

Blind Quantum Machine Learning with Quantum Bipartite Correlator

Changhao Li^{1,2,3,*} Boning Li^{2,4} Omar Amer,¹ Ruslan Shaydulin¹ Shouvanik Chakrabarti,¹
Guoqing Wang^{2,3,4} Haowei Xu,³ Hao Tang⁵ Isidor Schoch,³ Niraj Kumar,¹ Charles Lim,¹ Ju Li,^{3,5,†}
Paola Cappellaro,^{2,3,4,‡} and Marco Pistoia^{1,§}


¹Global Technology Applied Research, JPMorgan Chase, New York, New York 10017, USA

²Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA

³Department of Nuclear Science and Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA

⁴Department of Physics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA

⁵Department of Materials Science and Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA

 (Received 7 December 2023; revised 18 June 2024; accepted 16 August 2024; published 18 September 2024)

Distributed quantum computing is a promising computational paradigm for performing computations that are beyond the reach of individual quantum devices. Privacy in distributed quantum computing is critical for maintaining confidentiality and protecting the data in the presence of untrusted computing nodes. In this Letter, we introduce novel blind quantum machine learning protocols based on the quantum bipartite correlator algorithm. Our protocols have reduced communication overhead while preserving the privacy of data from untrusted parties. We introduce robust algorithm-specific privacy-preserving mechanisms with low computational overhead that do not require complex cryptographic techniques. We then validate the effectiveness of the proposed protocols through complexity and privacy analysis. Our findings pave the way for advancements in distributed quantum computing, opening up new possibilities for privacy-aware machine learning applications in the era of quantum technologies.

DOI: [10.1103/PhysRevLett.133.120602](https://doi.org/10.1103/PhysRevLett.133.120602)

Introduction—Quantum computation that leverages the principles of quantum mechanics has the potential to tackle problems that are beyond the reach of classical computers, revolutionizing fields ranging from cryptography [1] to finance [2] and drug discovery [3]. Distributed quantum computing has attracted a lot of attention in recent years [4–11] due to the rapid progress in quantum communication technologies. In distributed quantum computing, multiple quantum processors are connected over a network, enabling collaborative computation and resource sharing. This approach is crucial for scaling up quantum computing power and overcoming the limitations of individual quantum systems. Exploiting distributed quantum resources enables tackling larger and more computationally complex problems in domains such as optimization, simulation and quantum machine learning (QML). QML is especially suitable for distributed computation due to the need to process large datasets.

Privacy in distributed computing plays a vital role in ensuring the confidentiality and security of sensitive information processed by multiple parties. Distributed quantum computation involves sharing and transmitting

of quantum states across multiple nodes, making it paramount to protect the privacy of data and prevent unauthorized access. Furthermore, in practice, addressing privacy concerns in distributed quantum computing is essential for facilitating applications in fields such as finance and healthcare, where preserving the privacy of sensitive data is of utmost importance.

A number of protocols have been proposed in recent years that aim to implement private distributed quantum computing. For example, blind quantum computing [12–14] enables the client to execute a quantum computation using one or more remote quantum servers while keeping the structure of the computation hidden. Meanwhile, reducing the overhead in communication over blind quantum computation protocols has been an active research area since the first proposal of universal blind quantum computation [12]. However, for distributed quantum computing problems such as QML, ensuring the privacy of data from a certain party while reducing the overhead in both quantum communication and computation remains a challenge.

In this Letter, we introduce novel protocols for blind distributed quantum machine learning based on quantum bipartite correlator algorithm that can perform inner product estimation task, a core subroutine in many common machine learning applications including linear regression as we will show. Our protocols are communication-efficient compared with state-of-the-art classical and quantum blind distributed machine learning algorithms. Particularly, for

*Contact author: changhao@mit.edu

†Contact author: liju@mit.edu

‡Contact author: pcappell@mit.edu

§Contact author: marco.pistoia@jpmchase.com

the task of distributed inner product estimation, the protocols involve a communication complexity $O(\log N/\epsilon)$ with N and ϵ being the size of the vectors and standard estimation error, respectively. We demonstrate how our protocols allow the client to conceal its data from the server, and vice versa. We provide a detailed resource analysis for both communication and computation costs of our methods. Our Letter paves the way for performing quantum machine learning with an untrusted device, while maintaining the privacy and keeping the resource overhead low.

Formalism—We start by presenting the problem statement in distributed quantum computation. The basic setting includes two parties, Alice and Bob. We assume that Alice has more quantum computational resources than Bob, such as a larger number of qubits. In many distributed quantum computation applications such as a delegated computation setting, Alice can be considered as a quantum server with Bob being a client. Furthermore, there is a quantum channel where qubits can be transmitted between the two parties. For the distributed QML tasks studied in this Letter, we assume that \mathbf{X} and \mathbf{y} are only held locally by Alice and Bob, respectively. For example, in supervised learning, \mathbf{X} and \mathbf{y} could be feature data and labels, respectively [15], while in unsupervised learning, both \mathbf{X} and \mathbf{y} can be feature data with the objective to cluster them based on distance estimation [16].

We consider the task of blind quantum machine learning, such as linear regression or classification [17–20]. In machine learning, evaluating the inner product between two vectors is an important algorithmic building block, with applications in both linear and nonlinear fitting, as well as in the evaluation of cost functions within distributed neural networks [21]. The server holds the data vector \mathbf{X} of size N and the number of features for each data point is M , and the client holds a one-dimensional bitstring \mathbf{y} with the same size N . Note that transmitting the data classically to the server would introduce $O(N)$ complexity in communication.

