

# Blind Quantum Machine Learning with Quantum Bipartite Correlator

Changhao Li,<sup>1,2,3,\*</sup> Boning Li,<sup>2,4</sup> Omar Amer,<sup>1</sup> Ruslan Shaydulin,<sup>1</sup> Shouvanik Chakrabarti,<sup>1</sup>  
Guoqing Wang,<sup>2,3</sup> Haowei Xu,<sup>3</sup> Hao Tang,<sup>5</sup> Isidor Schoch,<sup>3</sup> Niraj Kumar,<sup>1</sup>  
Charles Lim,<sup>1</sup> Ju Li,<sup>3,5,†</sup> Paola Cappellaro,<sup>2,3,4,‡</sup> and Marco Pistoia<sup>1,§</sup>

<sup>1</sup>*Global Technology Applied Research, JPMorgan Chase, New York, NY 10017 USA*

<sup>2</sup>*Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*

<sup>3</sup>*Department of Nuclear Science and Engineering,  
Massachusetts Institute of Technology, Cambridge, MA 02139, USA*

<sup>4</sup>*Department of Physics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*

<sup>5</sup>*Department of Materials Science and Engineering,  
Massachusetts Institute of Technology, Cambridge, MA 02139, USA*

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 work, 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.

## I. 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 [4–10] 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 [11–13] 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 (UBQC) [11]. 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 work, we introduce novel protocols for blind distributed quantum machine learning based on quantum bipartite correlator algorithm that can perform inner product estimation tasks. Our protocols are communication-efficient compared with state-of-the-art classical and quantum blind distributed machine learning algorithms. Particularly, for the task of distributed inner product estimation, a core subroutine in machine learning applications, the protocols involve a communication complexity  $O(\log N/\epsilon)$  with  $N$  and  $\epsilon$  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 work paves the way for performing quantum machine learning with an untrusted device, while maintaining the privacy and

\* [changhao.li@jpmchase.com](mailto:changhao.li@jpmchase.com)

† [liju@mit.edu](mailto:liju@mit.edu)

‡ [pcappell@mit.edu](mailto:pcappell@mit.edu)

§ [marco.pistoia@jpmchase.com](mailto:marco.pistoia@jpmchase.com)

keeping the resource overhead low.

## II. 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 work, we assume that Alice holds the data  $\mathbf{X}$  and Bob holds  $\mathbf{y}$ . For example, in supervised learning,  $\mathbf{X}$  and  $\mathbf{y}$  could be feature data and labels, respectively [14], while in unsupervised learning, both  $\mathbf{X}$  and  $\mathbf{y}$  can be feature data with the objective to cluster them based on distance estimation [15].

We consider the task of blind quantum machine learning, such as linear regression or classification [16–19]. In machine learning, evaluating the inner product between two vectors is an important algorithmic building block. 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. Meanwhile, as we consider distributed quantum computation, the data  $\mathbf{X}$  and  $\mathbf{y}$  are only held locally by the server and client, respectively.

In classical settings, the goal of achieving distributed machine learning with privacy can be approached using various techniques, such as homomorphic encryption [20, 21], which allows computation over encrypted data. Specifically, for distributed bipartite correlation estimation, many methods could be employed, including linearly homomorphic encryption [22, 23], non-interactive inner product protocols [24] and oblivious-transfer-based secure computation [25]. 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  $\tilde{O}(N)$  would be a minimum requisite [24]. As a result, their practical applications become limited, especially when dealing with large data sizes.

## III. QUANTUM BIPARTITE CORRELATOR ALGORITHM AND ITS PRIVACY

In this section, we briefly introduce the quantum bipartite correlator (QBC) algorithm that can estimate the correlation between two bitstrings held by remote parties [8]. The algorithm can be easily generalized to perform other computation tasks, such as the Hamming dis-

tance estimation. We remark that estimating bipartite correlation or Hamming distance serves as the building block of a general class of machine learning problems, including least-square fitting and classification of discrete labels [26, 27].

Without loss of generality, we consider binary floating point numbers. We take the feature dimension  $M$  to be one 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  $\overline{xy} = \frac{1}{N} \sum_{i=1}^N x_i y_i$  within a standard deviation error  $\epsilon$ . 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  $\hat{U}_{\hat{x}}: |i\rangle_n |0\rangle \mapsto |i\rangle_n |x_i\rangle$  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  $\hat{U}_{\hat{y}}$  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 [28].

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, 29], as shown in the top of Fig. 1. The quantum counting algorithm consists of a Grover operator  $\hat{G}_{\hat{x}, \hat{y}} \equiv \hat{H}^{\otimes n} (2|0\rangle_n \langle 0|_n - \hat{I}) \hat{H}^{\otimes n} \hat{U}_{xy}$ , 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<sup>†</sup>) 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  $\theta = 2 \arcsin \sqrt{\overline{xy}}$ , with equivalent standard deviation:  $\Delta \hat{\theta} = 2^{-t+1}$  [8].

During the phase oracle  $\hat{G}_{\hat{x}, \hat{y}}$ , 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.,  $\hat{U}_{\hat{x}}$  and  $\hat{U}_{\hat{y}}$ .

