Necessary condition for local quantum operations and classical communication with extensive violation by separable operations

Scott M. Cohen*
Department of Physics, Portland State University, Portland, Oregon 97201, USA
(Received 25 November 2013; published 29 July 2014)

We give a conceptually simple necessary condition such that a separable quantum operation can be implemented by local operations on subsystems and classical communication between parties (LOCC), a condition which follows from a novel approach to understanding LOCC. This necessary condition applies to all LOCC protocols involving any number of parties and any finite number of rounds of communication. Furthermore, it demonstrates an extremely strong difference between separable operations and LOCC, in that there exist examples of the former for which the condition is extensively violated. More precisely, the violation by separable operations of our necessary condition for LOCC grows without limit as the number of parties increases.

DOI: 10.1103/PhysRevA.90.012336 PACS number(s): 03.67.Hk, 03.67.Ac, 03.65.Ta

I. INTRODUCTION

One of the most challenging goals of quantum information theory is to understand what can be accomplished by spatially separated parties, each performing local operations on quantum subsystems and exchanging classical communication among themselves, a process known as LOCC [1–10]. Unfortunately, this class of operations is extremely difficult to analyze, and while much progress has been made, we still lack a deep intuition about its inner workings. Separable operations (SEP, also known as separable channels) [11], of which LOCC is a strict subset [12], has a much simpler mathematical description, but while its study has provided powerful lessons about LOCC, no simple picture has emerged that would allow us to understand the difference between these two important classes of quantum operations in the most general terms.

The first work showing that SEP and LOCC differ [12] considered a quantum system distributed to two parties who know only that the full system is in one of a given set of mutually orthogonal product states, commonly referred to as domino states. Their task is to determine with certainty which state the system was prepared in. While this task can be readily achieved with SEP, it was shown in [12] that it is impossible using LOCC, even if the parties are allowed an infinite number of rounds of communication. More recent work on a certain random distillation problem [13] has demonstrated that the gap between LOCC and SEP can be quite significant [14,15].

There has also been much interest in the structure of LOCC itself. For example, several papers have looked at the extent to which using more rounds of communication may be advantageous. Early work [16] showed that for the task of transforming from one bipartite pure state to another, multiple rounds is not better than a single round. In contrast, certain mixed-state purification scenarios require at least two rounds of communication [17]. The question of whether an infinite number of rounds can be helpful has also been studied. In [18], it was shown that this is not helpful for perfect state discrimination of complete product bases, but there are random distillation tasks for which infinite rounds are required [19]. Random distillations were also used to show the existence of sequences of LOCC maps that converge to a map that cannot be implemented by LOCC [14].

The foregoing results all arose from studies of specific operational tasks. In this paper, we take a different approach, seeking general conditions that can be applied to all LOCC protocols. Our main result is proven only for the finite-round case, but see our conclusions for why we believe it may hold for infinite rounds as well. We uncover a conceptually simple picture that distinguishes between LOCC and SEP, and then show that it provides a very strong separation between these two classes of quantum operations. We prove a necessary condition for LOCC and then show that separable operations violate this condition by an arbitrarily large amount, an amount constrained only by the size of the system as measured by the number of parties involved. Previous work showing a gap between LOCC and SEP considered specific tasks that can be accomplished by SEP but cannot be closely approximated by LOCC [12–15,20,21]. The gap we show is of a different sort, one which does not directly address the important question of how closely a given separable operation can be approximated by LOCC. In contrast, our necessary condition is of an abstract, geometrical nature, which provides a more general (and one may hope, ultimately deeper) understanding of the difference between SEP and LOCC.

Before proceeding to our theorem, let us recall what LOCC involves. We may assume the parties have agreed in advance upon a protocol that they will follow. One of the parties, say party 1, whose system is described by states in Hilbert space $\mathcal{H}_1$, starts by locally performing a generalized measurement [22] with outcomes corresponding to Kraus operators K_{i_1}. That party broadcasts her outcome i_1 to the other parties, who according to the agreed-upon protocol, all know which of them (call this party 2) is to measure next. Party 2 then performs a measurement with outcome i_2, described by $K_{i_2}^{(i_1)}$ acting on $\mathcal{H}_2$ and conditioned on Alice's outcome i_1, after which he broadcasts his outcome i_2 to all the others. The next party to measure will be α (which could be party 1 again), performing $K_{i_3}^{(i_1,i_2)}$, and they may continue in this way for an arbitrary (but we assume here, finite) number of rounds. From the fact

*cohensm52@gmail.com

that the probabilities of outcomes obtained at each stage must always sum to unity, one has that for each and every n,

$$I_\alpha = \sum_{i_n} K_{i_n}^{(\mathcal{S}_n^\alpha)\dagger} K_{i_n}^{(\mathcal{S}_n^\alpha)}, \tag{1}$$

where I_α is the identity operator on $\mathcal{H}_\alpha$, and $\mathcal{S}_n^\alpha$ is a collection of indices $\mathcal{S}_n^\alpha = \{i_1, i_2, \ldots, i_{n-1}; \beta\}$, indicating all outcomes obtained in earlier measurements. The last index in the collection, β, indicates which party performed this measurement, and we refer to the node as a "β node." When $\beta \neq \alpha$ we define $K_{i_n}^{(\mathcal{S}_n^\alpha)} = I_\alpha$, reflecting the fact that party α does nothing when party β is measuring. When this is the case, the sum on the right has only this single term I_α, and (1) becomes trivial.

Given any LOCC protocol, we can represent it as a tree, each local measurement appearing as a branching to a set of nodes, with each of these nodes representing one outcome of that measurement. We will label each node by a positive operator $\mathcal{K}_{i_n}^{(\mathcal{S}_n^\alpha)}$, obtained as follows: starting from the ordered product, $K_{i_n}^{(\mathcal{S}_n^\alpha)} K_{i_{n-1}}^{(\mathcal{S}_{n-1}^\alpha)} \ldots K_{i_2}^{(\mathcal{S}_2^\alpha)} K_{i_1}^{(\mathcal{S}_1^\alpha)}$, of all Kraus operators implemented by the party (α) whose outcome is represented by the given node, multiply this by its Hermitian conjugate to obtain

$$\begin{aligned}\mathcal{K}_{i_n}^{(\mathcal{S}_n^\alpha)} = {} & K_{i_1}^{(\mathcal{S}_1^\alpha)\dagger} K_{i_2}^{(\mathcal{S}_2^\alpha)\dagger} \ldots K_{i_{n-1}}^{(\mathcal{S}_{n-1}^\alpha)\dagger} K_{i_n}^{(\mathcal{S}_n^\alpha)\dagger} K_{i_n}^{(\mathcal{S}_n^\alpha)} \\ & \times K_{i_{n-1}}^{(\mathcal{S}_{n-1}^\alpha)} \ldots K_{i_2}^{(\mathcal{S}_2^\alpha)} K_{i_1}^{(\mathcal{S}_1^\alpha)}.\end{aligned} \tag{2}$$

Note that in this product of Kraus operators, there is one for each round leading up to this node, but many of these operators will be the identity, as parties other than α will have measured at that round along this branch of the tree. We will sometimes refer to the operator $\mathcal{K}_{i_n}^{(\mathcal{S}_n^\alpha)}$ as an "outcome" of the associated measurement by party α.