In classical settings, the goal of achieving distributed machine learning with privacy can be approached using various techniques, such as homomorphic encryption [22,23], which allows computation over encrypted data. Specifically, for distributed bipartite correlation estimation, many methods could be employed, including linearly homomorphic encryption [24,25], noninteractive inner product protocols [26] and oblivious-transfer-based secure computation [27]. However, it is important to note that these classical methods often introduce considerable overhead in terms of computation and communication complexity. Particularly, a communication cost of $O(N)$ would be a minimum requisite [26]. As a result, their practical applications become limited, especially when dealing with large data sizes.

Quantum bipartite correlator algorithm and its privacy—We briefly introduce the quantum bipartite correlator (QBC) algorithm that can estimate the correlation between

two bitstrings held by remote parties [8]. As discussed, estimating bipartite correlation including inner product serves as the building block of a general class of machine learning problems [21,28,29]. Without loss of generality, we consider binary floating point numbers. We take the feature dimension $M = 1$ for simplicity hereafter unless specified. For two vectors $\mathbf{x}, \mathbf{y} \equiv [x_1, \dots, x_N]^T, [y_1, \dots, y_N]^T \in \{0, 1\}^N$, we are interested in evaluating $\overline{\mathbf{x}\mathbf{y}} = (1/N) \sum_{i=1}^N x_i y_i$ within a standard deviation error ϵ . To begin with, we assume that the two parties Alice and Bob hold a local oracle that can encode their own data using a unitary transformation. That is, for Alice, one has $\hat{U}_{\mathbf{x}}: |i\rangle_n |0\rangle \mapsto |i\rangle_n |x_i\rangle$ that encodes the data x_i , where $|i\rangle_n$ is an $n \equiv \lceil \log_2(N) \rceil$ -qubit (called index qubit hereafter) state $|i_1 i_2 \dots i_n\rangle$, representing the index of the queried component with $i_k \in \{0, 1\}$, $k \in [N]$, and $|x_i\rangle$ is a single-qubit state. Similarly, Bob has an oracle $\hat{U}_{\mathbf{y}}$ of the same type that encodes his local data y_i . These oracle operators, as well as the ones introduced later, could be implemented with various techniques such as quantum random access memory [30].

QBC is based on the quantum counting algorithm, where Alice and Bob send qubits via quantum channels and communicate with each other to realize the phase oracle [8,31], as shown in the top of Fig. 1. The quantum

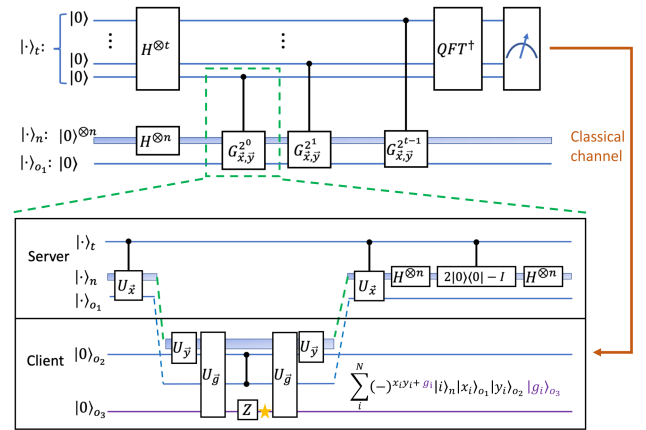


FIG. 1. Diagram for blind QBC with an untrusted server. The upper diagram shows the quantum counting algorithm consisting Grover phase oracles $\hat{G}_{\mathbf{x},\mathbf{y}}$ and inverse QFT, while the lower box panel shows the realization details of each phase oracle. Compared to the original QBC algorithm, we introduce an ancillary qubit o_3 on the client's side to add a phase g_i during the computation process. The phase can be introduced via applying a phase gate on qubit o_3 , which encodes a bitstring that is random and unknown to the server. The detailed phase encoding rule is explained in the text. The quantum state at the star point is shown in the inset of the figure. After the server finishes the quantum circuit, it sends the extracted modified bipartite correlation $(1/N) \sum_i (x_i y_i + g_i)$ to the client via a classical communication channel. We omit the normalization factor for index qubit states in the figures hereafter for simplicity.

counting algorithm consists of a Grover operator $\hat{G}_{\bar{x},\bar{y}} \equiv \hat{H}^{\otimes n} (2|0\rangle_n \langle 0|_n - \hat{I}) \hat{H}^{\otimes n} \hat{U}_{xy}$, where \hat{U}_{xy} is a unitary operator that encodes information of both parties as we will introduce below, and inverse quantum Fourier transform (QFT[†]) on register qubits $|\cdot\rangle_t$. When measuring the t register, one can project it into a state $|j\rangle_t$ with phase $2\pi j \cdot 2^{-t}$ which encodes either $\hat{\theta}$ or $2\pi - \hat{\theta}$, where $\theta = 2 \arcsin \sqrt{\bar{x}\bar{y}}$, with equivalent standard deviation: $\Delta\hat{\theta} = 2^{-t+1}$ [8].