Specifically, Alice encodes her local information  $\mathbf{X}$  into qubit  $o_1$  via  $\hat{U}_{\hat{x}}$  operator and sends the  $(n+1)$ -qubit state  $\frac{1}{\sqrt{N}} \sum_i |i\rangle_n |x_i\rangle_{o_1}$  to Bob via a quantum channel. After Bob applies his oracle and generates the state  $\frac{1}{\sqrt{N}} \sum_i |i\rangle_n |x_i\rangle_{o_1} |y_i\rangle_{o_2}$ , 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  $\frac{1}{\sqrt{N}} \sum_i (-1)^{x_i y_i} |i\rangle_n |x_i\rangle_{o_1} |y_i\rangle_{o_2}$ . The following local oracles would then yield the desired state  $\frac{1}{\sqrt{N}} \sum_i (-1)^{x_i y_i} |i\rangle_n$  on which Alice will apply the quantum counting algorithm to estimate  $\overline{xy} = \frac{1}{N} \sum_{i=1}^N x_i y_i$  with bounded error  $\epsilon$ . We note that the CZ gate might

be replaced with a different set of gates to estimate other types of correlations between  $\mathbf{X}$  and  $\mathbf{y}$ . For example, to calculate their Hamming distance, one can implement the XOR gate  $x_i \oplus y_i$  by replacing the CZ gate with a Z gate on  $o_2$  sandwiched by two CNOT gates between  $o_1$  and  $o_2$  [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. We remark that the above communication complexity is advantageous compared with the SWAP-test-based algorithm that has a scaling of  $O(\log_2(N)/\epsilon^2)$  [30] or LOCC-based algorithms with a scaling of  $O(\log_2(N) \max\{1/\epsilon^2, \sqrt{N}/\epsilon\})$  [31]. 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:

$$C_{\text{comp}} = 4(2^t - 1) = O\left(\frac{1}{\epsilon}\right). \quad (3)$$

We next consider the privacy of data in the QBC algorithm discussed above. From now on, we consider Alice as a server and Bob as a client. We first focus on the privacy of the client's information  $\mathbf{y}$  to a semi-honest adversary. In this type of adversary, the honest-but-curious server follows the protocol and does not do any malicious behavior, but it tries to violate the privacy of the client's input by scrutinizing the messages transmitted in the protocol. That is, the server tries to infer  $\mathbf{y}$  from the estimated  $\frac{1}{N} \sum_i^N x_i y_i$ .

In the trivial case when  $x_i = 0, \forall i \leq N$ , we have  $\overline{xy} = 0$  no matter what  $\mathbf{y}$  is and the protocol has the best privacy. While in the worst case where the  $x_i = 1, \forall i \leq N$  and  $\overline{xy} = 1$ , the server could infer that  $y_i = 1, \forall i \leq N$ . In general, for  $\mathbf{X}$  with Hamming weight  $d_x$ , the probability that the server gets the exact  $\mathbf{y}$  (that is, the Hamming distance between extracted and exact bitstring is  $d_0 = 0$ ) is given by

$$\Pr(d_x) = \frac{1}{2^{N-d_x}} \frac{\prod_{i=1}^{d_x} i}{\prod_{i=1}^{Nxy} i \prod_{i=1}^{d_x-Nxy} i}, \quad (4)$$

where the factor  $\frac{1}{2^{N-d_x}}$  comes from server having random guess on the indices  $j$  that satisfies  $x_j = 0$ . For a honest server in the original QBC protocol, however, the  $\mathbf{y}$  information is always hidden from the server and is private.

In addition to the semi-honest adversary scenario discussed above, we note that in the original QBC algorithm, the preservation of privacy is not assured when

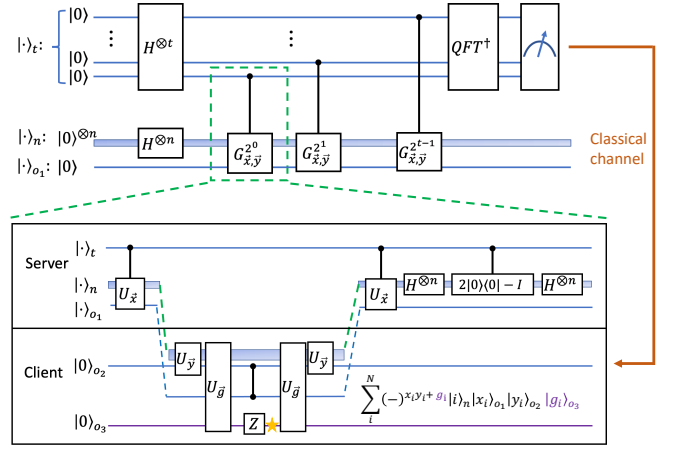


FIG. 1. Diagram for blind QBC with untrusted server. The upper diagram shows the quantum counting algorithm consisting Grover phase oracles  $\hat{G}_{x,y}$  and inverse QFT, while the lower box panel shows the realization details of each phase oracle. Compared to the original QBC algorithm, we introduce an ancillary qubit  $o_3$  on 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  $\frac{1}{N} \sum_i^N (x_i y_i + g_i)$  to the client via a classical communication channel. We omit the  $1/\sqrt{N}$  normalization factor for index qubit states  $\sum_i^N |i\rangle$  in the figures hereafter for simplicity.

we consider a malicious server Alice. The server has the capability to acquire, to a certain extent, Bob's strings  $\mathbf{y}$  by deviating from the expected quantum operations. We next discuss the designed blind QBC protocol with such an untrusted server.

#### IV. BLIND QBC WITH UNTRUSTED SERVER

A malicious server can get the client's information by deviating from the established QBC protocol. 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  $\frac{1}{\sqrt{N}} \sum_i^N (-1)^{x_i y_i} |i\rangle_n |x_i\rangle_{o_1}$  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  $\mathbf{X}$ . In principle, for each communication round, the server can acquire one bit of information of client's data  $\mathbf{y}$ . Then with the  $2^t - 1 = O(\frac{1}{\epsilon})$  Grover iterations, the server could get  $O(\frac{1}{\epsilon})$  bits of information in  $\mathbf{y}$ . Such an attack strategy might be implemented by preparing the  $o_1$  qubit in  $|+\rangle$  state and sending  $\frac{1}{\sqrt{N}} \sum_i^N |i\rangle_n |+\rangle_{o_1}$  to the client (Appendix A). Subsequent to the reception of the quantum

state from the client, the server undertakes an  $X$  basis measurement on qubit  $o_1$ . The server could perform the sampling procedure encompassing the bitstrings of the index qubits during the  $O(\frac{1}{\epsilon})$  communication rounds.