With this definition of the operators, it was observed in [23,24] that

$$\sum_{i_n} \mathcal{K}_{i_n}^{(\mathcal{S}_n^\alpha)} = \mathcal{K}_{i_{n-1}}^{(\mathcal{S}_{n-1}^\alpha)}. \tag{3}$$

This follows directly from (1) and (2), and will play an important role in what follows.

For clarity and use in the remaining discussion, the following is how we identify the SEP implemented by a given LOCC protocol. When $\{\mathcal{S}_n^\alpha, i_n\}$ denotes a leaf of the tree, it identifies a final outcome of the protocol. The operator $\hat{\mathcal{K}}_j^{(\alpha)} := \mathcal{K}_{i_n}^{(\mathcal{S}_n^\alpha)} / \mathrm{Tr}(\mathcal{K}_{i_n}^{(\mathcal{S}_n^\alpha)})$, with $\mathcal{S}_n^\alpha = \{i_1, i_2, \ldots, i_{n-1}; \alpha\}$ [case $\beta = \alpha$ in the definition of $\mathcal{S}_n^\alpha$ given just below (1)], is then defined to be party α's part of the jth outcome in the SEP implemented by this LOCC protocol. The closest γ node that is an ancestor to this leaf (ancestors are closer to the root, descendants are further) identifies the operator implemented by party γ for this final outcome of the LOCC, and is therefore proportional to $\hat{\mathcal{K}}_j^{(\gamma)}$ of the SEP (this ancestral relationship between operators will be important in what follows). By doing this for each party, we may determine the product operator $\hat{\mathcal{K}}_j = \hat{\mathcal{K}}_j^{(1)} \otimes \cdots \otimes \hat{\mathcal{K}}_j^{(P)}$ associated with each leaf node. The SEP implemented by this LOCC protocol is then defined by the collection of distinct operators $\{\hat{\mathcal{K}}_j\}_{j=1}^N$.[1] By Theorem 1 of [20] and the fact that we are restricting our discussion to finite-round protocols, we may, without loss of generality, assume N is finite.

The method of [23,24] constructs an LOCC protocol for a SEP by finding intersections of convex cones formed from subsets of the local operators $\hat{\mathcal{K}}_j^{(\alpha)}$ for each party α. Consideration of the extreme rays[2] of convex cones generated by these operators will lead us to our main result. The basic idea underlying this result is that too many extreme rays means that one cannot find enough intersections to piece together the full puzzle into a single protocol that incorporates all of the operators defining the given SEP.

Let us count the distinct extreme rays in the convex cone generated by the set of local operators $\{\hat{\mathcal{K}}_j^{(\alpha)}\}_{j=1}^N$ for each party α, and define this number to be e_α. Then we have the following theorem:

Theorem 1: For any finite-round LOCC protocol of P parties implementing a separable operation corresponding to the N distinct positive product operators $\{\hat{\mathcal{K}}_j = \hat{\mathcal{K}}_j^{(1)} \otimes \cdots \otimes \hat{\mathcal{K}}_j^{(P)}\}_{j=1}^N$, it must be that

$$\sum_{\alpha=1}^{P} e_\alpha \leqslant 2(N-1), \tag{4}$$

where e_α is the number of distinct extreme rays in the convex cone generated by operators $\{\hat{\mathcal{K}}_j^{(\alpha)}\}_{j=1}^N$, and the sum includes only those parties for which at least one of these local operators is not proportional to the identity. The upper bound in (4) can be achieved with equality when $N \leqslant 2^P$.

In Sec. III, we give a proof of this theorem. Prior to that, in Sec. II, we show that for proving necessary conditions for LOCC, one can restrict consideration to a special class of LOCC protocols, which we refer to as "canonical." In Sec. IV, we construct separable operations for every $P \geqslant 2$, each one having a unique representation in terms of product Kraus operators, and which satisfy $\sum_\alpha e_\alpha = PN$, the maximum possible value of this sum, showing that the bound in the theorem is extensively violated by separable operations. The uniqueness of the product representation for each of these separable operations implies that any implementation by LOCC must be in terms of the set of Kraus operators in that specific representation, since nonproduct representations cannot be implemented by LOCC. It therefore follows that in the sense of this theorem, these separable operations are as far from LOCC as possible. Finally, we offer our conclusions in Sec. V.

---

[1] See Theorem 2 and the accompanying discussion in [23] for an explanation of why all our results hold equally well when one's interest is in the Kraus operators implemented by the protocol, rather than just the $\hat{\mathcal{K}}_j$.

[2] A ray is a half-line of the form $\{\lambda \hat{\mathcal{K}}_j^{(\alpha)} | \lambda \geqslant 0\}$, and we will sometimes refer to $\hat{\mathcal{K}}_j^{(\alpha)}$ as a "ray," by which we will mean that this operator generates the ray through multiplication by non-negative scalars λ. An extreme ray of a convex cone is a ray that lies in the cone but cannot be written as a positive linear combination of other rays in that cone.

II. CANONICAL LOCC PROTOCOLS

Before proceeding to the proof of our main result, we show that in proving a necessary condition for LOCC, we can restrict consideration to the special class of "canonical" LOCC protocols, defined below. We begin by arguing that one need only consider SEPs having operators $\hat{\mathcal{K}}_j$ such that each of the P parties has at least one local operator $\hat{\mathcal{K}}_j^{(\alpha)}$ that is not the identity. Consider any LOCC protocol involving $\widetilde{P}$ parties, with the collection of final outcomes corresponding to the set of positive operators $\{\hat{\mathcal{K}}_j\}_{j=1}^N$. If for each $j = 1, \ldots, N$ a given party only does an isometry, so that for this party $\hat{\mathcal{K}}_j^{(\alpha)} \propto I_\alpha \forall j$, then that party can simply do those isometries at the end of the protocol. Then it is immediate that an LOCC exists for this SEP on $\widetilde{P}$ parties if and only if one exists for the SEP on $\widetilde{P} - 1$ parties obtained by simply deleting that one party's local operators from the $\hat{\mathcal{K}}_j$. Hence, in proving a necessary condition for LOCC, we need only consider SEPs having operators $\hat{\mathcal{K}}_j$ such that each of the P parties has at least one local operator $\hat{\mathcal{K}}_j^{(\alpha)}$ that is not the identity.

