Communication-Efficient Quantum Algorithm for Distributed Machine Learning

Hao Tang,[1,*] Boning Li,[2,3,*] Guoqing Wang,[2,4] Haowei Xu,[4] Changhao Li,[2,4]
Ariel Barr,[1] Paola Cappellaro,[2,3,4,†] and Ju Li[1,4,‡]

[1]Department of Materials Science and Engineering, Massachusetts Institute of Technology, Massachusetts 02139, USA
[2]Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA
[3]Department of Physics, Massachusetts Institute of Technology, Massachusetts 02139, USA
[4]Department of Nuclear Science and Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA

(Received 11 September 2022; accepted 21 March 2023; published 12 April 2023)

The growing demands of remote detection and an increasing amount of training data make distributed machine learning under communication constraints a critical issue. This work provides a communication-efficient quantum algorithm that tackles two traditional machine learning problems, the least-square fitting and softmax regression problems, in the scenario where the dataset is distributed across two parties. Our quantum algorithm finds the model parameters with a communication complexity of $O(\log_2(N)/\epsilon)$, where N is the number of data points and ϵ is the bound on parameter errors. Compared to classical and other quantum methods that achieve the same goal, our methods provide a communication advantage in the scaling with data volume. The core of our methods, the quantum bipartite correlator algorithm that estimates the correlation or the Hamming distance of two bit strings distributed across two parties, may be further applied to other information processing tasks.

DOI: 10.1103/PhysRevLett.130.150602

The amount of training data is critical for machine learning to achieve high accuracy, generalization capabilities, and predictive power. Nowadays, data collection is growing with unprecedented speed around the world, so it becomes a challenge for algorithms to exploit such large-scale data within feasible time and memory [1,2]. Distributed machine learning emerges as a promising solution, where the training data and learning process are allocated to multiple machines [1,3,4]. This scales up computational power and is also suitable for intrinsically distributed data when collected [5,6]. However, these algorithms require extensive communication between different machines, which usually becomes a rate-limiting step [7]. Therefore, efficient communication schemes for distributed machine learning tasks are attracting broad interest. The communication necessary between two machines in a computation task is quantified by its communication complexity, either within classical [8–11] or quantum channels [12–15]. Compared to classical communication, even though quantum algorithms have been shown to reduce the communication complexity in some scenarios [16], machine learning tasks were not included. Quantum algorithms have been generally studied as accelerators for the computational complexity [17] in problems such as least-square fitting [18], statistical inference [19], feature engineering [20], and classification problems [21]. Whether quantum algorithms can accelerate communication in distributed learning tasks remains an open question.

Here, we propose a quantum communication algorithm for two typical data fitting subroutines in machine learning: least-square fitting and softmax regression, which are the common output layers of predictors and classifiers, respectively [22]. In this Letter we assume a training dataset contains N independent identically distributed (iid) data points. Each data point has an M-dimensional input $\vec{x}$ and a scalar output y. In the basic communication scenario [4], the training dataset, comprising the input attributes and labels, is distributed across two parties, Alice and Bob. Both least-square fitting and softmax regression aim at fitting a model $y \approx f(\vec{x}, \boldsymbol{\lambda})$ to the data, by estimating the parameters $\hat{\boldsymbol{\lambda}}$ that minimize a given loss function. The goal of a communication algorithm is to minimize the number of bits [8,9] or qubits [14,15] exchanged between Alice and Bob during model fitting, while keeping the accuracy of $\hat{\boldsymbol{\lambda}}$ within a standard error ϵ.

Least-square fitting has been extensively studied in both classical distributed algorithms and single-party (no communication) quantum algorithms. Using a classical algorithm based on correlation estimation, it has been proved that the classical communication complexity cannot be below $O(1/\epsilon^2)$ [23,24]. However, to reach such a lower bound requires an exponentially large number of data points. In the case of finite datasets, since the accuracy of the fitting parameters should be at least as small as its error ϵ, a classical deterministic method requires $O[N\log_2(1/\epsilon)]$ bits to be exchanged between two parties within a precision ϵ [25]. When high accuracy is not required, only $1/\epsilon^2$ data points with random indexes need to be transferred, which yields a $O\{[\log_2(1/\epsilon) + \log_2(N)](1/\epsilon^2)\}$ communication complexity [23]. Then, to

achieve a statistical variance $\epsilon_s^2 = \mathrm{var}(|\lambda|) \propto 1/N$, these two classical algorithms have the same communication complexity $O[N\log_2(N)]$ or $O(\log_2(1/\epsilon_s)/\epsilon_s^2)$. In comparison, quantum computation methods for linear fitting based on the Harrow-Hassidim-Lloyd (HHL) algorithm [26] yield normalized parameters ($|\boldsymbol{\lambda}|^2 = 1$) from a quantum state $|\lambda\rangle = \sum_{j=1}^{M} \lambda_j |j\rangle$ with communication complexity of $O[\log_2(N)]$ [18,27,28]. However, to extract $\lambda_{j=1,\ldots,M}$, the HHL-based algorithm requires $O[M^2(1/\epsilon^2)]$ repeated measurements. In this case, the HHL-based fitting algorithm requires communicating $O(\log_2(N)/\epsilon^2)$ qubits [18,29], with no clear advantage over classical algorithms.

We designed a *quantum counting*-based [30,31] communication algorithm that achieves a reduced communication complexity of $O(\log_2(N)/\epsilon)$ for both least-square fitting and softmax regression (Table I). At its core, the direct action of our algorithm is to estimate the correlation or the Hamming distance of two bit strings distributed across two parties. Embedding this algorithm into a hybrid computing scheme enables the data fitting tasks beyond the theoretical limit of classical algorithms, and we expect it could benefit other scenarios not analyzed here.

Estimating correlation.—We first present the core subroutine of our methods, the quantum bipartite correlator (QBC) algorithm. The problem is stated as follows: Alice and Bob have N-dimensional vectors $\vec{x}^{\mathrm{b}}, \vec{y}^{\mathrm{b}} \in \{0,1\}^N$, respectively, that can only take binary values (denoted by superscript $^{\mathrm{b}}$). This is not as restrictive as it sounds, as real numbers can always be expanded as binary floating point numbers (see Sec. "*Least-square fitting*"). The task is to estimate the correlation $\hat{\rho} \equiv \left[\left(\overline{x^{\mathrm{b}}y^{\mathrm{b}}} - \overline{x^{\mathrm{b}}}\cdot\overline{y^{\mathrm{b}}}\right)/\sqrt{\overline{x^{\mathrm{b}}}(1-\overline{x^{\mathrm{b}}})\overline{y^{\mathrm{b}}}(1-\overline{y^{\mathrm{b}}})}\right]$, in which the communication-intensive step is to evaluate $\overline{x^{\mathrm{b}}y^{\mathrm{b}}} = (1/N)\sum_{i=1}^{N} x_i^{\mathrm{b}} y_i^{\mathrm{b}}$ within a standard deviation error ϵ [23].

We assume that Alice and Bob have access to quantum computers with oracles. The oracle of Alice's computer performs a unitary transformation $\hat{U}_{\vec{x}^{\mathrm{b}}}^{1,2}\colon |i\rangle_1|0\rangle_2 \mapsto |i\rangle_1|x_i^{\mathrm{b}}\rangle_2$ that encodes the data x_i^{b}, where $|i\rangle$ is an $n \equiv \lceil \log_2(N)\rceil$-qubit state $|i_1 i_2 \cdots i_n\rangle$, representing the index of the queried component, and $|x_i^{\mathrm{b}}\rangle$ is a single-qubit state. Bob has an oracle $\hat{U}_{\vec{y}^{\mathrm{b}}}$ of the same type that encodes the data y_i^{b}. This type of oracle is a common building block in quantum algorithms [18,26,36], which can be realized through quantum random access memory [37] or other data-loading procedures [38,39].

Estimating the correlation $\overline{x^{\mathrm{b}}y^{\mathrm{b}}}$ is based on quantum counting, in which the phase oracle is realized cooperatively by Alice and Bob through communication, as shown in Fig. 1. We sketch the framework here and provide the algorithm details in the Supplemental Material [40] (which includes Refs. [31,33,35,41]), Sec. I. The algorithm works on an n-qubit vector index space ($|\cdot\rangle_n$), a t-qubit register space ($|\cdot\rangle_t$), and a 2-qubit oracle workspace ($|\cdot\rangle_o$). Initially, all qubits are set to zero: $|\psi_0\rangle \equiv |0\rangle_t|0\rangle_n|00\rangle_o$. Hadamard gates are applied to create superposition in both t and n space $|\psi_1\rangle = 2^{-(t+n)/2}\sum_{i,\tau}|\tau\rangle_t|i\rangle_n|00\rangle_o$. A phase oracle on the state $|\cdot\rangle_n$ can be realized through the following unitary operation:

$$\hat{O}_{\vec{x}^{\mathrm{b}},\vec{y}^{\mathrm{b}}} \equiv \hat{U}_{\vec{x}^{\mathrm{b}}}^{n,o_1}\hat{U}_{\vec{y}^{\mathrm{b}}}^{n,o_2} CZ^{o_1,o_2}\hat{U}_{\vec{y}^{\mathrm{b}}}^{n,o_2}\hat{U}_{\vec{x}^{\mathrm{b}}}^{n,o_1}, \tag{1}$$

which yields $\hat{O}_{\vec{x}^{\mathrm{b}},\vec{y}^{\mathrm{b}}}|i\rangle_n|00\rangle_o = (-1)^{x_i^{\mathrm{b}}y_i^{\mathrm{b}}}|i\rangle_n|00\rangle_o$. Here o_1, o_2 are the two qubits in the oracle space, and CZ^{o_1,o_2} is a control-Z gate acting on them. Each oracle call requires about

TABLE I. Communication complexity of classical distributed algorithm, quantum counting-based algorithm developed in this work, and other quantum algorithms. Listed problems include estimating correlation and Hamming distance of two separate bit strings, distributed linear fitting, and distributed softmax regression. In the first column, (c) and (q) mean the problem requires output as classical data or quantum states, respectively. In the table, ϵ, N, and M are the standard error of solution, number of data points, and number of attributes in Alice's data; κ and s are the condition number and sparseness of the matrix $\boldsymbol{X}$ in linear regression problems; and q is the number of classes in softmax regression problems. (See derivation in Sec. III.). All the classical algorithms and the LOCC algorithm transfer classical bits, and the rest of the quantum algorithms transfer qubits.