We note that the server could not manipulate the index qubit states  $\frac{1}{\sqrt{N}} \sum_i^N |i\rangle$  to amplify the amplitude of a specific bitstring of interest, as the client is capable of verifying the received quantum state of index qubits by performing  $X$  basis measurements to check whether they have the same amplitude. On the other hand, it is possible to employ a redundant encoding strategy to further decrease the probability that the server attains a specific  $y_i$  corresponding to an intended index. However, this comes at the expense of increased communication complexity, as detailed in Appendix. B.

To counteract the aforementioned attack strategy, we need to devise a protocol enabling the server to execute machine learning tasks while remaining unaware of the exact label information  $\mathbf{y}$ , even when the malicious server does not follow the designed protocol. In this case, we consider an honest client, who is not interested in learning  $\mathbf{X}$ . This assumption might be removed if we consider further encoding privacy in  $\mathbf{X}$  when sending information to the client. To implement remote blind bipartite correlation estimation, a desired protocol should have 1) less overhead in quantum communication, 2) less requirements in the computational power of client, 3) a certified estimation result with error  $\epsilon$ .

We thus consider the revised QBC algorithm below (Fig. 1). Inspired by quantum one-time pad [11], the 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

$$\hat{U}_{xyg} |i\rangle_n |000\rangle_{o_1 o_2 o_3} = (-1)^{x_i y_i + g_i} |i\rangle_n |000\rangle_{o_1 o_2 o_3}. \quad (5)$$

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 server to create the state  $\frac{1}{\sqrt{N}} \sum_i^N |i\rangle_n |x_i\rangle_{o_1} |y_i\rangle_{o_2} |g_i\rangle_{o_3}$ , 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 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  $\frac{1}{N} \sum_i^N (x_i y_i + g_i) = \frac{1}{N} \sum_i^N x_i y_i + g_i$

mod 2) with error bound  $\epsilon$ . Finally, after the measurement, the server sends the estimated result back to the client via a classical channel, from which the client can extract  $\frac{1}{N} \sum_i^N x_i y_i$  using his local information of  $\frac{1}{N} \sum_i^N g_i$ . Alternatively, depending on the specific use cases, the client could directly share  $\frac{1}{N} \sum_i^N g_i$  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  $g_i$  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 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 bitstring  $\mathbf{y}'$  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

$$\Pr(d(\mathbf{y}, \mathbf{y}') = d_0) = \frac{C(d_y, d_0) C(N - d_y, \min(2^t - 1, d_y) - d_0)}{C(N, \min(2^t - 1, d_y))} \quad (6)$$

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,  $2^t - 1 = O(\frac{1}{\epsilon})$ . A less tight error bound  $\epsilon$  will reduce the number of communication rounds between server and client thus increasing the privacy of client's data.

We remark that the quantum communication complexity of the aforementioned algorithm for blind server is  $\mathcal{C}_{\text{comm}}^{b_s} = O(\frac{\log_2(N)}{\epsilon})$ , 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  $O(\frac{1}{\epsilon})$  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.

## V. BLIND QBC WITH UNTRUSTED CLIENT

We now discuss the scenario where the server would like to estimate  $\frac{1}{N} \sum_i^N x_i y_i$  while keeping  $\mathbf{X}$  hidden from the client at all times during the process. In practical applications such as model-as-a-service platforms [32, 33], 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 a 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

$$\hat{O}_f \frac{1}{\sqrt{N}} \sum_i^N |i\rangle_n |00\rangle_{o_1 o_2} = \frac{1}{\sqrt{N}} \sum_i^N (-1)^{x_i y_i} |i\rangle_n |00\rangle_{o_1 o_2}. \quad (7)$$

Inspired by quantum key distribution protocols [34] such as BB84 [35], we consider a modified local oracle operator  $\hat{U}_{X_1}$  held by the server, where the data information  $\mathbf{X}$  is encoded in 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  $|+\rangle(|-\rangle) = \frac{1}{2}(|0\rangle \pm |1\rangle)$  are the eigenstates of Pauli X operator. This oracle  $\hat{U}_{X_1}$  can be implemented with the original oracle  $\hat{U}_{\vec{x}}$  with Hadamard gates on  $o_1$  conditioned on index  $|i\rangle_n$ .

Then, the state received by the client at each time reads as  $\frac{1}{\sqrt{N}} \sum_i^N |i\rangle_n |X_i\rangle_{o_1}$  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  $|i\rangle_n$  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  $\frac{1}{\sqrt{N}} \sum_i^N |i\rangle_n |X_i\rangle_{o_1}$  state. Note that the server could pick different random numbers  $R_i$  at different communication rounds when executing the QBC algorithm.

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  $\frac{1}{\sqrt{N}} \sum_i^N |i\rangle_n (a_i |0\rangle + b_i (-1)^{y_i} |1\rangle)_{o_1}$  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  $\frac{1}{\sqrt{N}} \sum_i^N (-1)^{x_i y_i} |i\rangle_n$  for running the follow-up QBC algorithm. We consider a second oracle operator held by the server  $\hat{U}_{X_2}$ :

$$\begin{aligned} \hat{U}_{X_2} \frac{1}{\sqrt{N}} \sum_i^N |i\rangle_n (a_i |0\rangle + b_i (-1)^{y_i} |1\rangle)_{o_1} = \\ \frac{1}{\sqrt{N}} \sum_i^N (-1)^{x_i y_i} |i\rangle_n (a_i |0\rangle + b_i (-1)^{y_i} |1\rangle)_{o_1}. \end{aligned} \quad (8)$$

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 Appendix C 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

$$\begin{aligned} \hat{U}_{X_3} \frac{1}{\sqrt{N}} \sum_i^N (-1)^{x_i y_i} |i\rangle_n (a_i |0\rangle + b_i (-1)^{y_i} |1\rangle)_{o_1} = \\ \frac{1}{\sqrt{N}} \sum_i^N (-1)^{x_i y_i + h_i} |i\rangle_n (a_i |0\rangle + b_i (-1)^{y_i} |1\rangle)_{o_1}. \end{aligned} \quad (9)$$