We next recall the following fairly straightforward result, which is Lemma 4 in [23].

Lemma 2: For any LOCC protocol involving local measurements having two or more proportional outcomes, there is a corresponding LOCC protocol with no two outcomes of any measurement proportional to each other, but which implements the exact same separable operation as the original protocol, including reproducing the same weights for each positive operator $\hat{\mathcal{K}}_j$ defining the separable operation.

In addition, if a party performs an isometry ("measurement" with only one outcome) at any stage of an LOCC protocol, they could just as well have absorbed that isometry into their subsequent measurement, omitting the round in which they had implemented the isometry (see the paragraph following Lemma 1 in [23] for details). Therefore, we can restrict consideration to LOCC protocols for which every round involves a measurement having at least two outcomes.

It is then a simple matter to revise any such LOCC protocol into another for which every measurement has exactly two outcomes, with the latter implementing the exact same separable operation as the former. This can be done with a replacement of each measurement having more than two outcomes by a sequence of measurements having exactly two outcomes each. It is also readily demonstrated that if the original protocol has no two proportional outcomes in any individual measurement, then one can choose each replacement sequence of two-outcome measurements such that none of these measurements has its two outcomes proportional to each other.

These observations tell us that in proving necessary conditions for LOCC, we can (and will, in what follows) restrict consideration to the following special class of LOCC protocols, which we refer to as canonical LOCC protocols.

Definition 3: A "canonical" LOCC protocol is one where every local measurement has exactly two outcomes, and in each of these measurements, the two outcomes are not proportional to each other. Every such protocol can therefore be represented by a tree with every nonleaf node having exactly two child nodes (these are commonly known as full binary trees). Each node is labeled by a positive operator as defined in (2), and for any given node, the positive operators representing its two child nodes are not proportional to each other. We will refer to such trees as canonical LOCC trees.

We now turn to the proof of our theorem.

III. PROOF OF THEOREM 1

As we will be dealing with full binary trees, the following well-known theorem will be useful.

Theorem 4: For a full binary tree, the number of leaf nodes exceeds the number of nonleaf nodes by exactly 1 [25].

We will also need the following lemma, proved in Appendix A, concerning the location of nodes in a canonical LOCC tree that may be extreme rays.

Lemma 5: In a canonical LOCC tree, a node n representing an outcome of a measurement by party α cannot be an extreme ray if there is an α node that is a descendant of node n.

We are now ready to prove Theorem 1.

Proof of Theorem 1. Consider first the special case where, in a finite canonical LOCC tree, each distinct outcome (each $\hat{\mathcal{K}}_j$) appears once and only once as a leaf of the tree. Then, the number of leaves on the tree is equal to the number of outcomes, N. Since the protocol begins with all parties having yet to make a measurement, we may label the root of the tree as I_α for some α (it is immaterial for our purposes which α is chosen). Now, I_α is not an extreme ray except when it is the *only* ray in the cone, which cannot happen for the canonical protocols we are considering. Therefore, since the root node is not extreme, and since every extreme ray (indeed, every local operator $\hat{\mathcal{K}}_j^{(\alpha)}$) in the corresponding SEP is represented by a node in the tree, $\sum_\alpha e_\alpha$ cannot exceed one less than the number of nodes. Since the total number of nodes is $2N - 1$ by Theorem 4, we find that $\sum_\alpha e_\alpha \leqslant 2(N - 1)$ as claimed.

In general, however, there will be repeated outcomes: multiple leafs will correspond to the same outcome of the SEP (the same $\hat{\mathcal{K}}_j$). Therefore, we need a way to count extreme rays without counting repetitions of those already counted. We will do this by removing nodes in a way in which the tree remains a full binary tree at every stage of the process, and which leaves at least one instance of each extreme ray. Note that by removing nodes, the remaining tree will no longer correspond to an LOCC protocol, but this is unimportant as our only purpose is to count everything in an appropriate manner.

Depict the tree with the root at the top and branches extending downward to the right and left. Each nonleaf node is parent to two child nodes, each of which is, in turn, the root of what we may refer to as a child subtree of that parent. Notice that every pair of nodes has a closest common ancestor. If that ancestor is not one of the pair, the node that is in the right child subtree of this common ancestor is "to the right" of the other, which is necessarily in the left child subtree (according to this definition, a node that is an ancestor to another node is neither to the right nor to the left of that other one). For each j, there is therefore a rightmost $\hat{\mathcal{K}}_j$ leaf (a leaf is never ancestor to another leaf), which we choose as the "keeper" $\hat{\mathcal{K}}_j$ leaf. In this way we obtain N keeper leafs, where for each j

any nonkeeper $\hat{\mathcal{K}}_j$ leaf is to the left of its respective keeper leaf. All nonkeeper leafs will be removed in a way that leaves a full binary tree with N leaf nodes.

As discussed in the next paragraph, nonkeeper leafs will be removed as part of a subtree, and this is done in two different ways, which we now describe. Consider a subtree T of the full tree, where T has no keeper leafs in it. Denote the parent of (the root of) T as n_p, the other child of this parent as n_c. Of the collection of $\hat{\mathcal{K}}_j$ leafs in T (with j ranging over all values present in T), if at least one of the corresponding keeper $\hat{\mathcal{K}}_j$ leafs is not a descendant of n_p, then remove n_p along with the entire subtree, T. The tree is kept as a full binary tree by adding an edge from the parent of n_p to the remaining child node n_c. We will refer to this as a "type-1" removal. "Type-2" removals will be used when every leaf in T has its corresponding keeper leaf as a descendant of n_p, and thus of n_c, in which case we will remove n_c along with T, reattaching the children of n_c as children, still siblings, of n_p, and taking care to preserve the right-left relationship between these children. (Under these circumstances, it turns out that n_c cannot be a leaf; see Appendix B for a proof.) Again, the tree remains full binary. See Fig. 1 for an illustration of these two types of removals.[3] They are chosen to guarantee that there is at least one instance of every extreme ray still present in the fully pruned tree; this will be proved below.

