Data centers with quantum random access memory and quantum networks

Junyu Liu [1,2,3,4,5,*] Connor T. Hann [1,4,6,†] and Liang Jiang [1,3,6,‡]

[1] *Pritzker School of Molecular Engineering, University of Chicago, Chicago, Illinois 60637, USA*
[2] *Kadanoff Center for Theoretical Physics, University of Chicago, Chicago, Illinois 60637, USA*
[3] *Chicago Quantum Exchange, University of Chicago, Chicago, Illinois 60637, USA*
[4] *Institute for Quantum Information and Matter, California Institute of Technology, Pasadena, California 91125, USA*
[5] *Walter Burke Institute for Theoretical Physics, California Institute of Technology, Pasadena, California 91125, USA*
[6] *AWS Center for Quantum Computing, Pasadena, California 91125, USA*

(Received 10 October 2022; accepted 9 August 2023; published 20 September 2023)

In this paper we propose the Quantum Data Center (QDC), an architecture combining Quantum Random Access Memory (QRAM) and quantum networks. We give a precise definition of QDC and discuss its possible realizations and extensions. We discuss applications of QDC in quantum computation, quantum communication, and quantum sensing, with a primary focus on QDC for T-gate resources, QDC for multiparty private quantum communication, and QDC for distributed sensing through data compression. We show that QDC will provide efficient, private, and fast services as a future version of data centers.

DOI: 10.1103/PhysRevA.108.032610

I. INTRODUCTION

As. a frontier subject of physics and computer science, quantum information science is currently a rapidly developing and highly valued research area, with wide applications in computation [1–4], data science and machine learning [5,6], communication [7–13], and sensing [14–16]. In the near future, quantum computation may bring significant advantages to some specific algorithms; Quantum communication will strictly guarantee data security and privacy and boost transmission efficiency based on the laws of physics; and quantum sensing may boost the measurement precision significantly.

The generation, processing, and application of quantum data, and the treatment of those data together with their classical counterparts, are currently challenging theoretical and experimental problems in quantum science.

In this paper we propose the idea of the so-called *Quantum Data Center* (QDC), a unified concept referring to some specific quantum hardware that could efficiently deal with the quantum data and would provide an efficient interface between classical data and quantum processors. The key component of the proposed QDC is a Quantum Random Access Memory (QRAM) [17–25], which is a device that allows a user to access multiple different elements in superposition from a database (which can be either classical or quantum). At minimum, a QDC consists of a QRAM coupled to a quantum network.

We construct a theory of QDCs associated with original applications. We propose explicit constructions of examples, including QDCs as implementations of data-lookup oracles in fault-tolerant quantum computations; QDCs as mediators of so-called multiparty private quantum communications (defined below), which combines the Quantum Private Query (QPQ) [26] and Quantum Secret Sharing [8,9] protocols; and QDCs as quantum data compressors for distributed sensing applications. These three examples demonstrate that QDCs can provide significant advantages in the areas of quantum computing, quantum communication, and quantum sensing, respectively (see Fig. 1), with all other technical details and extra examples in the Appendixes.

II. GENERAL THEORY

A minimal definition of a QDC is a quantum or classical database, equipped with QRAM, connected to a quantum (communication) network. The minimal function of a QDC is that Alice (the customer) is able to upload and download information (classical or quantum) by providing the address to the database (Bob), and Bob will provide the information to Alice through QRAM, sending it via the quantum network (see Fig. 2).

About the role of QRAM, there are numerous quantum algorithms that claim potential advantages against their classical counterparts, but those algorithms often implicitly require an interface between classical data and the quantum processor. The advantages of the computational complexity are estimated usually from the query complexity where the oracle provides this interface (see, for instance, [5]). Quantum Random Access Memory (QRAM, see [17,25,27]) is a general-purpose architecture that could serve as a realization of such oracles. More specifically, QRAM allows a user to perform a superposition of queries to different elements of a data set stored in memory. The data itself can be either classical or quantum. In the case where the data are classical, the user provides an arbitrary superposition of addresses as

*junyuliu@uchicago.edu
†connor.t.hann@gmail.com
‡liang.jiang@uchicago.edu

 ©2023 American Physical Society

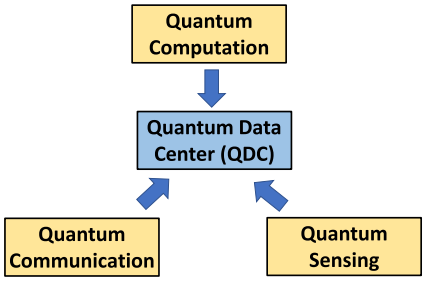

FIG. 1. A quantum data center (QDC) could potentially provide services about generation, processing, and application of quantum data, which could have wide applications in quantum computation, quantum communication, and quantum sensing.

input, and the QRAM returns an entangled state where the addresses are correlated with the corresponding data:

$$\sum_{i=0}^{N-1} \alpha_i |i\rangle^{Q_1} |0\rangle^{Q_2} \rightarrow \sum_{i=0}^{N-1} \alpha_i |i\rangle^{Q_1} |x_i\rangle^{Q_2}. \tag{1}$$

Here the superscripts Q_1 and Q_2, respectively, denote the input and output qubit registers, x_i denotes the ith element of the classical data set, α_i are generic coefficients, and N is the size of the database. We emphasize the distinction between QRAM, defined via Eq. (1), and so-called random-access quantum memories [28–30]; the latter do not allow for accessing a superposition of multiple different data elements and hence are not sufficient for our purposes. Indeed, the ability to perform a superposition of queries as in Eq. (1) is crucial to the applications we describe below. Moreover, in the Appendixes we will discuss in detail a quantum version of QRAM and a formal definition of QDCs.

Note that our definition means that a quantum architecture could be qualified as QDC only if it has both QRAM and quantum networks implemented. Thus, if a device does not have either of them, it cannot be called a QDC according to our definition. For example, a standard quantum memory, as described in, e.g., Refs. [29,30], coupled to a quantum network *cannot* be called a QDC. This is because QRAM is more than just a quantum memory; QRAM requires that different elements of the memory can be queried in superposition as in Eq. (1) (and the quantum version in the Appendixes). Moreover, there are extended parameters we could choose

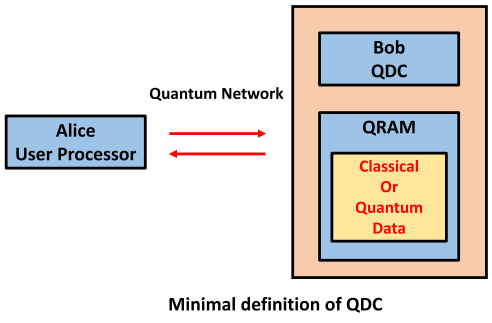

FIG. 2. The minimal definition of QDC contains the quantum network and QRAM. The data stored in QRAM can be either classical or quantum.

when we choose the circuit depth or the width of the QRAM implementation (see [22,24,25,31–35]), but for latter applications, we assume our QRAM circuits to be shallow. Further, we assume that QRAM has been built in the fault-tolerant way and has been error corrected. Building large-scale fault-tolerant QRAM is, in fact, a primary challenge in experiments [20].

About the role of the quantum networks, they might be realizable in the future due to the fast development of quantum communication technology in recent years. Here we are considering the service provided by QDC is centralized and has some physical distances from users. Thus, quantum states are supposed to be teleported through the quantum network from the user to the QDC or vice versa, where quantum teleportation technology includes the technologies of quantum satellites [36–38], quantum repeaters [10,12,13,39–41], etc.

III. QDC FOR QUANTUM COMPUTING: FAULT-TOLERANT RESOURCE SAVINGS

As the first example about QDCs applied for quantum computing, we show how a QDC can provide resource savings in a fault-tolerant cost model to users running query-based quantum algorithms. There are two aspects of resource savings induced by QDC: hardware outsourcing and communication costs (see the Appendixes for an unification of space and time costs). The reason for the hardware outsourcing is simple. Imagine that we are doing fault-tolerant quantum computation with a significant amount of T gates in the queries, which are considered to be expensive and require the magic-state distillation [42,43]. Rather than preparing the requisite magic states themselves, the user could instead ask the QDC to prepare the magic states, thereby reducing the resources required of the user. This naive approach, however, has a high communication cost since each magic state would need to be sent over the quantum network from the QDC to the user.

In contrast to this naive approach, we propose that the user outsources entire oracle queries to the QDC. Outsourcing entire queries provides a particularly efficient way for users to offload large amounts of magic-state distillation to the QDC with minimal communication cost.

More specifically, without the aid of a QDC, a user would be required to distill at least $O(\sqrt{N})$ [32] magic states in order to query a data set of size N as in Eq. (1). In contrast, with a QDC, a user can outsource the query to the QDC: the QDC is responsible for implementing the query and distilling the associated magic states, while the user incurs only a $O(\log N)$ (In this paper, log indicates $\log_2$) communication cost. This communication cost is due to the fact that the input and outputs of the query must be sent between the user and the QDC. The user also benefits in that they are no longer responsible for the potentially large amount of ancillary qubits needed to implement a query [22,32,33]; this hardware cost is paid by the QDC. Thus, this approach of outsourcing full queries to the QDC is exponentially more efficient than naive the approach described previously in terms of communication cost.

Both the savings from the hardware and the communication costs could quantify this benefit by the following example. Suppose, a user wishes to run a 100-qubit algorithm that requires 10^8 T gates when decomposed into Clifford $+T$

operations. Further, suppose that the user has a device with physical error rates of $p = 10^{-3}$ and that the target failure probability for the entire computation is $<1\%$. To achieve this failure probability, we assume error correction is used, and that gates are implemented fault tolerantly. Non-Clifford gates are implemented fault tolerantly with the aid of magic-state distillation.

A resource estimate for exactly this situation is performed in [44] for surface codes (see the Appendixes for a detailed review). The outsourcing can enable resource savings for the user (potentially in both the overall algorithm runtime and hardware cost). To estimate these savings, we suppose that these queries are responsible for 99% of the algorithm's T-state consumption (this is not an unreasonable supposition; see, for instance, [33]). According to [44] and the Appendixes, we observe that the user can now use a 15-to-1 magic-state distillation scheme, as the user need only produce $\sim 10^6$ magic states, and the total probability that any such state is faulty is $10^6 \times 35p^3 \sim 0.01$, which is within the allowed error tolerance. As described in [44], the number of surface code tiles required for computation and distillation with the 15-to-1 scheme is 164. With the distillation scheme selected, we can now estimate the required code distance d and algorithm run time. The code distance must satisfy

$$(\text{No. of tiles} = 164) \times [\text{No. of code cycles} \times (1 + \text{delay})] \times p_L(p, d) < 1\% \tag{2}$$

to guarantee that that the total error probability remains below 1%. Here p_L is the logical error probability of a distance d surface code with physical error probability p, which is approximately given by [45]. The parameter *No. of code cycles* is the number of surface code cycles required to distill 10^6 magic states, i.e., the minimum number of cycles required to run the algorithm assuming instantaneous data center queries. In practice, however, the data center queries will not be instantaneous. Thus we add a *delay* factor to the total number of cycles. This delay is related to the QDC's latency, τ, and the exact amount of the delay depends not only on how the oracle is implemented by the QDC, but also on the communication time overhead. Moreover, with the $\sqrt{N} \to \log N$ arguments from the communication cost, we could use assumptions, delay factor$\sim O(\sqrt{N})$, or delay factor$\sim O(\log N)$, respectively, referring to the protocols where oracle queries are implemented by the user (with magic states sent one-by-one from the QDC) or where full queries are implemented by the QDC. In Fig. 3 we plot the relative time costs depending on the data size N of QDCs, where we show that QDCs provide significant time savings in some ranges of data sizes, where the communication cost savings could be exponential. Finally, although our calculation is query-based, there are proposals where a query-based approach could provide a unified framework for all quantum algorithms [46], despite that the circuit depth for QRAMs should be shallow. Finally, we emphasize that the advantage is only for outsourcing users instead of users combined with QDC. Moreover, the exponential savings of communication costs are from QRAM, instead of specific quantum or classical algorithms. As a summary, combinations of QRAM and quantum networks might lead to significant benefits for outsourcing costs for T-gate preparations and magic-state distillations from unique features of QRAM, leading to useful applications in quantum computing.