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  $\frac{1}{\sqrt{N}} \sum_i^N (-1)^{x_i y_i + h_i} |i\rangle_n (a_i |0\rangle + b_i |1\rangle)_{o_1}$  from client and performs oracle  $\hat{U}_{X_4}$ :

$$\begin{aligned} \hat{U}_{X_4} \frac{1}{\sqrt{N}} \sum_i^N (-1)^{x_i y_i + h_i} |i\rangle_n (a_i |0\rangle + b_i |1\rangle)_{o_1} = \\ \frac{1}{\sqrt{N}} \sum_i^N (-1)^{x_i y_i} |i\rangle_n |0\rangle_{o_1}. \end{aligned} \quad (10)$$

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

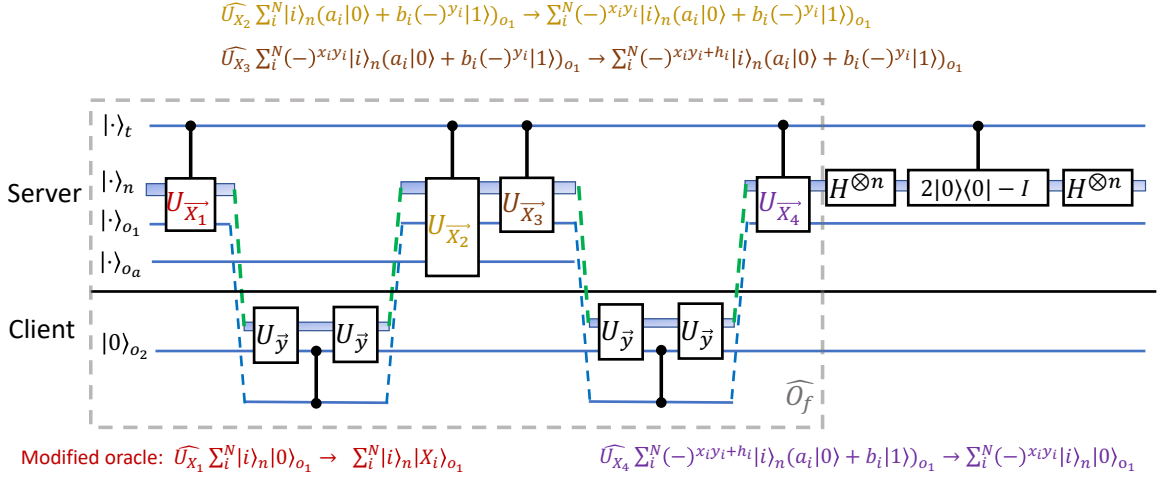


FIG. 2. Grover operator  $\hat{H}^{\otimes n} (2|0\rangle_n \langle 0| - \hat{I}) \hat{H}^{\otimes n} \hat{O}_f$  for blind quantum bipartite correlator protocol to hide server data  $\mathbf{X}$  from client. The operator starts with an oracle held by server (Alice) that encodes  $\mathbf{X}$  with 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 return 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  $\frac{1}{\sqrt{N}} \sum_i^N (-1)^{x_i y_i} |i\rangle_n$  by decoupling  $o_1$  qubit with index qubits ( $\hat{U}_{X_4}$ ).

the client gets  $\mathbf{X}'$  that is  $d_0$ -close to the true  $\mathbf{X}$  would simply be the same as a random guess.

To this end, we have described a phase encoding oracle  $\hat{O}_f$  that lets the server acquire the state  $\frac{1}{\sqrt{N}} \sum_i^N (-1)^{x_i y_i} |i\rangle_n$  for subsequent operations without leaking the information of data  $\mathbf{X}$  to an untrusted client. 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 [35, 36]. 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  $O(\frac{1}{\epsilon})$  as Eq. 3. The total communication cost of this blind client scheme is given by

$$\mathcal{C}_{\text{comm}}^{bc} = 4(n+1)(2^t - 1) = O\left(\frac{\log_2(N)}{\epsilon}\right), \quad (11)$$

which has the same complexity scaling as the original QBC algorithm. We summarize the proposed algorithms here and above in Table. I.

## VI. GENERALIZATION INTO MULTI-PARTY SETTINGS

The algorithms discussed above can be generalized into multi-party settings and find applications in secure multi-party computation and machine learning [37, 38], where parties collaboratively perform computations on their combined data sets without revealing the data they possess to untrusted parties. For example, to perform model

aggregation, an untrusted central server would like to perform linear regression or classification using its local data as well as labels that are distributed among multiple clients. Then, the protocol in Sec. IV can be applied in which the server can interact with each client to extract model parameters individually.

Here we provide an example of multi-party protocols. We consider a system consisting of a central server and  $m$  clients, where the server is untrusted by the clients. The task is to have the server evaluate  $f_m = \frac{1}{N} \sum_i^N (\sum_j^m x_i y_i^{(j)} \bmod 2)$  without leaking individual information of clients. Similar to the phase pad technique introduced in Sec. IV, one can protect each individual client's information by adding additional terms in the phase when running the QBC algorithm. Specifically, we consider a cascaded protocol where each client encodes its information into the phase of index qubits and passes the state into the next client. In each communication round, the  $k$ -th client would receive the state  $\frac{1}{\sqrt{N}} \sum_i^N (-1)^{x_i y_i^{(1)} + x_i y_i^{(2)} + \dots + x_i y_i^{(k-1)}} |i\rangle_n |x_i\rangle$  from the  $(k-1)$ -th client. Then, by applying CZ gate between  $o_1$  and its local qubit, the  $j$ -th client sends the state  $\frac{1}{\sqrt{N}} \sum_i^N (-1)^{x_i y_i^{(1)} + x_i y_i^{(2)} + \dots + x_i y_i^{(k-1)} + x_i y_i^{(k)}} |i\rangle_n |x_i\rangle$  to the next client. The final  $m$ -th client will pass the state  $\frac{1}{\sqrt{N}} \sum_i^N (-1)^{\sum_j^m x_i y_i^{(j)}} |i\rangle_n |x_i\rangle$  to the server which can then perform the remaining part of the original QBC algorithm to extract the desired  $f_m$ .