Problem (output)	Classical algorithm	Quantum counting	Other quantum algorithm
Correlation (c)	$O(1/\epsilon^2)$ lower-bound) Ref. [23]	$O(\log_2(N)/\epsilon)$	$O(\log_2(N)/\epsilon^2)$ (swap-test, [32]) $O[\log_2(N)\max\{(1/\epsilon^2),(\sqrt{N}/\epsilon)\}]$ (LOCC, [33])
Hamming distance (c)	$O(N)$ [34]	$O(\log_2(N)/\epsilon)$	$O(\log_2(N)/\epsilon^2)$ (classical shadows, [35])
Linear-fitting (c)	$O[N\log_2(\kappa^2/\epsilon)]$ (deterministic [25]) $O\{[\log_2(N)+\log_2(\kappa^2/\epsilon)]/(\epsilon/\kappa^2)^2\}$ (stochastic [23])	$O(M\kappa(\log_2(N)/\epsilon))$	$O(M^2\kappa^5(\log_2(N)/\epsilon^2))$ (HHL, [18])
Linear-fitting (q)	$\cdots$	$O(M\kappa(\log_2(N)/\epsilon))$	$O[\kappa^5\log_2(N)]$ (HHL, [18])
Softmax regression (c)	$O(N\log_2 q)$	$O\{Mq\kappa[\log_2(N)/\epsilon]\}$	$\cdots$

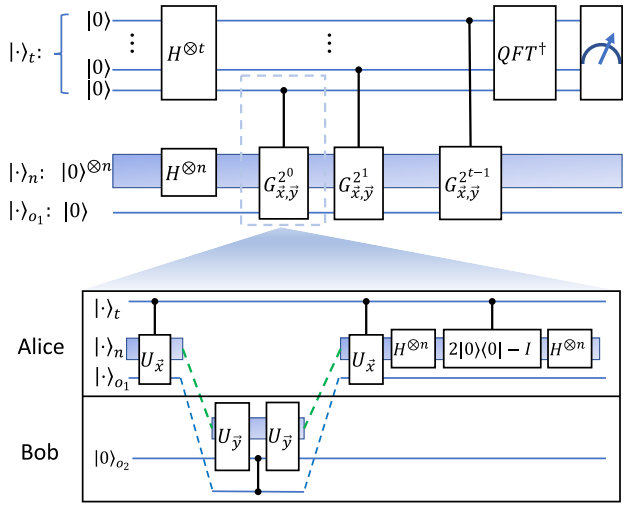

FIG. 1. Quantum circuits for the distributed quantum counting or QBC scheme. H, G, and $\mathrm{QFT}^\dagger$ represent the Hadamard gate, the Grover operator, and the inverse QFT, respectively. The t-qubit register is measured after the inverse QFT. The inset shows the biparty distributed scheme of the Grover operation, where U_{x_l} and V_{y_k} are defined in Eqs. (7) and (8).

$2n$-qubit communication, as Alice needs to send the $(n+1)$ qubits to Bob after applying $\hat{U}^{n,o_1}_{\vec{x}^{\mathrm{b}}}$ and Bob needs to send the $(n+1)$ qubits back after applying $\hat{U}^{n,o_2}_{\vec{y}^{\mathrm{b}}} CZ^{o1,o2} \hat{U}^{n,o_2}_{\vec{y}^{\mathrm{b}}}$; finally, Alice applies $\hat{U}^{n,o_1}_{\vec{x}^{\mathrm{b}}}$ to finish the whole oracle $\hat{O}_{\vec{x}^{\mathrm{b}},\vec{y}^{\mathrm{b}}}$. The Grover operation needed for counting is then constructed as $\hat{G}_{\vec{x}^{\mathrm{b}},\vec{y}^{\mathrm{b}}} \equiv \hat{H}^{\otimes n}(2|0\rangle_n\langle 0|_n - \hat{I})\hat{H}^{\otimes n}\hat{O}_{\vec{x}^{\mathrm{b}},\vec{y}^{\mathrm{b}}}$. The QBC scheme applies the Grover operation iteratively on the initial state:

$$|\psi_2\rangle = \frac{1}{2^{(t+n)/2}} \sum_\tau |\tau\rangle_t \otimes (\hat{G}_{\vec{x}^{\mathrm{b}},\vec{y}^{\mathrm{b}}})^\tau \sum_i |i\rangle_n |00\rangle_o. \quad (2)$$

Expanding the Grover operator in its eigenbasis gives $(\hat{G}_{\vec{x}^{\mathrm{b}},\vec{y}^{\mathrm{b}}})^\tau \sum_i |i\rangle_n = (e^{i\tau\theta}|\phi_+\rangle\langle\phi_+| + e^{-i\tau\theta}|\phi_-\rangle\langle\phi_-|)\sum_i |i\rangle_n$, where $|\phi_\pm\rangle$ are the two eigenstates of $\hat{G}_{\vec{x}^{\mathrm{b}},\vec{y}^{\mathrm{b}}}$, and $\theta = 2\arcsin\left(\sqrt{\overline{x^{\mathrm{b}}y^{\mathrm{b}}}}\right)$. Applying the inverse quantum Fourier transform ($\mathrm{QFT}^\dagger$) to $|\cdot\rangle_t$ yields the final state:

$$|\psi_3\rangle = \frac{1}{\sqrt{2^{t+n}}} \sum_{\eta=\pm,i} \langle\phi_\eta|i\rangle |\phi_\eta\rangle_n |00\rangle_o \mathrm{QFT}^\dagger \left(\sum_\tau |\tau\rangle_t e^{i\eta\tau\theta}\right). \quad (3)$$

Measuring the t register will project into a state $|j\rangle_t$ resulting in the phase $2\pi j \cdot 2^{-t}$ which encodes either $\hat{\theta}$ or $2\pi - \hat{\theta}$ with an equivalent standard deviation: $\Delta\hat{\theta} = 2^{-t+1}$.

Both cases give the same estimated correlation $\widehat{\overline{x^{\mathrm{b}}y^{\mathrm{b}}}} = \sin^2(\hat{\theta}/2)$, with standard deviation $\epsilon = \sqrt{\overline{x^{\mathrm{b}}y^{\mathrm{b}}}(1-\overline{x^{\mathrm{b}}y^{\mathrm{b}}})}2^{-t+1}$ (see the Supplemental Material [40], Sec. II, for details). The overall communication complexity $\mathcal{C}$ is the Grover operation's $2(n+1)$ qubit communication repeated for $2^t - 1$ iterations:

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

where we choose t to satisfy the desired error bound. The computational complexity is the total number of oracle calls by Alice and Bob, which is $\mathcal{C}_{\mathrm{comp}} = 4(2^t - 1) = O(1/\epsilon)$.

We note that the QBC algorithm solves the problem of estimating $x^{\mathrm{b}}y^{\mathrm{b}}$, which is equivalent to computing the inner product. The inner product of quantum states is usually accomplished by the swap test algorithm [32,33]. However, the swap test method costs $O(\log_2(N)/\epsilon^2)$ bits of communication, due to the requirement of repeated measurements. Recently, Anshu *et al.* [33] proposed an algorithm to estimate the inner product of two quantum states using local quantum operations and classical communication (LOCC). With respect to communication complexity, neither the original SWAP test that transfers qubits, nor LOCC that transfers bits, achieves an advantage over the classical algorithms. The QBC algorithm achieves the communication advantage by utilizing quantum counting and a distributed implementation of the Grover iterator.

Estimating the Hamming distance.—The QBC algorithm can be used to estimate the Hamming distance d between $\vec{x}^{\mathrm{b}}$ and $\vec{y}^{\mathrm{b}}$ (that is, the number of positions i where $x^{\mathrm{b}}_i \neq y^{\mathrm{b}}_i$). The key is to replace the oracle in Eq. (1) by

$$\hat{O}'_{\vec{x}^{\mathrm{b}},\vec{y}^{\mathrm{b}}} \equiv \hat{U}^{n,o_1}_{\vec{x}^{\mathrm{b}}} \hat{U}^{n,o_2}_{\vec{y}^{\mathrm{b}}} C^{o_1,o_2}_{\mathrm{NOT}} Z^{o_2} C^{o_1,o_2}_{\mathrm{NOT}} \hat{U}^{n,o_2}_{\vec{y}^{\mathrm{b}}} \hat{U}^{n,o_1}_{\vec{x}^{\mathrm{b}}}, \quad (5)$$

where $C^{o_1,o_2}_{\mathrm{NOT}}$ represents a Control-NOT (CNOT) gate gate with o_1 as control qubit, and Z^{o_2} represents a σ_Z gate acting on the o_2 qubit. This phase oracle acts as $\hat{O}'_{\vec{x}^{\mathrm{b}},\vec{y}^{\mathrm{b}}}|i\rangle_n|00\rangle_o = (-1)^{x^{\mathrm{b}}_i \oplus y^{\mathrm{b}}_i}|i\rangle_n|00\rangle_o$, and the QBC scheme counts the number of indexes i such that $x^{\mathrm{b}}_i \oplus y^{\mathrm{b}}_i = 1$, returning (d/N) with the same communication complexity as for estimating the correlation.