The following is how we will prune the tree. At every stage including the first, consider the leftmost nonkeeper leaf. If the sibling subtree of this nonkeeper leaf has no keepers, consider instead the entire subtree for which the parent of these subtrees is the root (that is, consider both sibling subtrees and their parent as a single subtree). Then, if the sibling subtree of this larger subtree has no keepers, combine these two subtrees with their parent, and consider this larger subtree. Continue in this fashion until a keeper leaf is encountered in the sibling subtree, having thus found a "maximal" keeperless subtree. Remove this entire maximal subtree as either type 1 or type 2, whichever is appropriate. Then, find the leftmost nonkeeper leaf in the tree that remains, and repeat this process until all nonkeeper leafs have been removed.

Having removed all nonkeeper leafs, the N keepers are the only leafs remaining in the fully pruned tree, as desired. Furthermore, a nonleaf node is only removed if it is within a subtree that has no keeper leaf in it, or if it is that extra nonleaf node that is removed along with one of those subtrees. This implies that the root of the entire original tree is never removed: the only subtree it is within is the full original tree, which obviously has a keeper; and if either child subtree of the root has no keeper leaf, then the other subtree has in it every keeper corresponding to the nonkeepers in that first subtree,

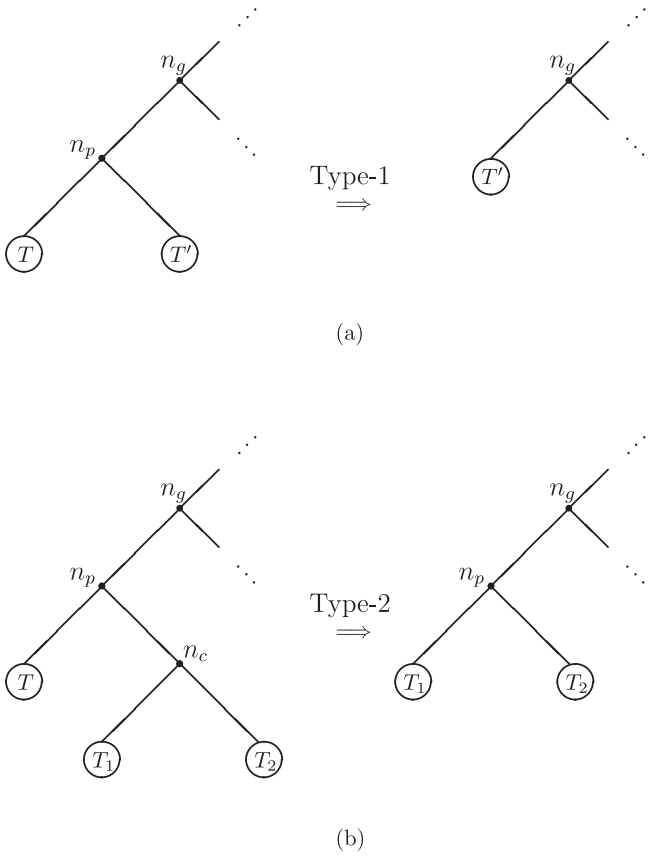

FIG. 1. Illustration of the two types of removals used in pruning a canonical LOCC tree. (a) A type-1 removal, where the parent n_p is removed along with the maximal keeperless subtree T. This type of removal is used when there is at least one $\hat{\mathcal{K}}_j$ leaf in T whose corresponding keeper is not in T's sibling subtree, T'. The root of T' is n_c, which may be the only node in T', in which case it turns out that n_c is itself a keeper leaf. (b) A type-2 removal where n_c, which is the root of the sibling subtree of T, is removed with T. This type of removal is used when every leaf in T has its corresponding keeper leaf in either T_1 or T_2. Under these circumstances, n_c cannot be a leaf.

so the first subtree is removed as type 2, in which case it is the root of the other subtree that is removed as the extra nonleaf, rather than the root of the entire tree. Hence, the root of the entire original tree is still present as the root of the entire fully pruned tree.

We now prove that at least one instance of each extreme ray always remains in the fully pruned tree. Suppose, by contradiction, that for some fixed α and each $j \in \mathcal{J}$ with index set $\mathcal{J} \subseteq \{1,2,\ldots,N\}$, $\hat{\mathcal{K}}_j^{(\alpha)} =: \hat{\mathcal{K}}_*^{(\alpha)}$ is a (single) extreme ray whose every appearance is removed in our procedure for pruning the tree. This means that no keeper leaf can be $\hat{\mathcal{K}}_*^{(\alpha)}$, since keeper leafs are never removed. Therefore, each $\hat{\mathcal{K}}_j$ keeper leaf with $j \in \mathcal{J}$ has a $\hat{\mathcal{K}}_*^{(\alpha)}$ node as its closest α-node ancestor in the original tree. Indeed, since by Lemma 5 each $\hat{\mathcal{K}}_*^{(\alpha)}$ node has no α-node ancestors itself, it is also the case that each $\hat{\mathcal{K}}_*^{(\alpha)}$ node is a closest α-node ancestor to at least one $\hat{\mathcal{K}}_j$ leaf, $j \in \mathcal{J}$ (at least two such leafs, actually, in this canonical tree). Recall that each nonkeeper is to the left of its respective

[3]Notice that the tree structure induces a partial order among the nodes, having to do with whether or not two nodes are an ancestor descendant of one another. As should be clear from Fig. 1, if a pair of nodes are (are not) an ancestor (descendant) of one another after a removal, then they were (were not) an ancestor (descendant) before the removal.