We note that a malicious server could only get the  $\sum_j^m y_i^{(j)}$  and the individual  $y_i^{(j)}$  information is not leaked, as the phase added by each client servers as a random pad of other clients. For the same reason, the  $j$ -th ( $j \geq 3$ ) client cannot get previous clients' informa-

TABLE I. Privacy and communication complexity of proposed distributed inner product estimation algorithms.

Adversaries	Protocol	Privacy mechanism	Privacy	Communication complexity
Honest-but-curious server	original QBC algorithm [8]	-	worst scenario in Eq. 4	$O((\log_2 N)/\epsilon)$
Malicious server	blind QBC for untrusted server (Sec. IV)	random phase padding	worst scenario in Eq. 6	$O((\log_2 N)/\epsilon)$
Malicious client	blind QBC for untrusted client (Sec. V)	random basis encoding, random phase padding	information-theoretic secure	$O((\log_2 N)/\epsilon)$

tion as it can only extract  $\sum_{k=1}^{j-1} y_i^{(k)}$ . The first client ( $j = 1$ ) can further add a random pad  $g_i^{(1)}$  to protect its information against the second client ( $j = 2$ ). The protocol here is similar to incremental learning [39], where the model aggregation is performed while preserving privacy. We remark that the total communication cost scales as  $O\left(\frac{m \log_2(N)}{\epsilon}\right)$  and the privacy mechanism does not introduce additional communication cost. To this end, our work paves the way for communication-efficient private machine learning for multi-party system, such as quantum federated learning [40–42].

## VII. DISCUSSION AND CONCLUSION

As mentioned above, the proposed blind distributed inner product estimation protocols can be applied in distributed machine learning where a central task is to evaluate correlations between remote matrices or vectors. Here we give an example of such applications. In linear regression problems, one is interested in finding the coefficient vector  $\boldsymbol{\lambda}$  with standard error  $\epsilon$  that satisfies  $\mathbf{X}_{N \times M} \boldsymbol{\lambda}_{M \times 1} = \mathbf{y}_{N \times 1}$ , where the  $N$ -by- $M$  matrix  $\mathbf{X}$  and  $N$ -by-1 vector  $\mathbf{y}$  are separately held by two remote parties, a server and a client, respectively. We consider the case where the server would like to estimate  $\boldsymbol{\lambda}$  without letting the client extract its local information  $\mathbf{X}_{N \times M}$ . The  $l$ -th component of  $\boldsymbol{\lambda}$  reads  $\lambda_l = \sum_{i=1}^N X_{li}^\dagger y_i$ , where  $l$  and  $i$  labels the index of the element in the matrix or vector. The problem can be reduced to estimate product of distributed numbers  $a_{li} = X_{li}^\dagger$  and  $b_i = y_i$ . They can be expanded as binary floating point numbers using, for example,  $a_{li} = \sum_{k=0}^{\infty} 2^{u-k} x_{li}^{(k)}$  and  $b_i = \sum_{k=0}^{\infty} 2^{v-k} y_i^{(k)}$ , for which  $u$  and  $v$  denote the highest digits of  $a$  and  $b$ , respectively [8, 43]. Then, the target coefficient  $\lambda_l$  can be written as  $\lambda_l = \sum_{i=1}^N a_{li} b_i = \sum_{r=0}^{\infty} 2^{u+v-r} \sum_{k=0}^r \sum_{i=1}^N x_{li}^{(k)} y_i^{(r-k)}$ , where the blind QBC algorithm introduced in Sec. V can be directly applied. In this case, the untrusted client can neither directly extract the information of  $\mathbf{X}_{N \times M}$  during the blind QBC communication, nor indirectly have an estimation on  $\mathbf{X}_{N \times M}$  from the knowledge of coefficient  $\boldsymbol{\lambda}_{M \times 1}$ . To this end, our proposed algorithms exhibit direct applicability within the domain of distributed blind machine learning tasks, particularly in scenarios involv-

ing matrix or vector multiplication operations.

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  $O\left(\frac{\log N}{\epsilon}\right)$  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.

In summary, this study introduces novel blind quantum machine learning protocols that utilize a quantum bipartite correlator estimation algorithm for distributed parties. By addressing the potential threat of malicious parties attempting to extract information from others, we propose two distinct settings that ensure privacy preservation for each party in the QBC algorithm. Leveraging the advantageous properties of quantum phases and the flexibility of encoding data in various bases, our protocols can effectively safeguard information. The developed blind QML algorithm offers notable advantages, including low communication and computational complexity. This work contributes to the advancement of secure and efficient QML protocols, thus presenting an efficient pathway for distributed quantum computing.

## ACKNOWLEDGMENTS

JL acknowledges support by DTRA (Award No. HDTRA1-20-2-0002) Interaction of Ionizing Radiation with Matter (IIRM) University Research Alliance (URA).

## DISCLAIMER

This paper was prepared for informational purposes with contributions from the Global Technology Applied Research center of JPMorgan Chase & Co. This paper 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 paper, 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.



