Distributed Quantum Interactive Proofs

François Le Gall
Graduate School of Mathematics, Nagoya University, Japan

Masayuki Miyamoto
Graduate School of Mathematics, Nagoya University, Japan

Harumichi Nishimura
Graduate School of Informatics, Nagoya University, Japan

Abstract

The study of distributed interactive proofs was initiated by Kol, Oshman, and Saxena [PODC 2018] as a generalization of distributed decision mechanisms (proof-labeling schemes, etc.), and has received a lot of attention in recent years. In distributed interactive proofs, the nodes of an n-node network G can exchange short messages (called certificates) with a powerful prover. The goal is to decide if the input (including G itself) belongs to some language, with as few turns of interaction and as few bits exchanged between nodes and the prover as possible. There are several results showing that the size of certificates can be reduced drastically with a constant number of interactions compared to non-interactive distributed proofs.

In this paper, we introduce the quantum counterpart of distributed interactive proofs: certificates can now be quantum bits, and the nodes of the network can perform quantum computation. The first result of this paper shows that by using distributed quantum interactive proofs, the number of interactions can be significantly reduced. More precisely, our result shows that for any constant k, the class of languages that can be decided by a k-turn classical (i.e., non-quantum) distributed interactive protocol with $f(n)$-bit certificate size is contained in the class of languages that can be decided by a 5-turn distributed quantum interactive protocol with $O(f(n))$-bit certificate size. We also show that if we allow to use shared randomness, the number of turns can be reduced to three. Since no similar turn-reduction *classical* technique is currently known, our result gives evidence of the power of quantum computation in the setting of distributed interactive proofs as well.

As a corollary of our results, we show that there exist 5-turn/3-turn distributed quantum interactive protocols with small certificate size for problems that have been considered in prior works on distributed interactive proofs such as [Kol, Oshman, and Saxena PODC 2018, Naor, Parter, and Yogev SODA 2020].

We then utilize the framework of the distributed quantum interactive proofs to test closeness of two quantum states each of which is distributed over the entire network.

2012 ACM Subject Classification Theory of computation → Distributed algorithms; Theory of computation → Quantum computation theory

Keywords and phrases distributed interactive proofs, distributed verification, quantum computation

Digital Object Identifier 10.4230/LIPIcs.STACS.2023.42

Related Version *Full Version*: https://arxiv.org/abs/2210.01390 [9]

Funding FLG was supported by the JSPS KAKENHI grants JP16H01705, JP19H04066, JP20H00579, JP20H04139, JP20H05966, JP21H04879 and by the MEXT Q-LEAP grants JPMXS0118067394 and JPMXS0120319794. MM would like to take this opportunity to thank the "Nagoya University Interdisciplinary Frontier Fellowship" supported by Nagoya University and JST, the establishment of university fellowships towards the creation of science technology innovation, Grant Number JPMJFS212. HN was supported by the JSPS KAKENHI grants JP19H04066, JP20H05966, JP21H04879, JP22H00522 and by the MEXT Q-LEAP grants JPMXS0120319794.

40th International Symposium on Theoretical Aspects of Computer Science (STACS 2023).
Editors: Petra Berenbrink, Patricia Bouyer, Anuj Dawar, and Mamadou Moustapha Kanté;
Article No. 42; pp. 42:1–42:21

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

1.1 Distributed Interactive Proofs

In distributed computing, efficient verification of graph properties of the network is useful from both theoretical and applied aspects. The study of this notion of verification in the distributed setting has lead to the notion of "distributed NP" in analogy with the complexity class NP in centralized computation: A powerful prover provides certificates to each node of the network in order to convince that the network has a desired property; If the property is satisfied, all nodes must output "accept", otherwise at least one node must output "reject". This concept of "distributed NP" has been formulated in several ways, including *proof-labeling schemes* (PLS) [19], *non-deterministic local decision* (NLD) [5], and *locally checkable proofs* (LCP) [10].

As a motivating example, consider the problem of verifying whether the network is bipartite or not. While this problem cannot be solved in $O(1)$ rounds without prover [29], it can easily be solved as following: The prover tells each node which part it belongs to, which requires only a 1-bit certificate per node, and then each node broadcasting this information to its adjacent nodes (here the crucial point is that if the network is non-bipartite, then at least one node will be able to detect it). On the other hand, it is known that there exist properties that require large certificate size to decide: Göös and Suomela [10] have shown that recognizing symmetric graphs (Sym) and non 3-colorable graphs ($\overline{\text{3Col}}$) require $\Omega(n^2)$-bit certificates per node in the framework of LCP (which is tight since all graph properties are locally decidable by giving the $O(n^2)$-bit adjacency matrix of the graph).

To reduce the length of the certificate for such problems, the notion of distributed interactive proofs (also called distributed Arthur-Merlin proofs) was recently introduced by Kol, Oshman and Saxena [18] as a generalization of distributed NP. In this model there are two players, the prover (often called Merlin), who has unlimited computational power and sees the entire network but is untrusted (i.e., can be malicious), and the verifier (often called Arthur) representing all the nodes of the network, who can perform only local computation and brief communication with adjacent nodes. Generalizing the concept of distributed NP, the nodes are now allowed to engage in multiple turns of interaction with the prover. As for distributed NP, there are two requirements of the protocol: if the input is legal (yes-instance) then all nodes must accept with high probability (*completeness*), and if the input is illegal then at least one node must reject with high probability (*soundness*).

In the setting of [18], each node has access to a private source of randomness, and sends generated random bits to the prover in Arthur's turn. For instance, a 2-turn protocol contains two interactions: Arthur first queries Merlin by sending a random string from each node, and then Merlin provides a certificate to each node. After that, nodes exchange messages with adjacent nodes to decide their outputs. The main complexity measures when studying distributed interactive protocols are the size of certificates provided to each node, the size of the random strings generated at each node and the size of the messages exchanged between nodes. Let us denote $\mathsf{dAM}[k](f(n))$ the class of languages that have k-turn distributed Arthur-Merlin protocols where Merlin provides $O(f(n))$-bit certificates, Arthur generates $O(f(n))$-bit random strings at each node and $O(f(n))$-bit messages are exchanged between nodes. Kol et al. [18] showed the power of interaction by giving a $\mathsf{dMAM}(\log n) = \mathsf{dAM}[3](\log n)$ protocol for graph symmetry (Sym) and a $\mathsf{dAMAM}(n \log n) = \mathsf{dAM}[4](n \log n)$ protocol for graph non-isomorphism (GNI), which are known to require $\Omega(n^2)$-bit certificate in LCP (see Appendix A for the definition of these problems).

This model has been further studied in several works. Naor, Parter and Yogev [25] showed that any $O(n)$-time centralized computation can be converted into a $\mathsf{dMAM}(\log n) = \mathsf{dAM}[3](\log n)$ protocol. Using this compiler, for instance, they constructed a $\mathsf{dMAMAM}(\log\log n) = \mathsf{dAM}[5](\log\log n)$ protocol for SetEquality and a special case of Sym. Crescenzi, Fraigniaud and Paz [3] initiated the study of distributed Arthur-Merlin protocols with shared randomness: in each Arthur's turn, Arthur generates a random string that can be seen from all nodes. In order to distinguish the two models we use dAM for the (standard) private randomness setting and dAM^{sh} for the shared randomness setting. They showed that dAM protocols can simulate dAM^{sh} protocols by giving additional $O(\log n)$-size certificates. The role of shared randomness was further investigated by Montealegre, Ramírez-Romero and Rapaport [23], who showed the computational power of small-certificate dAM^{sh} protocols without private randomness is relatively weak: for any constant k, $\mathsf{dAM}^{sh}[k]$ protocols with message size m can be converted to locally checkable proofs (LCPs) with message size $O(2^m + \log n)$.

Lower bounds on distributed Arthur-Merlin protocols for some concrete problems are known. Kol, Oshman and Saxena [18] showed that if the language Sym is in the class $\mathsf{dAM}[2](f(n))$, then $f(n) \in \Omega(\log\log n)$. As mentioned in [6], this lower bound can actually be improved to $f(n) \in \Omega(\log n)$. On the other hand, there is no known method to prove lower bounds when the number of turns is three or more.

1.2 Quantum Interactive Proofs

Quantum interactive proofs (QIP) were introduced by Watrous [31] in the centralized setting as a variant of classical interactive proofs (IP) in which the verifier can perform polynomial-time quantum computation (instead of polynomial-time classical computation), and the prover and verifier can exchange quantum bits (instead of classical bits). Kitaev and Watrous [17] first showed that QIP, the class of languages that can be decided by a quantum interactive protocol with polynomial number of interactions, is contained in EXP, the class of languages decided in exponential time. This containment was improved by Jain, Ji, Upadhyay, and Watrous [14], who showed that QIP is actually contained in PSPACE, which implies that QIP collapses to the complexity class IP ($\mathsf{QIP} = \mathsf{IP} = \mathsf{PSPACE}$).

While the above result shows that quantum interactive proofs are not more powerful than classical interactive proofs, there is a striking property of quantum interactive proofs that is not expected to hold for classical interactive proofs: in the quantum case the number of interactions can be significantly reduced. More precisely, Watrous first showed that any language in PSPACE can be decided by a three-turn QIP protocol [31]. After that, Kitaev and Watrous [17] showed that any QIP protocol with a polynomial number of interaction can be parallelized to three turns ($\mathsf{QIP} = \mathsf{QIP}[3]$). Marriott and Watrous [22] additionally showed that the verifier's turn in $\mathsf{QIP}[3]$ protocols can be replaced by a 1-bit coin flip ($\mathsf{QIP}[3]$=QMAM). Kempe, Kobayashi, Matsumoto, and Vidick [16] showed an alternative proof of $\mathsf{QIP} = \mathsf{QIP}[3]$.

1.3 Our Results

In this paper we introduce the quantum counterpart of distributed interactive proofs, which we call distributed quantum interactive proofs (or sometimes distributed quantum interactive protocols) and write dQIP, and show their power. Roughly speaking, distributed quantum interactive proofs are defined similarly to the classical distributed interactive proofs (i.e., distributed Arthur-Merlin proofs) defined above, but the messages exchanged between the prover and the nodes of the network can now contain quantum bits (qubits), the nodes can

now do any (local) quantum computation (i.e., each node can apply any unitary transform to the registers it holds), and each node can now send messages consisting of qubits to its adjacent nodes. In analogy to the classical case, the main complexity measures when studying distributed quantum interactive protocols are the size of registers exchanged between the prover and the nodes, and the size of messages exchanged between the nodes. We give the formal definition of dQIP in Section 2. The class $\mathsf{dQIP}[k](f(n))$ is defined as the set of all languages that can be decided by a k-turn dQIP protocol where both the size of the messages exchanged between the prover and the nodes, and the size of the messages exchanged between the nodes are $O(f(n))$ qubits.