keeper leaf and consider the rightmost appearance of $\hat{\mathcal{K}}_*^{(\alpha)}$ in the original tree.[4] This rightmost appearance will be ancestor to a keeper $\hat{\mathcal{K}}_j$ leaf with $j \in \mathcal{J}$, since otherwise the nonkeeper it is ancestor to would be to the right of its respective keeper. Since this $\hat{\mathcal{K}}_*^{(\alpha)}$ node is ancestor to a keeper, it is not removed as part of an entire subtree T (T is only removed if it has no keepers), so it must be removed as the extra nonleaf that is removed along with T. For type-2 removals, it turns out that the extra nonleaf n_c is not extreme (see Appendix B for a proof), so this $\hat{\mathcal{K}}_*^{(\alpha)}$ node must be removed as type 1, that is, as parent n_p of T. For the (generally, partially pruned) tree to which this type-1 removal is applied, then according to how we decide which type of removal to use, there exists a $\hat{\mathcal{K}}_i$ leaf in T, which is a descendant of n_p, whose corresponding keeper is not a descendant of n_p. Since ancestral relationships are not altered during pruning[3] (for those nodes that remain in the tree), this means that in the original tree, this nonkeeper $\hat{\mathcal{K}}_i$ leaf in T is a descendant of n_p and its corresponding keeper is not. Thus, the keeper $\hat{\mathcal{K}}_i$ leaf is to the right of n_p in the original tree, since that keeper is to the right of a descendant of n_p (that nonkeeper) and is not itself a descendant of n_p, implying that keeper is in the right child subtree of its closest common ancestor with n_p, and n_p is not that closest common ancestor. Also in the original tree, this keeper $\hat{\mathcal{K}}_i$ leaf is either $\hat{\mathcal{K}}_i^{(\alpha)}$ or else has a $\hat{\mathcal{K}}_i^{(\alpha)}$ ancestor, and in either case this $\hat{\mathcal{K}}_i^{(\alpha)}$ node cannot be a descendant of n_p (or else the keeper $\hat{\mathcal{K}}_i$ leaf would be a descendant of n_p), so is either ancestor to n_p or is to the right of n_p. If it is ancestor to n_p, $\hat{\mathcal{K}}_i^{(\alpha)} \neq \hat{\mathcal{K}}_*^{(\alpha)}$, because $\hat{\mathcal{K}}_i^{(\alpha)}$ has n_p as an α-node descendant, so by Lemma 5 cannot be extreme, which $\hat{\mathcal{K}}_*^{(\alpha)}$ is, by assumption. If, on the other hand, this $\hat{\mathcal{K}}_i^{(\alpha)}$ is to the right of n_p, we also have that $\hat{\mathcal{K}}_i^{(\alpha)} \neq \hat{\mathcal{K}}_*^{(\alpha)}$, because n_p is the rightmost $\hat{\mathcal{K}}_*^{(\alpha)}$ node. In either case, the nearest α node to that nonkeeper $\hat{\mathcal{K}}_i$ leaf in the child subtree T of n_p is also $\hat{\mathcal{K}}_i^{(\alpha)} \neq \hat{\mathcal{K}}_*^{(\alpha)}$, so is not n_p, implying T must have an α node in it, which is thus a descendant of n_p. Since n_p is $\hat{\mathcal{K}}_*^{(\alpha)}$, Lemma 5 then tells us that $\hat{\mathcal{K}}_*^{(\alpha)}$ is not extreme, a contradiction, proving that every extreme ray is present at least once in the fully pruned tree.

The fully pruned tree is a full binary tree, so has N leaf nodes and $2N-1$ nodes in all. Since every extreme ray is present as one of its nodes, and since the root of the original tree is not extreme and is never removed, there can therefore be no more than $2(N-1)$ extreme rays. It is shown in Appendix C that this bound can be saturated when $N \leqslant 2^P$, which completes the proof. ■

Having proved the bound of Theorem 1, we now proceed to the next section, where we demonstrate the existence of separable channels that maximally violate this bound.

[4]With $\hat{\mathcal{K}}_*^{(\alpha)}$ assumed to be extreme, no $\hat{\mathcal{K}}_*^{(\alpha)}$ node can have an α-node descendant, according to Lemma 5. Therefore, no one of these nodes can be ancestor to another one, implying that each $\hat{\mathcal{K}}_*^{(\alpha)}$ node is either to the right or left of every other one, so there is one of them that is furthest to the right.

IV. SEPARABLE CHANNELS ON P PARTIES WITH $\sum_\alpha e_\alpha = PN$

Here, we construct SEPs as sets of positive operators $\{\hat{\mathcal{K}}_j\}$ for every P and for which Theorem 1 is violated maximally, having $\sum_\alpha e_\alpha = PN$. Define operators $\hat{\mathcal{K}}_j = |\Psi_j\rangle\langle\Psi_j|, j = 1,\ldots,N$, where $|\Psi_j\rangle = (D/N)^{1/2}|\psi_j^{(1)}\rangle \otimes \cdots \otimes |\psi_j^{(P)}\rangle$, $D = d_1 d_2 \ldots d_P$, d_α is the dimension of Hilbert space $\mathcal{H}_\alpha$ with parties ordered such that $d_1 \leqslant d_2 \leqslant \cdots \leqslant d_P$, and the state on party α's subsystem is

$$\left|\psi_j^{(\alpha)}\right\rangle = \frac{1}{\sqrt{d_\alpha}} \sum_{m_\alpha=1}^{d_\alpha} e^{2\pi i j p_\alpha m_\alpha/N} |m_\alpha\rangle. \tag{5}$$

Here, $p_1 = 1$ and for $\alpha \geqslant 2$, $p_\alpha = d_1 d_2 \ldots d_{\alpha-1}$, $|m_\alpha\rangle$ is the standard basis for party α, and N is chosen as any prime number exceeding D. Since for each positive operator $\hat{\mathcal{K}}_j$, the local parts are the rank-1 projectors $\hat{\mathcal{K}}_j^{(\alpha)} = |\psi_j^{(\alpha)}\rangle\langle\psi_j^{(\alpha)}|$, each $\hat{\mathcal{K}}_j^{(\alpha)}$ is thus an extreme ray of its respective convex cone. This means that $e_\alpha = N \forall \alpha$ and $\sum_\alpha e_\alpha = PN$, an extensive violation of Theorem 1 and the maximal possible value of this sum.

We need to show that the set $\{\hat{\mathcal{K}}_j\}$ satisfies closure, $\sum_j \hat{\mathcal{K}}_j = I$. We have

$$\begin{aligned}\sum_{j=1}^{N} \hat{\mathcal{K}}_j &= \frac{1}{N} \sum_{m_1,n_1=1}^{d_1} \cdots \sum_{m_P,n_P=1}^{d_P} \left(\sum_{j=1}^{N} e^{2\pi i j \sum_\alpha p_\alpha (m_\alpha - n_\alpha)/N} \right) \\ &\quad \times |m_1 \ldots m_P\rangle\langle n_1 \ldots n_P| \\ &= \sum_{m_1,n_1=1}^{d_1} \cdots \sum_{m_P,n_P=1}^{d_P} \delta\left(\sum_{\alpha=1}^{P} p_\alpha m_\alpha, \sum_{\alpha=1}^{P} p_\alpha n_\alpha \right) \\ &\quad \times |m_1 \ldots m_P\rangle\langle n_1 \ldots n_P|, \end{aligned} \tag{6}$$

where $\delta(\cdot,\cdot)$ is the Kronecker delta, vanishing unless its two arguments are equal, in which case it is equal to unity. Recall that $p_1 = 1$ and $p_\alpha = d_1 d_2 \ldots d_{\alpha-1}, \alpha \geqslant 2$. Suppose $m_P \neq n_P$. Then, $p_P |m_P - n_P| \geqslant p_P = d_1 d_2 \ldots d_{P-1} > \sum_{\alpha=1}^{P-1} p_\alpha |m_\alpha - n_\alpha|$, since the right-hand side of this inequality is no greater than $d_1 - 1 + d_1(d_2 - 1) + d_1 d_2 (d_3 - 1) + \cdots + d_1 d_2 \ldots d_{P-2}(d_{P-1} - 1) = d_1 d_2 \ldots d_{P-1} - 1$. We conclude that equality of the two arguments in the Kronecker delta in the last line of (6) requires $m_P = n_P$. Similar arguments, proceeding sequentially with decreasing α starting next from $\alpha = P - 1$, shows that $m_\alpha = n_\alpha \forall \alpha$. Thus, the right-hand side of (6) is equal to I, the identity on the full input Hilbert space, as desired.