During the phase oracle $\hat{G}_{\bar{x},\bar{y}}$, the following unitary circuit is applied to achieve encoding of x_i and y_i

$$\hat{U}_{xy} |i\rangle_n |00\rangle_{o_1 o_2} = (-1)^{x_i y_i} |i\rangle_n |00\rangle_{o_1 o_2}, \quad (1)$$

where o_1, o_2 are two qubits locally held by Alice and Bob, respectively. The above unitary operator can be implemented with the local oracles that Alice and Bob hold, i.e., $\hat{U}_{\bar{x}}$ and $\hat{U}_{\bar{y}}$. Specifically, Alice encodes her local information x into qubit o_1 via $\hat{U}_{\bar{x}}$ operator and sends the $(n+1)$ -qubit state $(1/\sqrt{N}) \sum_i |i\rangle_n |x_i\rangle_{o_1}$ to Bob via a quantum channel. After Bob applies his oracle and generates the state $(1/\sqrt{N}) \sum_i |i\rangle_n |x_i\rangle_{o_1} |y_i\rangle_{o_2}$, a controlled-Z (CZ) gate between qubit o_1 and o_2 is applied to encode the correlation information into the phase of the quantum state. That is, the bipartite quantum state is described by $(1/\sqrt{N}) \sum_i (-1)^{x_i y_i} |i\rangle_n |x_i\rangle_{o_1} |y_i\rangle_{o_2}$. The following local oracles would then yield the desired state $(1/\sqrt{N}) \sum_i (-1)^{x_i y_i} |i\rangle_n$ for Eq. (1). We note that the CZ gate might be replaced with a different set of gates to estimate other types of correlations such as Hamming distance between x and y [8].

In the QBC algorithm, the communication complexity, i.e., the qubits transmitted during the overall process, is given by the Grover operation's $2(n+1)$ qubits communication repeated for $2^t - 1$ iterations:

$$C_{\text{comm}} = 2(n+1)(2^t - 1) = O\left(\frac{\log_2(N)}{\epsilon}\right), \quad (2)$$

where the number of register qubits t is chosen to satisfy the desired error bound ϵ . We remark that the above communication complexity is advantageous compared with the SWAP-test-based algorithm that has a scaling of $O(\log_2(N)/\epsilon^2)$ [32] or LOCC-based algorithms with a scaling of $O(\log_2(N) \max\{1/\epsilon^2, \sqrt{N}/\epsilon\})$ [33]. This advantage is achieved by utilizing the distributed Grover operations. The computational complexity, on the other hand, is the total number of oracle calls by Alice and Bob $C_{\text{comp}} = 4(2^t - 1) = O(1/\epsilon)$.

We now examine data privacy within the QBC algorithm discussed previously, designating Alice as the server and Bob as the client. We first focus on safeguarding client information y against a semihonest adversary. In this

scenario, the *honest-but-curious* server adheres to the protocol without malicious intent yet attempts to compromise the privacy of the client's input by inferring y from the estimated $(1/N) \sum_i x_i y_i$. In the trivial case when $x_i = 0$, $\forall i \leq N$, we have $\bar{x}\bar{y} = 0$ no matter what y is and the protocol has the best privacy. While in the worst case where the $x_i = 1$, $\forall i \leq N$ and $\bar{x}\bar{y} = 1$, the server could infer that $y_i = 1$, $\forall i \leq N$. In general, for x with Hamming weight d_x , the probability that the server gets the exact y (that is, the Hamming distance between extracted and exact bit-string is $d_0 = 0$) is given by

$$\Pr(d_x) = \frac{1}{2^{N-d_x}} \frac{\prod_{i=1}^{d_x} i}{\prod_{i=1}^{N-\bar{x}\bar{y}} i \prod_{i=1}^{d_x - N\bar{x}\bar{y}} i}, \quad (3)$$

where the factor $(1/2^{N-d_x})$ comes from the server having a random guess on the indices j that satisfies $x_j = 0$.

Blind QBC with an untrusted server—In addition to the semihonest adversary scenario discussed above, we note that in the original QBC algorithm, the preservation of privacy is not assured when we consider a malicious server Alice. The server has the capability to acquire, to a certain extent, Bob's strings y by deviating from the expected quantum operations. One example is that the server could perform quantum gate operations and measurements to extract the phase information instead of following the expected Grover steps after receiving $(1/\sqrt{N}) \sum_i (-1)^{x_i y_i} |i\rangle_n |x_i\rangle_{o_1}$ from the client Bob. Alternatively, a malicious server could potentially manipulate the state of qubit o_1 sent to the client, rather than genuinely encoding the information of x . With the $2^t - 1 = O(1/\epsilon)$ Grover iterations, the server could get $O(1/\epsilon)$ bits of information in y [21,34]. While it is possible to employ a redundant encoding strategy to decrease the probability that the server attains a specific y_i corresponding to an intended index, this approach comes at the expense of increased communication complexity, as detailed in Sec. II of [21].

We thus consider a revised QBC algorithm to counteract the aforementioned attack by the malicious server and protect the privacy of information held by the honest client. A desired protocol for remote blind QBC should have (i) less overhead in quantum communication, (ii) fewer requirements in the computational power of client, (iii) a certified estimation result with error ϵ .