Our first result is the following theorem.

▶ **Theorem 1.** *For any constant $k \geq 5$, $\mathsf{dAM}[k](f(n)) \subseteq \mathsf{dQIP}[5](f(n))$.*

Theorem 1 shows that by using distributed quantum interactive proofs, the number of interactions in distributed interactive proofs can be significantly reduced. To prove this result, we develop a generic *quantum* technique for turn reduction in distributed interactive proofs. Since no similar turn-reduction *classical* technique is currently known, our result gives evidence of the power of quantum computation in the setting of distributed interactive proofs as well.

We also show that if we allow to use randomness shared to all nodes (we denote this model by dQIP^{sh}), the number of turns can be further reduced to three turns.

▶ **Theorem 2.** *For any constant $k \geq 3$, $\mathsf{dAM}[k](f(n)) \subseteq \mathsf{dQIP}^{sh}[3](f(n))$.*

On the other hand, in the classical case, it is known that allowing shared randomness does not change the class [3]: $\mathsf{dAM}^{sh}[k](f(n)) \subseteq \mathsf{dAM}[k](f(n))$ for all $k \geq 3$.[1]

As mentioned above, for (classical) dAM protocols increasing the number of turns is helpful to reduce the complexity (in particular, the certificate size) for many problems. Our result thus shows if we allow quantum resource, such protocols can be simulated in five turns, and in three turns if we allow shared randomness. More precisely, we obtain the following corollary (see Appendix A for the precise definitions of these problems and Theorems 12 and 13 in Section 4 for a statement of the corresponding classical results):

▶ **Corollary 3.**

1. *There exist*
   - *a $\mathsf{dQIP}^{sh}[3](\log n)$ protocol for Asym,*
   - *a $\mathsf{dQIP}^{sh}[3](\log n)$ protocol for GNI,*
   - *a $\mathsf{dQIP}^{sh}[3](\log \log n)$ protocol for SetEquality,*
   - *a $\mathsf{dQIP}^{sh}[3](\log \log n)$ protocol for DSym,*
   - *a $\mathsf{dQIP}[5](\log n)$ protocol for GNI.*
2. *There exists a constant δ such that if a language $\mathcal{L}$ can be decided in $\mathrm{poly}(n)$ time and n^{δ} space, then $\mathcal{L} \in \mathsf{dQIP}[5](\log n)$ and $\mathcal{L} \in \mathsf{dQIP}^{sh}[3](\log n)$.*

We also introduce a *quantum* problem (i.e., the inputs are quantum states) which arises naturally when considering distributed quantum networks. More specifically, we consider the following task: There are two quantum states $|\psi\rangle$ and $|\phi\rangle$ as the inputs, each of which is distributed over the entire network (each node $u \in V$ has N_u-qubit of $|\psi\rangle$ and $|\phi\rangle$, where

[1] In fact, the authors of [3] showed $\mathsf{dAM}^{sh}[k](f(n)) \subseteq \mathsf{dAM}[k](f(n) + \log n)$ for all $k \geq 1$ where the additional $\log n$ comes from constructing a spanning tree, but for $k \geq 3$, a spanning tree can be constructed with $O(1)$-sized messages between the prover and the nodes in three turns [25], so $\log n$ can be removed.

$\sum_{u \in V} N_u = N$). The goal of the task is to measure closeness of these states. We call this problem N-qubit Distributed Quantum Closeness Testing (DQCT_N). For this task, we show the following theorem (see Appendix B for the definition of the trace distance).

▶ **Theorem 4.** *There is a* $\mathsf{dQIP}[5](O(1))$ *protocol for* DQCT_N*, where the completeness and the soundness conditions are defined as follows:*

- ***Completeness:*** *If* $|\psi\rangle = |\phi\rangle$ *and the prover is honest, the protocol accepts with probability 1.*
- ***Soundness:*** *If the protocol accepts with probability* $1 - 1/z$*,* $\mathrm{dist}(|\psi\rangle, |\phi\rangle) \leq \sqrt{2/z} + \varepsilon$ *for any small constant* $\varepsilon > 0$*. Here* $\mathrm{dist}(,)$ *is the trace distance.*

Without the prover, a naive approach is to accumulate all of the input to the leader node, and perform local operations at the leader node to measure their closeness. Obviously this approach is inefficient in the following sense: (1) it requires $\Omega(D)$-round of communication where D is the diameter of the network; (2) the amount of communication is linear in N, the size of the input quantum states. Theorem 4 means that in the dQIP setting, (1) it only needs 1-round of communication between the nodes; (2) the amount of communication (the size of messages per edge, and the size of messages between each node and the prover) is $O(1)$, regardless of the input size. Note that the main result of the recent paper [4] (see Section 1.5 for their result) immediately shows that if the two input quantum states are hold by some specific two nodes respectively and there is no input for the other nodes, DQCT_N in an n-node network can be done with $O(N \cdot \mathrm{poly}(n))$ size of quantum proofs and 1-round of communication between nodes, in the non-interactive setting. Our setting is more general in the sense that the input quantum states can be distributed over the entire network.

Lastly, in Appendix C, we show how to transform dQIP protocols with two-sided (i.e., completeness and soundness) bounded error into dQIP protocols with perfect completeness. We show that if we allow the communication between nodes in the middle of interaction (we call this model as dQIP^{com}), achieving perfect completeness is possible. Thus dQIP^{com} protocols can be converted to 5-turn protocols with perfect completeness using parallel repetition with a fairly small increase of the message size.

1.4 Organization and Overview of our Approach

We start by considering a more powerful model than dQIP, which allows nodes to use shared randomness. That is, at each turn the network can send a shared random string of limited length to the prover. We call this model dQIP^{sh}. In Section 3.1, we first show how to reduce the number of turns by half in the dQIP^{sh} model. This is shown by adapting to the distributed setting the method of Kempe et al. [16], which reduces the number of turns of QIP by half. More precisely, we show that for any $\ell \geq 1$, $(4\ell + 1)$-turn dQIP^{sh} protocols can be parallelized to $(2\ell + 1)$-turn.

The main idea of [16] is the following. In the first turn the honest prover provides the verifier with a snapshot state of the $(2\ell + 1)$-th turn, which includes the state of its private register and the message register in the original protocol. In the second turn the verifier flips a fair coin and sends it to the prover. In the remaining turns they perform the forward or backward simulation of the original protocol, according to the result of the coin flip.

We then go back to the dQIP model in Section 3.2 and show how to reduce the number of turns by half in the dQIP model by using the same argument as in the case of dQIP^{sh}, by using two additional turns in order to share the result of the coin flip. We can thus parallelize $(4\ell + 1)$-turn dQIP protocols to $(2\ell + 3)$-turn. Applying recursively this approach makes possible to reduce the number of turns down to 7 (corresponding to $\ell = 2$), but not lower.

After that, we focus on how to parallelize 7-turn dQIP protocols to 5-turn. Starting from 7-turn, we can reduce the number of turns to 5 in the dQIP^{sh} model, as in the QIP model. In dQIP model we need additional two turns, so we need a different approach to turn-reduction when we start 7-turn protocols. To parallelize 7-turn to 5-turn, we use a protocol similar to the Marriott-Watrous protocol [22]. Their protocol is used to show that $\mathsf{QIP}[3] \subseteq \mathsf{QMAM}$, where QMAM is the subclass of QIP[3] in which messages Arthur can send to Merlin are random bits. We construct a similar protocol, which can be used to parallelize 7-turn dQIP protocols to 5-turn dQIP protocols.

In Section 4 we then prove Theorem 1 and Theorem 2. We first discuss how to convert $\mathsf{dAM}[k](f(n))$ protocols to $\mathsf{dQIP}[k](f(n))$ protocols. This is achieved by doing all the computation of the dAM protocol in a reversible manner (i.e., unitary computation). Since the verification phase remains classical, the probability of being fooled is as low as the original dAM protocol, no matter what entangled state the malicious prover sends. This converted $\mathsf{dQIP}[k](f(n))$ protocol is then parallelized to 5-turn (3-turn in dQIP^{sh} model, respectively) by repeatedly using the technique we show in Section 3. We also have to discuss the size of messages. Since in $\mathsf{dAM}[k](f(n))$ protocols, the private registers of the nodes are used only to store a copy of certificates, the size of the private registers of nodes in the converted $\mathsf{dQIP}[k]$ protocol is $O(f(n))$. Therefore the size of the snapshot states given by the prover is also $O(f(n))$.

In Section 5, we tackle with the task to test closeness of two distributed quantum states (DQCT_N in Section 1.3). In the full version [9], we present a dQIP protocol for this task. The main difficulty is the implementation of the controlled SWAP gate, since there is no prior shared entanglement in the network. To resolve this issue, we utilize the protocol of Zhu and Hayashi [34] to make the nodes share the GHZ state $\frac{1}{\sqrt{2}}(|0^n\rangle + |1^n\rangle)$. Using the prover, we show the tests (described by some POVM measurement) in the protocol of [34] can be implemented in the distributed setting. Another difficulty is to avoid to be fooled by the malicious prover. This is achieved by carefully constructing the protocol, which ensures that the malicious prover cannot fraudulently increase the acceptance probability.

1.5 Related Works

Although there is no previous result about distributed interactive proofs with quantum resources, Fraigniaud, Le Gall, Nishimura and Paz [4] investigated the quantum version of *randomized proof-labeling schemes* (or equivalently, dMA protocols). They considered the following problem: N-bit inputs (where N is sufficiently larger than the number of nodes n) are given to several nodes in a network and the goal is to check if all inputs are equal. They gave a dQMA protocol with $O(\log N)$ certificate size, and showed that any classical dMA protocol requires $\Omega(N)$ certificate size, which shows the superiority of quantum certification in this setting. Another example is a very recent work by Le Gall, Miyamoto, and Nishimura [8], which studied *quantum* problems in the dQMA framework.