Finally, for each P, we show the existence of sets of Kraus operators corresponding to the positive operators given in the previous paragraph, which are the unique product Kraus representation for their associated quantum channel. Define Kraus operators $\hat{K}_j = |\Phi_j\rangle\langle\Psi_j|$, with $|\Psi_j\rangle$ defined above (5). Define normalized states $|\Phi_j\rangle = |\phi_j^{(1)}\rangle \otimes |\Phi'_j\rangle$ with $\{|\Phi'_j\rangle\}_{j=1}^N$ a set of linearly independent product states on the $P-1$ parties excluding party 1. (This requires that at least one output dimension of those last $P-1$ parties exceeds its input, in order that the overall output dimension is not less than N, the number of these independent states.) Then, since no two of the $|\psi_j^{(1)}\rangle$ are proportional to each other, the conditions of Theorem 2 of

[26] are met for this set of Kraus operators (with a bipartite split between party 1 and all the rest), which implies that no linear combination of the $\hat{K}_j$ is a product operator (apart from the $\hat{K}_j$'s themselves). This, in turn, implies that the set $\{\hat{K}_j\}_{j=1}^N$ is the unique product representation for the given channel. Therefore, there is no other Kraus representation that could possibly be LOCC, and these channels are as far from LOCC as possible, in the sense of Theorem 1, as claimed.

V. CONCLUSIONS

In summary, we have proved a necessary condition for any finite-round LOCC protocol, stated in Theorem 1. We have also demonstrated that this necessary condition is violated extensively by separable operations, this violation growing without bound as the number of parties increases. Considering any given product Kraus representation of a separable channel, violating our bound becomes a sufficient condition that this representation cannot be exactly implemented by LOCC.

One direction for future work is to take a more detailed look at extreme rays. As an example, suppose $\hat{\mathcal{K}}_1^{(1)}$ and $\hat{\mathcal{K}}_1^{(2)}$ are each extreme and are not proportional to any of the other local operators, $\hat{\mathcal{K}}_j^{(\alpha)}, j \neq 1, \alpha = 1,2$, respectively. We may say that $\hat{\mathcal{K}}_1$ contains two "singly extreme" rays. Then by Corollary 1 of [23], this SEP cannot be exactly implemented by any finite-round LOCC protocol. The reason is that in order for $\hat{\mathcal{K}}_1$ to be part of an LOCC tree involving the other $\hat{\mathcal{K}}_j$, either the ray $\hat{\mathcal{K}}_1^{(1)}$ must intersect the cone of the other $\hat{\mathcal{K}}_j^{(1)}$ or the ray $\hat{\mathcal{K}}_1^{(2)}$ must intersect the cone of the other $\hat{\mathcal{K}}_j^{(2)}$, neither of which is the case. Although we have no proof, we suspect that this SEP cannot be implemented even with the use of an infinite number of rounds, because if it could, then the SEP could be approximated as closely as desired by using a large enough (but finite) number of rounds. However, if the approximation is close enough, one expects that the $\hat{\mathcal{K}}_j'^{(\alpha)}$ of this approximation will partition into two sets—one close to the extreme ray $\hat{\mathcal{K}}_1^{(\alpha)}$, the other close to the other $\hat{\mathcal{K}}_j^{(\alpha)}, j \neq 1$—such that there will be no nontrivial intersection of the cones from these two sets, and this will be so for each of the parties $\alpha = 1,2$. By an argument similar to that just used for the $\hat{\mathcal{K}}_j$, one could then conclude that this approximation cannot be implemented by finite-round LOCC, a contradiction. Therefore, we conjecture that SEPs of this sort cannot be approximated arbitrarily closely by LOCC. This is especially interesting in the bipartite case, for which a violation of the conditions of our Theorem 1 necessarily implies that at least one $\hat{\mathcal{K}}_j$ contains two singly extreme rays. Hence, a proof of this conjecture would also extend our Theorem 1 to infinite-round LOCC, in the case of two parties. While the same conclusion does not immediately follow for $P > 2$, it would certainly be suggestive that our theorem may well apply to infinite-round LOCC for any number of parties.

Another avenue we are pursuing is to look at the number of distinct rays D_α in the cones of the various parties, rather than just counting those rays that are extreme. For two parties, it is not difficult to see from the results of [24] that a necessary condition for finite-round LOCC is $\sum_\alpha D_\alpha \leqslant 2N - 1$. This bound, which one can show is tight for any $N > 1$, provides the first general way of distinguishing between finite- and infinite-round LOCC, as demonstrated by the fact we have succeeded in constructing bipartite SEPs that can be exactly implemented using an infinite number of rounds, but which satisfy $\sum_\alpha D_\alpha = 2N$. For $P = 3$, we have proved that $\sum_\alpha D_\alpha \leqslant 3N - 5$ with finite-round LOCC, a tight bound that applies if (and only if) $N \geqslant 6$. We suspect that this upper bound can also be exceeded using infinite rounds, a question we are presently studying, and we are attempting to extend these results to larger P, as well.

ACKNOWLEDGMENTS

The author would like to thank Li Yu for helpful suggestions and especially Dan Stahlke for his numerous and very helpful comments and suggestions. This work has been supported in part by the National Science Foundation through Grant No. 1205931.