This result provides a quantum solution to the widely studied gap-Hamming problem in theoretical computer science [34,42]. Multiple proofs conclude that it is impossible for a classical protocol to output the Hamming distance d within $\sqrt{N}$ using less than $O(N)$ bits of communication [23,34,43]. By setting $\epsilon = (1/\sqrt{N})$, our quantum scheme performs the estimation using $O[\sqrt{N}\log_2(N)]$ qubits of communication, exhibiting a square-root speedup over classical algorithms. The Hamming distance can also be estimated via the "classical shadows" algorithm [35] (an established quantum algorithm) with communication complexity of $O(\log N/\epsilon^2)$ (see the Supplemental Material [40], Sec. V, for details), which has a higher order to $(1/\epsilon)$ than the QBC algorithm. As estimating the Hamming distance under communication

constraints has applications in database searching [42], networking [44], and streaming algorithms [45], the QBC algorithm may be embedded into other diverse applications in the future.

Least-square fitting.—When machine learning models are used to predict the central value of Gaussian distributed continuous variables, the common setting is a linear output layer $f(\boldsymbol{x_i}, \boldsymbol{\lambda}) = \lambda_0 + \vec{\lambda} \cdot \vec{x} = \boldsymbol{\lambda}^T \boldsymbol{x}$ (where $\boldsymbol{x_i} \equiv (1, x_{i,1}, \ldots, x_{i,M-1})^T$ and $\boldsymbol{\lambda} \equiv (\lambda_0, \lambda_1, \ldots, \lambda_{M-1})^T$) that performs the least-square fitting. The model fitting is reduced to solving a linear least-square problem $\boldsymbol{X\lambda} = \boldsymbol{y}$, where $\boldsymbol{X} \equiv (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)^T$ is an $N \times M$ matrix belonging to Alice and $\boldsymbol{y}$ is Bob's $N \times 1$ column vector, both of which have real-number components. The goal is to estimate $\hat{\boldsymbol{\lambda}}$ with standard error ϵ using minimal communications. Here we assume $M \ll N$, as the number of model parameters or attributes is usually much smaller than the number of data points to avoid overfitting.

The least-square solution of the equation is $\boldsymbol{\lambda} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} = (1/N)(N\boldsymbol{X}^\dagger)\boldsymbol{y}$, where $\boldsymbol{X}^\dagger$ is the Moore-Penrose pseudoinverse of $\boldsymbol{X}$, and $N\boldsymbol{X}^\dagger$ should scale as $O(N^0)$ in the case of the iid dataset. As $N\boldsymbol{X}^\dagger$ can be computed by Alice locally, only the calculation of $(1/N)(N\boldsymbol{X}^\dagger)\boldsymbol{y}$ involves communication. The jth component of λ can be represented by correlations (inner product) $\lambda_j = (1/N)\sum_i (NX^\dagger_{ji}) y_i, j = 0, \ldots, M-1$, which can be calculated by expanding the real numbers as binary floating point numbers. For example, following the IEEE 754 standard [46], each $NX^\dagger_{ji}$ and y_i can be written as binary floating point numbers: $NX^\dagger_{ji} \equiv \sum_{k=0}^{\infty} 2^{u-k} x^{\mathrm{b}k}_{ji}$, $y_i \equiv \sum_{k=0}^{\infty} 2^{v-k} y^{\mathrm{b}k}_i$, where u and v are the highest digits of the elements of $NX^\dagger_{ji}$ and y_i, and $x^{\mathrm{b}k}_{ji}$ and $y^{\mathrm{b}k}_i$ are the kth digits, respectively. Then λ_j can be written as

$$\lambda_j = \frac{1}{N}\sum_{r=0}^{\infty} 2^{u+v-r} \sum_{k=0}^{r}\sum_{i=1}^{N} x^{\mathrm{b}k}_{ji} y^{\mathrm{b}(r-k)}_i = 2^{u+v}\sum_{r=0}^{\infty} 2^{-r}(r+1) f_{jr}. \quad (6)$$

As $x^{\mathrm{b}k}_{ji}$ and $y^{\mathrm{b}k}_i$ are binary quantity, the inner product $f_{jr} = [1/N(r+1)]\sum_{k=0}^{r}\sum_{i=1}^{N} x^{\mathrm{b}k}_{ji} y^{\mathrm{b}(r-k)}_i$ can be directly estimated by the QBC algorithm. The overall communication complexity is $\mathcal{C} = \sum_{j=1}^{M}\sum_{r=0}^{\infty} 2[\log_2(N)/\epsilon_{jr}]$, where ϵ_{jr} is the standard deviation error of f_{jr}. The infinite series in r is cut off according to the target accuracy ϵ of each component λ_j, setting ϵ_{jr} to $\epsilon_{jr} = \epsilon[(0.449/2^{u+v})(r+1)^{2/3}]2^{(2/3)r}$. If r is large enough so that $\epsilon_{jr} > 1$, the quantum algorithm is no longer pertinent, as the number t of ancilla qubits in the quantum phase estimation algorithm drops to less than one, since $\epsilon_{jr} = 2^{-t+1}$. In that case, f_{jr} can be simply dropped because these f_{jr} terms are multiplied by 2^{-r} in Eq. (6); they do not contribute substantially to the total error of λ_j. Rewriting $\mathcal{C}$ in terms of the condition number $\kappa = \|\mathbf{A}^{-1}\|_\infty \|\mathbf{A}\|_\infty$ of the matrix $\mathbf{A} = (1/N)\boldsymbol{X}^T\boldsymbol{X}$ gives

$$\mathcal{C} = 11.026 \times 2^{v+1} 2^u M \frac{\log_2(N)}{\epsilon} = O\left(\frac{M\kappa \log_2(N)}{\epsilon}\right), \quad (7)$$

where the absolute magnitude of 2^{v+u} in $\mathcal{C}$ is on the same order of $(\kappa|y|_\infty/\|X\|_\infty)$ (see the Supplemental Material [40], Sec. III, for details). The total number of oracle queries is $\mathcal{C}_{\mathrm{comp}} = (M\kappa/\epsilon)$.

An HHL-based quantum algorithm has been previously developed for data fitting without the communication bottleneck [18]. The algorithm produces a quantum state $|\lambda\rangle \equiv \sum_j \lambda_j |j\rangle$ with $O[(s^3\kappa^6/\epsilon)\log_2(N)]$ computational complexity, where $0 \le s \le 1$ is the sparseness of the matrix A. As explained above, this method is, however, inefficient in extracting classical data from the quantum states. In the communication-restricted scenario, the HHL-based algorithm requires sharing $O[(\kappa^5 M^2/\epsilon^2)\log_2(N)]$ qubits. For a target statistical precision $\epsilon = 1/\sqrt{N}$, the QBC based scheme again obtains a square-root speedup from $O(N)$ to $O[\sqrt{N}\log_2(N)]$ compared with the classical theoretical limit. A summary of the communication complexity of different schemes is presented in Table I.

After demonstrating that the QBC algorithm can reduce the communication complexity to N, we numerically assess the practical conditions when the quantum algorithm shows

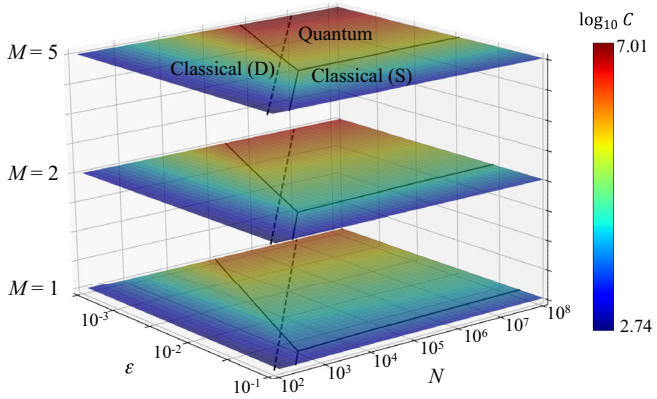

FIG. 2. Communication complexity phase diagram of the QBC algorithm, deterministic, and stochastic classical algorithms in parameter space of N, ϵ, and M. Without loss of generality, we assume that both $\vec{x}$ and y are normalized and different components of $\vec{x}$ are iid. The color map represents the minimal communication complexity of the three algorithms in the logarithmic scale. Black lines divide the space into three regions denoted as Classical (D), Classical (S), and Quantum, representing the region where the deterministic classical, stochastic classical, and QBC algorithm have the smallest communication complexity. The black dashed line in each layer indicates the statistical variance $\epsilon = 1/\sqrt{N}$.

an advantage compared with classical algorithms (Fig. 2). In general, the QBC algorithm starts showing an advantage when $N \geq 10^3 \sim 10^4$, which is a reasonable range in fitting problems. The quantum advantage requires ϵ to be in an intermediate level: too-small or too-large ϵ make deterministic or stochastic classical algorithms have a lower communication complexity.

The quality of a fitted model can be characterized by the mean square error $E \equiv (1/N)(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\lambda}})^2 = (1/N)(\boldsymbol{y}^2 + \hat{\boldsymbol{y}}^2 - 2\boldsymbol{y}^T\hat{\boldsymbol{y}})$. Only the calculation of $(1/N)\boldsymbol{y}^T\hat{\boldsymbol{y}}$ involves communication, which can again be realized through the correlation estimation scheme, requiring $O(\log_2(N)/\epsilon)$-qubit communication.