## Appendix A: Extraction of $y_i$ information in QBC by malicious server

We discuss a feasible attack protocol for a malicious server to extract information of  $\mathbf{y}$  with the received state  $\sum_i^N (-1)^{x_i y_i} |i\rangle_n |x_i\rangle$  in the original QBC algorithm. In this protocol, the server prepares the  $o_1$  qubit simple in the  $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$  state. The quantum state sent to client would then be

$$\frac{1}{\sqrt{2}} \left( \frac{1}{\sqrt{N}} \sum_i^N |i\rangle_n |0\rangle_{o_1} + \frac{1}{\sqrt{N}} \sum_i^N |i\rangle_n |1\rangle_{o_1} \right) \quad (\text{A1})$$

The honest client then encodes  $y_i$  information in the phase with his own local oracle, leading to state

$$\begin{aligned} & \frac{1}{\sqrt{2N}} \left( \sum_i^N |i\rangle_n |0\rangle_{o_1} + |1\rangle_{o_1} \sum_i^N (-1)^{y_i} |i\rangle_n |1\rangle_{o_1} \right) \\ &= \frac{1}{\sqrt{2N}} \sum_i^N |i\rangle_n (|0\rangle_{o_1} + (-1)^{y_i} |1\rangle_{o_1}) \end{aligned} \quad (\text{A2})$$

that is sent back to server.

Then, it's clear to see that to extract client's information, the server could perform measurement on qubit  $o_1$  in the X basis and extract the  $y_j$  information depending on the measured index qubit bitstring  $j$ . In this case, by performing sampling on the  $N$  index qubit states during the  $2^t - 1 = O(1/\epsilon)$  communication rounds, the malicious server could get  $O(1/\epsilon)$  information of  $\mathbf{y}$ . Indeed, given the state Eq. A2 received by the server, the upper bound of information that the server could get at each round by performing measurement on index qubits and qubit  $o_1$  is determined by the Holevo's bound [44]:

$$H(C : S) \leq S(\rho) - \frac{1}{N} \sum_i^N S(\rho(i)) = \log 2N, \quad (\text{A3})$$

where  $S(\rho)$  denotes the von Neumann entropy for density matrix  $\rho$  that corresponds to Eq. A2, and  $\rho_i = |a_i\rangle \langle i| \langle a_i|$  ( $a_i = +, -$ ) forms the POVM set that server performs.

One might argue that the server could amplify the probability of sampling a particular index qubit bitstring  $j$  by reducing the amplitude of other index qubit bitstrings. That is, the quantum state sent to client could be

$$\frac{1}{\sqrt{2}} \left( \sum_i^N A_i |i\rangle_n |0\rangle_{o_1} + \sum_i^N A_i |i\rangle_n |1\rangle_{o_1} \right) \quad (\text{A4})$$

where  $|A_{i=j}|^2 \gg |A_{i \neq j}|^2$  and  $\sum_i^N |A_i|^2 = 1$ . However, the client can add an additional verification on the  $\lceil \log_2(N) \rceil$  index qubits upon receiving them by performing measurements on X basis. This should yield +1 for all index qubits, as the state  $\frac{1}{\sqrt{N}} \sum_i^N |i\rangle$  can be rewritten as  $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)^{\otimes \lceil \log_2(N) \rceil}$ . While for the manipulated

state outlined in Eq. A4, there exists a nonzero probability of producing a measurement outcome of  $-1$  for at least a portion of the measurements.

## Appendix B: Redundant encoding against malicious server

We describe a redundant encoding approach aimed at reducing the probability that a malicious server acquiring a specific  $y_i$  information with  $i$  being the pertinent index of interest using the attack strategy in Appendix. A.

Given that the server is restricted to preparing the index qubits in a manner where each index bitstring holds identical probability, after receiving the state back from client, the probability that server samples a specific index bitstring  $|i\rangle_n$  is simply  $\frac{1}{N}$ . That is, in each iteration of communication during the execution of QBC algorithm, the server is constrained to attain a specific  $y_i$  corresponding to the intended index with a probability of  $\frac{1}{N}$ ; and for  $1/\epsilon$  iterations needed for QBC algorithm, this will cause a total amount of information being extracted to be  $\frac{1}{N\epsilon}$ . Following this, we can consider a protocol where both the client and server encode their single bit local information  $y_i$  and  $x_i$  into bitstrings  $[y'_{i,1}, y'_{i,2}, \dots, y'_{i,M}]'$  and  $[x'_{i,1}, x'_{i,2}, \dots, x'_{i,M}]$  with size  $M$ , where  $M > 1$ . The total amount of bits increases from  $N$  to  $MN$ . The encoding rule is shown as follows:

$$\mathbf{x}'_{i,j} = x_i; \quad i = 1, 2, \dots, N; j = 1, 2, \dots, M; \quad (\text{B1})$$

which is a simply copy the bit  $x_i$  for  $M$  times. As for  $\mathbf{y}'$ , the client can hide the information  $y_i$  randomly in one of the  $M$  digits and let the other  $M - 1$  digits to be all zero or one. That is, client chooses either

$$\begin{aligned} \mathbf{y}'_{i,j} &= \delta_{j,J_i} \cdot y_i; \\ i &= 1, 2, \dots, N; j = 1, 2, \dots, M, J_i \in \{1, 2, \dots, M\}. \end{aligned} \quad (\text{B2})$$

or

$$\begin{aligned} \mathbf{y}'_{i,j} &= (1 - \delta_{j,J_i}) \cdot y_i; \\ i &= 1, 2, \dots, N; j = 1, 2, \dots, M, J_i \in \{1, 2, \dots, M\}. \end{aligned} \quad (\text{B3})$$

where  $J_i$  is an random number and  $\delta_{j,J_i}$  is the Kronecker symbol. In these cases, the server would get  $\frac{1}{NM} \sum_i^N x_i y_i$  or  $\frac{1}{NM} \sum_i^N x_i y_i + \frac{M-1}{NM} \sum_i^N x_i$  by executing the QBC algorithm, depending on whether the client chooses encoding method Eq. B2 or Eq. B3. Afterwards, the client can send an one-bit message via classical channel to the server and let server knows which one was used.