The study of distributed interactive proofs for some concrete problems has been developed recently, including two or three turn distributed Merlin-Arthur protocols for recognition of cographs, distance-hereditary graphs, and some geometric intersection graph classes [24, 15].

Research on quantum distributed algorithms that outperform classical distributed algorithms in several standard distributed models has been very active recently: there have been investigations showing the superiority of quantum distributed algorithm in the CONGEST model [20, 12, 2], the CONGEST CLIQUE model [13] and the LOCAL model [21].

2 Definitions

2.1 Distributed Interactive Proofs

In this section we describe classical distributed interactive proofs, following the definition by Kol, Oshman and Saxena [18].

In distributed interactive proofs the verifier consists of a network represented by connected graph $G = (V, E)$ with $|V| = n$ nodes, and each node $u \in V$ is given its input label $I(u)$ where $I : V \to \{0,1\}^*$ is a function. We let $\mathcal{G}$ be the set of all connected graphs on vertices V, and $\mathcal{I}$ be the set of functions that maps V to $\{0,1\}^*$. Define a language $\mathcal{L}$ by a subset

$$\mathcal{L} \subseteq \mathcal{G} \times \mathcal{I}.$$

Given a network configuration $(G, I) \in \mathcal{G} \times \mathcal{I}$ and a language $\mathcal{L}$, we consider an interactive protocol that consists of a series of interactions between a prover (*Merlin*) and a distributed verifier (*Arthur*). The goal of the protocol is to decide if $(G, I) \in \mathcal{L}$. The prover Merlin has unlimited computational power, and knows all information about (G, I). The verifier Arthur is distributed; it consists of the nodes of the network G, and initially each node u only knows its input $I(u)$. Merlin is not trusted and tries to convince Arthur that $(G, I) \in \mathcal{L}$ by sending bit strings (certificates). Arthur provides Merlin some random queries. There exist two types of randomness that can be used by Arthur: private randomness and shared randomness. In the private randomness setting, each node can generate random bits that cannot be seen by the other nodes. In the shared randomness setting, all random bits generated by Arthur are shared among all nodes. We denote the private randomness setting by dAM and the shared randomness setting by dAM^{sh}.

A k-turn distributed interactive protocol (also called k-turn distributed Arthur-Merlin protocol in [18]) has the interaction phase and the verification phase. The interaction phase begins with Merlin's turn if k is odd, and Arthur's turn otherwise. In the first turn, if k is odd, Merlin chooses a function $c_1 : \{0,1\}^* \to \{0,1\}^*$ determined by the network configuration (G, I), and sends $c_1(u)$ to each node u. In the second turn, Arthur picks a random string $r_2(u)$ at each node u, and sends them to Merlin. (In dAM^{sh} interactive protocols Arthur picks one random string and it can be seen by all nodes.) This series of interactions continues for k turns. More precisely, if the j-th turn is Merlin's turn, he sends a certificate $c_j(u)$ to each node u where $c_j : \{0,1\}^* \to \{0,1\}^*$ is a function of (G, I) and all of random strings $\{r_i(u)\}_{u \in V, i \in \{2,4,\ldots,j-1\}}$ received from Arthur, and if the j-th turn is Arthur's, he picks a random string $r_j(u)$ at each node u, and sends them to Merlin. If k is even then the interaction phase begins with Arthur's turn (i.e., Arthur first picks a random string $r_1(u)$ at each node u).

The protocol completes with the verification phase. In this phase every node u broadcasts a message M_u to its neighbors which may depend on its input $I(u)$, random strings u picked, and the certificates u received. Finally, u decides its output (accept or reject) by all information accumulated by u. Arthur accepts if and only if all nodes accept. We say that a protocol has completeness c and soundness s for a language $\mathcal{L}$ if the following conditions hold for the verifier G and the input label I:

1. (Completeness) If $(G, I) \in \mathcal{L}$, then there exists a prover P such that $\Pr(\text{all nodes accept}) \geq c$.
2. (Soundness) If $(G, I) \notin \mathcal{L}$, then for any prover P, $\Pr(\text{all nodes accept}) \leq s$.

As in [18], we define the class $\mathsf{dAM}[k](f(n))$ as the class of languages accepted by such k-turn distributed interactive protocols in which in each turn the prover and the verifier exchange $O(f(n))$ bits per node, and each node exchanges $O(f(n))$ bits with its neighbors during the verification procedure. The formal definition is as follows.

▶ **Definition 5** ([18]). *The class* $\mathsf{dAM}[k](f(n))$ *is the class of languages* $\mathcal{L} \subseteq \mathcal{G} \times \mathcal{I}$ *that have a* k*-turn distributed Arthur-Merlin protocol with completeness* $2/3$ *and soundness* $1/3$ *satisfying the following conditions:*

- *At each Merlin's turn, Merlin sends certificates of* $O(f(n))$ *bits per node, and at each Arthur's turn, each node sends* $O(f(n))$ *random bits to Merlin.*
- *The size of messages exchanged between two adjacent nodes in the verification phase is* $O(f(n))$ *bits.*

2.2 Distributed Quantum Interactive Proofs

In this section we define the quantum counterpart of distributed interactive proofs, which we call distributed quantum interactive proofs. We assume the reader is familiar with the basic notions of quantum computation such that bra-ket notation of qubits, quantum circuits, and density operators (see [27], for instance, for a good reference).

Distributed quantum interactive proofs are defined similarly to the classical distributed interactive proofs of Section 2.1, but now the messages exchanged between the prover and the nodes consist of qubits, the nodes can do any (local) quantum computation, and each node can send messages consisting of qubits to its adjacent nodes. To make this rigorous and define the complexity of the protocol, we need to carefully specify how the messages are encoded using quantum registers and who owns the registers during the computation.[2] Here is the formal definition.

▶ **Definition 6.** *A* k*-turn distributed quantum interactive proof (*dQIP*) is a protocol between a prover and a distributed verifier who interact in the following way:*

- ***The configuration:*** *The verifier consists of an* n*-node network* $G = (V, E)$*. Each node* u *begins with a quantum register* V_u*. We denote* V *the set of registers* $\{\mathsf{V}_u\}_{u \in V}$*. The prover begins with a quantum register* P*. In addition, there is a quantum message register* M_u *for each* u*. Let* M *be the set of registers* $\{\mathsf{M}_u\}_{u \in V}$*. The prover initially has the register* M *if* k *is odd, otherwise the node* u *initially has the register* M_u*. The initial state in* V *and* M *is the all-zero pure state* $|0 \cdots 0\rangle$*.*
- ***The interaction:*** *The interaction of a* dQIP *system is the repetition of prover's turn and verifier's turn. In the prover's turn, the prover performs arbitrary unitary transform denoted* P_i *to* (M, P) *and sends each* M_u *to* u *in the* i*-th turn. In the verifier's turn, the verifier can do any local (quantum) computation. More precisely, each node* u *performs an arbitrary unitary transform denoted* $V_{u,i}$ *to* $(\mathsf{V}_u, \mathsf{M}_u)$*. Then, each node* u *sends* M_u *to the prover in the* i*-th turn. We let* $V_i = \bigotimes_{u \in V} V_{u,i}$ *be the unitary transform applied by the verifier in the* i*-th turn.*
- ***The verification:*** *After the interaction phase, each node* u *prepares registers* $\mathsf{W}_{u,v}$ *for* $(u, v) \in E$ *which are initialized to* $|0 \cdots 0\rangle$ *and used for communication. Then* u *performs an arbitrary unitary transform on the registers* $\mathsf{V}_u, \mathsf{M}_u$ *and all* $\mathsf{W}_{u,v}$ *for* $(u, v) \in E$*. After that, each node communicates with its neighbors, performs a measurement, and then decides reject/accept based on the outcome of the measurement (a more formal description of this step is given to Section 2.2.1).*

[2] Since quantum information differs from classical information in several fundamental ways (in particular, quantum information cannot be copied and quantum message can share "entanglement"), when studying quantum communication complexity or quantum distributed computation, a quantum message is represented as a quantum register (i.e., a physical system comprising multiple qubits) and the action of sending a quantum message is represented by sending this quantum register. The message size corresponds to the size of the register.

Note that distributed quantum interactive proofs defined above can simulate random bits.[3] The size of the certificate sent from the prover to node u at each prover's turn is the size of the register M_u. The size of the message sent from node u to the prover at each verifier's turn is also the size of the register M_u. At the verification phase, the size of the message exchanged between node u and v is the size of the register $\mathsf{W}_{u,v}$. This leads to the following definition of the complexity class $\mathsf{dQIP}[k](f(n))$, as the natural quantum variant of the complexity class $\mathsf{dAM}[k](f(n))$ of Definition 5.

▶ **Definition 7.** *The class* $\mathsf{dQIP}[k](f(n))$ *is the class of languages* $\mathcal{L}$ *such that there exists a* k*-turn* dQIP *protocol for* $\mathcal{L}$ *with completeness* $\frac{2}{3}$ *and soundness* $\frac{1}{3}$ *satisfying the following conditions:*

- *The size of register* M_u *for each node* u *is* $O(f(n))$.
- *The size of register* $\mathsf{W}_{u,v}$ *exchanged between* u *and* v *in the verification phase is* $O(f(n))$ *for any* $(u,v) \in E$.
- *The conditions of completeness and soundness is the same as those of* dAM *protocols.*

2.2.1 Technical Details about the Verification Phase

We now give a more formal (and more technical) description of the verification phase of a k-turn dQIP protocol in Definition 6. The communication and measurement operations can be specifically described as follows: for any $(u,v) \in E$, the two registers $\mathsf{W}_{u,v}$ and $\mathsf{W}_{v,u}$ are swapped by the SWAP gate (see Appendix B for the definition of the SWAP gate). Here, we let V_{k+1} be the unitary transform that is performed in the verification phase. If k is odd then the interaction begins with the prover's turn, and the entire unitary transform is written by $Q = V_{k+1}P_k \cdots V_2 P_1$. If k is even then Q is written by $Q = V_{k+1}P_k \cdots P_2 V_1$. After that, each node u performs a POVM measurement $(\Pi_{\mathrm{acc},u}, \Pi_{\mathrm{rej},u} = I - \Pi_{\mathrm{acc},u})$ on register $\mathsf{V}_u, \mathsf{M}_u$ and $\mathsf{W}_{u,v}$ for $(u,v) \in E$ to obtain its output (see Appendix B for the definition of POVMs). Without loss of generality, we can assume $\Pi_{\mathrm{acc},u} = |0\rangle\langle 0| \otimes I$ for all $u \in V$, i.e., node u accepts the protocol iff the first qubit of register V_u is in the state $|0\rangle$.