The applications of the QBC algorithm are not restricted to fitting linear functions, as a general function of $\vec{x}$ can be expanded as a linear combination of a series of basis functions $y = \sum_j \lambda_j f_j(\vec{x})$. The matrix $F_{ij} \equiv f_j(\vec{x}_i)$ can be computed locally, and the problem is then reduced to the linear fitting problem $\boldsymbol{F\lambda} = \boldsymbol{y}$. Furthermore, this scheme can be used as the linear output layer of a neural network in high-expressivity machine learning models [22].

Softmax classifier.—Besides fitting continuous data, the QBC scheme can also be used for fitting discrete labels (classification). A common output layer of classification models is the softmax classifier. The basic scenario is that the data of Bob y_i has discrete possible values in a set of classes $Y = \{c_1, c_2, \ldots, c_q\}$. The model outputs the probabilities for a given data point $\vec{x}$ to be in each class $P(y = c_j|\boldsymbol{x}, \boldsymbol{\Lambda})$ with ansatz $P(y = c_j|\boldsymbol{x}, \boldsymbol{\Lambda}) = (e^{\boldsymbol{\lambda}_j{}^T\boldsymbol{x}} / \sum_l e^{\boldsymbol{\lambda}_l{}^T\boldsymbol{x}})$, where the coefficient matrix is $\boldsymbol{\Lambda} \equiv (\boldsymbol{\lambda}_0, \ldots, \boldsymbol{\lambda}_q)$. The cross-entropy loss function $L(\boldsymbol{\Lambda}) \equiv -\sum_{ij} 1_{y_i=c_j} \log_2 P(y_i = c_j|\boldsymbol{x}_i, \boldsymbol{\Lambda})$ is to be minimized, where $1_{y=c_j}$ is a 1 when $y = c_j$ and 0 otherwise. $\hat{\boldsymbol{\lambda}}$ can be obtained from a set of equations:

$$\sum_{i=1}^{N} \frac{\boldsymbol{x}_i e^{\hat{\boldsymbol{\lambda}}_j^T \boldsymbol{x}_i}}{\sum_{k=1}^{q} e^{\hat{\boldsymbol{\lambda}}_k^T \boldsymbol{x}_i}} = \sum_{i=1}^{N} 1_{y_i=c_j} \boldsymbol{x}_i, \qquad j = 1, 2, \ldots, q. \tag{8}$$

The equation's right-hand side can be estimated as the inner product between $1_{y=c_j}$ and the vector $\boldsymbol{x}$ following our previous scheme, with communication complexity $\mathcal{C} = O(qM\log_2(N)/\epsilon)$ (see the Supplemental Material [40], Sec. IV, for details). As the left-hand side of the equations does not involve y, the equations can be solved without any further communication. We note that logistic regression for the two-class classification problems can be derived as a special case of the softmax regression scheme with $q = 2$.

We can further quantify the communication complexity of evaluating the quality of a fitted classifier. The quality can be determined by comparing the model outputs $\hat{y}_i = \mathrm{argmax}_{c_j} P(y_i = c_j|x_i, \boldsymbol{\Lambda})$ and labels y_i on the training or testing dataset. Alice and Bob encode $\hat{y}_i$ and y_i into Nq-bit strings $\hat{b}_{ij} \equiv 1_{\hat{y}_i=c_j}$ and $b_{ij} \equiv 1_{y_i=c_j}$, respectively. Then the correctness of the model can be determined by estimating the Hamming distance d between $\hat{b}$ and b as $1 - (d/2N)$ (as each error in classification contributes a two-bit difference). The communication complexity is $\mathcal{C} = O(\log_2(Nq)/\epsilon)$, showing no dependence on dimension M and insensitive dependence on the number of classes q.

Conclusion and outlook.—In this work, we developed a distributed Grover-quantum counting-based scheme that performs distributed least-square fitting or softmax regression with a communication complexity $O(\log_2(N)/\epsilon)$, a square-root improvement over classical algorithms. The quantum advantage comes from reduced communication requirements by encoding information in the phases of a superposition state, a unique attribute of quantum systems. Some previous quantum schemes [18,29,32] encode the information in the weight of superposition: as extracting the superposition weight by state tomography also requires $O(1/\epsilon^2)$ repetitions of state preparation and measurements, these methods do not show significant advantage in deriving classical fitting parameters compared with classical schemes. The core of our algorithm, a communication-efficient "quantum bipartite correlator," is expected to be useful in other communication and information-processing contexts as well. This method is expected to preserve privacy between two parties. Neither Alice nor Bob can determine the other party's attributes of a specific data point, as only the statistical average is encoded in the phase during communication. This meets the security requirement of distributed computing [47].

We thank Prof. Isaac Chuang for insightful comments. This work was supported by NSF CMMI-1922206 and DTRA (Grant No. HDTRA1-20-2-0002) Interaction of Ionizing Radiation with Matter (IIRM) University Research Alliance (URA). The calculations in this work were performed in part on the Texas Advanced Computing Center (TACC) and MIT engaging cluster.

*These authors contributed equally.
†pcappell@mit.edu
‡liju@mit.edu

[1] M. Gheisari, G. Wang, and M. Z. A. Bhuiyan, in *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)* (IEEE, Guangzhou, 2017), Vol. 2, pp. 173–180.
[2] L. Bottou and O. Bousquet, Adv. Neural Inf. Process. Syst. **20**, 161 (2007).
[3] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, and J. S. Rellermeyer, ACM Comput. Surv. (CSUR) **53**, 1 (2020).
[4] D. Peteiro-Barral and B. Guijarro-Berdiñas, Prog. Artif. Intell. **2**, 1 (2013).
[5] J. Erickson, *Database Technologies: Concepts, Methodologies, Tools, and Applications: Concepts, Methodologies, Tools, and Applications* (IGI Global, Hershey PA, 2009).

[6] S. E. Haupt and B. Kosovic, in *2015 IEEE Symposium Series on Computational Intelligence* (IEEE, Cape Town, 2015), pp. 496–501.

[7] S. Li, M. A. Maddah-Ali, Q. Yu, and A. S. Avestimehr, IEEE Trans. Inf. Theory **64**, 109 (2017).

[8] H. Abelson, J. Appl. Comput. Mech. **27**, 384 (1980).

[9] A. C.-C. Yao, in *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing* (Association for Computing Machinery, New York, 1979), pp. 209–213.

[10] E. Kushilevitz, in *Advances in Computers* (Elsevier, New York, 1997), Vol. 44, pp. 331–360.

[11] A. Rao and A. Yehudayoff, *Communication Complexity: And Applications* (Cambridge University Press, Cambridge, England, 2020).

[12] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, Contemp. Math. **305**, 53 (2002).

[13] D. Martínez, A. Tavakoli, M. Casanova, G. Canas, B. Marques, and G. Lima, Phys. Rev. Lett. **121**, 150504 (2018).

[14] G. Brassard, Found. Phys. **33**, 1593 (2003).

[15] H. Buhrman, R. Cleve, and A. Wigderson, in *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing* (Association for Computing Machinery, New York, 1998), pp. 63–68.

[16] H. Buhrman, R. Cleve, S. Massar, and R. de Wolf, Rev. Mod. Phys. **82**, 665 (2010).

[17] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Nature (London) **549**, 195 (2017).

[18] N. Wiebe, D. Braun, and S. Lloyd, Phys. Rev. Lett. **109**, 050505 (2012).

[19] G. H. Low, T. J. Yoder, and I. L. Chuang, Phys. Rev. A **89**, 062315 (2014).

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

[21] P. Rebentrost, M. Mohseni, and S. Lloyd, Phys. Rev. Lett. **113**, 130503 (2014).

[22] Y. LeCun, Y. Bengio, and G. Hinton, Nature (London) **521**, 436 (2015).

[23] U. Hadar, J. Liu, Y. Polyanskiy, and O. Shayevitz, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing* (Association for Computing Machinery, New York, 2019), pp. 792–803.

[24] D. A. Freedman, *Statistical Models: Theory and Practice* (Cambridge University Press, Cambridge, England, 2009).

[25] R. L. Burden, J. D. Faires, and A. M. Burden, *Numerical Analysis* (Cengage Learning, Boston MA, 2015).

[26] A. W. Harrow, A. Hassidim, and S. Lloyd, Phys. Rev. Lett. **103**, 150502 (2009).

[27] D.-B. Zhang, Z.-Y. Xue, S.-L. Zhu, and Z. D. Wang, Phys. Rev. A **99**, 012331 (2019).

[28] M. Schuld, I. Sinayskiy, and F. Petruccione, Phys. Rev. A **94**, 022342 (2016).

[29] G. Wang, Phys. Rev. A **96**, 012335 (2017).

[30] G. Brassard, P. HØyer, and A. Tapp, in *Automata, Languages and Programming*, edited by K. G. Larsen, S. Skyum, and G. Winskel (Springer Berlin Heidelberg, Berlin, Heidelberg, 1998), pp. 820–831.

[31] M. A. Nielsen and I. L. Chuang, Phys. Today **54**, No. 11, 60 (2001).

[32] M. Fanizza, M. Rosati, M. Skotiniotis, J. Calsamiglia, and V. Giovannetti, Phys. Rev. Lett. **124**, 060503 (2020).

[33] A. Anshu, Z. Landau, and Y. Liu, in *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing* (Association for Computing Machinery, New York, 2022), pp. 44–51.

[34] A. Chakrabarti and O. Regev, SIAM J. Comput. **41**, 1299 (2012).

[35] H.-Y. Huang, R. Kueng, and J. Preskill, Nat. Phys. **16**, 1050 (2020).

