data in the execution log and final output, but if the length of the log grows indefinitely, then this data can be compressed into fewer bits than the entropy. The strong converse of Shannon's noiseless coding theorem (see e.g. Cover and Thomas [2006]) imposes an exponential suppression of the probability of compressing below the entropy, implying a linear bound on the expected length of the log, hence also the run-time. This compression argument can be extended to the general Lovász Local Lemma, and it is reasonably clear [Moser, 2010] that with a little effort it could even give tight constants in the general LLL (Theorem 1). Indeed, the witness trees and coupling argument of Moser and Tardos [2010] essentially contains an implicit compression argument, but elegantly side-steps the necessity of designing an explicit compression scheme for the log. Our witness tree and quantum coupling argument again contains an implicit compression argument. But the compression argument can also be generalised to the quantum case explicitly, either making use of the strong converse of the Schumacher noiseless coding theorem [Ogawa and Nagaoka, 1999, Winter, 1999], or using a more general information-theoretic analysis [Sattath and Arad].

Acknowledgements We greatfully acknowledge insightful discussions with Julia Kempe on the topic of the QLLL, and particularly Or Sattath both for valuable discussions and for repeatedly pointing out errors in earlier versions of our proofs. We would also like to thank Robin Moser for taking the time to discuss his beautiful proof with us, and for sharing valuable insights into the classical LLL, and Itai Arad for sharing additional resuts on the QLLL. TSC would like to thank Andreas Winter, Aram Harrow and David Perez-Garcia for valuable discussions, James Martin for instruction in coupling arguments, and Frank Verstraete and the University of Vienna for their hospitality throughout the visit during which some of this work were carried out. Early parts of this work were carried out whilst TSC was at the University of Bristol, supported by a Leverhulme Early-Career fellowship. TSC is now supported by a Juan de la Cierva fellowship, the EU project QUEVADIS, and by Spanish grants QUITEMAD, I-MATH, and MTM2008-01366. MS acknowledges support by the Austrian AMS Bildungskarenz programme and the Austrian SFB project FoQuS (F4014).

References

D. Aharonov and L. Eldar. On the commuting local hamiltonian problem, and tight conditions on topological order. arXiv:1102.0770, 2011.

D. Aharonov, A. Ambainis, J. Kempe, and U. Vazirani. Quantum walks on graphs. In *Proceedings of the ACM Symposium on the Theory of Computation (STOC 2001)*, pages 50–59, July 2001.

N. Alon and J. H. Spencer. *The probabilistic method*. Wiley, 2008.

A. Ambainis, J. Kempe, and O. Sattath. A quantum Lovász Local Lemma. arXiv:0911.1696[quant-ph], 2009.

J. Beck. An algorithmic approach to the lovász local lemma. *Random Structures and Algorithms*, 2(4):343–365, 1991.

S. Bravyi. Efficient algorithm for a quantum analogue of 2-SAT. arXiv:quant-ph/0602108, 2006.

S. Bravyi and M. Vyalyi. Commutative version of the k-local hamiltonian problem and common eigenspace problem. arXiv:quant-ph/0308021, 2003.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, second edition, 2006.

T. S. Cubitt, J. Eisert, and M. M. Wolf. Deciding whether a quantum channel is Markovian is NP-hard. arXiv:0908.2128[math-ph], 2009.

P. Erdös and L. Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. *Infinite and finite sets*, 2:609–627, 1975.

M. Grötschel, L. Lovász, and A. Schrijver. *Geometric algorithms and combinatorial optimization*. Springer, 1988. ISBN 038713624x.

B. Haeupler, B. Saha, and A. Srinivasan. New constructive aspects of the lovasz local lemma. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS 2011)*, pages 397–406, October 2010.

D. Janzing, P. Wocjan, and T. Beth. Cooling and low energy state preparation for 3-local Hamiltonians are FQMA-complete. arXiv:quant-ph/0303186, 2003.

K. Kolipaka and M. Szegedy. Moser and tardos meet lovász. In *Proceedings of the ACM Symposium on the Theory of Computation (STOC 2011)*, pages 235–244, June 2011.

T. Lindvall. *Lectures on the Coupling Method*. Dover Publications, 2002.

R. Moser and G. Tardos. A constructive proof of the general Lovász Local Lemma. *Journal of the ACM (JACM)*, 57(2):1–15, 2010.

R. A. Moser. A constructive proof of the Lovász Local Lemma. In *Proceedings of the 41st annual ACM symposium on Theory of computing (STOC 2009)*, 2009.

R. A. Moser. An information theoretic view upon the constructive proof of the Lovász Local Lemma. Private communication, 2010.

M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, 2000.

T. Ogawa and H. Nagaoka. Strong converse to the quantum channel coding theorem. *IEEE Trans. Inform. Theory*, 45(7):2486, 1999.

M. Sanz, D. Perez-Garcia, M. M. Wolf, and J. I. Cirac. A quantum version of Wielandt's inequality. arXiv:0909.5347[quant-ph], 2009.

O. Sattath and I. Arad. In preparation.

N. Schuch. Complexity of commuting hamiltonians on a square lattice of qubits. arXiv:1105.2843[quant-ph], 2011.

M. Schwarz, K. Temme, and F. Verstraete. Contracting tensor networks and preparing peps on a quantum computer. arXiv:1104.1410[quant-ph], 2011.

T. Tao. Moser's entropy compression argument. http://terrytao.wordpress.com/2009/08/05/mosers-entropy-compression-argument/, 2009.

K. Temme, T. J. Osborne, , K. G. Vollbrecht, D. Poulin, and F. Verstraete. Quantum Metropolis Sampling. *Nature*, 471:87, 2011.

H. Thorisson. *Coupling, stationarity, and regeneration*. Springer, 2000.

F. Verstraete, M. M. Wolf, and J. I. Cirac. Quantum computation and quantum-state engineering driven by dissipation. *Nature Physics*, 5:633–636, 2009.

A. Winter. *Coding Theorems of Quantum Information Theory*. PhD thesis, Universität Bielefeld, 1999.