As shown in Fig. 1, inspired by a quantum one-time pad [12], the proposed protocol utilizes phase padding to preserve privacy. The client Bob now has one or more qubits at hand, where he can encode a bit string $|g_i\rangle$ that is blind to the server. That is, the client has an oracle $\hat{U}_{\bar{g}}$ for the extra qubit (denoted as o_3 hereafter), and the modified phase oracle of Eq. (1) reads as

$$\hat{U}_{xyg} |i\rangle_n |000\rangle_{o_1 o_2 o_3} = (-1)^{x_i y_i + g_i} |i\rangle_n |000\rangle_{o_1 o_2 o_3}. \quad (4)$$

To implement the above unitary \hat{U}_{xyg} similar to the \hat{U}_{xy} , the client performs $\hat{U}_{\bar{y}}$ and $\hat{U}_{\bar{g}}$ oracle after receiving state from the server to create the state $(1/\sqrt{N}) \sum_i^N |i\rangle_n |x_i\rangle_{o_1} |y_i\rangle_{o_2} |g_i\rangle_{o_3}$, followed by a controlled-Z gate between o_1 and o_2 . Then a local Z gate can be applied on qubit o_3 to add the phase $(-1)^{g_i}$ that is random to the server.

Since the phase term $(-1)^{x_i y_i + g_i}$ is binary here with modular addition between $x_i y_i$ and g_i , we design the following rule for the application of random phase g_i . For a given index i , when $y_i = 0$, the client chooses a random number from $\{0, 1\}$; while when $y_i = 1$, the client sets $g_i = 0$. Under this setting, the server cannot get y_i in general from direct measurement of the parity at each Grover step, even if the server knows exactly the circuit that the client performs.

The above phase encoding rule on g_i guarantees that $x_i y_i + g_i \in \{0, 1\}$. The quantum counting algorithm can then estimate $(1/N) \sum_i^N (x_i y_i + g_i) = (1/N) \sum_i^N (x_i y_i + g_i \bmod 2)$ with error bound ϵ . Finally, after the measurement, the server sends the estimated result back to the client via a classical channel, from which the client can extract $(1/N) \sum_i^N x_i y_i$ using his local information of $(1/N) \sum_i^N g_i$. Alternatively, depending on the specific use cases, the client could directly share $(1/N) \sum_i^N g_i$ with the server and let it extract the bipartite correlation between \mathbf{x} and \mathbf{y} .

We emphasize that in principle, the aforementioned protocol could still inadvertently leak a portion of the information in \mathbf{y} to the server. As can be seen from the scheme, in the case where $x_j = 1$ and the final phase term is $x_j y_j + g_j = 0$, if the server knows the above application rule of g_j and extracts the phase corresponding to the index qubit $|i\rangle_{i=j}$, it could infer that $y_j = 0$. We consider the worst scenario where the malicious server picks $x_i = 1$, $\forall i \leq N$ and has the client's local phase encoding rule. The server's attack strategy is to measure the phase of a randomly picked index $|i\rangle$ to extract $x_i y_i + g_i$ at each Grover iteration. Then, for \mathbf{y} with Hamming weight d_y , the probability that the server extracts a bit string \mathbf{y}' that is d_0 close ($d_0 \leq d_y$) to \mathbf{y} using the information of the measured phases and without doing random guess is simply given by

$$\Pr(d(\mathbf{y}, \mathbf{y}') = d_0) = \frac{C(d_y, d_0) C[N - d_y, \min(2^t - 1, d_y) - d_0]}{C[N, \min(2^t - 1, d_y)]}, \quad (5)$$

where $C(\cdot, \cdot)$ denotes the binomial coefficient. As can be seen from the analysis above, even in the worst case, the probability that the server can successfully extract part of \mathbf{y} information becomes considerably low when the data size becomes large, particularly when $N \geq 2^t - 1$, while in the original QBC a malicious server could get $2^t - 1$ bits of information from the client during the communication

round. Note that the iteration number $2^t - 1$ yields the standard deviation of the estimated correlation, that is, $2^t - 1 = O(1/\epsilon)$. A less tight error bound ϵ will reduce the number of communication rounds between server and client, thus increasing the privacy of the client's data.

We remark that the quantum communication complexity of the aforementioned algorithm for a blind server is $C_{\text{comm}}^{b_s} = O[\log_2(N)/\epsilon]$, which is the same as the original QBC as depicted in Eq. (2). Moreover, akin to the QBC algorithm, a classical communication channel is needed at the end of QBC to deliver estimation results to the client. In terms of computational overhead experienced by the client, introducing the ancilla qubit o_3 only adds $O(1/\epsilon)$ number of two-qubit phase gates and as a result, does not alter the inherent computational complexity. To this end, the blind QBC protocol proposed here could enable communication-efficient blind distributed machine learning tasks between a server and a client without presupposing substantial quantum resources on the client.

Blind QBC with an untrusted client—We now discuss the scenario where the server would like to estimate $(1/N) \sum_i^N x_i y_i$ while keeping \mathbf{x} hidden from the client at all times during the process. In practical applications such as model-as-a-service platforms [35,36], the server's information, including the model's parameters or training data, should remain hidden from the clients. By hiding the server-side information, they can prevent the client from reverse engineering or extracting valuable information about the underlying model architecture or training data. Under this setting, the protocol should be secure against not only an honest-but-curious client, but also a malicious client who tries to get \mathbf{x} by deviating from the original quantum algorithm.