[36] N. Wiebe, D. W. Berry, P. Høyer, and B. C. Sanders, J. Phys. A **44**, 445308 (2011).

[37] V. Giovannetti, S. Lloyd, and L. Maccone, Phys. Rev. Lett. **100**, 160501 (2008).

[38] X.-M. Zhang, M.-H. Yung, and X. Yuan, Phys. Rev. Res. **3**, 043200 (2021).

[39] J. A. Cortese and T. M. Braje, arXiv:1803.01958.

[40] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevLett.130.150602 for details of the algorithm and complexities' derivations.

[41] H. Abdi and L. J. Williams, Wiley Interdiscip. Rev.: Syst. Biol. Med. **2**, 433 (2010).

[42] P. Indyk and D. Woodruff, in *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings* (IEEE, New York, 2003), pp. 283–288.

[43] A. A. Sherstov, Theor. Comput. Sci. **8**, 197 (2012).

[44] A. Akella, A. Bharambe, M. Reiter, and S. Seshan, in *Proceedings of the Workshop on Management and Processing of Data Streams* (Citeseer, New York, 2003).

[45] A. Chakrabarti, G. Cormode, and A. McGregor, ACM Trans. Algorithms (TALG) **6**, 1 (2010).

[46] IEEE Standard for Floating-Point Arithmetic, in IEEE Std 754-2019 (Revision of IEEE 754-2008) (2019), pp. 1-84, 10.1109/IEEESTD.2019.8766229.

[47] M. Al-Rubaie and J. M. Chang, IEEE Secur. Privacy **17**, 49 (2019).

Supplementary Materials: Communication-efficient Quantum Algorithm for Distributed Machine Learning

Hao Tang,[1,*] Boning Li,[2,3,*] Guoqing Wang,[2,4] Haowei Xu,[4]
Changhao Li,[2] Ariel Barr,[1] Paola Cappellaro,[2,3,4,†] and Ju Li[1,4,‡]

[1]*Department of Materials Science and Engineering, Massachusetts Institute of Technology, MA 02139, USA*
[2]*Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*
[3]*Department of Physics, Massachusetts Institute of Technology, MA 02139, USA*
[4]*Department of Nuclear Science and Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*

I. DETAILS OF THE QUANTUM COUNTING SCHEME

At the core of our algorithm is the estimation of the correlation

$$f = \overline{x^{\mathrm{b}} y^{\mathrm{b}}} = \frac{1}{N}\sum_i x_i^{\mathrm{b}} y_i^{\mathrm{b}} \tag{1}$$

between two bit-strings $\vec{x}^{\mathrm{b}}$ and $\vec{y}^{\mathrm{b}}$, distributed across two parties. Without loss of generality, we assume $N = 2^n$, since for any bit string with arbitrary length N, one can construct a new bit string with $N' = 2^{\lceil \log N \rceil}$ by appending zeros without changing the inner product. At first, Alice prepare an equally weighted n-qubit state: $|\psi_0\rangle_n = (H|0\rangle)^{\otimes n} = \frac{1}{\sqrt{N}}\sum_{i=1}^{N}|i\rangle_n$. Alice then applies the Grover operations to this initial state with an oracle $O_{\vec{x}^{\mathrm{b}},\vec{y}^{\mathrm{b}}}$ realized by the following communication scheme. For a given input state $|\psi\rangle_n = \sum_i c_i|i\rangle_n$, Alice attaches a single qubit $|0\rangle_{o_1}$ to $|\psi\rangle_n$ and applies a unitary transformation $\hat{U}_{x^{\mathrm{b}}}^{n,o_1}$ to qubits n and o_1 according to the bit string x_i^{b}, where $\hat{U}_{x^{\mathrm{b}}}^{n,o_1}|i\rangle_n|s\rangle_{o_1} = |i\rangle_n|s \oplus x_i^{\mathrm{b}}\rangle_{o_1}$. This yields the state:

$$|\psi_1\rangle = \sum_{i=1}^{N} c_i|i\rangle_n|x_i^{\mathrm{b}}\rangle_{o_1}. \tag{2}$$

Alice sends this (n+1)-qubits string to Bob, and Bob appends another qubit $|0\rangle_{o_2}$ and applies the unitary transformation $\hat{U}_{y^{\mathrm{b}}}^{n,o_2} C_Z^{o_1,o_2} \hat{U}_{y^{\mathrm{b}}}^{n,o_2}$ using his bit string y^{b}, where $\hat{U}_{y^{\mathrm{b}}}^{n,o_2}$ acts similarly to $\hat{U}_{x^{\mathrm{b}}}^{n,o_1}$. As $\hat{U}_{y^{\mathrm{b}}}^{n,o_2} C_Z^{o_1,o_2} \hat{U}_{y^{\mathrm{b}}}^{n,o_2}|i\rangle_n|s\rangle_{o_1}|0\rangle_{o_2} = (-1)^{y_i^{\mathrm{b}} \wedge s}|i\rangle_n|s\rangle_{o_1}|0\rangle_{o_2}$, the $|0\rangle_{o_2}$ qubit is decoupled from other qubits both before and after this operation, so the operation can be considered as a unitary transformation acting on $|\psi_1\rangle$:

$$|\psi_2\rangle = \hat{U}_{y^{\mathrm{b}}}^{n,o_2} C_Z^{o_1,o_2} \hat{U}_{y^{\mathrm{b}}}^{n,o_2}|\psi_1\rangle = \sum_{i=1}^{N}(-1)^{x_i^{\mathrm{b}} y_i^{\mathrm{b}}} c_i|i\rangle|x_i^{\mathrm{b}}\rangle. \tag{3}$$

Bob then sends the n+1 qubits $|\psi_2\rangle$ back to Alice. By applying the unitary transformation $\hat{U}_{x^{\mathrm{b}}}$ again, Alice obtains the state $|\psi_3\rangle = \hat{U}_{x^{\mathrm{b}}}^{n,o_1}|\psi_2\rangle = (\sum_{i=1}^{N}(-1)^{x_i^{\mathrm{b}} y_i^{\mathrm{b}}} c_i|i\rangle_n)|0\rangle_{o_1}$, which decouples the attached qubit and realizes the oracle $O_{\vec{x}^{\mathrm{b}},\vec{y}^{\mathrm{b}}}$ that adds a phase of π at position i where $x_i^{\mathrm{b}} y_i^{\mathrm{b}} = 1$. The Grover operation is then completed by Alice through the standard procedure [1]: $|\psi_4\rangle = H^{\otimes n}(2|0\rangle_n\langle 0|_n - I)H^{\otimes n}|\psi_3\rangle$. The overall Grover operator is then $G_{\vec{x}^{\mathrm{b}},\vec{y}^{\mathrm{b}}} = (2|\psi_0\rangle_n\langle\psi_0|_n - I)\hat{O}_{\vec{x}^{\mathrm{b}},\vec{y}^{\mathrm{b}}}$, yielding

$$G_{\vec{x}^{\mathrm{b}},\vec{y}^{\mathrm{b}}}\sum_{i=1}^{N} c_i|i\rangle_n = (2|\psi_0\rangle\langle\psi_0| - I)\sum_{i=1}^{N}(-1)^{x_i^{\mathrm{b}} y_i^{\mathrm{b}}} c_i|i\rangle_n. \tag{4}$$

The state $|0\rangle_{o_1}$ is again omitted as it is decoupled both before and after the transformation. In the quantum counting scheme, the Grover operation acts iteratively on ψ_0. Note that ψ_0 can be decomposed as $|\psi_0\rangle = \sqrt{1-f}|\alpha_n\rangle + \sqrt{f}|\beta\rangle_n$, where $|\beta\rangle_n = \frac{1}{\sqrt{Nf}}\sum_{i,x_i^{\mathrm{b}} y_i^{\mathrm{b}}=1}|i\rangle_n$, $|\alpha\rangle_n = \frac{1}{\sqrt{N(1-f)}}\sum_{i,x_i^{\mathrm{b}} y_i^{\mathrm{b}}=0}|i\rangle_n$, and f is defined as the mean of correlation of x and y in Eq. (1). In the subspace spanned by $|\alpha\rangle_n$ and $|\beta\rangle_n$, $G_{\vec{x}^{\mathrm{b}},\vec{y}^{\mathrm{b}}}$ is then represented as a simple 2D rotation matrix

$$R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{5}$$

with rotation angle

$$\theta = 2\arcsin\sqrt{f}. \tag{6}$$

As R_θ has two eigenvalues $e^{\pm i\theta}$, one can use the quantum phase estimation (QPE) scheme to derive θ and then calculate f.