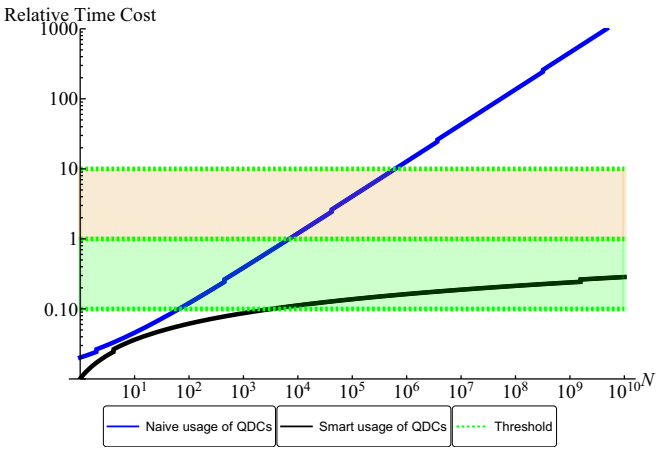

FIG. 3. The QDC-assisted relative time cost (the time cost from the user side with QDCs, divided by the one without QDCs) depending on the size of the data N. The dashed lines represent the threshold where QDC has the comparable performance as the situation without QDC, with the relative ratio 0.1, 1, and 10. Different thresholds correspond to prefactors relating the delay factor and the data N, and the relative ratio choices might correspond to different physical hardware. We consider the situations where two different methods of usages of QDCs, and both of them prepare the magic-state distillation in the QDC side. We replace the delay factor by the function $\sqrt{N}$ or $\log N$ directly where N is the size of the data in those two methods. For the solid lines from up to down, we have naive (blue) and smart (black) usages of QDCs. Some sudden jumps in the plot are because of the even integer values of the code distance, and we defer more precise discussions to the Appendixes.

IV. QDC FOR QUANTUM COMMUNICATION: MULTIPARTY PRIVATE QUANTUM COMMUNICATION

Quantum Private Query (QPQ) [26], a protocol combining QRAM and quantum networks, could already serve as an important application of QDC for quantum communication. Furthermore, the application of QDCs could be much broader to provide the users with fast and secure service. Based on QPQ and Quantum Secret Sharing from [8,9], we propose an original protocol, so-called multiparty private quantum communication, as an example of applications of QDCs. We provide discussions of QPQ and related concepts in the Appendixes as well as [47].

We present a protocol for multiparty private quantum communication using QDCs. We consider the situation in Fig. 4, where many sending users (denoted A_1, A_2, etc.) want to communicate privately to a set of receiving users (denoted B_1, B_2, etc.; A_i communicates with B_j, where $i \neq j$ in general). The communication occurs through two or more untrusted (but noncooperating) QDCs. Importantly, it is assumed that the users do not share any initial secret keys or entangled qubits, and that the users do not possess any secure communication links between them (either classical or quantum); all communication takes place over the untrusted quantum network shared with the QDCs.

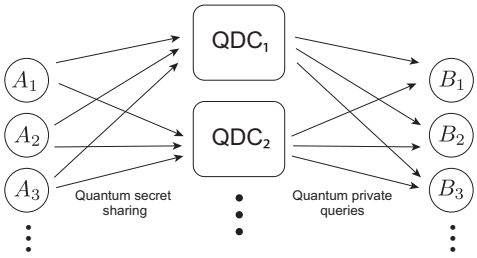

FIG. 4. Multiparty private quantum communication protocol. A set of sending users A_i communicates privately to a set of receiving users B_i through untrusted, noncooperating QDCs. The use of quantum secret sharing and quantum private queries guarantees that no QDC can learn what information was communicated or where the information was sent.

The protocol is as follows. First, each sending user A_i takes their quantum message, and decomposes it into several distinct parts using a quantum secret sharing protocol [8,9]. In isolation, each part of the secret message looks like a maximally mixed state, but when sufficiently many parts are assembled together, the original message can be perfectly recovered. Second, each sending user A_i sends parts of their secrets to the QDCs, where they are stored in QRAM at a publicly announced address. No one QDC should receive enough parts of a secret to reconstruct the original message. Finally, the receiving users interrogate the QDCs using the quantum private queries protocol. Each user B_j interrogates sufficiently many QDCs in order to retrieve enough parts of the secret to reconstruct A_i's original message. Advantages of this protocol is explained in the Appendixes. Moreover, a final note is that our protocol does not offer the communication between A_k and B_k protection from interception by $B_{j\neq k}$, rather it only provides privacy from untrusted QDCs. As a summary, combinations of QRAM and quantum networks could lead significant benefits for secure and private quantum communications, where quantum secret sharing and quantum private queries could serve as components.

V. QDC FOR QUANTUM SENSING: DATA COMPRESSION AND DISTRIBUTIVE SENSING

In the context of quantum sensing, QDCs can be used to compress quantum data and signals, enabling more efficient communication in distributed sensing tasks (see Fig. 5).

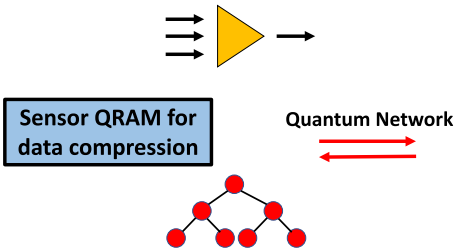

FIG. 5. Illustration of QDC for quantum sensing: we use QDCs to perform quantum data compression that enables distributive sensing.

To start, we illustrate how QDCs can be used to compress quantum data through a simple example. Suppose that the quantum data held by the QDC is confined to the single-expectation subspace, spanned by states where only one of the N qubits in the QDC's quantum memory is in the $|1\rangle$ state and all others are in $|0\rangle$. The state of the memory can then be written as

$$|\psi_{\text{unary}}\rangle = \sum_{i=0}^{N-1} \alpha_i \bigotimes_{j=1}^{N} |\delta_{ij}\rangle^{D_j}, \tag{3}$$

where D_j indicates the jth qubit in the N-qubit quantum memory, and δ_{ij} is the Kronecker delta ($\delta_{ij} = 1$ for $i = j$ and $\delta_{ij} = 0$ otherwise). Though the entire Hilbert space of the N-qubit quantum memory has the dimension 2^N, the single-excitation subspace has only dimension N. Thus, one could equivalently represent the above state using only $\log N$ qubits, as

$$|\psi_{\text{binary}}\rangle = \sum_{i=0}^{N-1} \alpha_i |i\rangle^{Q_1}, \tag{4}$$

where Q_1 denotes a $\log N$-qubit register, and $|i\rangle^{Q_1}$ denotes the ith basis state of this register. The two states $|\psi_{\text{unary}}\rangle$ and $|\psi_{\text{both}}\rangle$ contain the same quantum information (the N complex coefficients α_i) but encode this information in different ways.

A QDC can be used to realize the unary-to-binary compression described above, where the precise form implemented using QRAM is originally constructed in our work. The compression proceeds in two steps: first, the QDC performs an operation U (defined below) that encodes the location of the single excitation into a $\log N$-qubit address register, then a single QRAM query is performed in order to extract the excitation from the memory. In detail, the unitary U enacts the operation

$$U\left(|0\rangle^{Q_1} \sum_{i=0}^{N-1} \alpha_i \left[\bigotimes_{j=1}^{N} |\delta_{ij}\rangle^{D_j}\right]\right) = \sum_{i=0}^{N-1} \alpha_i |i\rangle^{Q_1} \left[\bigotimes_{j=1}^{N} |\delta_{ij}\rangle^{D_j}\right]. \tag{5}$$

We note that the operation U is not equivalent to a QRAM query, so U falls outside the scope of operations that a QDC can perform per the minimal definition. As we describe in the Appendixes, however, the operation U can be straightforwardly implemented using only minor modifications to standard QRAM architectures. Next, a QRAM query extracts the single excitation from the quantum memory and stores it in an output register Q_2,

$$\sum_{i=0}^{N-1} \alpha_i |i\rangle^{Q_1} |0\rangle^{Q_2} \left[\bigotimes_{j=1}^{N} |\delta_{ij}\rangle^{D_j}\right] \rightarrow \sum_{i=0}^{N-1} \alpha_i |i\rangle^{Q_1} |1\rangle^{Q_2} \left[\bigotimes_{j=1}^{N} |0\rangle^{D_j}\right]. \tag{6}$$

After this step, the Q_2 and D_j registers are disentangled from the Q_1 register. The state of the Q_1 register is $|\psi_{\text{binary}}\rangle$, which constitutes the compressed representation of the quantum data originally stored in the QDC's memory. This compressed

data may subsequently be stored, transmitted, or measured, depending on the application [48]. Thus, QDCs can be used to reduce the entanglement cost for distributed sensing applications. See details in the Appendixes for explanations. As a summary, combinations of QRAM and quantum networks could lead significant benefits for distributed quantum sensing, where QRAM designs could be helpful to perform quantum data compression and benefit entanglement cost reduction in distributive sensing applications.

VI. OUTLOOK AND CONCLUSION

Our research on QDCs opens up a promising direction in quantum information science (see details in the Appendixes for more applications). Recently we have analyzed the favorable error scaling of QRAM that only scales poly-log with the size of the database [24], which implies that QDC might be an intermediate-term application without the requirement of full error correction. QDC provides an example of application-specific efficient architectural design, taking full advantage of shallow QRAM circuits and small overhead in quantum communication. Given the treelike structure of QRAM, it will be interesting to explore the future possibility of distributed QDC so that we may decentralize the QRAM and perform the entire QRAM over the distributed quantum networks.

ACKNOWLEDGMENTS

We thank Gideon Lee, John Preskill, Nicolas Sawaya, Xiaodi Wu, Xiao Yuan, Pei Zeng, and Sisi Zhou for useful discussions. We specifically thank Isaac Chuang for inspiring discussions. J.L. is supported in part by International Business Machines (IBM) Quantum through the Chicago Quantum Exchange, and the Pritzker School of Molecular Engineering at the University of Chicago through AFOSR MURI (FA9550-21-1-0209). L.J. acknowledges support from the ARO (W911NF-18-1-0020, W911NF-18-1-0212), ARO MURI (W911NF-16-1-0349, W911NF-21-1-0325), AFOSR MURI (FA9550-19-1-0399, FA9550-21-1-0209), AFRL (FA8649-21-P-0781), DoE Q-NEXT, NSF (OMA-1936118, EEC-1941583, OMA-2137642), NTT Research, and the Packard Foundation (2020-71479). This work was done prior to C.T.H.'s joining the AWS Center for Quantum Computing.

APPENDIX A: MORE ABOUT QDCs

In these Appendixes we provide necessary information and related results about QDCs. Their structure is the following. In Appendix A we give an introduction on several basic aspects of QDCs. Specifically, in Sec. A 3 we give some perspectives about the writing function in QDCs. In Appendixes B, C, and D, we discuss several explicit examples of QDCs applied in quantum computing, quantum communication, and quantum sensing, including Secs. B 1 and B 2 with a review of quantum simulation and qubitization algorithms [57–59] where QDCs are used to provide the oracle, Sec. B 3 with QDCs for computing with multiple users, Sec. B 4 with a short introduction of the surface code and T-gate counting formalism developed in [44] and used in the T-gate example

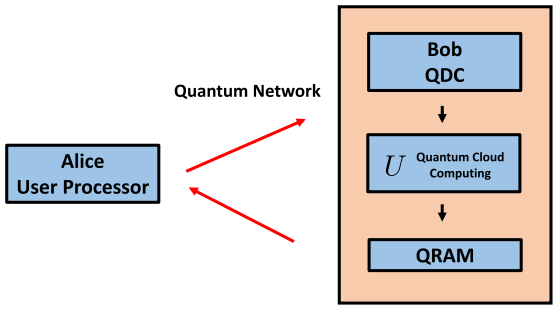

FIG. 6. QDCs could allow quantum cloud computing on the server.

in the main text, Secs. C 1 and C 2 with details of Quantum Private Query and blind quantum computing, Sec. D 1 with some details of quantum data compression, and Sec. D 2 with channel discrimination in quantum sensing and QDCs.