APPENDIX A: PROOF OF LEMMA 5

If α-node n has another α node that is its descendant, then party α made a measurement after the measurement that produced node n. Then, by (3) of the main text, the positive operator labeling node n is a sum of those positive operators labeling the descendant α nodes produced by the later measurement. In turn, some of those descendants may themselves have other descendant α nodes produced by a subsequent measurement, and so are sums of those operators corresponding to the latter nodes. Eventually, if we continue toward the leaves, we end up with nodes that are labeled by the $\hat{\mathcal{K}}_j^{(\alpha)}$ defining the separable operation, so that node n is seen to be labeled by a sum of those $\hat{\mathcal{K}}_j^{(\alpha)}$. This means node n cannot be an extreme ray in the convex hull of the $\hat{\mathcal{K}}_j^{(\alpha)}$ unless all the $\hat{\mathcal{K}}_j^{(\alpha)}$ entering this sum are proportional to each other, which is impossible in the canonical LOCC trees we are considering.■

APPENDIX B: PROOF THAT n_c IS A NONLEAF THAT IS NOT EXTREME FOR ALL TYPE-2 REMOVALS

For type-2 removals (of subtree T and its present sibling n_c), we need to show that n_c is a nonleaf that is not extreme. Our argument will utilize the fact that n_c and the root of T were siblings in the original tree, so we need to be sure this is always the case, even following earlier removals. The only way siblings change during the pruning process is via type-1 removals, since type-2 removals do not change sibling relationships. Suppose there was a type-1 removal of $\widetilde{T}$, child subtree of $\tilde{n}_p$, where the other child subtree of $\tilde{n}_p$ was $\widetilde{T}'$ and the sibling of $\tilde{n}_p$ before this removal was $\tilde{n}'_p$. This removal only changes sibling relationships by changing the sibling of $\tilde{n}'_p$ from $\tilde{n}_p$ to the root of $\widetilde{T}'$, which we denote as $\tilde{n}'_c$ (see Fig. 2).

We now argue that there will never be a type-2 removal involving nodes at the positions of this newly created sibling pair, even if their identities change further through subsequent pruning. Since $\widetilde{T}'$ presently has a keeper leaf (because otherwise the removed $\widetilde{T}$ was not a maximal keeperless subtree, which are the only ones we remove), it will always have a keeper no matter how much further pruning occurs, since keepers are never removed. Hence, a type-2 removal involving

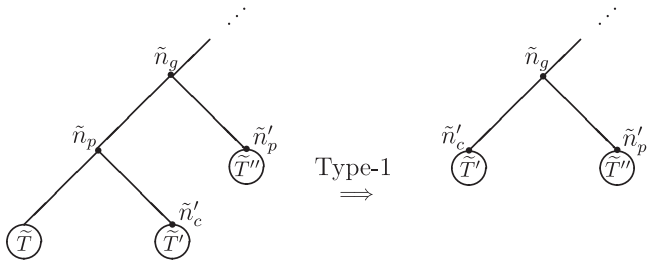

FIG. 2. A type-1 removal creates a new sibling pair, $\tilde{n}'_c$ and $\tilde{n}'_p$, but this pair will never be involved in a subsequent type-2 removal, showing that type-2 removals always involve siblings that were siblings in the original tree.

these nodes can only occur if the new sibling sub-tree of $\widetilde{T}'$ (call this sub-tree $\widetilde{T}''$) is keeperless, which means it was keeperless to begin with, even before the previous type-1 removal that changed the sibling pair. This means that $\widetilde{T}''$ is to the right of $\widetilde{T}'$, because otherwise $\widetilde{T}''$ would have been a maximal keeperless subtree, removed before that previous type-1 removal, since the pruning proceeds from left to right. However, if it is to the right, then the keeper leafs corresponding to the nonkeepers in $\widetilde{T}''$ are further to the right and therefore not in $\widetilde{T}'$, so removal of $\widetilde{T}''$ will be via a type-1 removal, not type 2. This conclusion is true no matter what pruning takes place between the previous type-1 removal and this one, because no pruning can change the fact that the keepers corresponding to the nonkeepers in $\widetilde{T}''$ are to the right of $\widetilde{T}''$ and not in $\widetilde{T}'$. Therefore, all type-2 removals involve siblings that were siblings in the original tree.

Consider a type-2 removal of T, a child subtree of n_p, with n_p's other child n_c removed along with T. We can now prove that n_c is not extreme. From the discussion above, we know that the root of T was sibling to n_c in the original (canonical) tree, so these two are not proportional and are both α nodes for the same α. Assume, by contradiction, n_c is extreme. Then by Lemma 5, n_c is an α node with no α-node descendants, so n_c must be proportional to $\hat{\mathcal{K}}_j^{(\alpha)}$ for every j such that $\hat{\mathcal{K}}_j$ is one of the keeper leafs that are descendants of n_c. Since every leaf in T has its corresponding keeper leaf as a descendant of n_c, the closest α node to each and every leaf in T must also be proportional to this same $\hat{\mathcal{K}}_j^{(\alpha)}$ (since the root of T is an α node, there is at least one such node in T). Therefore, by (3) of the main text, every α node in T, including its root node, is proportional to this same $\hat{\mathcal{K}}_j^{(\alpha)}$. In other words, the root of T is proportional to n_c, a contradiction, proving that n_c is not extreme. Repeating the exact same argument starting from the assumption n_c is a leaf leads to the same contradiction, thus proving n_c is a nonleaf, and we are done.

APPENDIX C: SATURATING THE BOUND IN THEOREM 1

We will now show that any LOCC protocol satisfying the following two conditions saturates the bound in Theorem 1, $\sum_\alpha e_\alpha = 2(N-1)$:

(1) Each party measures once and only once with the same ordering of the parties no matter which outcomes were obtained by the preceding parties. In addition, each of these measurements has exactly two outcomes.

Along any branch of the associated LOCC tree, party 1 starts with a two-outcome measurement, followed by a two-outcome measurement by party 2, which is followed by a two-outcome measurement by party 3, and so on until each party has measured once, party P always making the final measurement. For the entire protocol, party α has $2^{\alpha-1}$ different measurements, which measurement that party makes being determined by the outcomes of all previous parties' measurements. Party α has a total of $2\times 2^{\alpha-1} = 2^\alpha$ measurement outcomes.

(2) Each of the 2^α outcomes for party α is a distinct extreme ray in the cone generated by the collection of these outcomes. Note that for a protocol satisfying the preceding condition 1, many of the $\hat{\mathcal{K}}_j^{(\alpha)}$ (fixed α but different j) will be equal to each other, which explains why party $\alpha \neq P$ has only 2^α extreme rays, rather than the maximum possible number, $N = 2^P$.