The QPE scheme is implemented through a set of consecutive controlled Grover operations, as shown in Fig. 1. A t-qubit register is first prepared in an equal superposition state. The quantum state of the overall system is $|\psi^{init}\rangle = \frac{1}{2^{t/2}}(|0\rangle + |1\rangle)^{\otimes t}\left(\sqrt{1-f}|\alpha\rangle_n + \sqrt{f}|\beta\rangle_n\right)$. Alice then applies the controlled Grover operation using the circuit shown in the inset of Fig. 1, where the $U_{x^{\mathrm{b}}}$ and phase gate $2|0\rangle\langle 0| - I$ are replaced by the controlled gates in the Grover search scheme we described above. After-

* These authors contributed equally.
† pcappell@mit.edu
‡ liju@mit.edu

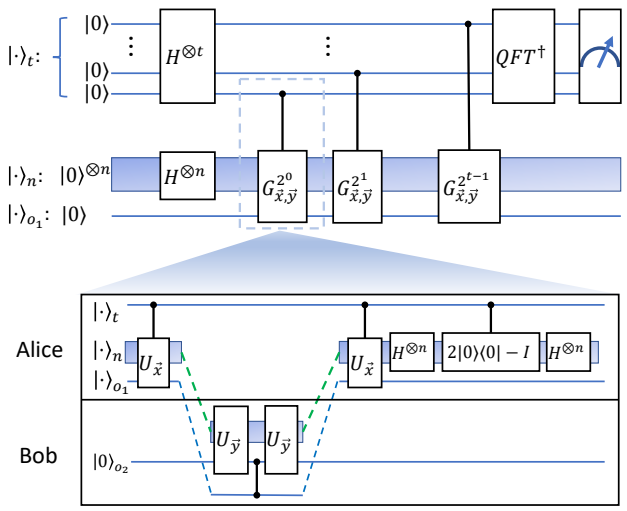

FIG. 1. Quantum circuits for the biparty quantum counting scheme. H, $G_{\vec{x}^{\rm b},\vec{y}^{\rm b}}$, and FT^{+} represent the Hardmard gate, the grover operator, and the inverse QFT, respectively. The t-qubit register is measured after the inverse QFT. The inset shows the biparty scheme of the grover operation, where $U^{1,2}_{\vec{x}^{\rm b}}$ is the phase oracle.

ward, the state becomes

$$|\psi'\rangle = \frac{1}{2^{t/2+1}} \sum_{\tau=0}^{2^t-1} [e^{-i\theta(\tau+1/2)}|\tau\rangle_t(|\alpha\rangle_n + i|\beta\rangle_n) + e^{i\theta(\tau+1/2)}|\tau\rangle_t(|\alpha\rangle_n - i|\beta\rangle_n)]. \quad (7)$$

The inverse quantum Fourier transformation gives

$$|\psi_f\rangle = \frac{1}{2^{t+1}} \sum_{\nu=0}^{2^t-1} \frac{1-e^{-i(2^t\theta+2\pi\nu)}}{1-e^{-i(\theta+2\pi\nu/2^t)}} e^{-i\theta/2}|\nu\rangle_t(|\alpha\rangle_n + i|\beta\rangle_n) + \frac{1-e^{i(2^t\theta-2\pi\nu)}}{1-e^{i(\theta-2\pi\nu/2^t)}} e^{i\theta/2}|\nu\rangle_t(|\alpha\rangle_n - i|\beta\rangle_n). \quad (8)$$

When Alice measures the t-qubit state $|\cdot\rangle_t$, the probability of obtaining $|\nu\rangle_t$ is

$$P_\nu = \frac{1}{2^{2t+1}} \left[\left(\frac{\sin 2^t \frac{X_+}{2}}{\sin \frac{X_+}{2}} \right)^2 + \left(\frac{\sin 2^t \frac{X_-}{2}}{\sin \frac{X_-}{2}} \right)^2 \right], \quad (9)$$

where $X_\pm = \pm\theta + 2\pi\nu/2^t$ and ν is a binary integer $\nu_{t-1}\nu_{t-2}\cdots\nu_0$. Alice's measurement outcome on the state $|\cdot\rangle_t$ is either $\hat{\nu} \simeq \nu_1$ or $\hat{\nu} \simeq \nu_2$, and Alice can obtain $\hat{\theta}$ or $2\pi - \hat{\theta}$ from the result. Considering the range of θ is $[0, \pi]$ according to the definition in Eq. (6), Alice does not need to distinguish these two outcomes. The correlation f is then estimated as:

$$f = \sin^2 \frac{\theta}{2} \simeq \sin^2 (\pi \frac{\hat{\nu}}{2^t}). \quad (10)$$

The distribution of f during a quantum counting measurement for different accuracy parameters t is shown in Fig. 2a.

Up to now, we described the procedure to compute $f = \overline{xy}$, which is only one ingredient to obtain the correlation $\rho_{x,y}$. As $\bar{y}$ can be computed by Bob and sent to Alice with communicate of $O(1)$, the communication complexity to obtain $\hat{\rho} = \frac{\overline{x^{\rm b}y^{\rm b}} - \bar{x^{\rm b}}\bar{y^{\rm b}}}{\sqrt{\bar{x^{\rm b}}(1-\bar{x^{\rm b}})\bar{y^{\rm b}}(1-\bar{y^{\rm b}})}}$ is approximately equal to that to finding f.

II. CORRELATION AND HAMMING DISTANCE ESTIMATION

A. Estimation error and communication complexity

The error in estimating the correlation $f = \overline{xy}$ can be obtained from the distribution (9), which sets the standard deviation of θ is $\Delta\theta = 2^{-(t-1)}$. This error propagates through to the correlation:

$$\Delta f = \frac{df}{d\theta}\Delta\theta = \sqrt{f(1-f)}2^{-(t-1)} \ . \quad (11)$$

We then evaluate the communication complexity. Each Grover operation communicates $2\lceil \log N \rceil + 2$ qubits, and the scheme repeats the Grover operation $2^t - 1$ times. To estimate the correlation, Bob needs to send Alice $\bar{y}$ with t' bits to keep t'-digits accuracy. the overall communication complexity is then $\mathcal{C} = 2(\lceil \log N \rceil + 1)(2^t - 1) + t'$. As the communication complexity of t' does not depend on N and has only logarithmic relation with ϵ, it is small compared with the overall communication complexity $\mathcal{C}$. Therefore, we can assume that t' is sufficiently large without significantly increasing $\mathcal{C}$, so we do not need to consider the rounded error from $\bar{y}$ (and similarly for $\bar{x}$).

The error (standard deviation) of the correlation is then

$$\epsilon_\rho = \frac{\Delta f}{\sqrt{\bar{x}(1-\bar{x})\bar{y}(1-\bar{y})}} = \frac{\sqrt{f(1-f)}2^{-(t-1)}}{\sqrt{\bar{x}(1-\bar{x})\bar{y}(1-\bar{y})}}. \quad (12)$$

Substituting the error into the communication complexity and keeping the leading order, we obtain:

$$\mathcal{C} \simeq \frac{4\sqrt{f(1-f)}}{\sqrt{\bar{x}(1-\bar{x})\bar{y}(1-\bar{y})}} \frac{\log N}{\epsilon_\rho}. \quad (13)$$

The coefficient $\frac{4\sqrt{f(1-f)}}{\sqrt{\bar{x}(1-\bar{x})\bar{y}(1-\bar{y})}}$ does not depend on N and is usually $\sim O(1)$ (as the extreme case when $\bar{x}$ or $\bar{y}$ approximates 1 or 0 is typically not of interests). Thus, the overall communication complexity is $\mathcal{C} = O(\frac{\log N}{\epsilon_\rho})$.

In the scenario of statistical inference, we assume we have independent identically distributed (iid) data points (x_i, y_i), and we can set the number of data points N according to our target accuracy. The overall accuracy of the inferred correlation is the combination of the error from the quantum counting $\epsilon_q \sim 2^{-t}$ (as derived above) and the statistical error ϵ_s from the finite number of

data points. The statistical error is well known to be $\epsilon_s \propto \frac{1}{\sqrt{N}}$, so for an overall target accuracy ϵ, one will restrict both ϵ_q and ϵ_s within $O(\epsilon)$, which means sampling $N \propto \frac{1}{\epsilon^2}$ data points. The communication complexity then becomes $\mathcal{C} = O(\frac{-\log\epsilon}{\epsilon})$. The query complexity (the number of oracle calls) is 4 in each Grover operation, so the overall query complexity is $4(2^t-1) = O(\frac{1}{\epsilon})$. The swap test in communication scenario requires $\log N$-qubit communication for each measurement and requires $O(\frac{1}{\epsilon^2})$ repetitions to get the target accuracy, so the overall communication complexity is $O(\frac{-\log\epsilon}{\epsilon^2})$ substituting $N \propto \frac{1}{\epsilon^2}$ into the expression.

In the typical scenario of the gap-Hamming problem, the length of bit string N is given and the target accuracy of the Hamming distance is $\sqrt{N}$. Then, we require that the estimation of $\frac{1}{N}\sum_i x_i \oplus y_i$ has an error $\epsilon = \frac{1}{\sqrt{N}}$, which gives a communication complexity of $O(\sqrt{N}\log N)$ when substituting ϵ with $1/\sqrt{N}$. Similarly, the query complexity is 4 times of the number of Grover operations, which is $O(\sqrt{N})$.

B. Lower bound

In the previous section we derived the quantum communication complexity based on typical (or worst-case) bit strings $\overline{x}, \overline{y}$. In particular, Eq. (13) shows that when the number of bits N is fixed, the communication complexity is determined by the number of qubits t that stores the correlation result at the end of the algorithm. The number $2^t - 1$ of qubit transmissions cannot be reduced when using the quantum counting algorithm. This number t and the property of the data $\overline{x}, \overline{y}, \overline{xy}$, will determine the standard deviation of the final estimation. Neglecting pathological cases of the data set ($\overline{x}, \overline{y}$ either zero or one), and without prior knowledge of the data set, one needs to assume $\sqrt{\overline{xy}(1-\overline{xy})}$ is on the order of $O(1)$, as stated above. Then the procedure to solve the problem will be: (1) Based on the required ϵ, one calculates t from $t = \log(1/\epsilon)$; (2) Run the quantum counting based algorithm and obtain the final result. This standard procedure achieves the error and communication complexity proved above, Eq. (13,12).

There are however special cases that when the data set is such that $\overline{xy}$ is close to zero or one. In this case, if there's prior knowledge of this property, one only needs $t = \frac{2}{3}\log\frac{1}{\epsilon}$ qubits to store the final result and the communication complexity will be $O(\frac{\log N}{\epsilon^{2/3}})$. Here we show the derivation of the case $\overline{xy}$ is close to 0, while the derivation and result of $\overline{xy} \to 1$ is the same because of the symmetry between $\overline{xy}$ and $1-\overline{xy}$.