1. Comments

Here we comment on some general perspectives about QDCs. We note also that some architectures without either QRAM or quantum networks could still be defined as QDCs. In Sec. D 2 we describe an application in quantum sensing where QRAMs are not necessarily used, but one could still use QDC architectures to realize it.

One might be curious about how QDCs are different from a generalized version of quantum computers. Here we should clarify that, of course, we could develop a universal quantum computing (UQC) device that is associated with QDC. However, it is not necessary, and our QDC construction could directly serve remote users with their own quantum computation architectures. In fact, some of our examples do not require UQC power for QDC, for instance, QDCs for the T-gate counting that have been discussed in the main text.

Moreover, QDCs could take not only just the above minimal definition, but also more general forms. For example, QDCs equipped with UQC could also perform quantum cloud computation (see Fig. 6). Quantum computers are hard to realize, and it is natural to consider remote cloud services running in QDCs and provide the results of computations to remote users. Moreover, if users wish to keep the privacy, quantum blind computation [60,61] could be performed in QDCs with the help of quantum networks [61]; see further discussion in Section C.

Finally, we also note that our proposal is also closely related to the idea of *disposable quantum software* proposed in [62] by Preskill. This is a fragile quantum state that is hard to maintain by users, so they prefer to buy such a state through the quantum network. The early quantum teleportation scheme based on [63] provides significant power for quantum devices by combining UQC and quantum communication, inspiring the observation in [62] for quantum software. Our definition of QDC, including the functioning of UQC, could be a particular realization of disposable quantum software systems. However, we are more emphasizing the ingredient from QRAM, enabling extra capabilities for computation, communication, and sensing.

Finally, we give an explicit definition of QRAM for quantum data and provide a formal definition of QDCs. In this case, the user provides an arbitrary superposition of addresses as input, but the QRAM now returns the quantum state that was stored in the memory location specified by the address. More precisely, if the QRAM holds an arbitrary product state, $\bigotimes_{j=1}^{N} |\psi_j\rangle^{D_j}$, where D_j denotes the jth cell of the memory, then a QRAM query enacts the operation

$$\sum_{i=0}^{N-1} \alpha_i |i\rangle^{Q_1} |0\rangle^{Q_2} \left[\bigotimes_{j=1}^{N} |\psi_j\rangle^{D_j} \right] \rightarrow \sum_{i=0}^{N-1} \alpha_i |i\rangle^{Q_1} |\psi_i\rangle^{Q_2} \left[\bigotimes_{j=1}^{N} |\overline{\psi}_j^{(i)}\rangle^{D_j} \right], \tag{A1}$$

which, conditioned on register Q_1 being in state $|i\rangle$, swaps the state of register Q_2 and the ith cell of the quantum memory. Here $|\overline{\psi}_j^{(i)}\rangle = |0\rangle$ for $i = j$ and $|\overline{\psi}_j^{(i)}\rangle = |\psi_j\rangle$ otherwise. By linearity, this definition also defines the query operation when the QRAM holds an entangled state. Note that, when the data are quantum, a QRAM query generally leaves the Q_1 and Q_2 registers entangled with the data. The difference between the classical and quantum operations will be manifest when we try to "write" the data in QDCs (see Appendix A for a detailed discussion).

We now give a formal definition of a QDC, and we expand on aspects of this definition below.

Definition 1. A QDC, $\mathcal{D} = \{\mathcal{R}, \mathcal{I}\}$, consists of a QRAM, $\mathcal{R}$, coupled to a quantum communication network, $\mathcal{I}$, and queries to the QDC can be performed in three steps: (1) a remote user uses $\mathcal{I}$ to send a quantum query to the QDC; (2) the QDC executes query using $\mathcal{R}$, as in either classical (in the main text) or Eq. (A1); and (3) the QDC uses $\mathcal{I}$ to send input and output qubit registers, Q_1 and Q_2, back to the user. $\mathcal{D}$ is characterized by four key parameters: the size of the database N, the error in the query ϵ, the latency τ (time cost of a single query), and the throughput T (number of queries performed per unit time).

2. Cost estimation for QDCs

For QDCs defined above, how could we estimate the cost of time and hardware with a given requirement of error and privacy? Here we establish a general theory to estimate the hardware-time cost for QDCs and determine the optimal parameters according to the cost function.

In general, we define a cost function $F_{\text{cost}}^{\text{QDC}}$ for a given QDC architecture. The cost function could be written as

$$F_{\text{cost}}^{\text{QDC}} = F_{\text{cost}}^{\text{QDC}}(T_{\text{total}}, N_{\text{total}}, P_{\text{total}}). \tag{A2}$$

Here the cost function $F_{\text{cost}}^{\text{QDC}}$ includes the time cost T_{cost}, space (hardware) cost N_{cost}, and the privacy cost P_{total}. (The privacy cost here means a quantity the represents the level of consumption for the QDC users. We will provide an example in the situation of Sec. C 1.) For instance, one could simply assume that the above cost function is linear,

$$F_{\text{cost}}^{\text{QDC}} = \alpha_T T_{\text{total}} + \alpha_N N_{\text{total}} + \alpha_P P_{\text{total}}, \tag{A3}$$

with fixed positive coefficients α_T, α_N, and α_P. More generally, $F_{\text{cost}}^{\text{QDC}}$ could be defined as a monotonic function of T_{cost}, N_{cost}, and P_{total}. Moreover, T_{cost}, N_{cost}, and P_{total} are given by one collection of throughput parameters and the other collection of hyperparameters (latency) θ. The optimal hyperparameters could be determined by

$$\theta^* = \text{argmin}_\theta F_{\text{cost}}^{\text{QDC}} \tag{A4}$$

for given requirements of hardware. Similar analysis could be done for their counterparts without QDC, with the cost function $F_{\text{cost}}^{\overline{\text{QDC}}}$, and if QDCs have advantages, we want $F_{\text{cost}}^{\overline{\text{QDC}}} > F_{\text{cost}}^{\text{QDC}}$.

The cost analysis examples discussed later could be understood as precise instances of the above framework. In Sec. B we understand the hardware cost as the qubit cost, and we manifest the contribution of both T_{total} and N_{total}. In the T-gate example, we understand the entanglement cost as another form of the hardware cost, and we find significant advantages of QDCs in some cases. In Sec. C 1 we emphasize T_{total}, N_{total}, and P_{total}, the privacy cost in quantum communication. In general QDCs, all terms might be included based on the practical usage, time cost, and hardware. For a more systematic mathematical treatment of privacy in the complexity theory, see *differential privacy* discussed in the machine learning community [64].

3. Writing data in QRAM

In this section we more precisely define what it means to *write* data to QRAM. The definition of writing depends on whether the data being written are classical or quantum, and also on whether the addressing scheme is classical or quantum. We elaborate on these four different situations below.

a. Classical data, classical addressing

In this situation the QRAM holds a classical data vector x, and the writing operation consists of specifying a classical address i and a new classical value y_i, then overwriting the value ith cell of the QRAM's memory, $x_i \rightarrow y_i$. This writing process is entirely classical; it can be implemented simply by performing classical operations on the classical data.

Even though this definition of writing to QRAM is completely classical, it is still useful in the context of quantum algorithms. In particular, after writing, the modified classical data in the QRAM can subsequently be read in superposition (i.e., with quantum addressing). For example, if each element in the database is replaced as $x_i \rightarrow y_i$, then reading the QRAM consists of the operation

$$\sum_{i=0}^{N-1} \alpha_i |i\rangle^{Q_1} |0\rangle^{Q_2} \rightarrow \sum_{i=0}^{N-1} \alpha_i |i\rangle^{Q_1} |y_i\rangle^{Q_2}; \tag{A5}$$

cf. Eq. (1). Thus, the same QRAM can be reused to perform a superposition of queries to a different data set. This is particularly useful in the context of QDCs, as multiple users could be running different algorithms that require access to different classical data sets. The QDC can cater to all of these users by overwriting the QRAM's classical data between queries from different users.

b. Classical data, quantum addressing

In contrast to the previous definition, writing to QRAM for the case of classical data and quantum addressing is not well defined. To illustrate this, we propose a possible definition for writing in this situation, then show that it ultimately reduces to a probabilistic version of the classical writing procedure described above.

We suppose that the QRAM's classical data are stored in a quantum memory, i.e., each classical datum $x_i \in \{0, 1\}$ is encoded in a qubit as $|x_i\rangle$, so that the full database consists of the product state $\bigotimes_i |x_i\rangle^{D_i}$, where D_i denotes the ith cell of the memory. The writing procedure consists of first specifying a quantum address $\sum_i \alpha_i |i\rangle^{Q_1}$. Then, coherently conditioned on the state of the Q_1 register, one prepares another qubit Q_2 in the state $|y_i\rangle$ with $y_i \in \{0, 1\}$, then swaps this state with the ith cell of the memory,

$$\sum_i \alpha_i |i\rangle^{Q_1} |y_i\rangle^{Q_2} \left[\bigotimes_{j=1}^{N} |x_j\rangle^{D_j} \right] \rightarrow \sum_i \alpha_i |i\rangle^{Q_1} |x_i\rangle^{Q_2} \left[|y_i\rangle^{D_i} \bigotimes_{j\neq i} |x_j\rangle^{D_j} \right]. \tag{A6}$$

In general, this operation leaves the data registers entangled with the Q_1 and Q_2 registers. As such, tracing out the Q_1 and Q_2 registers leaves the database in a mixed state, where with probability $|\alpha_i|^2$ one finds that ith entry has been overwritten as $x_i \rightarrow y_i$. To achieve the same result, one could instead simply have randomly chosen to overwrite the ith element according to the distribution $|\alpha_i|^2$. Therefore, the use of quantum addressing and classical data does not confer an advantage over the case of classical addressing and classical data.

c. Quantum data, classical addressing

In this situation the QRAM holds quantum data, i.e., an N-qubit quantum state. The writing operation consists of specifying a classical address i and a new single-qubit state $|\phi\rangle^{Q_2}$, then swapping this state with the ith qubit in the QRAM's memory. In particular, if the QRAM initially holds a product state $\bigotimes_j |\psi_j\rangle^{D_j}$, then this writing procedure enacts the operation

$$|\phi\rangle^{Q_2} \left[\bigotimes_j |\psi_j\rangle^{D_j} \right] \rightarrow |\psi_i\rangle^{Q_2} \left[|\phi\rangle^{D_i} \bigotimes_{j\neq i} |\psi_j\rangle^{D_j} \right]. \tag{A7}$$

Note that in the case where the quantum data consists of a product state, this operation does not entangle the Q_2 and D_i registers. For general quantum data, however, this operation may leave these registers entangled, such that the data can be left in a mixed state when the Q_2 register is traced out.

d. Quantum data, quantum addressing

In this situation the QRAM holds quantum data, i.e., an N-qubit quantum state. The writing operation consists of first specifying a quantum address $\sum_i \alpha_i |i\rangle$. Then, coherently conditioned on the state of the Q_1 register, one prepares another register Q_2 in the state $|\phi_i\rangle$, then swaps this state with the ith cell of the memory. In the case where the QRAM initially holds a product state $\bigotimes_j |\psi_j\rangle^{D_j}$, then this writing procedure enacts the operation

$$\sum_i \alpha_i |i\rangle^{Q_1} |\phi_i\rangle^{Q_2} \left[\bigotimes_j |\psi_j\rangle^{D_j} \right] \rightarrow \sum_i \alpha_i |i\rangle^{Q_1} |\psi_i\rangle^{Q_2} \left[|\phi_i\rangle^{D_i} \bigotimes_{j\neq i} |\psi_j\rangle^{D_j} \right]. \tag{A8}$$

We note that reading quantum data is a special instance of the above process where $|\phi_i\rangle = |0\rangle$ for all i. This operation generally leaves the Q_1 and Q_2 registers entangled with the data registers D_i.

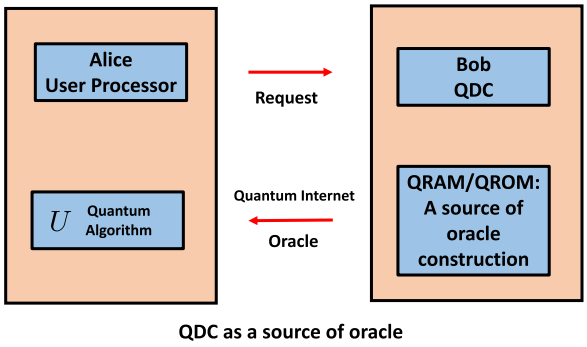

FIG. 7. QDC for a general algorithm U. In this example, QDC could serve as a pool of the explicit oracle construction that is needed for the algorithm U.