Here we assume an honest server that follows the protocol exactly without trying to get the label information \mathbf{y} . The goal is then to encode \mathbf{x} when the server sends qubits to the client while running the QBC algorithm. That is, we are interested in designing a privacy-preserving operator \hat{O}_f such that

$$\hat{O}_f \frac{1}{\sqrt{N}} \sum_i^N |i\rangle_n |00\rangle_{o_1 o_2} = \frac{1}{\sqrt{N}} \sum_i^N (-1)^{x_i y_i} |i\rangle_n |00\rangle_{o_1 o_2}. \quad (6)$$

Inspired by quantum key distribution protocols [37] such as BB84 [38], we consider a modified local oracle operator \hat{U}_{X_1} held by the server, where the data information \mathbf{x} is encoded in a different basis (Fig. 2). Specifically, at each iteration of quantum counting algorithm, for a given index i , the server chooses a random number R_i from $\{0, 1\}$. When $R_i = 0$, the server encodes x_i using the Z basis, i.e., $|i\rangle_n |0\rangle_{o_1}$ or $|i\rangle_n |1\rangle_{o_1}$, depending on whether x_i being 0 or 1; if $R_i = 1$, x_i is encoded in the X basis and now the state reads $|i\rangle_n |+\rangle_{o_1}$ or $|i\rangle_n |-\rangle_{o_1}$. Here $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ are the eigenstates of Pauli X operator. This oracle \hat{U}_{X_1} can

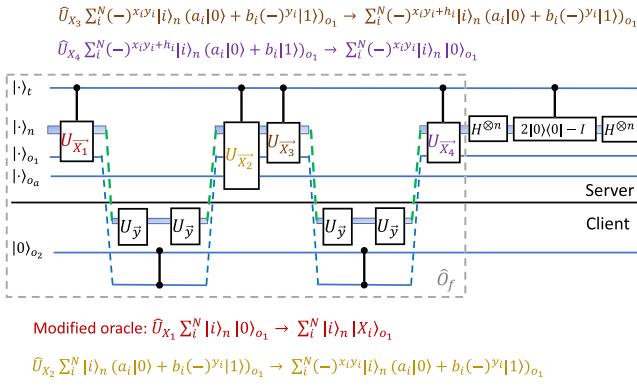


FIG. 2. Grover operator $\hat{H}^{\otimes n} (2|0\rangle_n \langle 0|_n - \hat{I}) \hat{H}^{\otimes n} \hat{O}_f$ for blind quantum bipartite correlator protocol to hide server data \mathbf{x} from client. The operator starts with an oracle held by server (Alice) that encodes \mathbf{x} with random basis (oracle \hat{U}_{X_1}). After receiving the state returned by client (Bob), the server extracts the desired phase term $(-1)^{x_i y_i}$ (\hat{U}_{X_2}) and returns an encoded state back to client (\hat{U}_{X_3}) to remove the phase in o_1 qubit that the server does not know. Finally, the server reaches the target state $(1/\sqrt{N}) \sum_i^N (-1)^{x_i y_i} |i\rangle_n$ by decoupling o_1 qubit with index qubits (\hat{U}_{X_4}).

be implemented with the original oracle $\hat{U}_{\bar{x}}$ with Hadamard gates on o_1 conditioned on index $|i\rangle_n$.

Then, the state received by the client at each time reads as $(1/\sqrt{N}) \sum_i^N |i\rangle_n |X_i\rangle_{o1}$ with X_i being 1(0) or $+(-)$. As the client does not know which basis the server chooses for given i , at each Grover iteration, measurement of qubit o_1 on index $|i\rangle$ will have the probability of yielding both 0 or 1, hence the client cannot infer the x_i information from the single copy of the received $(1/\sqrt{N}) \sum_i^N |i\rangle_n |X_i\rangle_{o1}$ state. Note that the server could pick different random numbers R_i at different communication rounds when executing the QBC algorithm. As shown in Fig. 2, in order to achieve the target \hat{O}_f operator, we utilize oracle operators $\hat{U}_{X_{2,3,4}}$ to generate the phase term $(-1)^{x_i y_i}$ while hiding x_i from the client via random basis encoding and phase padding (see detailed descriptions in the Appendix).

The scheme is based on a random encoding of \mathbf{x} and is information-theoretic secure against an untrusted client, with the proof of security following directly from the corresponding proof for the BB84 protocol [38,39]. The total number of oracle calls by server and client only increases by a constant at each iteration, thus leading to the same computation complexity $O(1/\epsilon)$ as in the original QBC algorithm. The total communication cost of this blind client scheme is given by

$$c_{\text{comm}}^b = 4(n+1)(2^t - 1) = O\left(\frac{\log_2(N)}{\epsilon}\right), \quad (7)$$

which has the same complexity scaling as the original QBC algorithm.

Discussions—We emphasize that our proposed algorithms exhibit direct applicability within the domain of distributed blind machine learning tasks, particularly in scenarios involving matrix or vector multiplication operations. In the Supplemental Material [21], we present examples demonstrating how the blind QBC algorithms can be applied to both linear and nonlinear fitting problems involving continuous label data \mathbf{y} . These algorithms can further be utilized to evaluate fitting performance by estimating the mean square error cost function. Additionally, for classification tasks where the label data is discrete, the proposed algorithms are suitable for estimating the cross-entropy cost function. Furthermore, these algorithms can be generalized to multiparty settings [21,40–42] and find applications in distributed QML tasks such as quantum federated learning [21,43–46]. These applications have demonstrated the broad applicability of secure and communication-efficient inner product estimation, paving the way for future practical implementations in blind quantum machine learning.