2.2.2 Variants of the Definition

The above definition corresponds to the distributed quantum interactive proofs with private randomness where communication between nodes of the networks only happens after the interaction with the prover. This is the natural quantum analog of the definition of classical distributed interactive proofs by [18] given in Section 2.1.

A possible variant is distributed quantum interactive proofs with shared randomness, in which nodes are allowed to use shared randomness. In order to distinguish this model with the settings of private randomness, we denote it dQIP^{sh}. We denote $\mathsf{dQIP}^{sh}[k](f(n))$ the complexity class defined for this variant similarly to Definition 5, with the additional condition that at each turn the size of (shared) random bits sent to the prover is also $O(f(n))$-bit (i.e., each node u can send its message register and a random string s of size $O(f(n))$, but s must be the same as those of the other nodes).

Another variant, which we call dQIP^{com}, is the variant where nodes can communicate with each other in the middle of interaction with the prover. While in this paper we do not focus on this variant (since the classical version did not consider communication in the

[3] Concretely, simulating one random bit can be done by using the Bell pair $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$ and keeping one qubit of the pair.

middle of the interaction with the prover either), we present a general result about this natural setting in Appendix C. We denote $\mathsf{dQIP}^{com}[k](f(n))$ the complexity class defined for this variant similarly to Definition 5.

3 General Turn Reduction Technique for Distributed Quantum Interactive Proofs

In this section we show a general reduction technique to reduce the number of turns by half while keeping the soundness parameter relatively low. The complexity only increases by the size of the private register (i.e., the amount of quantum memory used for local computation at each node).

3.1 Distributed Quantum Interactive Proofs with Shared Randomness

We first consider the case of dQIP^{sh} model, and show the following theorem.

▶ **Theorem 8.** *Let $\ell \geq 1$ be an integer, $\mathcal{L} \subseteq \mathcal{G}\times\mathcal{I}$ be a language that has a $\mathsf{dQIP}^{sh}[4\ell+1](f(n))$ protocol with completeness c and soundness s for some $c^2 > s$ where the protocol uses $g(n)$ space register at each node u in the interaction phase. Then $\mathcal{L}$ has a $\mathsf{dQIP}^{sh}[2\ell+1](f(n)+g(n))$ protocol with completeness $\frac{1+c}{2}$ and soundness $\frac{1+\sqrt{s}}{2}$.*

Proof. We will prove this theorem by adapting the method halving the number of turns of quantum interactive proofs given by [16] into the distributed setting. In their method, the prover first provides the snapshot state of the message register and the private register at (almost) half of turns in the original protocol. Then the verifier flips a coin and decides to execute either a forward-simulation (the simulation of the later half of the turns; $r = 0$ case in Figure 1) or a backward-simulation of the original protocol (the simulation of the inverse transformation of the first half of the turns; $r = 1$ case in Figure 1). The honest prover can perform the simulation according to the verifier's coin flip. On the other hand, due to the randomness of the verifier's choice, the malicious prover cannot fool the verifier. In order to implement this in the distributed setting, we only need to simulate the verifier's coin flip.

Let $\mathcal{L}$ be a language that has a k-turn dQIP^{sh} protocol where $k = 4\ell + 1$ for some integer $\ell \geq 1$, and let $Q = V_{k+1}P_k \cdots V_2P_1$ be the unitary transform that is applied to register $(\mathsf{V}, \mathsf{M}, \mathsf{P})$ in the protocol. We show a $(2\ell + 1)$-turn dQIP^{sh} protocol in Figure 1.

$(2\ell + 1)$-turn dQIP^{sh} protocol.

1. The prover sends $(\mathsf{V}_u, \mathsf{M}_u)$ to the node u for all $u \in V$.
2. The network generates a random bit r and sends it to the prover.
3. **a.** The network applies $V_{2\ell+2}^{\dagger}$ to (V, M) if $r = 1$, and sends M.
   b. For $j = 1$ to $\ell - 1$ do the following:
   Each node u receives M_u from the prover. The network applies $V_{2\ell+2j+2}$ if $r = 0$, and $V_{2\ell-2j+2}^{\dagger}$ if $r = 1$. Then each node u sends M_u to the prover.
   c. Each node u receives M_u from the prover. The network applies $V_{4\ell+2}$ if $r = 0$, and applies $V_2^{\dagger}$ if $r = 1$. If $r = 0$, then each node u performs the POVM measurement $\{\Pi_{\mathrm{acc},u}, I - \Pi_{\mathrm{acc},u}\}$ on register $(\mathsf{V}_u, \mathsf{M}_u)$ and obtains its output. If $r = 1$, each node u measures the content of V_u and accepts iff the qubits in V_u are in $|0\cdots0\rangle$.

Figure 1 $(2\ell + 1)$-turn dQIP^{sh} protocol.

It is obvious that the protocol in Figure 1 is a $(2\ell+1)$-turn dQIP^{sh} protocol. The analysis of completeness and soundness can be shown in the same way as in Lemma 4.1 of [16]. ◀

Applying recursively Theorem 8 makes possible to reduce the number of turns down to 3: Let m be the minimum integer that satisfies $k \leq 2^m + 1$. Then, the number of turns can be reduced from k to 3 by applying Theorem 8 $m-1$ times. In Section 4 we discuss it more rigorously with mentioning error reduction.

3.2 Distributed Quantum Interactive Proofs without Shared Randomness

Next, we consider dQIP protocols and show the analogous result of Theorem 8 in the dQIP model. We can implement the protocol of Figure 1 in the dQIP model by simulating the step (2) of Figure 1 (the verifier's coin flip) without shared randomness. In order to simulate it, we need additional two turns: In the first turn the prover sends the information that represents a rooted spanning tree along with the snapshot state. The root (denoted by ℓ) of the spanning tree creates a Bell pair $\frac{1}{\sqrt{2}}|0\rangle_{\mathsf{M}_\ell}|0\rangle_{\mathsf{V}_\ell} + \frac{1}{\sqrt{2}}|1\rangle_{\mathsf{M}_\ell}|1\rangle_{\mathsf{V}_\ell}$ using one qubit of its message register and one qubit of its private register, then sends the message register to the prover in the second turn. The prover in the third turn creates $\frac{1}{\sqrt{2}}|0^n\rangle_{\mathsf{M}} + \frac{1}{\sqrt{2}}|1^n\rangle_{\mathsf{M}}$ using the CNOT gate, sends the register M_u to each u except the root ℓ, and keeps one-qubit. The construction of a rooted spanning tree in 1-turn requires $\Omega(\log n)$ witness size [19], but we can construct a rooted spanning tree in $O(1)$ witness size in 3-turn by the result of [25]. Therefore the size of witnesses is unchanged without a constant factor. (Note that the soundness error of the protocol of [25] is bounded by any small constant $\varepsilon > 0$, and this ε appears in Theorems 9 and 10.) We thus obtain the following theorem.

▶ **Theorem 9.** *Let $\ell \geq 1$ be an integer, $\mathcal{L} \subseteq \mathcal{G} \times \mathcal{I}$ be a language that has a $\mathsf{dQIP}[4\ell+1](f(n))$ protocol with completeness c and soundness s for some $c^2 > s$ where the protocol uses $g(n)$ space register at each node u in the interaction phase. Then $\mathcal{L}$ has a $\mathsf{dQIP}[2\ell+3](f(n)+g(n))$ protocol with completeness $\frac{1+c}{2}$ and soundness $\frac{1+\sqrt{s}}{2} + \varepsilon$ for arbitrary small constant $\varepsilon > 0$.*

Note that applying recursively Theorem 9 makes possible to reduce the number of turns down to 7 (corresponding to $\ell = 2$), but not lower: Let ℓ be the minimum integer that satisfies $k \leq 4\ell + 1$. Starting from k-turn, using Theorem 9, it is reduced to $2\ell + 3$. For $\ell' = \lfloor \frac{\ell}{2} \rfloor$, we have $4(\ell'+1)+1 \geq 2\ell+3$. Using Theorem 9 again, it is reduced down to $2(\ell'+1)+3$, which is at most $2\ell+1$ if $\ell \geq 3$. However if $\ell \leq 2$, we cannot reduce from $4\ell+1$ to $2\ell+1$ using Theorem 9, that is, the recursion stops at 7-turn. We now show the following theorem, which enables us to parallelize 7-turn protocols.

▶ **Theorem 10.** *Let $\mathcal{L} \subseteq \mathcal{G} \times \mathcal{I}$ be a language that has a $\mathsf{dQIP}[7](f(n))$ protocol with completeness c and soundness s where the protocol uses $g(n)$ space register at each node u in the interaction phase. Then $\mathcal{L}$ has a $\mathsf{dQIP}[5](f(n)+g(n))$ protocol with completeness $\frac{1+c}{2}$ and soundness $\frac{1+\sqrt{s}}{2} + \varepsilon$ for arbitrary small constant $\varepsilon > 0$.*

Proof. Fix the input x and a $\mathsf{dQIP}[7](f(n))$ protocol π for $\mathcal{L}$ described by a sequence of unitaries (corresponding to the interactions between the verifier and the honest prover) $P_1, V_2, P_3, V_4, P_5, V_6, P_7, V_8$ in this order, which has completeness c and soundness s. Our converted protocol is shown in Figure 2 (we call this protocol π'). We denote $\mathsf{R}_1 = \{\mathsf{R}_{u,1}\}_{u \in V}$ and $\mathsf{R}_2 = \{\mathsf{R}_{u,2}\}_{u \in V}$. In π', we consider the entire register is $(\mathsf{P}, \mathsf{R}_1, \mathsf{R}_2)$ where P is the prover's private register: Initially, there is no verifier's private register, and after receiving R_1, the private register of each node u is $\mathsf{R}_{u,1}$. Here we analyze the completeness and the soundness of π'. Define two quantum states $|\psi_4\rangle = V_4 P_3 V_2 P_1 |0 \cdots 0\rangle_{(\mathsf{P},\mathsf{R}_1,\mathsf{R}_2)}$ and $|\psi_5\rangle = P_5 |\psi_4\rangle$.(Here we abuse the notation by thinking unitaries V_i act on both (M, V) and $(\mathsf{R}_1, \mathsf{R}_2)$, and also unitaries P_i act on both (P, M) and $(\mathsf{P}, \mathsf{R}_2)$ since they have the same size.)