APPENDIX B: COMPUTING

1. General discussions about oracles

One of the most important applications of QDC is quantum computation. In the minimal definition of QDCs, we could use the QRAM as a remote service center providing oracles for the user. Many famous quantum algorithms, such as Quantum Principle Component Analysis, require the construction of oracles to reach the quantum advantage [65]. QRAM could provide substantial benefits regarding interfaces between the classical and the quantum world, serving as a natural hardware realization of the quantum oracle. Moreover, a hybrid QRAM-QROM construction will provide an optimal choice of the hardware-time overhead. Thus, one could imagine that the quantum computation is performed on the user side, and QDCs will serve as the source of the oracle. Connected by quantum networks, the user will call QDCs multiple times to complete the algorithm.

Here we will give a general discussion about the hardware-time cost of using QDC as a resource of oracle in a minimal setup (see Fig. 7).

If we assume that a general quantum algorithm U has the time cost and the qubit cost given by

$$\begin{aligned} T_U &= T_U(L, \epsilon, \theta_0, \theta_U), \\ N_U &= N_U(L, \epsilon, \theta_0, \theta_U), \end{aligned} \tag{B1}$$

depending on the problem size L, the precision ϵ, and the collections of other hyperparameters θ_U and problem parameters θ_0. Say that the time cost of the algorithm itself is expressed by the query complexity, and the corresponding oracle is prepared by QRAM itself multiple times. We assume the QRAM cost as

$$\begin{aligned} T_Q &= T_Q(L, \epsilon, \theta_Q), \\ N_Q &= N_Q(L, \epsilon, \theta_Q), \end{aligned} \tag{B2}$$

with the QRAM parameter θ_Q. Finally, we define the quantum network cost

$$\begin{aligned} T_I &= T_I(L, \epsilon, L_{\text{tot}}, \theta_I), \\ N_I &= N_I(L, \epsilon, L_{\text{tot}}, \theta_I), \end{aligned} \tag{B3}$$

with the total length L_{tot} and the quantum network parameter θ_I. Here we are assuming that the oracles are prepared remotely with the total length L_{tot}, and transformed to the user with the quantum network. So the total cost is given by

$$\begin{aligned} &T_{\text{total}}(L, \epsilon, \theta_0, L_{\text{tot}}, \theta_U, \theta_Q, \theta_I) \\ &\quad = T_U(L, \epsilon, \theta_U) \times (T_Q(L, \epsilon, \theta_Q) + T_I(L, \epsilon, L_{\text{tot}}, \theta_I)), \\ &N_{\text{total}}(L, \epsilon, \theta_0, L_{\text{tot}}, \theta_U, \theta_Q, \theta_I) \\ &\quad = N_U(L, \epsilon, \theta_U) + N_Q(L, \epsilon, \theta_Q) + N_I(L, \epsilon, L_{\text{tot}}, \theta_I). \end{aligned} \tag{B4}$$

Note that T_U is the query complexity for the quantum algorithm U. The time cost is a product, and the qubit cost is additive. Thus, for given L, ϵ, and θ_0, we could determine the optimal choice of QDC by

$$(L^*_{\text{tot}}, \theta^*_U, \theta^*_Q, \theta^*_I)_{L,\epsilon,\theta_0} = \text{argmin}_{(L_{\text{tot}},\theta_U,\theta_Q,\theta_I)} F_{\text{cost}}, \tag{B5}$$

where

$$F_{\text{cost}} = F_{\text{cost}}(T_{\text{total}}, N_{\text{total}}) \tag{B6}$$

is a given cost function based on the architecture of QDC. This is a specific example of the cost function algorithm equation (A2) for quantum computation.

Here we maintain the quantum algorithm mentioned here to be abstract. All quantum algorithms with the oracle required in the QRAM form could be adapted here. In Sec. B 2 we discuss a specific quantum algorithm, quantum signal processing (QSP) for Hamiltonian simulation [57–59], where quantum oracles are needed to address the information of the Hamiltonian. According to Sec. B 2, if we use the qubitization algorithm, and L is the number of Pauli terms appearing in the Hamiltonian, we have

$$\begin{aligned} &\frac{\text{QDC}_{\text{hardware cost}}}{\overline{\text{QDC}}_{\text{hardware cost}}} \\ &\quad = \frac{O(\log L) + O(\log \max_i \dim \Pi_i)}{O(\log L) + O(\log \max_i \dim \Pi_i) + O\left(\frac{L}{M} + \log L\right)} \\ &\quad \approx \frac{O(M \log L)}{O(L)}. \end{aligned} \tag{B7}$$

We are comparing the hardware cost completely from the user side: in the $\overline{\text{QDC}}$ case, since the user does not have QDC, the user has to implement QRAM or QROM by himself or herself. M is the parameter for the hybrid QRAM or QROM architecture. Moreover, we assume that L is large (note that this will happen if we are assuming nonlocal Hamiltonians and the Hamiltonian might be dense, which is not always true in the quantum chemistry tasks). In this case, using QDC, we could provide a significant hardware cost saving from the user side: when M does not scale with L, the saving could be even exponential.

Moreover, we make some discussions about QDCs used for multiple users in Sec. B 3. Furthermore, another potential saving of the hardware could come from the fact that the entanglement cost of accessing an N-element data set with a QDC is only $\log N$, where we have implicitly used in the above example, and it has been already manifest in the T-gate example.

2. Quantum simulation and oracles from QDCs

A perfect example of running QDCs as oracle resources could be the quantum simulation algorithm, which has wide applications in quantum many-body physics, quantum field theory, and quantum computational chemistry with potential advantages compared to classical computers. Aside from the so-called Trotter simulation scheme [66–68], many quantum simulation algorithms are oracle-based, such as algorithms based on quantum walks [69,70], multiproduct formula [71,72], Taylor expansion [73,74], fractional-query models [75], and qubitization and quantum signal processing (QSP) [57–59]. Those oracles could naturally be implemented by the QRAM model (see, for instance, [25]).

We will give a short introduction to the qubitization and QSP algorithms and discuss their costs. We will consider the linear combination of unitaries (LCU) decomposition as the input. We assume that the Hamiltonian is given by the following unitary strings:

$$H = \sum_{i=1}^{L} \alpha_i \Pi_i. \tag{B8}$$

For simplicity, we will assume that $\alpha_i > 0$. This is called the LCU model, and Π_is are usually the Pauli matrices. We introduce the ancilla states $|i\rangle : i = 1, 2, \ldots, L$ with the number of qubits $\log L$. Furthermore, we implement the following state $|G\rangle$:

$$|G\rangle = \sum_{i=1}^{L} g_i |i\rangle, \quad |g_i|^2 = \frac{\alpha_i}{\lambda}, \quad \lambda = \sum_{i=1}^{L} \alpha_i \tag{B9}$$

and

$$\begin{aligned} R &= 2|G\rangle\langle G| - I, \\ U &= \sum_{i=1}^{L} |i\rangle\langle i| \otimes \Pi_i, \\ W &= RU. \end{aligned} \tag{B10}$$

One could show that

$$\frac{H}{\lambda} = (\langle G| \otimes I) U (|G\rangle \otimes I),$$
$$\langle G|W^n|G\rangle = T_n\left(\frac{H}{\lambda}\right), \tag{B11}$$

where T_n is the nth Chebychev polynomial. The Hamiltonian evolution e^{-iHt} could be given by

$$e^{-iHt} = J_0(-\lambda t) + 2\sum_{n=1}^{+\infty} i^n J_n(-\lambda t) T_n\left(\frac{H}{\lambda}\right). \tag{B12}$$

Namely, one could separately add all terms together, and it requires

$$O\left(\lambda t \log \frac{1}{\epsilon}\right), \tag{B13}$$

number of queries to the operation U and $|G\rangle$. Here t is the Hamiltonian evolution time, and ϵ is the error. Implementing $|G\rangle$ is a quantum oracle operation, which could be operated by QRAM or QROM. For simplicity, we will mostly discuss the query complexity made by $|G\rangle$, and U itself would cost $O(LC_1)$ primitive gates, where C_1 is the maximal complexity of implementing a single Pauli term Π_i. In terms of gate counting, G itself would cost $O(L)$ primitive gates. A more complicated construction, which is called the quantum signal processing (QSP) [57], could reduce the above product in query complexity to addition

$$O\left(\lambda t + \log \frac{1}{\epsilon}\right). \tag{B14}$$

Another ingredient of our analysis would combine the quantum network. Quantum network, based on hardware realizations of quantum teleportation and quantum cryptography, is expected to be efficient for transferring quantum states and their associated quantum data across long distances with guaranteed security [10]. Specifically, we will discuss the quantum repeaters, architectures that could significantly overcome the loss errors and depolarization errors for quantum communication with photons (see, for instance, [12]). For cost estimation, we will follow the discussion in [13]. There are three different generations of quantum repeaters, and we will, for simplicity, discuss them together. A universal measure of cost overhead for those quantum repeaters is the cost coefficient C_2 (C' used in [12]), which could be understood as the qubit × time cost for the transmission of one Bell pair per unit length. Now, we will assume that for the quantum teleportation task of the data center, we use L_{tot} length. The characteristic time is given by t_{ch} (which is different from three different generations of quantum repeaters).

For our minimal definition of the quantum data center, with QDCs serving as the remote oracle resources, one could compute the total time cost T_{total} and the qubit cost N_{total} as the following:

$$T_{\text{total}} = O\left(\lambda t + \log \frac{1}{\epsilon}\right) \times [O(LC_1) + O(M \log^2 L) + t_{\text{ch}}],$$
$$N_{\text{total}} = O(\log L) + O(\log \max_i \dim \Pi_i) + O\left(\frac{L}{M} + \log L\right)$$
$$+ O\left(\log L \times \frac{C_2}{t_{\text{ch}}} L_{\text{tot}}\right). \tag{B15}$$

We will give the following explanations to the above formula:

(1) The first term in the time cost, $O(\lambda t + \log \frac{1}{\epsilon})$, is exactly the query complexity of QSP. Based on our minimal definition of the quantum data center, the cost of each query, including the quantum communication cost and the QRAM/QROM cost.

(2) The term $O(LC_1)$ in the time cost corresponds to the cost of each U in the QSP algorithm.

(3) The parameter M corresponds to the parameter of the hybrid QRAM-QROM construction [24,25], which is a way to unify the hardware-time cost. A pure QRAM would cost $O(L)$ qubits in $O(\log L)$ time, while QROM would cost $O(\log L)$ qubits in $O(L \log L)$ time. With the tunable parameter M, the hybrid construction would cost $O(\log L + L/M)$ qubits within $O(M \log L)$ time, which could reduce to QRAM with $M = 1$ and QROM with $M = L$. This is how the $O(\frac{L}{M} + \log L)$ term comes in the second term.

(4) The form of the oracle $|G\rangle$ is identical to the amplitude encoding oracle, which could reduce to the QRAM definition (the data-lookup oracle) with $O(\log L)$ time cost overhead (without postselection) or $O(1)$ time cost overhead (with postselection) [25]. Here, for simplicity, we are using the case without postselection. Thus, aside from the hybrid QRAM-QROM time cost $O(M \log L)$, we have an extra $O(\log L)$ factor, which gives the $O(M \log^2 L)$ factor in T_{tot}.

(5) The term $O(\log L \times \frac{C_2}{t_{\text{ch}}} L_{\text{tot}})$ comes from the definition of C_2 in quantum repeaters, followed by the actual qubits and the corresponding maximal possible Bell pairs we are using when doing teleportation [13].