We further remark that the proposed quantum algorithms offer many benefits for practical applications with large data sizes. Notably, the quantum communication cost in estimating the bipartite correlation scales as $O(\log N/\epsilon)$ and additionally, the discussed data privacy mechanism does not impose any additional overhead in terms of communication cost. Furthermore, the protocols eliminate the need for a trusted third party and necessitate only a minimal quantum resource allocation from the participating clients, encompassing the number of qubits and gate operations. The protocols can be naturally integrated with standard quantum communication techniques [37,47] to address privacy concerns from a third party [21].

Acknowledgments—J.L. acknowledges support by DTRA (Grant No. HDTRA1-20-2-0002) Interaction of Ionizing Radiation with Matter (IIRM) University Research Alliance (URA).

This Letter was prepared for informational purposes with contributions from the Global Technology Applied Research center of JPMorgan Chase & Co. This Letter is not a product of the Research Department of JPMorgan Chase & Co. or its affiliates. Neither JPMorgan Chase & Co. nor any of its affiliates makes any explicit or implied representation or warranty and none of them accept any liability in connection with this position Letter, including, without limitation, with respect to the completeness, accuracy, or reliability of the information contained herein and the potential legal, compliance, tax, or accounting effects thereof. This document is not intended as investment research or investment advice, or as a recommendation, offer, or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction.