Proof of completeness. Assume that $x \in \mathcal{L}$. The honest prover does the following.

- **Turn 1:** Prepare the quantum state $|\psi_4\rangle$ in the register $(\mathsf{P}, \mathsf{R}_1, \mathsf{R}_2)$. Send the register R_1.
- **Turn 3:** Broadcast b. If $b = 0$, apply the honest operation P_5 and send the register R_2. If $b = 1$, send the register R_2.
- **Turn 5:** If $b = 0$, apply the honest operation P_7. If $b = 1$, apply $P_3^\dagger$ (the inverse operation of the operation P_3).

After Step 5 of Figure 2, if $b = 0$, the entire quantum state is $V_8P_7V_6P_5V_4P_3V_2P_1 |0\cdots0\rangle$ and if $b = 1$, the entire quantum state is $P_1 |0\cdots0\rangle = (P_1 |0\cdots0\rangle_{(\mathsf{P},\mathsf{R}_2)}) \otimes |0\cdots0\rangle_{\mathsf{R}_1}$. Thus the acceptance probability of π' is $\frac{1+c}{2}$.

Proof of soundness. Assume that $x \notin \mathcal{L}$. Let $|\psi\rangle$ be the initial state in $(\mathsf{P}, \mathsf{R}_1, \mathsf{R}_2)$, that is, in Turn 1 of π', the verifier receives the register R_1 and its reduced state is $\mathrm{tr}_{(\mathsf{P},\mathsf{R}_2)}(|\psi\rangle\langle\psi|)$. Assume that, when the random bit in Turn 2 is $b = i$, the prover applies $U_i \otimes I_{\mathsf{R}_1}$ and sends R_2 in Turn 3, and applies $W_i \otimes I_{\mathsf{R}_1}$ and sends R_2 in Turn 5. Define unitaries Q_0 and Q_1 by $Q_0 = (I_{(\mathsf{P},\mathsf{R}_2)} \otimes V_8)(W_0 \otimes I_{\mathsf{R}_1})(I_{(\mathsf{P},\mathsf{R}_2)} \otimes V_6)(U_0 \otimes I_{\mathsf{R}_1})$ and $Q_1 = (I_{(\mathsf{P},\mathsf{R}_2)} \otimes V_2^\dagger)(W_1 \otimes I_{\mathsf{R}_1})(I_{(\mathsf{P},\mathsf{R}_2)} \otimes V_4^\dagger)(U_1 \otimes I_{\mathsf{R}_1})$, and let

$$|\alpha\rangle = \frac{1}{\|\Pi_{acc}Q_0 |\psi\rangle\|}\Pi_{acc}Q_0 |\psi\rangle \text{ and } |\beta\rangle = \frac{1}{\|\Pi_{init}Q_1 |\psi\rangle\|}\Pi_{init}Q_1 |\psi\rangle,$$

where Π_{acc} is the projection onto the acceptance state of π, and $\Pi_{init} = I_{(\mathsf{P},\mathsf{R}_2)} \otimes |0\cdots0\rangle\langle0\cdots0|_{\mathsf{R}_1}$.

Let p_i be the acceptance probability of π' when the random bit in Turn 2 is $b = i$. Then we have

$$\begin{aligned}
p_0 &= \|\Pi_{acc}Q_0 |\psi\rangle\|^2 = \frac{1}{\|\Pi_{acc}Q_0 |\psi\rangle\|}|\langle\psi| Q_0^\dagger\Pi_{acc}Q_0 |\psi\rangle|^2 = F(|\alpha\rangle\langle\alpha|, Q_0 |\psi\rangle\langle\psi| Q_0^\dagger)^2 \\
&= F(Q_0^\dagger |\alpha\rangle\langle\alpha| Q_0, |\psi\rangle\langle\psi|)^2, \\
p_1 &= \|\Pi_{init}Q_1 |\psi\rangle\|^2 = \frac{1}{\|\Pi_{init}Q_1 |\psi\rangle\|}|\langle\psi| Q_1^\dagger\Pi_{init}Q_1 |\psi\rangle|^2 = F(|\beta\rangle\langle\beta|, Q_1 |\psi\rangle\langle\psi| Q_1^\dagger)^2 \\
&= F(Q_1^\dagger |\beta\rangle\langle\beta| Q_1, |\psi\rangle\langle\psi|)^2.
\end{aligned}$$

Therefore, using Lemma 21 the acceptance probability p_{acc} of π' is bounded by

$$\begin{aligned}
p_{acc} = \frac{1}{2}(p_0 + p_1) &\leq \frac{1}{2}(1 + F(Q_0^\dagger |\alpha\rangle\langle\alpha| Q_0, Q_1^\dagger |\beta\rangle\langle\beta| Q_1)) \\
&= \frac{1}{2}(1 + F(|\alpha\rangle\langle\alpha|, Q_0Q_1^\dagger |\beta\rangle\langle\beta| Q_1Q_0^\dagger)).
\end{aligned}$$

We also have

$$F(|\alpha\rangle\langle\alpha|, Q_0Q_1^\dagger |\beta\rangle\langle\beta| Q_1Q_0^\dagger) = |\langle\alpha| Q_0Q_1^\dagger |\beta\rangle| = |\langle\alpha|\Pi_{acc}Q_0Q_1^\dagger |\beta\rangle| \leq \|\Pi_{acc}Q_0Q_1^\dagger |\beta\rangle\|$$

from the fact that $\Pi_{acc} |\alpha\rangle = |\alpha\rangle$. The reduced state of $|\beta\rangle$ satisfies $\mathrm{tr}_{(\mathsf{P},\mathsf{R}_2)}(|\beta\rangle\langle\beta|) = |0\cdots0\rangle\langle0\cdots0|_{\mathsf{R}_1}$ since $\Pi_{init} |\beta\rangle = |\beta\rangle$. Therefore, from the soundness of π, for any U_0, U_1, W_0, W_1 acting on $(\mathsf{P}, \mathsf{R}_2)$,

$$\begin{aligned}
&\|\Pi_{acc}(I_{(\mathsf{P},\mathsf{R}_2)} \otimes V_8)(W_0 \otimes I_{\mathsf{R}_1})(I_{(\mathsf{P},\mathsf{R}_2)} \otimes V_6)(U_0U_1^\dagger \otimes I_{\mathsf{R}_1})(I_{(\mathsf{P},\mathsf{R}_2)} \otimes V_4)(W_1 \otimes I_{\mathsf{R}_1})(I_{(\mathsf{P},\mathsf{R}_2)} \otimes V_2) |\beta\rangle\|^2 \\
&= \|\Pi_{acc}Q_0Q_1^\dagger |\beta\rangle\|^2 \leq s.
\end{aligned}$$

Thus we have $p_{acc} \leq \frac{1}{2} + \frac{\sqrt{s}}{2}$, which completes the proof of soundness. ◀

dQIP[5]($f(n)+g(n)$) protocol π'.

1. **Turn 1:** The prover gives a $g(n)$-qubit quantum register $\mathsf{R}_{u,1}$ to each node u. The prover chooses arbitrary one node as a leader node leader.
2. **Turn 2:** leader chooses a bit $b \in \{0,1\}$ uniformly at random and sends it to the prover.
3. **Turn 3:** The prover sends one bit b_u and a $f(n)$-qubit quantum register $\mathsf{R}_{u,2}$ to each node u.
4. **Turn 4:** If $b_u = 0$, each node u applies $V_{u,6}$ to $(\mathsf{R}_{u,1}, \mathsf{R}_{u,2})$. If $b_u = 1$, u applies $V_{u,4}^{\dagger}$. u sends $\mathsf{R}_{u,2}$ to the prover.
5. **Turn 5:** The prover sends $\mathsf{R}_{u,2}$ to each node u.
6. **The verification phase:** If $b_u = 0$, each node u applies $V_{u,8}$ to $(\mathsf{R}_{u,1}, \mathsf{R}_{u,2})$ and u does the same verification as in the original 7-turn protocol. u outputs "accept" iff the output of the original protocol is "accept" and all random bits b_v where $v \in N(u)$ are the same as b_u. If $b_u = 1$, u applies $V_{u,2}^{\dagger}$ and outputs "accept" iff the register $\mathsf{R}_{u,1}$ is set to all-zero state and all random bits b_v where $v \in N(u)$ are the same as b_u.

Figure 2 dQIP[5]($f(n)+g(n)$) protocol π'.

dQIP protocol simulating a dAM protocol.

1. **The prover's turn:** The prover applies arbitrary unitary U_j to the register (P, M). Then M_u is sent to the node u.
2. **The verifier's turn:** Each node u stores the state in M_u to its private register V_u by the SWAP gate, and creates $\frac{1}{\sqrt{2^m}} \sum_{r\in\{0,1\}^m} |r\rangle_{\mathsf{M}_u} |r\rangle_{\mathsf{V}_u}$ in M_u and a fresh part of V_u. Then M_u is sent to the prover.
3. **The verification phase:** Each node measures its private register in the computational basis, then broadcasts the outcome to its neighbors, and decides the output of the protocol (accept or reject).

Figure 3 dQIP protocol simulating a dAM protocol.

4 Quantum Simulation of Distributed Arthur-Merlin Interactive Protocols

In this section we see how to convert dAM protocols to dQIP protocols, and parallelize the converted dQIP protocol to 5-turn. Assume the size of each witness and random bits is m. Let c_j be a function that represents the witnesses provided by Merlin at the j-th turn. That is, if random bits generated by Arthur in the i-th turn is $r_i = r_i(u_1)r_i(u_2)\cdots r_i(u_n) \in \{0,1\}^{mn}$, $c_j(r_2, r_4, \ldots, r_{j-1}) = c_j(u_1)c_j(u_2)\cdots c_j(u_n) \in \{0,1\}^{mn}$ represents the witness where $c_j(u)$ is provided to u. In order to simulate k-turn dAM protocols, each computation by the prover has to be converted to a form of reversible computation. Thus c_j must be realized by a unitary transform

$$U_{c_j} : |r_2, ..., r_{j-1}, b\rangle \to |r_2, ..., r_{j-1}, b \oplus c_j\rangle .$$

The protocol proceeds as Figure 3. Let c and s be the completeness and the soundness of the original dAM[k] protocol, respectively. We show the following theorem.