As long as we know the exact setups of QDCs, we could decide the resources easily based on our requirement as described by the general setup. Assuming a cost function F_{cost}, one could determine the set of hyperparameters both in quantum communication and quantum simulation by

$$M^*, L^*_{\text{tot}}, \text{other hyperparameters}, \ldots = \arg\min F_{\text{cost}}. \tag{B16}$$

Finally, we mention that QDCs could potentially provide transducers to transform different types of quantum data, for instance, from digital qubits to analog qubits. Since various different forms of qubits have their own advantages and challenges, it is necessary to consider hybrid quantum systems. For example, if we wish to combine quantum computation performed in the superconducting qubit systems, and quantum communication provided by transformations of optical photons across long distances in QDC and its users, quantum transducers might be necessary; see, for instance, [76]. In this example, since the quantum simulation algorithms could be performed by superconducting qubits, while the quantum network could be realized by optical photons, the quantum transducer is needed.

Finally, we consider the case where we count only the hardware cost from users. In the case where we do not have QDCs, the users have to implement QRAM or QROM by themselves in the quantum simulation algorithm. Thus, in the case where users have access to QDCs, we could subtract the hardware contribution from QDC. We could compute the hardware cost ratio between the case where we have QDCs,

and the case where we do not have QDCs ($\overline{\text{QDC}}$). The answer is

$$\frac{\text{QDC}_{\text{hardware}}}{\overline{\text{QDC}}_{\text{hardware}}} = \frac{O(\log L) + O(\log \max_i \dim \Pi_i)}{O(\log L) + O(\log \max_i \dim \Pi_i) + O\left(\frac{L}{M} + \log L\right)} \approx \frac{O(M \log L)}{O(L)}, \tag{B17}$$

where we take the large L limit. Thus the L-dependent term will be dominant. We could see that, especially when M is not scaling with L, this will be an exponential saving of the hardware cost for QDC users.

3. QDC for computing: Multiple users

In this section we discuss a simple situation where QDC has multiple users and discuss its usage.

Consider the case where multiple users want the same answer of a quantum algorithm. For simplicity, we assume the answer should be classical such that it is able to be copied to multiple users (the result could also be quantum, but then we have to use approximate quantum cloning). We define the hardware cost of the quantum algorithm U for a single user as $f_U(\theta_0, \theta_U)$ where θ_0 is the problem parameter, and θ_U is the hyperparameter of the algorithm. Say that we have k users, and for each user, the network cost of the hardware is $f_I(\theta_0, \theta_I)$ where θ_I is the hyperparameter of the algorithm. Thus, without QDC, calculations are performed independently from each user, and the total hardware cost scales as

$$f(\theta_0, \theta_U) = k f_U(\theta_0, \theta_U). \tag{B18}$$

With the QDC, the hardware cost will scale as

$$f(\theta_0, \theta_U, \theta_I) = f_U(\theta_0, \theta_U) + k f_I(\theta_0, \theta_I). \tag{B19}$$

Thus, the condition of the advantage of QDC is given by

$$\frac{f_I(\theta_0, \theta_U)}{f_U(\theta_0, \theta_I)} \ll \frac{k-1}{k}, \tag{B20}$$

and so we could define the ratio

$$r = \frac{k}{k-1} \frac{f_I(\theta_0, \theta_I)}{f_U(\theta_0, \theta_U)}. \tag{B21}$$

The smaller r is, the more useful QDC should be. The optimal r could be given by

$$r^*(\theta_0, k) = \frac{k}{k-1} \min_{\theta_U, \theta_I} \frac{f_I(\theta_0, \theta_U)}{f_U(\theta_0, \theta_I)}. \tag{B22}$$

Here we make some comments about the above calculation. The observation of comparing the communication cost and the computational cost is one of the original motivations of QDCs: using teleportation, one could save computational costs for multiple users. The r coefficient we defined here, and its possible variant, could serve as a generic measure for such observations. However, the task we described before is not using the full features of QDCs. If we teleport quantum states using the quantum network, the state itself is not copiable to multiple users (even though we could copy the state approximately, but the error might be significant). One could use classical networks instead, or encode the classical output to quantum repeaters and make use of quantum networks. Thus, in this case, the quantum network may not necessarily be needed. It could serve as a version of QDC where QRAM is used, but the quantum network is not (see another example where we use the quantum network but not QRAM in Sec. D 2). Moreover, we expect that the above generic protocol could be improved and extended to more practical applications in the real science or business situation, and the simple analysis presented here could be general guidance towards those applications.

4. Surface code and the T-gate counting

First, we give a brief comment on the alternative "smart" usage of QDC-assisted quantum computing. In fact, the native approach would incur a prohibitive $O(\sqrt{N})$ communication cost per query. In contrast, by outsourcing entire queries to the QDC, one effectively funnels a large amount of "magic" [the $O(\sqrt{N})$ magic states required to implement a query] into a very small number of transmitted qubits [the $O(\log N)$ qubits comprising the query's output]. In this way, the user receives maximal assistance from the QDC at minimal communication cost.

Moreover, we give a brief review of the T-gate counting techniques that are developed in [44] about surface-code quantum computation. Those techniques are based on a formalism of executions in a fault-tolerant surface-code architecture from a given quantum circuit (quantum algorithm). Estimations of hardware-time trade-off for given quantum algorithms, using this formalism, are based on the hardware and algorithm assumptions, which might be different compared to other protocols (see, for instance, [77]). Further details can be found in [44].

The formalism is established from making assumptions about basic qubit manipulations. Simple operations such as qubit initializations and single-patch measurements can be regarded as easy, and they will cost 0🕒, while operations like two or multiple-qubit measurements and patch deformation will cost 1🕒. Here, the time unit 1🕒 might be based on the real hardware. In the examples of [44] we can set 1🕒 $= 1\,\mu$s.

The procedure of estimating the hardware-time cost for a given quantum circuit is the following. First, we decompose the target unitary operation as Clifford $+T$-gates. Usually, we assume that the Clifford gates are cheap and T gates are expensive. In fact, T gates could be regarded as classical operations, but a given T-gate will require consumption of a single magic state, $|0\rangle + e^{i\pi/4}|1\rangle$. We need to use magic-state distillation [42] to generate high-quality magic states in the large-scale quantum computation.

Further treatment of a series of Clifford $+T$ gates will contain designing data blocks (blocks of tiles where the data qubits live), distillation blocks (blocks of tiles to distill magic states), and their combinations. In [44], several protocols are concretely discussed for hardware-time costs. Finally, for given large-scale quantum algorithms, precise designs are presented to minimize the hardware-time cost, especially the costs from T gates and magic-state distillation, and the costs could be pinned down to the number of qubits, gates, and even

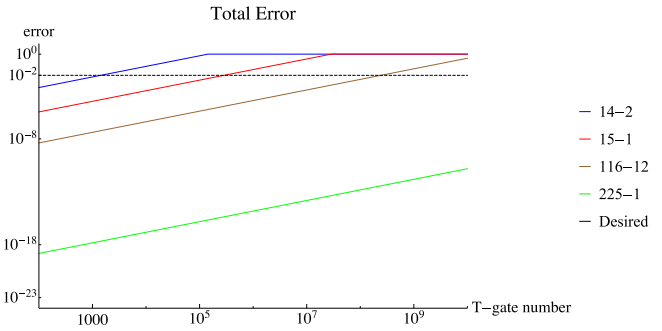

FIG. 8. Dependence of the total error on the physical error rate p for different magic-state distillation schemes according to the approximation in [44]. One could use this dependence to determine the optimal magic-state distillation scheme for a given number of T-gate costs. In our example we want the total error to be smaller than 10^{-2} (desired). Following the orders from up to down, we have 14-2 (blue), 15-1 (red), 116-12 (brown), and 225-1 (green) different distillation protocols showing in the plot as solid lines, compared to the dashed line 10^{-2}.

hours of time costs from assumptions of 🕐. In the T-gate example in the main text, we point out that QDCs could serve as a T-gate factory and could reduce the T-gate counts significantly.

Moreover, we discuss some details about the calculation in the main text with the help of [44]. Based on the setup of qubit numbers and the required target failure probability, the main takeaways from this analysis are as follows. First, a 116-to-12 magic-state distillation scheme is sufficient, as the probability of logical error in the distilled magic state is $<10^{-10}$, hence the total probability that any of the 10^8 magic states is faulty is $<1\%$. Once the distillation scheme is chosen, the total number of surface code tiles (210) and cycles ($11d \times 10^8$) required by the algorithm can be determined, and hence the minimum code distance can be calculated. For the above parameters, a distance $d = 27$ is required to keep the total logical error probability below 1%. This translates into a cost of 306 000 physical qubits and a runtime of 7 h (assuming each surface code cycle takes 1 μs). These costs constitute a baseline for our later comparisons. In Fig. 8 we are presenting the total failure probability of computations for different magic-state distillation schemes, depending on different error rates of devices p. Following [44], we are using the following formulas to estimate the total error:

$$\begin{aligned} \text{total error}(p)_{14-2} &\approx 7p^2, \\ \text{total error}(p)_{15-1} &\approx 35p^3, \\ \text{total error}(p)_{116-12} &\approx 41.25p^4, \\ \text{total error}(p)_{225-1} &\approx 35(35p^3)^3, \end{aligned} \tag{B23}$$

before the total error meets 1. This figure could directly reveal the proper choice of the magic-state distillation schemes. For instance, in our case, we are demanding the total error to be smaller than 10^{-2}. When the number of T gates is 10^8, the 116-to-12 magic-state distillation scheme is sufficient.

Now, in the same situation of the main text, we make an analysis on the pure hardware savings depending on the

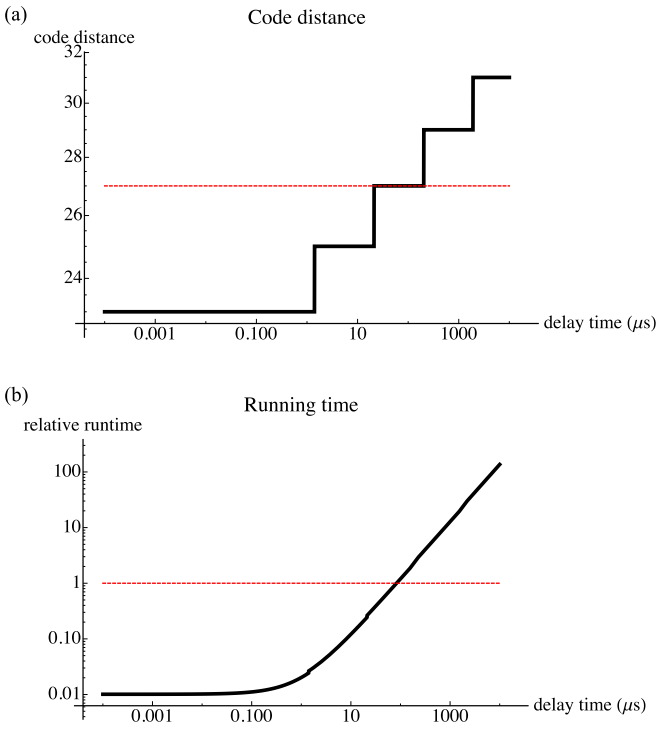

FIG. 9. QDC-assisted code distance (a) and the relative running time (b) depending on the delay time. The relative quantities are measured against the situation without QDC. The red dashed line represents the threshold where QDC has the same performance as the situation without QDC.

delay factor and the delay time. In Fig. 9 we investigate the code distance d, the hardware cost ratio (the number of qubits used in the QDC situation divided by the one without QDC), and the time cost ratio (the time cost used in the QDC situation divided by the one without QDC), depending on the delay factor or the actual delay time, assuming 1 μs per code cycle. Since the delay factor is not related to the choice of magic-state distillation schemes, in the QDC situation, we keep the scheme the same (the 15-1 protocol). We could see that when the delay factor is high, namely, we have a relatively large waiting time from the quantum communication, we are not able to obtain a significant advantage from using QDC. However, if we assume that the quantum communication is fast and the delay factor is small, we are able to save more hardware and time by using QDC. For instance, when the delay factor $\leqslant 21$ [$\leqslant O(10\ \mu s)$ for the delay time], we could set the code distance from $d = 27$ to $d = 25$. When the delay factor $\leqslant 1910$ ($\leqslant O(1$ ms) for the delay time), the QDC could outperform the situation without QDC measured by hardware overhead. When the delay factor $\leqslant 83$ [$\leqslant O(100\ \mu s)$ for the delay time], the QDC could outperform the situation without QDC measured by time overhead.