- [1] S. Pirandola, U.L. Andersen, L. Banchi, M. Berta, D. Bunandar, R. Colbeck, D. Englund, T. Gehring, C. Lupo, C. Ottaviani, J.L. Pereira, M. Razavi, J.S. Shaari, M. Tomamichel, V.C. Usenko, G. Vallone, P. Villoresi, and P. Wallden, *Advances in quantum cryptography*, *Adv. Opt. Photonics* **12**, 1012 (2020).
- [2] D. Herman, C. Googin, X. Liu, Y. Sun, A. Galda, I. Safro, M. Pistoia, and Y. Alexeev, Quantum computing for finance, *Nat. Rev. Phys.* **5**, 450 (2023).
- [3] Y. Cao, J. Romero, and A. Aspuru-Guzik, Potential of quantum computing for drug discovery, *IBM J. Res. Dev.* **62**, 6 (2018).
- [4] D. Cuomo, M. Caleffi, and A.S. Cacciapuoti, Towards a distributed quantum computing ecosystem, *IET Quantum Commun.* **1**, 3 (2020).
- [5] M. Caleffi, M. Amoretti, D. Ferrari, D. Cuomo, J. Illiano, A. Manzalini, and A.S. Cacciapuoti, Distributed quantum computing: A survey, [arXiv:2212.10609](https://arxiv.org/abs/2212.10609).
- [6] R. Beals, S. Brierley, O. Gray, A.W. Harrow, S. Kutin, N. Linden, D. Shepherd, and M. Stather, Efficient distributed quantum computing, *Proc. R. Soc. A* **469**, 20120686 (2013).
- [7] A.S. Cacciapuoti, M. Caleffi, F. Tafuri, F.S. Cataliotti, S. Gherardini, and G. Bianchi, Quantum internet: Networking challenges in distributed quantum computing, *IEEE Network* **34**, 137 (2020).
- [8] H. Tang, B. Li, G. Wang, H. Xu, C. Li, A. Barr, P. Cappellaro, and J. Li, Communication-efficient quantum algorithm for distributed machine learning, *Phys. Rev. Lett.* **130**, 150602 (2023).
- [9] A. Montanaro and C. Shao, Quantum communication complexity of linear regression, [arXiv:2210.01601](https://arxiv.org/abs/2210.01601).
- [10] D. Gilboa and J.R. McClean, Exponential quantum communication advantage in distributed learning, [arXiv:2310.07136](https://arxiv.org/abs/2310.07136).
- [11] Y.-B. Sheng and L. Zhou, Distributed secure quantum machine learning, *Sci. Bull.* **62**, 1025 (2017).
- [12] A. Childs, Secure assisted quantum computation, *Quantum Inf. Comput.* **5**, 456 (2005).
- [13] J.F. Fitzsimons, Private quantum computation: An introduction to blind quantum computing and related protocols, *npj Quantum Inf.* **3**, 23 (2017).
- [14] V. Giovannetti, L. Maccone, T. Morimae, and T.G. Rudolph, Efficient universal blind quantum computation, *Phys. Rev. Lett.* **111**, 230501 (2013).
- [15] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, and J.S. Rellermeier, A survey on distributed machine learning, *ACM Comput. Surv.* **53**, 1 (2020).
- [16] H.B. Barlow, Unsupervised learning, *Neural Comput.* **1**, 295 (1989).
- [17] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum algorithms for supervised and unsupervised machine learning, [arXiv:1307.0411](https://arxiv.org/abs/1307.0411).
- [18] P. Rebentrost, M. Mohseni, and S. Lloyd, Quantum support vector machine for big data classification, *Phys. Rev. Lett.* **113**, 130503 (2014).
- [19] T. Li, S. Chakrabarti, and X. Wu, Sublinear quantum algorithms for training linear and kernel-based classifiers, [arXiv:1904.02276](https://arxiv.org/abs/1904.02276).
- [20] X. Zhou and D. Qiu, Blind quantum machine learning based on quantum circuit model, *Quantum Inf. Process.* **20**, 363 (2021).
- [21] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/PhysRevLett.133.120602> for details of the proposed algorithms and their applications.
- [22] H. Fang and Q. Qian, Privacy preserving machine learning with homomorphic encryption and federated learning, *Future Internet* **13**, 94 (2021).
- [23] A. Wood, K. Najarian, and D. Kahrobaei, Homomorphic encryption for machine learning in medicine and bioinformatics, *ACM Comput. Surv.* **53**, 1 (2020).
- [24] P. Paillier, Public-key cryptosystems based on composite degree residuosity classes, in *Advances in Cryptology—EUROCRYPT '99* (Springer Berlin Heidelberg, Berlin, Heidelberg, 1999), pp. 223–238, [10.1007/3-540-48910-X_16](https://doi.org/10.1007/3-540-48910-X_16).
- [25] J.H. Cheon, A. Kim, M. Kim, and Y. Song, Homomorphic encryption for arithmetic of approximate numbers, in *Advances in Cryptology—ASIACRYPT 2017* (Springer International Publishing, New York, 2017), pp. 409–437, [10.1007/978-3-319-70694-8_15](https://doi.org/10.1007/978-3-319-70694-8_15).
- [26] G. Couteau and M. Zarezadeh, Non-interactive secure computation of inner-product from LPN and LWE, *Cryptology ePrint Archive*, Paper 2023/072 (2023), <https://eprint.iacr.org/2023/072>.
- [27] E. Boyle, G. Couteau, N. Gilboa, Y. Ishai, L. Kohl, and P. Scholl, Efficient pseudorandom correlation generators from ring-LPN, in *Advances in Cryptology—CRYPTO 2020* (Springer International Publishing, New York, 2020), pp. 387–416, [10.1007/978-3-030-56880-1_14](https://doi.org/10.1007/978-3-030-56880-1_14).
- [28] D. York, Least-squares fitting of a straight line, *Can. J. Phys.* **44**, 1079 (1966).
- [29] G. Tsoumakas and I. Katakis, Multi-label classification, *Int. J. Data Warehousing Mining* **3**, 1 (2007).
- [30] V. Giovannetti, S. Lloyd, and L. Maccone, Quantum random access memory, *Phys. Rev. Lett.* **100**, 160501 (2008).
- [31] G. Brassard, P. Høyer, and A. Tapp, Quantum counting, in *Automata, Languages and Programming* (Springer Berlin Heidelberg, Berlin, Heidelberg, 1998), pp. 820–831, [10.1007/BFb0055105](https://doi.org/10.1007/BFb0055105).
- [32] M. Fanizza, M. Rosati, M. Skotiniotis, J. Calsamiglia, and V. Giovannetti, Beyond the swap test: Optimal estimation of quantum state overlap, *Phys. Rev. Lett.* **124**, 060503 (2020).
- [33] A. Anshu, Z. Landau, and Y. Liu, Distributed quantum inner product estimation, in *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing* (Association for Computing Machinery, New York, 2022), pp. 44–51, [10.1145/3519935.3519974](https://doi.org/10.1145/3519935.3519974).
- [34] A.S. Holevo, Bounds for the quantity of information transmitted by a quantum communication channel, *Prob. Peredachi Inf.* **9**, 3 (1973).
- [35] T. Hunt, C. Song, R. Shokri, V. Shmatikov, and E. Witchel, Chiron: Privacy-preserving machine learning as a service, *CoRR* abs/1803.05961 (2018), [arXiv:1803.05961](https://arxiv.org/abs/1803.05961).
- [36] E. Hesamifard, H. Takabi, M. Ghasemi, and R.N. Wright, Privacy-preserving machine learning as a service, *Proc. Privacy Enhancing Technol.* **2018**, 123 (2018).
- [37] F. Xu, X. Ma, Q. Zhang, H.-K. Lo, and J.-W. Pan, Secure quantum key distribution with realistic devices, *Rev. Mod. Phys.* **92**, 025002 (2020).
- [38] C.H. Bennett and G. Brassard, Quantum cryptography: Public key distribution and coin tossing, *Theor. Comput. Sci.* **560**, 7 (2014).

- [39] R. Renner, N. Gisin, and B. Kraus, Information-theoretic security proof for quantum-key-distribution protocols, *Phys. Rev. A* **72**, 012332 (2005).
- [40] C. Crepeau, D. Gottesman, and A. Smith, Secure multi-party quantum computing, [arXiv:quant-ph/0206138](https://arxiv.org/abs/quant-ph/0206138).
- [41] B. Knott, S. Venkataraman, A. Y. Hannun, S. Sengupta, M. Ibrahim, and L. van der Maaten, Crypten: Secure multi-party computation meets machine learning, CoRR abs/2109.00984 (2021), [arXiv:2109.00984](https://arxiv.org/abs/2109.00984).
- [42] M. J. Sheller, B. Edwards, G. A. Reina, J. Martin, S. Pati, A. Kotrotsou, M. Milchenko, W. Xu, D. Marcus, R. R. Colen, and S. Bakas, Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data, *Sci. Rep.* **10**, 12598 (2020).
- [43] C. Li, N. Kumar, Z. Song, S. Chakrabarti, and M. Pistoia, Privacy-preserving quantum federated learning via gradient hiding, *Quantum Sci. Technol.* **9**, 035028 (2024).
- [44] W. Li, S. Lu, and D.-L. Deng, Quantum federated learning through blind quantum computing, *Sci. China Phys. Mech. Astron.* **64**, 100312 (2021).
- [45] H. T. Larasati, M. Firdaus, and H. Kim, Quantum federated learning: Remarks and challenges, in *Proceedings of the 2022 IEEE 9th International Conference on Cyber Security and Cloud Computing (CSCloud)/2022 IEEE 8th International Conference on Edge Computing and Scalable Cloud (EdgeCom)* (IEEE, New York, 2022), [10.1109/CSCloud-EdgeCom54986.2022.00010](https://doi.org/10.1109/CSCloud-EdgeCom54986.2022.00010).
- [46] N. Kumar, J. Heredge, C. Li, S. Eloul, S. H. Sureshbabu, and M. Pistoia, Expressive variational quantum circuits provide inherent privacy in federated learning, [arXiv:2309.13002](https://arxiv.org/abs/2309.13002).
- [47] H.-K. Lo, X. Ma, and K. Chen, Decoy state quantum key distribution, *Phys. Rev. Lett.* **94**, 230504 (2005).