Eq. (13) and (12) show that as $\overline{xy}$ is close to zero, the standard deviation of the final result will decrease. However, with maximum t digits precision, the lower bound of the final result is $\overline{xy} < 2^{-t-1}$ in the case that all the first t digits of $\overline{xy}$ are zero. Therefore, with t digits to store the final result, the standard deviation of the correlation can reach the lower bound by replacing $\overline{xy} \approx 2^{-t-1}$, as shown in Eq. (14)

$$\epsilon \geq \frac{2\sqrt{2^{-t-1}(1-2^{-t-1})}}{\sqrt{\overline{x}(1-\overline{x})\overline{y}(1-\overline{y})}} 2^{-t} \sim o(2^{-\frac{3}{2}t}). \tag{14}$$

By Replacing the 2^t term in Eq. (13) with Eq. (14), we obtain the communication complexity for this special case: $\mathcal{C} \sim O(\frac{\log N}{\epsilon^{2/3}})$, which represents the minimum communication complexity that this algorithm can reach.

However, when solving the linear regression problem for float numbers, the assumption that $\overline{xy}$ is close to zero or one is no longer valid since the digits of the binary expansion of a float number can be regarded as uniformly distributed. Therefore, there is not such a lower bound for the linear regression problem as for the correlation of bit strings.

Because the lower bound can only be reached for special cases of the data set (and when prior information about it is available), the communication complexity for the general case is as previously stated. We note that if no prior information is available, one might still attempt to solve the problem with a smaller t and verify that indeed $\overline{xy}$ is small or close to 1.

In the main text, we compared classical algorithms that transfers qubits and quantum algorithms that transfers eithor qubits or classical bits. Here in table. I, we clarify such distinction between different algorithm and the communication channels.

III. DETAILED ANALYSIS OF LEAST-SQUARE FITTING

We first derive the overall communication complexity equation **(7)** in the main text. From equation **(6)** in the main text, the error of λ_j, ϵ_j, is:

$$\epsilon_j^2 = 2^{2(u+v)} \sum_r (r+1)^2 e^{-2r} \epsilon_{jr}^2 \ . \tag{15}$$

The overall communication complexity $\mathcal{C}$ is the summation of $\mathcal{C}_{js}$ for all bit-string inner product estimation:

$$\begin{aligned} \mathcal{C} &= \sum_{jr} C_{jr} \simeq \sum_j \sum_{r=0}^{\infty} 4\sqrt{f_{jr}(1-f_{jr})}\frac{\log N}{\epsilon_{jr}} \\ &\leq \sum_j \sum_{r=0}^{\infty} 2\frac{\log N}{\epsilon_{jr}}. \end{aligned} \tag{16}$$

The optimal strategy for communication is to select positive real numbers ϵ_{jr} subject to the restriction that all ϵ_j in Eq. (15) are smaller than a given error bound ϵ, and minimize Eq. (16). This problem can be solved by the Lagrange multiplier method, yielding

$$\epsilon_{jr} = \epsilon \frac{0.449}{2^{u+v}(r+1)^{2/3}} 2^{\frac{2}{3}r} \ ; \tag{17}$$

TABLE I. Distinction between algorithms and communication channels. *: LOCC (local operation with classical communication) proposed by Ref. [2].

		Algorithm to calculate inner product (correlation) of two bit strings	
		Encode as coefficient	Encode as phase
Quantum algo.	Quantum Comm.	SWAP-test ($O(\log N/\epsilon^2)$)	Quantum counting ($O(\log N/\epsilon)$)
	Classical Comm.	LOCC*(max $\{O(\log N/\epsilon^2), O(\sqrt{N}\log N/\epsilon)\}$)	None
Classical algo.	Classical Comm.	$O(1/\epsilon^2)$ (lower-bound)	

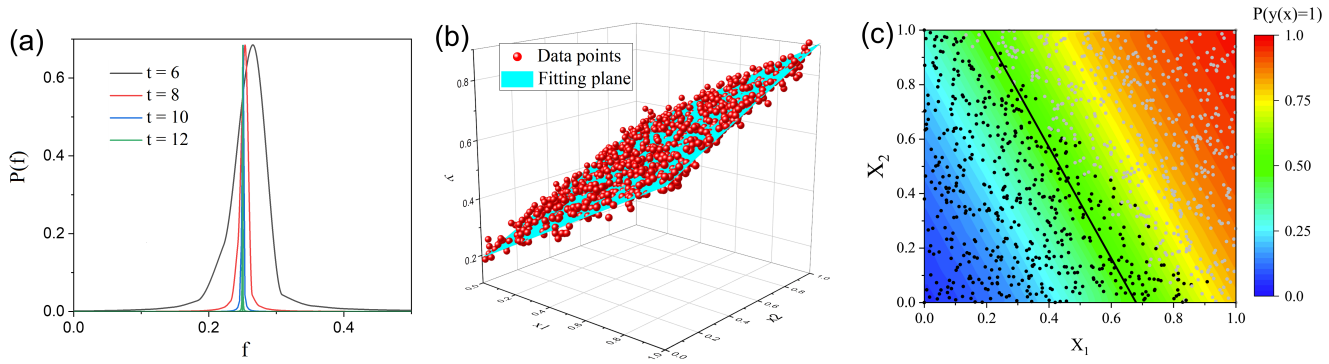

FIG. 2. Illustrative examples of the numerical results of the quantum counting algorithm. (a) Probability distribution function of f from QPE for different values of t. (b) Simulation of multiple linear regression. We set the value of t for the first digits as 12 and $N = 1024$. The data points are generated through the iid distribution $P(X_1, X_2, Y) = P(X_1)P(X_2)P(Y|X_1, X_2)$, where X_1, X_2 follow an uniform distribution $\mathcal{U}(0, 1)$, and $P(Y|X_1, X_2)$ follows a normal distribution $\mathcal{N}(k_1X_1+k_2X_2+b, \epsilon^2\mathbb{I}_{2\times 2})$ with $k_1 = 0.4, k_2 = 0.3, b = 0.2, \epsilon = 0.02$. (c) Numerical simulation of the distributed quantum softmax regression scheme. Raw data points for the logistic regression: the input $\vec{x}$ is a two-component real-number vector, and the output y is a single binary number classifying the data points into two categories 0 (black) and 1 (grey). The fitted classifier that determines the probability $P_{y=1}(\vec{x})$ shown as a contour color plot. The data points are also generated by an iid distribution $P(X_1, X_2, Y) = P(X_1)P(X_2)P(Y|X_1, X_2)$, where X_1, X_2 have the same distribution with (b), and $P(Y = 1|X_1, X_2) = \frac{1}{2}erfc(\frac{k_1X_1+k_2X_2+b}{\epsilon})$ with $k_1 = 0.4, k_2 = 0.3, b = -0.35, \epsilon = 0.02$.

$$\mathcal{C} = 11.026 \times 2^{u+v+1} M \frac{\log N}{\epsilon}. \tag{18}$$

The factor 2^{u+v} is from the fact that the error ϵ is proportional to the magnitude of y and inversely proportional to the magnitude of $\vec{x}$. Without loss of generality, one can normalize all features x_j and y into the range $[0, 1)$. This directly gives $v = -1$ and $0 \le |\vec{x}_i|_\infty < 1$, where $|\cdot|_\infty$ represents the infinity norm of a vector. We then derive the upper bound of 2^u:

$$\begin{aligned} 2^u &= \frac{N}{2}\max_{ij} X^\dagger_{ji} = \frac{1}{2}\max_{ij}[(\frac{1}{N}\mathbf{X}^T\mathbf{X})^{-1}_{jk}X^T_{ki}] \\ &= \frac{1}{2}\max_i |(\frac{1}{N}\mathbf{X}^T\mathbf{X})^{-1}\cdot\vec{x}_i|_\infty \\ &\le \frac{1}{2}\parallel(\frac{1}{N}\mathbf{X}^T\mathbf{X})^{-1}\parallel_\infty \max_i|\vec{x}_i|_\infty \\ &\le \frac{1}{2}\parallel(\frac{1}{N}\mathbf{X}^T\mathbf{X})^{-1}\parallel_\infty = \frac{\kappa}{2\parallel(\frac{1}{N}\mathbf{X}^T\mathbf{X})\parallel_\infty} \\ &= O(\kappa) \end{aligned} \tag{19}$$

where $\|\cdot\|_\infty$ represents the infinity norm of a matrix, and $\kappa \equiv \|\frac{1}{N}\mathbf{X}^T\mathbf{X}\|_\infty \|(\frac{1}{N}\mathbf{X}^T\mathbf{X})^{-1}\|_\infty$ is the condition number of the correlation matrix $\frac{1}{N}\mathbf{X}^T\mathbf{X}$. In the last line, we use the fact that $\|\frac{1}{N}\mathbf{X}^T\mathbf{X}\|_\infty = \max_j \sum_k \overline{x_j x_k} = O(1)$, as the magnitude of feature $\vec{x}$ is normalized. For example, if each component of $\vec{x}$ follows *iid* uniform distribution, then $\|\frac{1}{N}\mathbf{X}^T\mathbf{X}\|_\infty = 1/4$ and $2^u = 2\kappa$. Therefore, the communication complexity

$$\mathcal{C} = 11.026 \times O(\kappa) M \frac{\log N}{\epsilon} = O(\frac{\kappa M \log N}{\epsilon}). \tag{20}$$