▶ **Theorem 11.** *The protocol in Figure 3 has completeness c and soundness s.*

Proof.

Completeness. Assume that $(G, I) \in \mathcal{L}$ and the prover receives the M_u part of the quantum state $\frac{1}{\sqrt{2^m}} \sum_{r_{j-1}(u) \in \{0,1\}^m} |r_{j-1}(u)\rangle_{\mathsf{M}_u} |r_{j-1}(u)\rangle_{\mathsf{V}_u}$ from the node u in the $(j-1)$-th turn. At the j-th turn the honest prover applies the SWAP gate to (P, M), obtaining

$$\frac{1}{\sqrt{2^{\frac{j-1}{2}mn}}} \left(\sum_{r_2, r_4, ..., r_{j-1} \in \{0,1\}^{mn}} |r_2, r_4, ..., r_{j-1}\rangle_{\mathsf{P}} |0\rangle_{\mathsf{M}} |c_1, r_2, c_3, r_4, ..., c_{j-2}, r_{j-1}\rangle_{\mathsf{V}} \right).$$

Then, the prover also applies U_{c_j} to $(\mathsf{P}, \mathsf{M}_u)$, obtaining the following state

$$\begin{aligned} &\frac{1}{\sqrt{2^{\frac{j-1}{2}mn}}} \left(\sum_{r_2, r_4, ..., r_{j-1} \in \{0,1\}^{mn}} |r_2, r_4, ..., r_{j-1}\rangle_{\mathsf{P}} |c_j\rangle_{\mathsf{M}} |c_1, r_2, ..., c_{j-2}, r_{j-1}\rangle_{\mathsf{V}} \right) \\ &= \frac{1}{\sqrt{2^{\frac{j-1}{2}mn}}} \left(\sum_{r_2, r_4, ..., r_{j-1} \in \{0,1\}^{mn}} |r_2, r_4, ..., r_{j-1}\rangle_{\mathsf{P}} \bigotimes_{u \in V} |c_j(u)\rangle_{\mathsf{M}_u} |c_1, ..., r_{j-1}\rangle_{\mathsf{V}} \right). \end{aligned}$$

At the verification phase, the quantum state in V is a mixed state

$$\frac{1}{2^{\frac{k-1}{2}mn}} \sum_{r_2, ..., r_{k-1} \in \{0,1\}^{mn}} |c_1, r_2, c_3, ..., r_{k-1}, c_k\rangle \langle c_1, r_2, c_3, ..., r_{k-1}, c_k|_{\mathsf{V}},$$

and u obtains one of the state $|c_1(u), r_2(u), c_3(u), ..., r_{k-1}(u), c_k(u)\rangle$ uniformly at random as the outcome of its measurement. Then u broadcasts the outcome to its adjacent nodes. From the completeness of the original $\mathsf{dAM}[k]$ protocol the acceptance probability of this verification phase is at least c.

Soundness. Assume that $(G, I) \notin \mathcal{L}$. A malicious prover may use some other unitary instead of U_{c_j} at the j-th turn. Let $\sum_{r_{j-1} \in \{0,1\}^{mn}} |r_{j-1}\rangle_{\mathsf{M}} |r_{j-1}\rangle_{\mathsf{V}}$ be the Bell pairs created by the verifier in the $(j-1)$-th turn. The witness provided by the prover in the j-th turn is stored into the private register V. In the verification phase, node u_i obtains $|x_i, r_{j-1}(u)\rangle$ for some $x_i \in \{0,1\}^m$ as the outcome of its measurement. From the soundness of dAM protocols, the original protocol is accepted for at most s of all random strings generated by Arthur. On the other hand, each node u obtains $|r_2(u), r_4(u), ..., r_{k-1}(u)\rangle$ uniformly at random by its measurement regardless of the prover's action. Thus the acceptance probability of this dQIP protocol is at most s. ◀

Using this theorem and the results in Section 3, we can show Theorems 1 and 2.

Proofs of Theorem 1 and Theorem 2. Assume without loss of generality $k = 4\ell+1$ for $\ell \geq 1$ and $m = f(n)$. For any $\mathsf{dAM}[k](m)$ protocol, we have a $\mathsf{dQIP}[k](m)$ protocol which simulates it using Theorem 11. Applying Theorem 9 we can parallelize it to a $\mathsf{dQIP}[2\ell+3](m)$ protocol with completeness $\frac{1+c}{2}$ and soundness $\frac{1+\sqrt{s}}{2}$. Note that we can assume $g(n) = O(m)$ since the register provided by the honest prover at the first turn of the parallelized protocol contains a $(4\ell+1)mn$-qubit state in registers V and M such that the total state is represented as

$$\frac{1}{\sqrt{2^{(\ell+1)mn}}} \sum_{r_2, ..., r_{2\ell+2} \in \{0,1\}^{mn}} |r_2, \ldots, r_{2\ell+2}\rangle_{\mathsf{P}} |c_1, r_2, \ldots, c_{2\ell+1}, r_{2\ell+2}, 0^{mn}, \cdots, 0^{mn}\rangle_{(\mathsf{V},\mathsf{M})}$$

and each node u receives its reduced state of $(4\ell+1)m = O(m)$-qubit on $(\mathsf{V}_u, \mathsf{M}_u)$. Thus the size of witnesses and the size of messages in the verification phase are both $O(m)$. We assume that the parameters c and s of the original $\mathsf{dAM}[k](m)$ protocol are $c = 1-\varepsilon$ and $s = \delta$ for small constant $\varepsilon > 0$ and $\delta > 0$ since we can use the standard technique of parallel repetition by [3] (the protocol is executed in parallel a constant number of times, and the leader node, which is determined by the prover as a root of a spanning tree, adopts the majority of the outcomes in the verification phase). Note that the witness size does not change by parallel repetition since the protocol has $k \geq 3$ turn and the construction of a spanning tree can be done with $O(1)$-size witnesses using three turns [25]. (Here, the soundness error ε' of the construction of the spanning tree affects the analysis, so we set the soundness $s+\varepsilon'$ for the parallel repetition.) Now we assume that the completeness and the soundness of the original $\mathsf{dAM}[k](m)$ protocol are $c = 1-\frac{1}{12a^2}$ and $s = \frac{1}{12a^2}$ for $a = k-1$, respectively. By Theorem 11, the converted $\mathsf{dQIP}[k](m)$ protocol has the same completeness and soundness. By Theorem 9, which reduces the number of turns down to 7, and Theorem 10, which reduces the number of turns down to 5, and the same analysis as Lemma 4.2 of [16], we get a $\mathsf{dQIP}[5](m)$ protocol with completeness $1-\frac{2c}{k-1} = 1-\frac{1}{6a^3}$ and soundness $1-\frac{1-s}{(k-1)^2} < 1-\frac{1}{2a^2}$. Then we use another parallel repetition for quantum interactive protocols developed in [11]. (The parallel repetition in [11] accepts iff all outcomes in repetitions are "accept". See Theorem 4.9 in [11].) More precisely, using $2a^3$ time repetitions the completeness and soundness become $\left(1-\frac{1}{6a^3}\right)^{2a^3} > 1-\frac{1}{3} = \frac{2}{3}$ and $\left(1-\frac{1}{2a^2}\right)^{2a^3} < \frac{1}{e^a} < \frac{1}{3}$, which completes the proof of Theorem 1.

In the case of dQIP^{sh}, we can parallelize a $k = (4\ell+1)$-turn protocol to a $(2\ell+1)$-turn protocol by using Theorem 8. Therefore Theorem 2 can be shown similarly to the proof of Theorem 1 by applying Theorem 8, instead of Theorem 9 and Theorem 10. ◀

4.1 Applications of Theorem 1 and 2

We can apply Theorem 1 and Theorem 2 to the following dAM protocols by [25] (see Appendix A for the definition of these problems), obtaining Corollary 3:

▶ **Theorem 12** ([25]). *There exist*

- *a* $\mathsf{dAM}[4](\log n)$ *protocol for* Asym,
- *a* $\mathsf{dAM}[O(1)](\log n)$ *protocol for GNI,*
- *a* $\mathsf{dAM}[5](\log\log n)$ *protocol for* SetEquality,
- *a* $\mathsf{dAM}[5](\log\log n)$ *protocol for* DSym.

▶ **Theorem 13** ([25]). *There exists a constant δ such that if a language $\mathcal{L}$ can be decided in* $\mathrm{poly}(n)$ *time and n^δ space, then* $\mathcal{L} \in \mathsf{dAM}[O(1)](\log n)$.

▶ **Corollary 3.** 1. *There exist*

- *a* $\mathsf{dQIP}^{sh}[3](\log n)$ *protocol for* Asym,
- *a* $\mathsf{dQIP}^{sh}[3](\log n)$ *protocol for GNI,*
- *a* $\mathsf{dQIP}^{sh}[3](\log\log n)$ *protocol for* SetEquality,
- *a* $\mathsf{dQIP}^{sh}[3](\log\log n)$ *protocol for* DSym.
- *a* $\mathsf{dQIP}[5](\log n)$ *protocol for GNI.*

2. *There exists a constant δ such that if a language $\mathcal{L}$ can be decided in* $\mathrm{poly}(n)$ *time and n^δ space, then* $\mathcal{L} \in \mathsf{dQIP}[5](\log n)$ *and* $\mathcal{L} \in \mathsf{dQIP}^{sh}[3](\log n)$.

5 Testing Closeness of Two Quantum States

Verification of the GHZ state

In this section we briefly explain how to show Thereom 4. Technically, our protocol can be viewed as the distributed implementation of the SWAP test [1]. To do this, we need to implement the controlled SWAP gate, but it is not possible by local operations at each node if the inputs are distributed since there is no prior entanglement in our setting. To resolve this issue, we create the quantum state that is called the GHZ state using the prover. Let $|GHZ\rangle$ be the n-qubit GHZ state

$$|GHZ\rangle = \frac{1}{\sqrt{2}}(|0^n\rangle + |1^n\rangle).$$

The detail of our approach is omitted from the main body of the paper due to space constraint. It can be found in the full version [9]. Ultimately, we present a dQIP protocol for verification of the GHZ state, and show the following theorem.