We remark that at each communication round, the probability that the server samples a specific bit reduces from  $\frac{1}{N}$  to  $\frac{1}{NM}$ . Even though that  $M$ -times more communication round will be needed to achieve the same error bound  $\epsilon$  as in the original QBC case, the server would not know which digit encodes the correct  $y_i$  information as here  $J_i$ s are random numbers. Therefore, using the

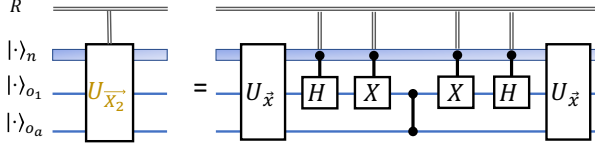


FIG. 3. Circuit diagram for implementing  $U_{\bar{X}_2}$  with the help of an ancilla qubit  $o_a$ . The first control line shows the classical control decided by the random number  $R_i, i = 1, \dots, N$ .

attack strategy detailed in the Appendix. A, the probability that the server successfully gets a specific bit  $y_i$  would be  $\frac{1}{NM} \times \frac{M}{\epsilon} \times \frac{1}{M} = \frac{1}{NM\epsilon}$ , where the second term  $\frac{M}{\epsilon}$  is the total number of communication rounds and the third term is  $\frac{1}{M}$  is due to the randomness in  $J_i$ . It's clear to see that a larger value of  $M$  corresponds to a decreased probability for the server to successfully extract valuable information from the client through the attack strategy. The flexibility that the client can independently choose encoding method also protects the majority information of  $\mathbf{y}$ , i.e., the client may choose Eq. B2 to encode data if the majority of  $\mathbf{y}$  is 1 to decrease the probability that 1s are being detected. Nevertheless, the trade-off for employing this redundant encoding approach manifests as an augmented quantum communication complexity, which reads  $O(\frac{\log(NM)}{\epsilon})$ .

### Appendix C: Construction of oracle operator $\hat{U}_{X_2}$ for blind QBC with untrusted client

In this section, we give the details for the implementation of  $\hat{U}_{X_2}$  operator mentioned in Sec. V. Recall that  $\hat{U}_{X_2}$  is applied to extract the phase term  $(-1)^{x_i y_i}$ , as shown in Eq. 8. The quantum state before applying  $\hat{U}_{X_2}$

is given by

$$\frac{1}{\sqrt{N}} \sum_i^N |i\rangle_n (a_i |0\rangle + b_i (-1)^{y_i} |1\rangle)_{o_1}, \quad (C1)$$

where  $a_i$  and  $b_i$  depends on  $x_i$  and the encoding basis  $R_i$ . For the data  $j$  encoded in Z basis, i.e.,  $R_j = 0$ , one has  $a_j b_j = 0$  and the phase  $(-1)^{x_i y_i}$  naturally shows up as in the original QBC algorithm. While for the data encoded in X basis, i.e.,  $R_j = 1$ , we target to extract the  $(-1)^{x_i y_i}$  term by transforming it back to Z basis.

For this purpose, we consider the following protocol. Firstly,  $\hat{U}_{\bar{x}}$  oracle is called to generate the state  $\frac{1}{\sqrt{N}} \sum_i^N |i\rangle_n (a_i |0\rangle + b_i (-1)^{y_i} |1\rangle)_{o_1} |x_i\rangle_{o_a}$  where the additional qubit  $o_a$  encodes  $x_i$  in Z basis. Secondly, a Hadamard gate is applied on qubit  $o_1$  conditioned on index qubit state  $|i\rangle_n = |j\rangle_n$  that satisfies  $R_j = 1$  (i.e., encoding in X basis). This will transform the X basis encoding to Z basis. Then, a NOT gate on qubit  $o_1$  conditioned on those index qubit states followed by a controlled-Z gate between  $o_1$  and  $o_a$  is applied. With the above steps, a phase  $(-1)$  is generated unless  $x_i = y_i = 1$ . The state now reads:

$$\frac{1}{\sqrt{N}} \sum_i^N (-1)^{x_i y_i} |i\rangle_n |m_i\rangle_{o_1} |x_i\rangle_{o_a}. \quad (C2)$$

Here  $m_i = x_i$  when  $R_i = 0$  or when  $R_i = 1$  and  $y_i = 0$ .

Now that as the phase term  $(-1)^{x_i y_i}$  has already been extracted, we transform the  $o_1$  qubit state  $|m_i\rangle_{o_1}$  back to the initial  $(a_i |0\rangle + b_i (-1)^{y_i} |1\rangle)_{o_1}$  by applying the controlled Hadamard and NOT gate again, and then decouple the ancillary qubit by calling the  $\hat{U}_{\bar{x}}$ . The resulting quantum state reads  $\frac{1}{\sqrt{N}} \sum_i^N (-1)^{x_i y_i} |i\rangle_n (a_i |0\rangle + b_i (-1)^{y_i} |1\rangle)_{o_1}$ .

- 
- [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 Optics and Photonics* **12**, 1012 (2020).
- [2] D. Herman, C. Googin, X. Liu, Y. Sun, A. Galda, I. Safro, M. Pistoia, and Y. Alexeev, *Nature Reviews Physics* **5**, 450 (2023).
- [3] Y. Cao, J. Romero, and A. Aspuru-Guzik, *IBM Journal of Research and Development* **62**, 6:1 (2018).
- [4] D. Cuomo, M. Caleffi, and A. S. Cacciapuoti, *IET Quantum Communication* **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," (2022), [arXiv:2212.10609 \[quant-ph\]](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, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **469**, 20120686 (2013).
- [7] A. S. Cacciapuoti, M. Caleffi, F. Tafuri, F. S. Cataliotti, S. Gherardini, and G. Bianchi, *IEEE Network* **34**, 137 (2020).
- [8] H. Tang, B. Li, G. Wang, H. Xu, C. Li, A. Barr, P. Cappellaro, and J. Li, *Phys. Rev. Lett.* **130**, 150602 (2023).
- [9] A. Montanaro and C. Shao, "Quantum communication complexity of linear regression," (2023), [arXiv:2210.01601 \[quant-ph\]](https://arxiv.org/abs/2210.01601).
- [10] D. Gilboa and J. R. McClean, "Exponential quantum communication advantage in distributed learning," (2023), [arXiv:2310.07136 \[quant-ph\]](https://arxiv.org/abs/2310.07136).
- [11] A. Childs, *Quantum Information and Computation* **5**, 456 (2005).
- [12] J. F. Fitzsimons, *npj Quantum Information* **3** (2017), [10.1038/s41534-017-0025-3](https://doi.org/10.1038/s41534-017-0025-3).