APPENDIX C: COMMUNICATION

In the context of quantum communication, QDCs can be used to guarantee privacy, with a variety of potential applications. The essential feature of the QDC that enables this privacy is the ability of QRAM to perform queries to data

in superposition. By secretly choosing to perform classical queries or superposed queries, then examining the results, users can determine whether other parties (including the QDC) may have tampered with the queries.

This basic idea is operationalized in the Quantum Private Queries (QPQ) protocol of [26], which we describe below. This protocol allows users to access classical data with privacy guarantees, and this same idea can be applied to enable efficient blind quantum computation [60] (also described below). Both of these protocols can be directly implemented using a QDC.

1. Quantum Private Queries

A QDC can be directly used to implement the quantum private queries protocol of [26]. In the protocol, a user (Alice) wants to access some classical data that is stored in a remote database (held by Bob). Alice wishes to access the data without revealing to Bob which data elements she has accessed. At the same time, Bob wants to maintain the privacy of his database, sending Alice only the information she requests.

The protocol of [26], Quantum Private Query (QPQ), guarantees both user and database privacy by storing the data in QRAM. To access the ith element of a length-N database, Alice prepares a $\log N$-qubit register in the state $|i\rangle$ and transmits this state to Bob. Then Bob uses this state as input to a QRAM query, so that the corresponding classical data, x_i, are encoded in an output qubit register. Both the input and output registers are then returned to Alice. As such, database privacy is guaranteed because Bob must transmit only one element of the data back to Alice. To guarantee user privacy, Alice randomly chooses to send either initial state $|i\rangle$ or a lure state $(|i\rangle + |0\rangle)/\sqrt{2}$ to Bob (which she chooses is unknown to him). By performing measurements on the states Bob returns, Alice can ascertain whether or not Bob has attempted to learn the value of i. Thus, Alice can guarantee her privacy.

The implementation of this protocol with a QDC is not hard. The QDC consists of a QRAM, so the QDC simply plays the role of Bob in the protocol. Moreover, QDC provides an application of the QPQ protocol through the quantum network.

Now we quantify the protocol more precisely. In fact, instead of considering the states $|i\rangle$ and $(|i\rangle + |0\rangle)/2$, we could consider more general states [26]. Bob needs to make choices in one of the two following scenarios:

$$|S_A\rangle = |j\rangle_{Q_1} \otimes \frac{1}{\sqrt{2}}(|j\rangle_{Q_2} + |r\rangle_{Q_2}),$$
$$|S_B\rangle = \frac{1}{\sqrt{2}}(|j\rangle_{Q_1} + |r\rangle_{Q_1}) \otimes |j\rangle_{Q_2}, \tag{C1}$$

where $S_{A,B}$ are made by the joint states of two queries: Q_1 and Q_2. All possible operations from Bob could be summarized by two unitaries: U_1 and U_2. U_1 (U_2) acts on the query space Q_1 (Q_2), the associated register system R_1 (R_2), and Bob's ancillary system B (now we could understand it as Bob's QDC). If Bob is honest, the algorithm of Bob is to make use of QRAM, uploading the information from Q_2 to registers, and the states in Q_2 will not be changed. If not, Bob's remaining system Q_2 will be entangled with the rest at the end. One could compute the final state of Alice:

$$\rho_\ell(j) \equiv \mathrm{Tr}_B[U_2U_1|\Psi_\ell(j)\rangle\langle\Psi_\ell(j)|U_1^\dagger U_2^\dagger], \tag{C2}$$

where $\ell = A, B$, and

$$|\Psi_\ell(j)\rangle = |S_\ell\rangle_{Q_1Q_2}|0\rangle_{RB}. \tag{C3}$$

Moreover, the final state of Q_2 is given by

$$\sigma_\ell(j) \equiv \mathrm{Tr}_{Q_1Q_2R_1R_2}[U_2U_1|\Psi_\ell(j)\rangle\langle\Psi_\ell(j)|U_1^\dagger U_2^\dagger]. \tag{C4}$$

One could quantify the amount of information Bob could obtain from Alice by the mutual information I_B. We will use the Holevo information associated with the ensemble $\{p_j, \sigma(j)\}$, where $p_j = 1/N$ is the probability for choosing j, and $\sigma(j) = [\sigma_A(j) + \sigma_B(j)]/2$ is the final state of Q_2, since Alice has an equal probability to choose $\ell = A, B$. Thus one could obtain [78]

$$I_B \leqslant c\epsilon_p^{1/4}\log_2 N. \tag{C5}$$

Here c is a constant, $c \leqslant 631$, and ϵ_p is the maximal probability where Alice finds that Bob is not cheating. Namely, if we use $1 - P_\ell(j)$ to denote the probability where Bob will pass Alice's test, then $P_\ell(j) \leqslant \epsilon_p$. As a summary, I is a measure of how *honest* Bob could actually be, and ϵ_p is a result of Alice's test. The above inequality is originated from the information-disturbance trade-off and the Holevo bound [27]. The $\epsilon_p^{1/4}$ dependence is coming from repetitively taking the square root between amplitudes and probabilities in quantum mechanics.

Now we relate ϵ_p by the number of queries appearing in the QPQ protocol. Let us assume that Alice has Q queries independently sent to Bob. Note that for multiple queries, if there is at least one time when Alice finds that Bob is cheating, Alice will know that privacy is not guaranteed. So the probability of *Alice cannot find Bob is cheating among Q_B times in all Q times*, is given by $(1 - \epsilon_p)^{Q_B}$, where ϵ_p is the maximal probability where Alice finds that Bob is cheating in a single time. When Q_B increases, $(1 - \epsilon_p)^{Q_B} = a$ will decay from 1 to an $O(1)$ number a where $1 - a$ is not ignorable, and we assume that $Q_B \ll 1/\epsilon_p$. In this case, we have

$$Q_B = \frac{\log a}{\log(1 - \epsilon_p)} \approx \frac{\log \frac{1}{a}}{\epsilon_p} = O\left(\frac{1}{\epsilon_p}\right). \tag{C6}$$

Now, we get

$$I_B \leqslant O\big(Q_B^{-1/4}\log N\big). \tag{C7}$$

Thus, we see that for larger Q_B, if Alice does not find Bob is cheating, then Alice could be more confident that Bob has less mutual information. One can also assume that Bob picks a cheating strategy by $Q_B \sim Q^\alpha$, where $0 \leqslant \alpha \leqslant 1$ will imply how many times Bob is cheating during the whole process. So we have

$$I_B \leqslant O(Q^{-\alpha/4}\log N). \tag{C8}$$

When designing the QDC associated with QPQ, we could introduce a joint cost measurement among time, space, and privacy. Similar to the analysis about quantum signal processing, we write the costs for the QDC when implementing QPQ

as

$$T_{\text{total}} = O(QM \log N) + O(Qt_{\text{ch}}),$$
$$N_{\text{total}} = O\left(\frac{N}{M} + \log N\right) + O\left(\log N \times \frac{C_2}{t_{\text{ch}}} L_{\text{tot}}\right), \quad \text{(C9)}$$

where N is the number of qubits, Q is the total number of queries Alice has sent, M is the parameter in the hybrid QRAM/QROM construction, t_{ch} is the teleportation time per query, C_2 is the teleportation qubit × time cost for the transmission of one Bell pair per unit length, and L_{tot} is the total length during teleportation. For the mutual information I_B, one could understand I_B as the privacy cost, by defining P_{total} as a monotonically decreasing function of I_B, since the smaller I_B, the larger privacy we are requiring. For simplicity, we could define $P_{\text{total}} = 1/I_B$. It does not matter how we choose the monotonic function, since a redefinition of the function could be absorbed to the definition of the cost function F_{cost}. Moreover, one could also understand I_B as part of the hardware and the time cost, since we could write

$$Q = O\left(I_B^{-4/\alpha} \log^{4/\alpha} N\right). \quad \text{(C10)}$$

If we demand a fixed value of I_B, we could adapt Q into the hardware and the time cost. The larger Q is, the higher costs are required:

$$T_{\text{total}} = O\left(M \times I_B^{-4} \log^{4/\alpha+1} N\right) + O\left(t_{\text{ch}} \times I_B^{-4} \log^{4/\alpha} N\right),$$
$$N_{\text{total}} = O\left(\frac{N}{M} + \log N\right) + O\left(\log N \times \frac{C_2}{t_{\text{ch}}} L_{\text{tot}}\right). \quad \text{(C11)}$$

The total cost estimation of QDC associated with QPQ will be a joint measurement among T_{total}, N_{total}, and P_{total}.

Note that in the main text, we discuss the combination of QPQ with the quantum secret sharing protocol. The original proposal about quantum secret sharing given in [9] is based on the entanglement property of the Greenberger-Horne-Zeilinger (GHZ) state, and it is a quantum scheme for sharing classical data since it relies on the measurement result. Moreover, in [8], a quantum scheme for sharing quantum data has been proposed, which is more suitable in our context. The paper [8] constructs a $[(k, n)]$ threshold scheme, where a quantum state is divided into n shares, while any k of them could completely reconstruct the state, but any $k-1$ of them cannot. It was shown that as long as $n < 2k$, the construction is possible, and the explicit scheme has been constructed. In our case, a single Alice could divide the information to n QDCs. We could assume arbitrary k such that $n < 2k$. Based on the practical purposes, a more specific setup of k might be used. A joint analysis of the multiparty quantum communication parameters might set the security standard for a given set of hardware, which we defer for future research.

Finally, we mention that there are studies about limitations and insecurity concerns about concepts that are related to QPQs. In fact, there are no-go theorems [49–51] about the imperfection of certain quantum computation and communication schemes. The QPQ protocol does not violate the no-go theorems [51] since the setup is different. In QPQ, we do not have the security requirement required for the no-theorem, where the user Alice cannot know the private key of QDCs, since QDCs do not have private keys. On the other hand, it will be interesting to understand better how those studies could potentially improve the capability of QDCs.

2. Efficient blind quantum computation

A QDC can be directly used to implement the efficient blind quantum computation protocol of [60]. In blind quantum computation, a user, Alice, wants to perform a quantum computation using Bob's quantum computer without revealing to Bob what computation has been performed. Reference [60] shows how this is possible through a simple application of the QPQ protocol. Bob holds a length-N database stored in QRAM, where each entry in the database corresponds to a different unitary operation that he can perform on his quantum computer. Alice tells him which operation to perform by sending a $\log N$-qubit quantum state $|i\rangle$, indicating that Bob should apply the ith unitary operation, U_i. Bob applies the operation without measuring the register (i.e., he applies a coherently controlled operation $U_{\text{Bob}} = \sum_i |i\rangle\langle i| \otimes U_i$) and then sends the state back to Alice. To protect her privacy, Alice periodically sends lure states and measures the states returned to her. If Bob attempts to cheat, Alice will be able to detect it.

The implementation of this protocol with a QDC is simple, as we have already shown that a QDC can implement the QPQ protocol. In this case, though, the QDC also requires a universal quantum computer in order to perform the computation. Thus, the efficient blind quantum computation protocol could be understood as an extension of QPQ with extra powers in quantum computation. When estimating the computational cost for QDC, one should include the computational complexity of the quantum operation U_i, while other analyses stay the same as QDC associated with QPQ.

3. Final comments on multiparty private quantum communication

Private quantum communication refers to the possibility of transmitting quantum information without revealing this information to eavesdroppers. If multiple parties are communicating over a quantum network, eavesdroppers may nevertheless be able to learn who has sent information and who has received it, even if they cannot determine what that information was. Multiparty private quantum communication refers to a stronger notion, where eavesdroppers can neither learn what information was communicated *nor which users were communicating to which others*. To our knowledge, this notion of *multiparty private quantum communication* and the corresponding protocol is unique in our paper.