Here is one specific example that does the trick. For each of party α's measurements, indexed by $m = 1,\ldots,2^{\alpha-1}$, let one Kraus operator be a projector onto (normalized) pure state $|\xi_m^{(\alpha)}\rangle$, with the other outcome $I_\alpha - |\xi_m^{(\alpha)}\rangle\langle\xi_m^{(\alpha)}|$. As these are both projectors, the $\hat{\mathcal{K}}_j^{(\alpha)}$ are equal to these Kraus operators:

$$\begin{aligned}\hat{\mathcal{K}}_{2m-1}^{(\alpha)} &= \left|\xi_m^{(\alpha)}\right\rangle\!\left\langle\xi_m^{(\alpha)}\right|,\\ \hat{\mathcal{K}}_{2m}^{(\alpha)} &= I_\alpha - \left|\xi_m^{(\alpha)}\right\rangle\!\left\langle\xi_m^{(\alpha)}\right|.\end{aligned} \tag{C1}$$

Choose the set of pure states $|\xi_m^{(\alpha)}\rangle$ in each $\mathcal{H}_\alpha$ such that no two are the same and no two are orthogonal to each other. Then the following argument shows that each and every $\hat{\mathcal{K}}_j^{(\alpha)}$ is an extreme ray in the convex cone generated by the collection of all of them (many are repeated, as explained above): The pure state projectors $\hat{\mathcal{K}}_{2m-1}^{(\alpha)}$ are each an extreme ray in the cone of the full set of positive operators acting on $\mathcal{H}_\alpha$, so are necessarily also extreme in the cone of the collection of $\hat{\mathcal{K}}_j^{(\alpha)}$. If $\mathcal{H}_\alpha$ is two-dimensional, then $\hat{\mathcal{K}}_{2m}^{(\alpha)}$ is also extreme for each m, by the same argument. Otherwise, $\hat{\mathcal{K}}_{2m}^{(\alpha)}$ has rank exceeding unity so is not extreme in the cone of all positive operators, but is nonetheless on the boundary of that set, since $\hat{\mathcal{K}}_{2m}^{(\alpha)}|\xi_m^{(\alpha)}\rangle = 0$. Note also that $p_j := \langle\xi_m^{(\alpha)}|\hat{\mathcal{K}}_j^{(\alpha)}|\xi_m^{(\alpha)}\rangle > 0$ for every $j \neq 2m$. Suppose $\hat{\mathcal{K}}_{2m}^{(\alpha)} = \sum_{j\neq 2m} c_j \hat{\mathcal{K}}_j$, with $c_j \geqslant 0$. Taking the diagonal element, $\langle\xi_m^{(\alpha)}|\cdots|\xi_m^{(\alpha)}\rangle$, of this equation leads to $0 = \sum_{j\neq 2m} c_j p_j$, or $c_j = 0 \forall j \neq 2m$, a contradiction since $\hat{\mathcal{K}}_{2m}^{(\alpha)} \neq 0$. Therefore, $\hat{\mathcal{K}}_{2m}^{(\alpha)}$ cannot be written as a positive linear combination of all the others and so is an extreme ray in the cone of their collection. We conclude that each and every $\mathcal{K}_j^{(\alpha)}$ is an extreme ray, $e_\alpha = 2^\alpha \forall \alpha$, and $\sum_\alpha e_\alpha = 2 + 2^2 + \cdots + 2^P = 2(2^P - 1) = 2(N-1)$, saturating the bound.

If the last party omits his measurement for one of the outcomes of the next-to-last party, this removes a pair of leaf nodes, replacing it with a single leaf. Therefore, this reduces N by unity to $2^P - 1$. The number of extreme rays is reduced by two, since the new leaf was counted as extreme before the pair of leafs was removed, so the bound is still saturated. By continually omitting single measurements by the last party to measure along a given branch, N is reduced by unity for each omission, and the number of extreme rays is reduced by two. In this way, we can obtain examples saturating the bound for any N satisfying $P + 1 \leqslant N \leqslant 2^P$ [N cannot be less than $P + 1$, since all P parties perform a nontrivial measurement

at least once in the full protocol; alternatively, one may note that according to the way we count parties, $e_\alpha \geqslant 2 \forall \alpha$, so $2P \leqslant \sum e_\alpha \leqslant 2(N-1)$, which also shows $N \geqslant P+1$]. This completes the proof of Theorem 1.

[1] J. Walgate, A. J. Short, L. Hardy, and V. Vedral, Phys. Rev. Lett. **85**, 4972 (2000).
[2] C. H. Bennett, G. Brassard, S. Popescu, B. Schumacher, J. A. Smolin, and W. K. Wootters, Phys. Rev. Lett. **76**, 722 (1996).
[3] C. H. Bennett, H. J. Bernstein, S. Popescu, and B. Schumacher, Phys. Rev. A **53**, 2046 (1996).
[4] J. I. Cirac, W. Dür, B. Kraus, and M. Lewenstein, Phys. Rev. Lett. **86**, 544 (2001).
[5] M. A. Nielsen, Phys. Rev. Lett. **83**, 436 (1999).
[6] F. Anselmi, A. Chefles, and M. B. Plenio, New J. Phys. **6**, 164 (2004).
[7] M. Hillery, V. Buzek, and A. Berthiaume, Phys. Rev. A **59**, 1829 (1999).
[8] D. W. Berry, Phys. Rev. A **75**, 032349 (2007).
[9] L. Yu, R. B. Griffiths, and S. M. Cohen, Phys. Rev. A **81**, 062315 (2010).
[10] A. Chefles, Phys. Rev. A **69**, 050307 (2004).
[11] E. M. Rains, Phys. Rev. A **60**, 173 (1999).
[12] C. H. Bennett, D. P. DiVincenzo, C. A. Fuchs, T. Mor, E. Rains, P. W. Shor, J. A. Smolin, and W. K. Wootters, Phys. Rev. A **59**, 1070 (1999).
[13] B. Fortescue and H.-K. Lo, Phys. Rev. Lett. **98**, 260501 (2007).
[14] E. Chitambar, W. Cui, and H.-K. Lo, Phys. Rev. Lett. **108**, 240504 (2012).
[15] E. Chitambar, W. Cui, and H.-K. Lo, Phys. Rev. A **85**, 062316 (2012).
[16] H.-K. Lo and S. Popescu, Phys. Rev. A **63**, 022301 (2001).
[17] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters, Phys. Rev. A **54**, 3824 (1996).
[18] M. Kleinmann, H. Kampermann, and D. Bruß, Phys. Rev. A **84**, 042326 (2011).
[19] E. Chitambar, Phys. Rev. Lett. **107**, 190502 (2011).
[20] E. Chitambar, D. Leung, L. Mančinska, M. Ozols, and A. Winter, Commun. Math. Phys. **328**, 303 (2014).
[21] A. M. Childs, D. Leung, L. Mančinska, and M. Ozols, Commun. Math. Phys. **323**, 1121 (2013).
[22] K. Kraus, *States, Effects, and Operations* (Spring-Verlag, Berlin, 1983).
[23] S. M. Cohen, Phys. Rev. A **87**, 052135 (2013).
[24] S. M. Cohen, Phys. Rev. A **84**, 052322 (2011).
[25] C. A. Shaffer, *Data Structures and Algorithm Analysis in C++*, 3rd ed. (Dover, New York, 2011), Chap. 5.
[26] S. M. Cohen, J. Math. Phys. **55**, 062202 (2014).