▶ **Theorem 16.** *There exists a* dQIP$[5](O(1))$ *protocol* $\mathcal{P}_{GHZ}$ *that satisfies the following properties:*

- *(completeness): If the prover is honest, the protocol accepts with probability 1 and the network outputs the GHZ state.*
- *(soundness): If the protocol accepts with probability* δ*, then the reduced state* ρ *of the output register satisfies*

$$\langle GHZ|\,\rho\,|GHZ\rangle \geq 1 - \varepsilon.$$

The dQIP protocol for DQCT_N

In the full version [9], using the protocol $\mathcal{P}_{GHZ}$, we present a protocol $\mathcal{P}_{\mathsf{DQCT}}$, which satisfies the desired conditions appeared in Theorem 4.

References

1 Harry Buhrman, Richard Cleve, John Watrous, and Ronald De Wolf. Quantum fingerprinting. *Physical Review Letters*, 87(16):167902, 2001.

2 Keren Censor-Hillel, Orr Fischer, François Le Gall, Dean Leitersdorf, and Rotem Oshman. Quantum Distributed Algorithms for Detection of Cliques. In *Proceedings of the 13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*, pages 35:1–35:25, 2022.

3 Pierluigi Crescenzi, Pierre Fraigniaud, and Ami Paz. Trade-Offs in Distributed Interactive Proofs. In *Proceedings of the 33rd International Symposium on Distributed Computing (DISC 2019)*, pages 13:1–13:17, 2019.

4 Pierre Fraigniaud, François Le Gall, Harumichi Nishimura, and Ami Paz. Distributed Quantum Proofs for Replicated Data. In *Proceedings of the 12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, pages 28:1–28:20, 2021.

5 Pierre Fraigniaud, Amos Korman, and David Peleg. Local distributed decision. In *Proceedings of the IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS 2011)*, pages 708–717, 2011.

6 Pierre Fraigniaud, Pedro Montealegre, Rotem Oshman, Ivan Rapaport, and Ioan Todinca. On distributed Merlin-Arthur decision protocols. In *Proceedings of the International Colloquium on Structural Information and Communication Complexity (SIROCCO 2019)*, pages 230–245, 2019.

7 Christopher A Fuchs and Jeroen Van De Graaf. Cryptographic distinguishability measures for quantum-mechanical states. *IEEE Transactions on Information Theory*, 45(4):1216–1227, 1999.

8 François Le Gall, Masayuki Miyamoto, and Harumichi Nishimura. Distributed Merlin-Arthur Synthesis of Quantum States and Its Applications, 2022. `arXiv:2210.01389`.

9 François Le Gall, Masayuki Miyamoto, and Harumichi Nishimura. Distributed Quantum Interactive Proofs, 2022. `arXiv:2210.01390`.

10 Mika Göös and Jukka Suomela. Locally checkable proofs in distributed computing. *Theory of Computing*, 12(1):1–33, 2016.

11 Gus Gutoski. Quantum strategies and local operations. *arXiv preprint*, 2010. `arXiv:1003.0038`.

12 Taisuke Izumi, François Le Gall, and Frédéric Magniez. Quantum Distributed Algorithm for Triangle Finding in the CONGEST Model. In *Proceedings of the 37th International Symposium on Theoretical Aspects of Computer Science (STACS 2020)*, pages 23:1–23:13, 2020.

13 Taisuke Izumi and François Le Gall. Quantum distributed algorithm for the all-pairs shortest path problem in the congest-clique model. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing (PODC 2019)*, pages 84–93, 2019.

14 Rahul Jain, Zhengfeng Ji, Sarvagya Upadhyay, and John Watrous. QIP= PSPACE. *Journal of the ACM*, 58(6):1–27, 2011.

15 Benjamin Jauregui, Pedro Montealegre, and Ivan Rapaport. Distributed interactive proofs for the recognition of some geometric intersection graph classes. In *Proceedings of 29th International Colloquium on Structural Information and Communication Complexity (SIROCCO 2022)*, pages 212–233, 2022.

16 Julia Kempe, Hirotada Kobayashi, Keiji Matsumoto, and Thomas Vidick. Using entanglement in quantum multi-prover interactive proofs. *Computational Complexity*, 18(2):273–307, 2009.

17 Alexei Kitaev and John Watrous. Parallelization, amplification, and exponential time simulation of quantum interactive proof systems. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing (STOC 2000)*, pages 608–617, 2000.

18 Gillat Kol, Rotem Oshman, and Raghuvansh R Saxena. Interactive distributed proofs. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing (PODC 2018)*, pages 255–264, 2018.

19 Amos Korman, Shay Kutten, and David Peleg. Proof labeling schemes. *Distributed Computing*, 22(4):215–233, 2010.

20 François Le Gall and Frédéric Magniez. Sublinear-time quantum computation of the diameter in congest networks. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing (PODC 2018)*, pages 337–346, 2018.

21 François Le Gall, Harumichi Nishimura, and Ansis Rosmanis. Quantum Advantage for the LOCAL Model in Distributed Computing. In *Proceedings of the 36th International Symposium on Theoretical Aspects of Computer Science (STACS 2019)*, pages 49:1–49:14, 2019.

22 Chris Marriott and John Watrous. Quantum Arthur–Merlin games. *Computational Complexity*, 14(2):122–152, 2005.

23 Pedro Montealegre, Diego Ramírez-Romero, and Ivan Rapaport. Shared vs Private Randomness in Distributed Interactive Proofs. In *Proceedings of the 31st International Symposium on Algorithms and Computation (ISAAC 2020)*, pages 51:1–51:13, 2020.

24 Pedro Montealegre, Diego Ramírez-Romero, and Ivan Rapaport. Compact distributed interactive proofs for the recognition of cographs and distance-hereditary graphs. In *Proceedings of the International Symposium on Stabilizing, Safety, and Security of Distributed Systems (SSS 2021)*, pages 395–409, 2021.

25 Moni Naor, Merav Parter, and Eylon Yogev. The power of distributed verifiers in interactive proofs. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2020)*, pages 1096–115, 2020.

26 Ashwin Nayak and Peter Shor. Bit-commitment-based quantum coin flipping. *Physical Review A*, 67(1):012304, 2003.

27 Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.

28 Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

29 Atish Das Sarma, Stephan Holzer, Liah Kor, Amos Korman, Danupon Nanongkai, Gopal Pandurangan, David Peleg, and Roger Wattenhofer. Distributed verification and hardness of distributed approximation. *SIAM Journal on Computing*, 41(5):1235–1265, 2012.

30 Robert W Spekkens and Terry Rudolph. Degrees of concealment and bindingness in quantum bit commitment protocols. *Physical Review A*, 65(1):012310, 2001.

31 John Watrous. PSPACE has constant-round quantum interactive proof systems. *Theoretical Computer Science*, 292(3):575–588, 2003.

32 John Watrous. *The Theory of Quantum Information*. Cambridge University Press, 2018.

33 Mark M. Wilde. *Quantum Information Theory*. Cambridge University Press, 2017.

34 Huangjun Zhu and Masahito Hayashi. Efficient verification of hypergraph states. *Physical Review Applied*, 12(5):054047, 2019.

A Problems

In this appendix we formally define the problems Set Equality, Graph Asymmetry, Dumbbell Symmetry and Graph Non-Isomorphism.

▶ **Definition 17** (Set Equality [25]). *Let G be a graph and I be an input such that the label $I(u)$ for each node u contains two lists of ℓ elements $\mathcal{A}_u = \{a_{u,1}, \ldots, a_{u,\ell}\}$ and $\mathcal{B}_u = \{b_{u,1}, \ldots, b_{u,\ell}\}$ where $\ell \leq n$ is an integer and each element in $\mathcal{A}_u$ and $\mathcal{B}_u$ can be represented in $O(\log n)$-bit. The language* SetEquality *is the set of graphs and labels such that $\{\mathcal{A}_u\}_{u\in V} = \{\mathcal{B}_u\}_{u\in V}$ as multisets.*

▶ **Definition 18** (Graph Asymmetry [25]). *The language* Asym *is the set of all connected graphs that do not have a nontrivial automorphism.*

▶ **Definition 19** (Dumbbell Symmetry [18]). *Let m, k be positive integers and let $n = 2m + 2k + 1$. An n-vertex connected graph $G = (\{0, 1, \ldots, n-1\}, E)$ is a dumbbell graph if it satisfies following conditions:*

- *Let G_0 be the vertex-induced subgraph of G on vertices $\{0, \ldots, m-1\}$ and G_1 be the vertex-induced subgraph of G on vertices $\{m, \ldots, 2m-1\}$.*
- *G_0 and G_1 are connected to each other by the following path of length $2k+2$*
$$0 - (2m) - (2m+1) - \cdots - (2m+2k) - (m).$$
- *E consists of all edges in G_0 and G_1, and the path-edges.*

The automorphism σ is given as follows:

$$\sigma(i) = \begin{cases} m+i & \text{if } i \in \{0, \ldots, m-1\} \\ i-m & \text{if } i \in \{m, \ldots, 2m-1\} \\ 4m+2k-i & \text{if } i \in \{2m, \ldots, 2m+2k\} \end{cases}$$

The language DSym *is the set of all dumbbell graphs G such that $\sigma(G)$ is isomorphic to G.*

▶ **Definition 20** (Graph Non-Isomorphism [18]). *The language GNI is the set of all pairs of graphs (G_0, G_1) where G_0 is not isomorphic to G_1. We assume that the communication graph is G_0, and nodes cannot communicate on G_1-edges.*

In [25], it was shown that SetEquality $\in$ dAM[2]($\log n$), SetEquality $\in$ dAM[4]($\log\log n$), Asym $\in$ dAM[4]($\log n$), DSym $\in$ dAM[4]($\log\log n$), GNI $\in$ dAM[k]($\log n$) for some constant $k > 4$. [4] For GNI, there also exists a dAM[4]($n \log n$) protocol showed by [18].

B Quantum information

We assume that the readers are familiar with basic concepts of quantum information such as density matrices, measurements, and quantum circuits (See, e.g., [28, 32, 33]).