This protocol constitutes private multiparty quantum communication because (1) the use of secret sharing means that no QDC can learn what information is being communicated, and (2) the use of Quantum Private Queries means that no QDC can learn which user B_j is accessing the information transmitted by A_i. A crucial assumption in the protocol is that the QDCs are noncooperating. If the QDCs cooperate, they could work together to reconstruct the secret. To mitigate this problem, the number of parts each secret is divided into can be increased (along with the number of QDCs). In this way,

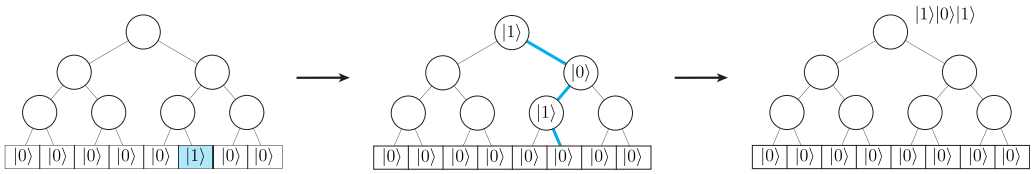

FIG. 10. Quantum compression with QRAM. (a) Compression procedure. The bucket-brigade QRAM consists of a binary tree of quantum routers [24], with the N-qubit quantum memory located at the bottom of the tree. An excitation initially stored in the memory is routed up and out of the tree, such that its initial position is first encoded in the states of the quantum routers, then in the state of an external register. The excitation is routed up through the tree using the routing circuit shown in (b). The action of this circuit is shown diagrammatically in (c). We note that this compression procedure is a straightforward extension of the routing procedure described in Ref. [24], which provides a more detailed description of QRAM and quantum routing.

revealing the secret would require cooperation between an increasingly large number of QDCs. We comment on the difference between multiparty private quantum communication and so-called covert quantum communication [79]. Covert quantum communication refers to a stronger notion, where eavesdroppers cannot even detect whether any information has been transmitted in the first place. In the protocol of [79], Alice and Bob are assumed to share a random, secret key, and the quantum information is sent via optical photons from Alice to Bob at one of N times specified by the key. The probability that an eavesdropper can distinguish between this situation and that where no information is communicated at all (i.e., when no photons are sent) is shown to decrease as $1/\sqrt{N}$. In the limit of large N, the eavesdropper cannot determine whether any information has been sent.

Note that the multiparty private quantum communication scheme is teleporting quantum states, not classical information. Those quantum states are naturally merged with quantum private queries where the security is guaranteed quantumly, making the usage of superposition of addresses in QRAM. Moreover, an important technicality regarding the last step of the protocol is that the QDCs are storing quantum data, and, in general, the act of accessing these data can perturb the quantum database (a consequence of the no-cloning theorem). The QDC could, in principle, detect this perturbation and use this information to infer which user B_j is accessing the information transmitted by A_i. To prevent this, we suppose that the quantum data are accessed as follows. In addition to sending a state $|i\rangle$ specifying which element to access, each user B_j also sends a quantum state ρ_j to the QDC. The QDC then swaps ρ_j with the state stored at location i in the memory. If ρ_j is chosen to be a maximally mixed state, this data access procedure has no backreaction on the database; from the perspective of each QDC, the states stored in the database always look maximally mixed.

APPENDIX D: SENSING

1. Quantum data compression

In this section we describe how QRAM architectures can be used to implement the unary-to-binary compression described in our main text, which could be used for quantum sensing. In particular, we show that the compression operation can be implemented using a modified version of the bucket-brigade QRAM archiecture [17,18]. In the following, we assume familiarity with this QRAM architecture; we refer unfamiliar readers to Ref. [24], which provides a recent, self-contained review.

The basic compression scheme is illustrated in Fig. 10. Suppose a single excitation is stored in one of N different cells in the QRAM's quantum memory (or in a superposition of multiple different cells). The procedure in the figure allows one to coherently extract the position of the excitation using a modified version of the bucket-brigade QRAM's binary-tree routing scheme. Specifically, the excitation is routed upward from the quantum memory at the bottom of the tree to the root node at the top. As the excitation is routed upward, its original position is encoded into the states of the quantum routers comprising the tree. This encoding is accomplished using a simple modification to the quantum routing circuit of Ref. [24], shown in Figs. 10 (b) and 10(c). Subsequently, the position information is extracted from the routers and stored in an external $\log N$-qubit register, exactly as in the usual bucket-brigade approach [17,18,24].

In this way, QRAM enables us to convert the unary information of the photon's position into a more compact binary representation of $\log N$ qubits. That is, the scheme in Fig. 10 implements the mapping $|\psi_{\text{unary}}\rangle \to |\psi_{\text{binary}}\rangle$ described in this Appendix. As a result, the state of N-mode memory (assumed to lie within the single-excitation subspace) can be compressed to $\log N$ qubits. Generalization to multiexcitation subspaces is straightforward; the procedure can be repeated to extract multiple excitations from the memory, such that the k-excitation subspace can be compressed into $k \log N$ qubits.

Additionally, the scheme in Fig. 10 can also be used to implement the operation U that coherently extracts the address of an excitation stored in the QRAM's quantum memory. To do so, first, the compression of procedure in Fig. 10(a) is applied:

$$\sum_{i=0}^{N-1} \alpha_i |i\rangle^{Q_1'} |0\rangle^{Q_2} \left[\bigotimes_{j=1}^{N} |\delta_{ij}\rangle^{D_j} \right] \to \sum_{i=0}^{N-1} \alpha_i |i\rangle^{Q_1'} |1\rangle^{Q_2} \left[\bigotimes_{j=1}^{N} |0\rangle^{D_j} \right], \tag{D1}$$

where here the labels Q_1, Q_2, and D_j, respectively, denote the external $\log N$-qubit register, an external qubit used to hold the extracted excitation (not shown in Fig. 10), and the jth cell of the quantum memory. Then a series of CNOT gates are used to copy the address information stored in register Q_1 into

another external $\log N$-qubit register, denoted Q_1,

$$\to \sum_{i=0}^{N-1} \alpha_i |i\rangle^{Q_1} |i\rangle^{Q_1'} |1\rangle^{Q_2} \left[\bigotimes_{j=1}^{N} |0\rangle^{D_j} \right]. \tag{D2}$$

Finally, the compression procedure is run in reverse in order to return the excitation to its original location in memory:

$$\to \sum_{i=0}^{N-1} \alpha_i |0\rangle^{Q_1} |i\rangle^{Q_1'} |0\rangle^{Q_2} \left[\bigotimes_{j=1}^{N} |\delta_{ij}\rangle^{D_j} \right]. \tag{D3}$$

The Q_1' and Q_2 registers can subsequently be discarded.

Finally, we comment on how QDC-assisted distributed sensing could provide benefits on the entanglement cost. Suppose that two or more physically separated users probe some system, such that a quantity of interest is encoded in an entangled state shared between them. If local operations do not suffice to measure this quantity, then quantum information must be transmitted between the users. If each user has N qubits of quantum data, then N entangled pairs will be required to transmit the information in general. In certain situations, this entanglement cost can be greatly reduced using a QDC. For example, if the N-qubit states are guaranteed to lie in the single-excitation subspace, then they can be transmitted using only $\log N$ entangled pairs using the unary-to-binary compression described above. This is the case for quantum-assisted telescope arrays [80], where quantum networks are used to enable optical interferometry. In this context, a single optical photon arrives at one of multiple telescopes in superposition, with its arrival time and frequency unknown. Supposing that the photon arrives at one of N unknown time-frequency bins, Refs. [81,82] show that unary-to-binary compression enables the optical phase difference to be extracted using only $\log N$ entangled pairs. QDCs could be directly used to implement this compression. In fact, using a QDC to implement the compression is more hardware efficient than the approach proposed in Ref. [82]. In that work the authors consider the case where a photon arrives at one of T_{bin} different time bins and in one of R different frequency bands, and they describe a procedure that uses $O(R \log T_{\text{bin}})$ qubits to compresses the photon's arrival time and frequency information. The same compression can be achieved with a QDC using only $O(R) + O(\log T_{\text{bin}})$ qubits as follows. At each time step, any incoming photon is stored in one of R different memory qubits according to its frequency band. These R qubits constitute the QDC's quantum memory, and the QDC performs unary-to-binary compression scheme described above to compress this which-frequency information. If the photon arrived at the present time step, it is now stored at a definite location [namely, register Q_2 in Eq. (6)]. The presence of this photon can then be used to control the binary encoding of the which-time information, as in Ref. [81]. Altogether, this procedure requires $O(R)$ qubits for the QDC and its quantum memory, plus an additional $O(\log T_{\text{bin}})$ qubits to hold the compressed which-time information, hence the total hardware cost is $O(R) + O(\log T_{\text{bin}})$ qubits. When counting the communication time cost, the savings will be more drastic. More sophisticated compression protocols can enable further reduction in entanglement cost. If each mode of an M-mode system is populated with a photon with probability p, then the quantum data can be transmitted as few as $MH(p)$ qubits, where $H(p)$ is the binary entropy, using a scheme for Schumacher coding [52]. Such schemes would further reduce the entanglement cost. Moreover, a QDC equipped with quantum sorting networks (a generalization of QRAM) can implement Schumacher coding in polylogarithmic time [56], enabling improved detector bandwidth.

2. Channel discrimination using QDCs

In the main text, we discuss various aspects of QDC realizations with both QRAM and quantum communications. However, QDCs could still be made without either QRAM or quantum communications. Here we give a simple example from quantum sensing, where QRAM is not necessarily needed.

The estimation and discrimination of quantum channels are natural problems in quantum sensing (see [83–87]). Following [87], one of the simplest problems is the following channel discrimination problem: say that we have two distributions Θ_b with $b = 0, 1$. For a given b, we wish to find out the value of b by accessing the quantum channel

$$\mathcal{E}_b = \exp(i\theta_b H) \qquad \theta_b \sim \Theta_b, \tag{D4}$$

with minimal numbers of times. (For the qubit setup in this paper, we could specify H as Pauli X. For higher dimensional channel discrimination, see [85].) One could define the channel discrimination protocol by the following circuits. The quantum (coherent) protocol corresponds to the following circuit:

$$Q = \mathcal{E}_{b,N} \prod_{\ell=1}^{N-1} (V_\ell \mathcal{E}_{b,\ell}), \tag{D5}$$

where N is the total number of queries, and $\mathcal{E}_{b,\ell}$ corresponds to ℓth copy of the channel $\mathcal{E}_b$. A series of unitaries V_ℓ will define the protocol (one could specify it by QSP angles; see [87]). Say that we define the input state to be $|\psi_{\text{input}}\rangle$ and the output state to be $|\psi_{\text{output}}\rangle$, the success probability is

$$p = \max_{V, \psi_{\text{input}}, \psi_{\text{output}}} |\langle \psi_{\text{output}} | Q | \psi_{\text{input}} \rangle|^2. \tag{D6}$$

For the incoherent (classical) protocol, we could perform N different measurement. Say that for the ℓth measurement, we expect the input $|\psi_{\text{input},\ell}\rangle$ and the output $|\psi_{\text{output},\ell}\rangle$, and we will have the ℓth probability,

$$p_\ell = |\langle \psi_{\text{output},\ell} | V_\ell \mathcal{E}_{b,\ell} | \psi_{\text{input},\ell} \rangle|^2, \tag{D7}$$

and the protocol could be specified by the majority vote [87]. Moreover, one could specify the ξ-hybrid protocol as performing a length-ξ coherent protocol N/ξ times, and the result could be determined by the majority vote of N/ξ trials. It is shown that when the channel is noiseless (Θ_b's are Dirac function distributions), coherent protocols always have the advantage over incoherent protocols. However, when the channel is too noisy, incoherent protocols might be better [87]. Thus, the hybrid protocols might be useful when we increase N.

If the user wants the quantum sensing to have high precision, they might have to go to the large N regime. In the

above example, precision δ is the difference between the mean of the distribution Θ_1 and Θ_2. It is proved that [87] in the noiseless case, the optimal N scales as $1/\delta$. Moreover, the unitaries $V = \{V_\ell\}$ might be hard to construct and design. In those cases, QDC might be useful.

The protocol with QDC could be defined as the following. On the user side, the user could generate the channel $\mathcal{E}_{b,j}$, and the user could also send the information about the distribution Θ_b to a QDC. The QDC could generate a series of unitaries $\{V_\ell\}$ and design the optimal hybrid ξ protocol. Each time when the state passes through the channel $\mathcal{E}_{b,j}$, one could teleport the state by the quantum network to the QDC, and QDC will apply V_ℓ on the state and teleport it back. The measurement could be done either by the QDC or by the user. The majority vote could either be done classically with $O(N/\xi)$ complexity, or quantumly by measuring $O(N/\xi)$ times in the computational basis. The QDC-assisted channel discrimination protocols will have advantages when the user finds it hard to design the optimal circuits $V = \{V_\ell\}$.