For HHL based least square fitting which requires one-time $\log_2(N)$ qubit transfer from Bob to Alice, and then Alice can locally estimate $y' = X^T y$ and $(\mathbf{X}^T\mathbf{X})^{-1}$, and produce a quantum state $|\lambda\rangle = \sum_{j=1}^M \lambda_j|j\rangle$. Here the error ϵ is defined as the distance between this result and the exact solution, which is equivalent with our definition. Such a results requires κ^5 times of repetition, which yields the total number of qubits transfer to be $O(\kappa^5\log_2(N))$. However, when one needs classical numbers as an output, M^2/ϵ^2 times of repeated measurement will increase the communication complexity to $O(M^2\kappa^5\log_2(N))/\epsilon^2$. For classical algorithms solving linear fitting problem, according to the definition of condition number we can obtain the relation between the precision of data y, ϵ_y and the error of each component in $\boldsymbol{\lambda}$, ϵ:

$$\kappa \equiv \frac{\|\delta\boldsymbol{\lambda}\|_\infty}{\|\boldsymbol{\lambda}\|_\infty} / \frac{\|\delta\boldsymbol{y}\|_\infty}{\|\boldsymbol{y}\|_\infty} = \frac{\epsilon}{|\lambda_j|_{\max}\epsilon_t}, \tag{21}$$

where λ_j is the maximum component in $\boldsymbol{\lambda}$. Considering $|\lambda_j|_{\max} \leq |y_{\max}| \parallel (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T \parallel_\infty < \kappa$, the transferred data precision should be $\epsilon_t > \frac{\epsilon}{\kappa^2}$. Therefore, the total bit for single data point y_i is $\log_2(1/\epsilon_t) < \log_2(\kappa^2/\epsilon)$. If the covariance of $\vec{x}$ is ill-conditioned, which means the matrix has a large condition number κ, this will lead to high communication complexity. However, this is from the problem itself regardless of the fitting scheme. In this case, a principle component analysis (PCA) [3] should be conducted locally by Alice: $Ucov(\vec{x},\vec{x})U^T = \mathrm{diag}(d_1, d_2, \cdots, d_{M_x})$. Then, all data points $\vec{x}_i$ are replaced by $\vec{x}'_i = U\vec{x}_i$ before being used for model fitting. As the PCA only involves Alice's information, it can be conducted with arbitrary accuracy without increasing the communication complexity. The components of $\vec{x}'$ corresponding to the eigenvalues d_i close to zero are decoupled from other components. Then, Alice can release the accuracy requirement for these components by setting $\kappa_{\max} = 1/d_{\min}$, where $d_{\min}$ equals the smallest but not close to zero diagonal components.

To provide readers an intuitive example, the quantum counting scheme for linear and softmax regression is numerically simulated. The simulated multiple linear regression is shown in Fig. 2b. The fitted plane goes through the central region of the data points distribution as expected, validating that the scheme performs accurate linear fitting. Similarly, the softmax regression is shown in Fig. 2c, where a classifier is obtained to classify input data points.

IV. DETAILS OF SOFTMAX REGRESSION

The error in estimating Λ in softmax regression is determined by the numerical sensitivity in Eq. (8) in the main text. As the model does not change if we subtract from all $\boldsymbol{\lambda}_j$ an arbitrary vector $\boldsymbol{u}$, we can set $\boldsymbol{\lambda}_q = 0$ without loss of generality. The equations then become

$$\sum_{i=1}^{N} \frac{\boldsymbol{x}_i e^{\boldsymbol{\lambda}_j^T \boldsymbol{x}_i}}{1+\sum_{k=1}^{q-1} e^{\boldsymbol{\lambda}_k^T \boldsymbol{x}_i}} = \sum_{i=1}^{N} 1_{y_i=c_j}\boldsymbol{x}_i, \quad j=1,\ldots,q-1. \tag{22}$$

The error from quantum counting appears on the right-hand side of the equation as ϵ_{jm} (for the mth vector component of the jth equation), and the overall communication complexity is:

$$\mathcal{C} = \frac{2^{u+1}}{(1-2^{-2/3})^2} \sum_{jm} \frac{\log N}{\epsilon_{jm}}, \tag{23}$$

where u is from the expansion of $x_k = 2^u \sum_{l=0}^{\infty} 2^{-l} x_{kl}$. The error of $\sum_{i=1}^{N} 1_{y_i=c_j}$ then propagates to λ_j through Eq. 22. The relation of $\delta\lambda_j$ with ϵ_{jm} can be derived by the Taylor expansion of Eq. 22 around its solution:

$$\begin{aligned} &\partial_{\boldsymbol{\lambda}_k}\left[\sum_{i=1}^{N} \frac{\boldsymbol{x}_i e^{\boldsymbol{\lambda}_j^T \boldsymbol{x}_i}}{1+\sum_{k=1}^{q-1} e^{\boldsymbol{\lambda}_k^T \boldsymbol{x}_i}}\right]\delta\boldsymbol{\lambda}_k \equiv N A_{jm,kn}\delta\boldsymbol{\lambda}_k \\ &= \delta\left(\sum_{i=1}^{N} 1_{y_i=c_j}\boldsymbol{x}_i\right), \end{aligned} \tag{24}$$

where we introduced the matrix $A_{jm,kn}$,

$$\begin{aligned} A_{jm,kn} = \frac{1}{N}\sum_{k=1}^{q-1}\sum_{i} &\frac{x_{i,m}x_{i,n}^T e^{\frac{\boldsymbol{\lambda}_j^T+\boldsymbol{\lambda}_k^T}{2}\boldsymbol{x}_i}}{1+\sum_{l=1}^{q-1} e^{\boldsymbol{\lambda}_l^T\boldsymbol{x}_i}} \\ &\times\left[(1+\sum_{l=1}^{q-1} e^{\boldsymbol{\lambda}_l^T\boldsymbol{x}_i})\delta_{jk} - e^{\frac{\boldsymbol{\lambda}_j^T+\boldsymbol{\lambda}_k^T}{2}\boldsymbol{x}_i}\right], \end{aligned}$$

where $x_{i,m}$ is the mth component of the ith data point $\boldsymbol{x}_i$. Defining $g_{jm} = \frac{1}{N}\sum_{i=1}^{N} 1_{y_i=c_j} x_{i,m}$ and λ_{kn} as the nth component of $\boldsymbol{\lambda}_k$, we have:

$$A_{jm,kn}\delta\lambda_{kn} = \delta g_{jm}. \tag{25}$$

The overall communication complexity with a bounded relative error of the vector $\boldsymbol{g}$ is then obtained through the Lagrange multiplier scheme:

$$\mathcal{C} = \frac{2^{u+1}(qM)^{3/2}}{(1-2^{-2/3})^2|\boldsymbol{g}|}\frac{\log N}{\epsilon_g} \simeq O(\frac{Mq\log N}{\epsilon_g}). \tag{26}$$

In the last step, we utilize that the 2-norm of a qM-dimension vector $\boldsymbol{g}$ has an order of $O((qM)^{1/2})$. As the relative error of $\boldsymbol{\lambda}$ is related to the relative error of $\boldsymbol{g}$ by:

$$\epsilon_\lambda \leq \kappa\epsilon_g, \tag{27}$$

where κ is the condition number of the matrix $|A|$, we then have the communication complexity as a function of the relative error of λ:

$$\mathcal{C} = O(\frac{Mq\kappa\log N}{\epsilon_\lambda}). \tag{28}$$

V. CLASSICAL SHADOWS ALGORITHM FOR GAP HAMMING PROBLEM

The Hamming distance of two distributed bit strings can also be estimated via the classical shadows algorithm. The procedure is as follows:

1. Alice and Bob encode their bit strings x_i and y_i as $|\psi_x\rangle = \frac{1}{\sqrt{N}}\sum_{i=1}^{N}(-1)^{x_i}|i\rangle$ and $|\psi_y\rangle = \frac{1}{\sqrt{N}}\sum_{i=1}^{N}(-1)^{y_i}|i\rangle$, respectively.

2. Alice sends M copies of his states ψ_x to Bob, costing communication complexity of $O(M\log N)$.

3. Now, Bob has a known state $|\psi_y\rangle$ and an unknown state $|\psi_x\rangle$ from Alice. He can use the classical shadows algorithm to estimate their inner product $\langle\psi_x|\psi_y\rangle$ using

$M = O(\frac{1}{\epsilon_{IP}^2})$ copies of $|\psi_x\rangle$, as discussed in [2] and elaborated in [4]. The estimated inner product is:

$$\langle\psi_x|\psi_y\rangle = \frac{1}{N}\sum_{i=1}^{N}(-1)^{x_i+y_i} = \frac{1}{N}\sum_{i=1}^{N}(1-2x_i\oplus y_y), \quad (29)$$

with an error of ϵ_{IP}.

4. The estimated Hamming distance d can then be calculated as:

$$\frac{d}{N} = \frac{1}{N}\sum_{i=1}^{N} x_i \oplus y_y = \frac{1}{2}(1 - \langle\psi_x|\psi_y\rangle) \quad (30)$$

The overall communication complexity of this scheme is $M \log N = O(\frac{\log N}{\epsilon_{IP}^2}) = O(\frac{\log N}{\epsilon^2})$, as we showed in the table in the main text. Here, the ϵ represents the error of d/N, which is $\frac{1}{2}\epsilon_{IP}$, one half of the error of the inner product.

[1] M. A. Nielsen and I. L. Chuang, Phys. Today **54**, 60 (2001).

[2] A. Anshu, Z. Landau, and Y. Liu, in *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing* (2022) pp. 44–51.

[3] H. Abdi and L. J. Williams, Wiley interdisciplinary reviews: computational statistics **2**, 433 (2010).

[4] H.-Y. Huang, R. Kueng, and J. Preskill, Nature Physics **16**, 1050 (2020).