End Matter

Appendix: On oracle operators in untrusted client scenario—We describe the detailed implementation of the \hat{O}_f operator in the untrusted client scenario. As mentioned in the main text, the server first sends the state with random basis encoding to the client. Then, as in the original QBC algorithm, the client performs CZ gate between the received qubit o_1 and local qubit o_2 sandwiched by $\hat{U}_{\vec{y}}$ operators. Then, the state received by the server from the quantum channel is $(1/\sqrt{N}) \sum_i^N |i\rangle_n (a_i|0\rangle + b_i(-1)^{y_i}|1\rangle)_{o_1}$, where $a_i(b_i)$ is decided by x_i and the encoding basis R_i thus is known to the server. We next discuss how the server could perform operations to reach the target state $(1/\sqrt{N}) \sum_i^N (-1)^{x_i y_i} |i\rangle$ for running the follow-up QBC algorithm. We consider a second oracle operator held by the server \hat{U}_{X_2} :

$$\begin{aligned} \hat{U}_{X_2} & \frac{1}{\sqrt{N}} \sum_i^N |i\rangle_n (a_i|0\rangle + b_i(-1)^{y_i}|1\rangle)_{o_1} \\ & = \frac{1}{\sqrt{N}} \sum_i^N (-1)^{x_i y_i} |i\rangle_n (a_i|0\rangle + b_i(-1)^{y_i}|1\rangle)_{o_1}. \end{aligned} \quad (\text{A1})$$

This can be achieved via the help of an additional qubit o_a held by the server that encodes the \mathbf{x} information in the normal Z basis (see Ref. [21] for details of circuit implementation).

Note that the server cannot decouple the o_1 qubit with an unknown state, as the honest server only has the information of a_i and b_i but doesn't have the information of \mathbf{y} . In order to reset the state of qubit o_1 , the server could return the state back to client to have the client remove the phase $(-1)^{y_i}$. Before doing so, the server would like to first hide its information by adding a random phase padding by applying \hat{U}_{X_3} which is defined as

$$\begin{aligned} \hat{U}_{X_3} & \frac{1}{\sqrt{N}} \sum_i^N (-1)^{x_i y_i} |i\rangle_n (a_i|0\rangle + b_i(-1)^{y_i}|1\rangle)_{o_1} \\ & = \frac{1}{\sqrt{N}} \sum_i^N (-1)^{x_i y_i + h_i} |i\rangle_n (a_i|0\rangle + b_i(-1)^{y_i}|1\rangle)_{o_1}. \end{aligned} \quad (\text{A2})$$

Here, $h_i \in \{0, 1\}$ is blind to the client and could change in different communication rounds, therefore the client would not be able to extract x_i information. The client performs a controlled-Z gate again between its local qubit o_2 and the received qubit o_1 , after which the phase term $(-1)^{y_i}$ becomes $(-1)^{y_i + y_i} = 1$. Then, the server receives the state $(1/\sqrt{N}) \sum_i^N (-1)^{x_i y_i + h_i} |i\rangle_n (a_i|0\rangle + b_i|1\rangle)_{o_1}$ from client and performs oracle \hat{U}_{X_4} :

$$\begin{aligned} \hat{U}_{X_4} & \frac{1}{\sqrt{N}} \sum_i^N (-1)^{x_i y_i + h_i} |i\rangle_n (a_i|0\rangle + b_i|1\rangle)_{o_1} \\ & = \frac{1}{\sqrt{N}} \sum_i^N (-1)^{x_i y_i} |i\rangle_n |0\rangle_{o_1}. \end{aligned} \quad (\text{A3})$$

It can be easily seen that to implement \hat{U}_{X_4} , the server could simply perform \hat{U}_{X_3} again to remove the added random phase term $(-1)^{h_i}$ and then reset the qubit o_1 to $|0\rangle_{o_1}$ as the server knows the all coefficients a_i and b_i .

We remark that the random numbers R_i and h_i can change in different Grover iterations. That is, the client will not get useful information by performing measurements on each iteration and using the joint results from a sequence of measurements to infer \mathbf{x} . The privacy of \mathbf{x} is guaranteed by the fact that measuring a single copy in a given basis cannot reveal both the basis information R_i and the data information x_i . The probability that the client gets \mathbf{x}' that is d_0 close to the true \mathbf{x} would simply be the same as a random guess.