Let $\mathcal{H}$ be a finite dimensional Hilbert space, and ρ, σ be any quantum states in $\mathcal{H}$. The fidelity of two quantum states ρ, σ is defined as $F(\rho,\sigma) = \mathrm{tr}\left[\sqrt{\sqrt{\rho}\sigma\sqrt{\rho}}\right]$. Note that for two pure states $\rho = |\psi_\rho\rangle\langle\psi_\rho|$ and $\sigma = |\psi_\sigma\rangle\langle\psi_\sigma|$ we have $F(\rho,\sigma) = |\langle\psi_\rho|\psi_\sigma\rangle|$. Let $\mathrm{dist}(\rho,\sigma) = \frac{1}{2}\|\rho-\sigma\|_{\mathrm{tr}}$ be the trace distance of ρ, σ, where $\|A\|_{\mathrm{tr}} = \mathrm{tr}\sqrt{A^\dagger A}$. For two pure states $|\psi\rangle\langle\psi|, |\phi\rangle\langle\phi|$ we denote $\mathrm{dist}(|\psi\rangle, |\phi\rangle)$. Here we summarize some useful inequalities about the fidelity and the trace distance, which are used multiple times in this paper.

▶ **Lemma 21.** *For any quantum states ρ, σ, ξ in $\mathcal{H}$, we have*

1. [7]: $1 - F(\rho,\sigma) \le \mathrm{dist}(\rho,\sigma) \le \sqrt{1-F(\rho,\sigma)^2}$,
2. [26, 30]: $F(\rho,\sigma)^2 + F(\xi,\sigma)^2 \le 1 + F(\rho,\xi)$.

A *POVM (Positive Operator Valued Measure)*, which is a general framework for quantum measurement, consists of a set of positive semi-definite matrices $\{M_m\}_m$ that satisfies $\sum_m M_m = I$. If a quantum state ρ is measured by a POVM $\{M_m\}_m$, the outcome i of the measurement is obtained with probability $\mathrm{tr}(M_i\rho)$. For arbitrary system of n-qubit, the quantum measurement corresponds to the POVM $\{M_m\}_m$ where $M_m = |m\rangle\langle m|$ for $m \in \{0, 1, \ldots, 2^n - 1\}$ is called the measurement in the computational basis.

The SWAP gate U_{swap} is the gate acting on two-qubit as $U_{swap}|a\rangle|b\rangle = |b\rangle|a\rangle$ for $a, b \in \{0, 1\}$. The SWAP test is a basic tool for quantum computing, which acts on the two registers $\mathsf{R}_1, \mathsf{R}_2$ as follows: (1) Prepare the state $|+\rangle$ on the 1-qubit ancilla register B; (2) Perform the SWAP gate on $(\mathsf{R}_1, \mathsf{R}_2)$ iff the qubit in B is $|1\rangle$; (3) Perform Hadamard gate on B and measure it in the computational basis $\{|0\rangle\langle 0|, |1\rangle\langle 1|\}$.

C Perfect completeness

In this appendix we consider the dQIP^{com} model, and show how to transform a protocol with two-sided bounded error into a protocol with perfect completeness. The number of turns of the transformed protocol increases by four turns and the size of message registers remains unchanged. This is shown by implementing the result of [17] in a distributed manner. The main difference from the centralized setting is the rejection condition.

In the case of the distributed setting, assume each node outputs 0 if it accepts otherwise outputs 1. Then the protocol rejects iff the output is not all-zero. The method of Kitaev and Watrous in the case of the centralized setting includes the operation that an additional register is prepared by the verifier, and it is incremented iff the private register is in the rejection state (i.e., $|1\rangle$). In the distributed setting, the prover determines the leader node, and the leader performs this operation. However, as mentioned above, in the case of distributed setting, the protocol rejects even if the leader is in the acceptance state but some other node is in the rejection state. Hence the leader needs the help of the prover to confirm that the protocol is in the rejection state. These operations require four additional turns but do not change the size of the message register.

[4] If we add the condition that the nodes can communicate on G_1-edges, there is a dAM[4]($\log n$) protocol [25].

Let $\mathcal{L}$ be a language that has a k-turn dQIP^{com} system with completeness c and soundness s where $c-s>\delta$ for some constant $\delta>0$. We show a distributed implementation of Kitaev and Watrous in Figure 4. Note that this protocol works for dQIP^{com} but does not work for dQIP (and dQIP^{sh}) since nodes need to communicate with each other in the middle of the protocol in order to check if its private register is in the acceptance state. The following theorem can be shown easily by adapting the proof of [17] to the distributed setting.

▶ **Theorem 22.** *Any $\mathsf{dQIP}^{com}[k](f(n))$ protocol with completeness c and soundness s where $c-s>\delta$ for some constant $\delta>0$ that uses $(g(n)+f(n)\deg(u))$-qubit private register at node u can be transformed to a $\mathsf{dQIP}^{com}[k+4](f(n)\Delta+g(n))$ protocol that uses $(g(n)+f(n)\deg(u))$-qubit private register at node u, with perfect completeness and soundness $1-\delta^2+\varepsilon$ where Δ is the maximum degree of the network and $\varepsilon>0$ is arbitrary small constant.*

Proof. We call the protocol of Figure 4 $\mathcal{P}$ in order to distinguish from the original protocol. The proof uses almost the same argument from Ref. [17]. We assume that in the original protocol each node u accepts iff the first qubit of its private register V_u is $|0\rangle$. Then, in step 3, $\mathcal{P}$ is rejected when one of the following conditions holds; (1) the output qubits is not in $|0^n\rangle$, but the state in $\mathsf{O}=\{\mathsf{O}_u\}_{u\in V}$, which is sent from the prover, is $|0^n\rangle$ (it means the case that at least one node rejects the original protocol); (2) the state in O is neither $|0^n\rangle$ nor $|1^n\rangle$. Now we only consider the case that the protocol $\mathcal{P}$ is not rejected in step 3. Let $|\psi\rangle=\alpha_{acc}|\psi_{acc}\rangle+\alpha_{rej}|\psi_{rej}\rangle$ be the state in (V,W) before the measurement of the original protocol where $|\psi_{acc}\rangle$ represents the state that the output qubits of all nodes is $|0\rangle$ and $|\psi_{rej}\rangle$ represents the state that the output qubit of at least one node is $|1\rangle$. Thus, after step 3, the state[5] in $(\mathsf{O},\mathsf{V},\mathsf{W})$ is written as

$$\alpha_{acc}|0^n\rangle_{\mathsf{O}}|\psi_{acc}\rangle+\alpha_{rej}|1^n\rangle_{\mathsf{O}}|\psi_{rej}\rangle.$$

Let U be the unitary that is applied by the prover between step 5 and 6. The state in $(\mathsf{O}',\mathsf{O},\mathsf{V},\mathsf{W})$ before step 6 is then

$$\alpha_{acc}|0\rangle_{\mathsf{O}'}U(|0^n\rangle_{\mathsf{O}}|\psi_{acc}\rangle)+\alpha_{rej}|1\rangle_{\mathsf{O}'}U(|1^n\rangle_{\mathsf{O}}|\psi_{rej}\rangle).$$

In step 6, the leader subtracts O' from O_ℓ, yielding the state

$$\alpha_{acc}|0\rangle_{\mathsf{O}'}|\phi_{acc}\rangle+\alpha_{rej}|1\rangle_{\mathsf{O}'}|\phi_{rej}\rangle,$$

where $|\phi_{acc}\rangle$ and $|\phi_{rej}\rangle$ are some states in $(\mathsf{O},\mathsf{V},\mathsf{W})$. The leader applies the operation T_c and accepts with probability $\|\alpha_{acc}\sqrt{c}|\phi_{acc}\rangle+\alpha_{rej}\sqrt{1-c}|\phi_{rej}\rangle\|^2$. For the completeness, we have $\alpha_{acc}=\sqrt{c}$ and $\alpha_{rej}=\sqrt{1-c}$. If the honest prover performs the operation that makes $|\phi_{acc}\rangle=|\phi_{rej}\rangle$, the acceptance probability is 1. For the soundness, observe that the acceptance probability is bounded by

$$(\alpha_{acc}\sqrt{c}+\alpha_{rej}\sqrt{1-c})^2+\varepsilon\leq 1-(c-\alpha_{acc}^2)^2+\varepsilon\leq 1-\delta^2+\varepsilon.$$

where ε is the soundness error of the spanning tree construction. ◀

Then we can use the dQIP^{com} variant of Theorem 9 and the parallel repetition of [11] repeatedly to reduce the number of turns to 5.

▶ **Corollary 23.** *Any $\mathsf{dQIP}^{com}[k](f(n))$ protocol that uses $(g(n)+f(n)\deg(u))$-qubit private register at node u can be transformed to a $\mathsf{dQIP}^{com}[5](f(n)\Delta+g(n))$ protocol with perfect completeness where Δ is the maximum degree of the network.*

[5] The malicious prover can make the state e.g., $\alpha_{acc}(\sqrt{\beta}|0^n\rangle+\sqrt{1-\beta}|1^n\rangle)_{\mathsf{O}}|\psi_{acc}\rangle+\alpha_{rej}|1^n\rangle_{\mathsf{O}}|\psi_{rej}\rangle$, but it only leads lower acceptance probability.

$(k+4)$-turn dQIP^{com} protocol with perfect completeness.

1. Run the original protocol except outputting accept or reject. Construct a spanning tree with the root ℓ.
2. Each node u prepares 1-qubit register O_u, perform the CNOT gate on O_u controlled on the first qubit of its private register which corresponds to the output, then sends O_u to the prover.
3. The prover sends the register O_u to each node u, then u rejects if O_u is in $|0\rangle$ and its output qubit is $|1\rangle$. The network checks if all of qubits provided by the prover are the same. If not, the network rejects the protocol.
4. Let ℓ be the leader node which is also the root of a spanning tree. The leader ℓ prepares one-qubit register O' in the state $|0\rangle$ and increments O' iff O_ℓ is $|1\rangle$.
5. Each node u sends the registers V_u, O_u and $\mathsf{W}_{u,v}$ for all $(u,v) \in E$ to the prover.
6. The prover sends O_ℓ to ℓ, and ℓ subtracts O_ℓ from O' (i.e., flips O' iff the content of O_ℓ is $|1\rangle$). The leader ℓ applies T_c to O_ℓ where T_c is given by

$$T_c(|0\rangle) = \sqrt{c}\,|0\rangle - \sqrt{1-c}\,|1\rangle$$
$$T_c(|1\rangle) = \sqrt{1-c}\,|0\rangle + \sqrt{c}\,|1\rangle\,.$$

   Then the leader measures O_ℓ and accepts iff the outcome is $|0\rangle$.

Figure 4 $(k+4)$-turn dQIP^{com} protocol with perfect completeness.