[1] R. P. Feynman, in *Feynman and Computation* (CRC Press, 2018), pp. 133–153.
[2] P. W. Shor, SIAM Rev. **41**, 303 (1999).
[3] L. K. Grover, Phys. Rev. Lett. **79**, 325 (1997).
[4] J. Preskill, Quantum **2**, 79 (2018).
[5] P. Wittek, *Quantum Machine Learning: What Quantum Computing Means to Data Mining* (Academic Press, 2014).
[6] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Nature (London) **549**, 195 (2017).
[7] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, Rev. Mod. Phys. **74**, 145 (2002).
[8] R. Cleve, D. Gottesman, and H.-K. Lo, Phys. Rev. Lett. **83**, 648 (1999).
[9] M. Hillery, V. Bužek, and A. Berthiaume, Phys. Rev. A **59**, 1829 (1999).
[10] H. J. Kimble, Nature (London) **453**, 1023 (2008).
[11] M. Caleffi, A. S. Cacciapuoti, and G. Bianchi, Quantum Internet: From Communication to Distributed Computing!, in *Proceedings of the 5th ACM International Conference on Nanoscale Computing and Communication, NANOCOM '18* (Association for Computing Machinery, Reykjavik, Iceland, 2018), pp. 1–4.
[12] S. Muralidharan, J. Kim, N. Lütkenhaus, M. D. Lukin, and L. Jiang, Phys. Rev. Lett. **112**, 250501 (2014).
[13] S. Muralidharan, L. Li, J. Kim, N. Lütkenhaus, M. D. Lukin, and L. Jiang, Sci. Rep. **6**, 20463 (2016).
[14] C. L. Degen, F. Reinhard, and P. Cappellaro, Rev. Mod. Phys. **89**, 035002 (2017).
[15] V. Giovannetti, S. Lloyd, and L. Maccone, Phys. Rev. Lett. **96**, 010401 (2006).
[16] V. Giovannetti, S. Lloyd, and L. Maccone, Nat. Photon. **5**, 222 (2011).
[17] V. Giovannetti, S. Lloyd, and L. Maccone, Phys. Rev. Lett. **100**, 160501 (2008).
[18] V. Giovannetti, S. Lloyd, and L. Maccone, Phys. Rev. A **78**, 052310 (2008).
[19] F.-Y. Hong, Y. Xiang, Z.-Y. Zhu, L.-Z. Jiang, and L.-N. Wu, Phys. Rev. A **86**, 010306(R) (2012).
[20] S. Arunachalam, V. Gheorghiu, T. Jochym-O'Connor, M. Mosca, and P. V. Srinivasan, New J. Phys. **17**, 123010 (2015).
[21] C. T. Hann, C.-L. Zou, Y. Zhang, Y. Chu, R. J. Schoelkopf, S. M. Girvin, and L. Jiang, Phys. Rev. Lett. **123**, 250501 (2019).
[22] O. Di Matteo, V. Gheorghiu, and M. Mosca, IEEE Trans. Quantum Eng. **1**, 4500213 (2020).
[23] A. Paler, O. Oumarou, and R. Basmadjian, Phys. Rev. A **102**, 032608 (2020).
[24] C. T. Hann, G. Lee, S. M. Girvin, and L. Jiang, PRX Quantum **2**, 020311 (2021).
[25] C. Hann, Practicality of quantum random access memory, Ph.D. thesis, Yale University (2021).
[26] V. Giovannetti, S. Lloyd, and L. Maccone, Phys. Rev. Lett. **100**, 230502 (2008).
[27] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information 10th Anniversary Edition* (Cambridge University Press, Cambridge, 2010).
[28] R. Naik, N. Leung, S. Chakram, P. Groszkowski, Y. Lu, N. Earnest, D. McKay, J. Koch, and D. Schuster, Nat. Commun. **8**, 1904 (2017).
[29] N. Jiang, Y.-F. Pu, W. Chang, C. Li, S. Zhang, and L.-M. Duan, npj Quantum Inf. **5**, 28 (2019).
[30] S. Langenfeld, O. Morin, M. Körber, and G. Rempe, npj Quantum Inf. **6**, 86 (2020).
[31] R. Babbush, C. Gidney, D. W. Berry, N. Wiebe, J. McClean, A. Paler, A. Fowler, and H. Neven, Phys. Rev. X **8**, 041015 (2018).
[32] G. H. Low, V. Kliuchnikov, and L. Schaeffer, arXiv:1812.00954.
[33] D. W. Berry, C. Gidney, M. Motta, J. R. McClean, and R. Babbush, Quantum **3**, 208 (2019).
[34] O. Di Matteo, *Methods for Parallel Quantum Circuit Synthesis, Fault-Tolerant Quantum RAM, and Quantum State Tomography* (University of Waterloo, 2019).
[35] K. C. Chen, W. Dai, C. Errando-Herranz, S. Lloyd, and D. Englund, PRX Quantum **2**, 030319 (2021).
[36] J. Yin, Y. Cao, Y.-H. Li, S.-K. Liao, L. Zhang, J.-G. Ren, W.-Q. Cai, W.-Y. Liu, B. Li, H. Dai *et al.*, Science **356**, 1140 (2017).
[37] S.-K. Liao, W.-Q. Cai, W.-Y. Liu, L. Zhang, Y. Li, J.-G. Ren, J. Yin, Q. Shen, Y. Cao, Z.-P. Li *et al.*, Nature (London) **549**, 43 (2017).
[38] Y.-A. Chen, Q. Zhang, T.-Y. Chen, W.-Q. Cai, S.-K. Liao, J. Zhang, K. Chen, J. Yin, J.-G. Ren, Z. Chen *et al.*, Nature (London) **589**, 214 (2021).
[39] H.-J. Briegel, W. Dur, J. I. Cirac, and P. Zoller, Phys. Rev. Lett. **81**, 5932 (1998).
[40] L.-M. Duan, M. D. Lukin, J. I. Cirac, and P. Zoller, Nature (London) **414**, 413 (2001).
[41] W. J. Munro, K. Azuma, K. Tamaki, and K. Nemoto, IEEE J. Sel. Top. Quantum Electron. **21**, 78 (2015).
[42] S. Bravyi and A. Kitaev, Phys. Rev. A **71**, 022316 (2005).
[43] S. Bravyi and J. Haah, Phys. Rev. A **86**, 052329 (2012).
[44] D. Litinski, Quantum **3**, 128 (2019).
[45] A. G. Fowler and C. Gidney, arXiv:1808.06709.

[46] J. M. Martyn, Z. M. Rossi, A. K. Tan, and I. L. Chuang, PRX Quantum **2**, 040203 (2021).

[47] There are studies about limitations and insecurity concerns about concepts that are related to QPQs. In fact, there are no-go theorems [49–51] about the imperfection of certain quantum computation and communication schemes. The QPQ protocol does not violate the no-go theorems [51] since the setup is different. In QPQ we do not have the security requirement required for the no-theorem, where the user Alice cannot know the private key of QDCs, since QDCs do not have private keys. On the other hand, it will be interesting to understand better how those studies could potentially improve the capability of QDCs.

[48] Moreover, we remark that numerous extensions of this simple unary-to-binary compression are possible. For example, it is straightforward to generalize the above procedure to the case of multiple excitations, such that a QDC can be used to compress multiexcitation subspaces as well (see related discussions in the Appendixes about quantum data compression). QDCs could even be used for Schmacher coding [52], universal source coding [53–55], or other quantum compression tasks. The ability to efficiently compress quantum data using QRAM and related architectures is studied more thoroughly in [56].

[49] D. Mayers, Phys. Rev. Lett. **78**, 3414 (1997).

[50] H.-K. Lo and H. F. Chau, Phys. Rev. Lett. **78**, 3410 (1997).

[51] H.-K. Lo, Phys. Rev. A **56**, 1154 (1997).

[52] B. Schumacher, Phys. Rev. A **51**, 2738 (1995).

[53] R. Jozsa, M. Horodecki, P. Horodecki, and R. Horodecki, Phys. Rev. Lett. **81**, 1714 (1998).

[54] M. Hayashi and K. Matsumoto, Phys. Rev. A **66**, 022311 (2002).

[55] C. H. Bennett, A. W. Harrow, and S. Lloyd, Phys. Rev. A **73**, 032336 (2006).

[56] C. T. Hann *et al.* (unpublished).

[57] G. H. Low and I. L. Chuang, Quantum **3**, 163 (2019).

[58] G. H. Low, T. J. Yoder, and I. L. Chuang, Phys. Rev. X **6**, 041067 (2016).

[59] G. H. Low and I. L. Chuang, Phys. Rev. Lett. **118**, 010501 (2017).

[60] V. Giovannetti, L. Maccone, T. Morimae, and T. G. Rudolph, Phys. Rev. Lett. **111**, 230501 (2013).

[61] J. F. Fitzsimons, npj Quantum Inf. **3**, 23 (2017).

[62] J. Preskill, Nature (London) **402**, 357 (1999).

[63] D. Gottesman and I. L. Chuang, Nature (London) **402**, 390 (1999).

[64] S. Vadhan, in *Tutorials on the Foundations of Cryptography* (Springer, 2017), pp. 347–450.

[65] S. Lloyd, M. Mohseni, and P. Rebentrost, Nat. Phys. **10**, 631 (2014).

[66] S. Lloyd, Science **273**, 1073 (1996).

[67] M. Suzuki, Commun. Math. Phys. **51**, 183 (1976).

[68] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe, and S. Zhu, Phys. Rev. X **11**, 011020 (2021).

[69] A. M. Childs and R. Kothari, in *Theory of Quantum Computation, Communication, and Cryptography*, edited by W. van Dam, V. M. Kendon, and S. Severini (Springer, Berlin, 2011), pp. 94–103.

[70] D. W. Berry, A. M. Childs, and R. Kothari, in *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on* (IEEE, New York, 2015), pp. 792–809.

[71] G. H. Low, V. Kliuchnikov, and N. Wiebe, arXiv:1907.11679.

[72] A. M. Childs and N. Wiebe, Quantum Info. Comput. **12**, 901 (2012).

[73] D. W. Berry and A. M. Childs, Quantum Info. Comput. **12**, 29 (2012).

[74] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Phys. Rev. Lett. **114**, 090502 (2015).

[75] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Exponential improvement in precision for simulating sparse Hamiltonians, in *Proceedings of the forty-sixth annual ACM symposium on Theory of computing, STOC'14* (Association for Computing Machinery, New York, 2014), pp. 283–292.

[76] M. Zhang, C.-L. Zou, and L. Jiang, Phys. Rev. Lett. **120**, 020502 (2018).

[77] D. Herr, F. Nori, and S. J. Devitt, New J. Phys. **19**, 013034 (2017).

[78] V. Giovannetti, S. Lloyd, and L. Maccone, IEEE Trans. Inf. Theory **56**, 3465 (2010).

[79] J. M. Arrazola and V. Scarani, Phys. Rev. Lett. **117**, 250503 (2016).

[80] D. Gottesman, T. Jennewein, and S. Croke, Phys. Rev. Lett. **109**, 070503 (2012).

[81] E. T. Khabiboulline, J. Borregaard, K. De Greve, and M. D. Lukin, Phys. Rev. Lett. **123**, 070504 (2019).

[82] E. T. Khabiboulline, J. Borregaard, K. De Greve, and M. D. Lukin, Phys. Rev. A **100**, 022316 (2019).

[83] A. Acín, Phys. Rev. Lett. **87**, 177901 (2001).

[84] R. Duan, Y. Feng, and M. Ying, Phys. Rev. Lett. **98**, 100503 (2007).

[85] R. Duan, Y. Feng, and M. Ying, Phys. Rev. Lett. **103**, 210501 (2009).

[86] S. Zhou and L. Jiang, PRX Quantum **2**, 010343 (2021).

[87] Z. M. Rossi, J. Yu, I. L. Chuang, and S. Sugiura, Phys. Rev. A **105**, 032401 (2022).