- [13] V. Giovannetti, L. Maccone, T. Morimae, and T. G. Rudolph, *Phys. Rev. Lett.* **111**, 230501 (2013).
- [14] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, and J. S. Rellermeier, *ACM Computing Surveys* **53**, 1 (2020).
- [15] H. B. Barlow, *Neural computation* **1**, 295 (1989).
- [16] S. Lloyd, M. Mohseni, and P. Rebentrost, “Quantum algorithms for supervised and unsupervised machine learning,” (2013), [arXiv:1307.0411](https://arxiv.org/abs/1307.0411) [quant-ph].
- [17] P. Rebentrost, M. Mohseni, and S. Lloyd, *Phys. Rev. Lett.* **113**, 130503 (2014).
- [18] T. Li, S. Chakrabarti, and X. Wu, “Sublinear quantum algorithms for training linear and kernel-based classifiers,” (2019), [arXiv:1904.02276](https://arxiv.org/abs/1904.02276) [quant-ph].
- [19] X. Zhou and D. Qiu, *Quantum Information Processing* **20** (2021), [10.1007/s11128-021-03301-y](https://doi.org/10.1007/s11128-021-03301-y).
- [20] H. Fang and Q. Qian, *Future Internet* **13**, 94 (2021).
- [21] A. Wood, K. Najarian, and D. Kahrobaei, *ACM Computing Surveys* **53**, 1 (2020).
- [22] P. Paillier, in *Advances in Cryptology — EUROCRYPT ’99* (Springer Berlin Heidelberg) pp. 223–238.
- [23] J. H. Cheon, A. Kim, M. Kim, and Y. Song, in *Advances in Cryptology – ASIACRYPT 2017* (Springer International Publishing, 2017) pp. 409–437.
- [24] G. Couteau and M. Zarezadeh, “Non-interactive secure computation of inner-product from lpn and lwe,” *Cryptography ePrint Archive*, Paper 2023/072 (2023), <https://eprint.iacr.org/2023/072>.
- [25] E. Boyle, G. Couteau, N. Gilboa, Y. Ishai, L. Kohl, and P. Scholl, in *Advances in Cryptology – CRYPTO 2020* (Springer International Publishing, 2020) pp. 387–416.
- [26] D. York, *Canadian Journal of Physics* **44**, 1079 (1966).
- [27] G. Tsoumakas and I. Katakis, *International Journal of Data Warehousing and Mining* **3**, 1 (2007).
- [28] V. Giovannetti, S. Lloyd, and L. Maccone, *Phys. Rev. Lett.* **100**, 160501 (2008).
- [29] G. Brassard, P. Høyer, and A. Tapp, in *Automata, Languages and Programming* (Springer Berlin Heidelberg, 1998) pp. 820–831.
- [30] M. Fanizza, M. Rosati, M. Skotiniotis, J. Calsamiglia, and V. Giovannetti, *Phys. Rev. Lett.* **124**, 060503 (2020).
- [31] A. Anshu, Z. Landau, and Y. Liu, in *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing* (2022) pp. 44–51.
- [32] T. Hunt, C. Song, R. Shokri, V. Shmatikov, and E. Witchel, *CoRR* [abs/1803.05961](https://arxiv.org/abs/1803.05961) (2018), 1803.05961.
- [33] E. Hesamifard, H. Takabi, M. Ghasemi, and R. N. Wright, *Proceedings on Privacy Enhancing Technologies* **2018**, 123 (2018).
- [34] F. Xu, X. Ma, Q. Zhang, H.-K. Lo, and J.-W. Pan, *Rev. Mod. Phys.* **92**, 025002 (2020).
- [35] C. H. Bennett and G. Brassard, *Theoretical Computer Science* **560**, 7 (2014).
- [36] R. Renner, N. Gisin, and B. Kraus, *Phys. Rev. A* **72**, 012332 (2005).
- [37] C. Crepeau, D. Gottesman, and A. Smith, “Secure multi-party quantum computing,” (2002), [arXiv:quant-ph/0206138](https://arxiv.org/abs/quant-ph/0206138) [quant-ph].
- [38] B. Knott, S. Venkataraman, A. Y. Hannun, S. Sengupta, M. Ibrahim, and L. van der Maaten, *CoRR* [abs/2109.00984](https://arxiv.org/abs/2109.00984) (2021), 2109.00984.
- [39] 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, *Scientific Reports* **10** (2020), [10.1038/s41598-020-69250-1](https://doi.org/10.1038/s41598-020-69250-1).
- [40] W. Li, S. Lu, and D.-L. Deng, *Science China Physics, Mechanics & Astronomy* **64** (2021), [10.1007/s11433-021-1753-3](https://doi.org/10.1007/s11433-021-1753-3).
- [41] H. T. Larasati, M. Firdaus, and H. Kim, in *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, 2022).
- [42] N. Kumar, J. Heredge, C. Li, S. Eloul, S. H. Sureshbabu, and M. Pistoia, “Expressive variational quantum circuits provide inherent privacy in federated learning,” (2023), [arXiv:2309.13002](https://arxiv.org/abs/2309.13002) [quant-ph].
- [43] “IEEE standard for floating-point arithmetic.”
- [44] A. S. Holevo, *Problemy Peredachi Informatsii* **9**, 3 (1973).