

Quantum-enhanced t-stochastic neighbor embedding for high-dimensional data visualization

Xin Zhang

School of Mathematics and Statistics, Fujian Normal University, Fuzhou 350117, China

Chao-Hua Yu*

School of Computing and Artificial Intelligence, Jiangxi University of Finance and Economics, Nanchang 330032, China

Gong-De Guo

Digital Fujian Internet-of-Things Laboratory of Environmental Monitoring, Fujian Normal University, Fuzhou 350117, China

Song Lin[†]

College of Computer and Cyber Security, Fujian Normal University, Fuzhou 350117, China



(Received 3 September 2025; accepted 6 April 2026; published 8 June 2026)

t-distributed stochastic neighbor embedding (t-SNE) is a nonlinear dimensionality reduction algorithm that preserves local similarity by matching probability distributions between high- and low-dimensional spaces. Owing to its ability to capture complex manifold structures, t-SNE has become a widely used technique for high-dimensional data visualization. However, it still faces significant computational challenges when dealing with large-scale datasets. In this paper, we present a quantum-enhanced t-SNE algorithm that achieves a quadratic speedup with respect to the number of data samples. Specifically, three unitary operators are constructed using quantum arithmetic and comparators to compute the probability distributions. By integrating them with the Hadamard transform, the gradients of the Kullback-Leibler divergence can be efficiently computed. Moreover, since a complete quantum procedure is provided, our method surpasses the performance of existing quantum variants.

DOI: [10.1103/w8b9-7zt9](https://doi.org/10.1103/w8b9-7zt9)

I. INTRODUCTION

Dimensionality reduction is a fundamental task in machine learning that aims to represent high-dimensional data in a lower-dimensional space while preserving essential structures and information [1]. It is commonly employed as a preprocessing step to improve the efficiency of subsequent data analysis. In particular, by mapping data into a low-dimensional (two- or three-dimensional) space, it facilitates visualization [2,3], enabling identification of trends and patterns that are difficult to discern in the original space. Among various dimensionality reduction methods, t-distributed stochastic neighbor embedding (t-SNE) [4] is one of the most widely used for data visualization. It seeks a suitable low-dimensional representation by minimizing the Kullback-Leibler (KL) divergence between the probability distributions in the high- and low-dimensional spaces. Compared to classical methods such as principal component analysis and linear discriminant analysis, t-SNE is more

effective in visualizing complex datasets. However, classical t-SNE has a time complexity of $O(N^2D)$ for a dataset with N samples of D dimensions. This quadratic dependence on sample size makes it scale poorly to large datasets, posing a significant limitation in the era of big data.

Quantum computing leverages the principles of superposition and entanglement in quantum mechanics to achieve significant speedups for certain computational problems [5–8]. These advancements have also motivated researchers to explore the potential of quantum computing for more efficient dimensionality reduction [9–13], including efforts to accelerate t-SNE. In 2022, Kawase *et al.* [14] proposed a parametrized t-SNE algorithm based on quantum neural networks. In this approach, high-dimensional data are input into a quantum neural network (i.e., the parametrized quantum circuits, PQCs), and a low-dimensional representation is obtained by measuring its output. Subsequently, they investigated another mapping approach [15], where the swap test was employed to calculate the inner products between high-dimensional data points, which are then incorporated into a kernel function to obtain the low-dimensional embeddings. While both approaches demonstrate the potential advantages of quantum computing in specific steps of the t-SNE algorithm, the overall time complexity is still largely influenced by the remaining classical steps. As a result, these methods still face challenges when applied to large-scale datasets. Inspired by their valuable contributions, we further investigate how quantum computing can accelerate t-SNE.

*Contact author: quantum.ych@gmail.com

†Contact author: lins95@fjnu.edu.cn

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

In this paper, a quantum t-SNE algorithm with a time complexity of $O(ND\text{polylog}(ND))$ is proposed, which achieves a speedup over its classical counterpart. As mentioned above, the key to t-SNE is to minimize the KL divergence, which requires efficient computation of its gradient. To achieve this, we design three unitary operators and give a gradient computation procedure. Specifically, the unitary operator U_s , constructed based on quantum arithmetic and phase estimation, is used to calculate the similarity between any two samples. The other two operators, U_p and U_q , are designed to compute the probability distributions of the high- and low-dimensional data, respectively. The gradient vector required for optimization is then calculated using these operators. In contrast to previous work [16] that computes gradients using the swap test, our approach employs quantum arithmetic and the Hadamard transform to achieve a more precise extraction of gradient components. Overall, a comprehensive analysis shows that the proposed algorithm provides a fully quantumized framework with explicit time complexity and error bounds, while comparative studies with Barnes-Hut t-SNE and existing quantum t-SNE approaches highlight its theoretical advantages.

The remainder of this paper is organized as follows. In Sec. II, a classical t-SNE algorithm and other related works are reviewed. Section III presents the quantum algorithm for t-SNE, followed by a detailed complexity analysis and comparison in Sec. IV. Finally, conclusions are drawn in Sec. V.

II. RELATED WORK

In this section, we first give an overview of the classical t-SNE with its main procedure and time complexity, and then introduce the existing two quantum t-SNE methods [14,15].

Given a dataset with N samples $\{\mathbf{x}_i\}_{i=1}^N$, where $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iD})^T \in \mathbb{R}^D$ represents a column vector of dimensions D . The objective of t-SNE is to map these data points into a low-dimensional space while maintaining similarity between the points. The map is determined by minimizing the KL divergence between the similarity-based data distribution in the high-dimensional space and that in the low-dimensional space. Suppose that the dataset mapped into the low-dimensional space is denoted as $\{\mathbf{y}_i\}_{i=1}^N$, where $\mathbf{y}_i = (y_{i1}, y_{i2}, \dots, y_{id})^T \in \mathbb{R}^d$ ($d \ll D$) is the data after projection of the i th original data $\mathbf{x}_i$ into the d -dimensional space. It is worth noting that the dimension of the low-dimensional space is usually 2 or 3 for visualization. Specifically, the t-SNE algorithm consists of the following steps:

(1) Calculate the similarity of the data points in high-dimensional space. For the given dataset, calculate the similarity between any two data points $\mathbf{x}_i$ and $\mathbf{x}_j$,

$$s_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma_i^2}\right), \quad (1)$$

where σ_i is the Gaussian variance for data $\mathbf{x}_i$. The value of σ_i is determined from the quantity called perplexity, $\text{Perp}_i = 2^{-\sum_j p_{j|i} \log_2 p_{j|i}}$, which is a user-specified value (typically between 5 and 50), where $p_{j|i}$ is the conditional probability distribution defined by Eq. (3).

(2) Construct a probability distribution P in the high-dimensional space. Compute probability distributions for high-dimensional data

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}, \quad (2)$$

where

$$p_{j|i} = \frac{\exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_k\|^2}{2\sigma_i^2}\right)} \equiv \frac{s_{ij}}{s_i} \quad (3)$$

is called the conditional probability distribution. Similarly, $p_{i|j} = \frac{s_{ji}}{s_j}$. Specifically, to avoid the disturbance of self-similarity for $\mathbf{x}_i$, we set $s_{ii} = 0$ ensuring that $p_{ii} = 0$.

(3) Initialization: Initialize the low-dimensional data randomly.

(4) Construct a probability distribution Q in the low-dimensional space. Unlike the probability distribution for high-dimensional data, the probability distribution for low-dimensional data is determined by t-distribution with one degree of freedom,

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_k \sum_{z \neq k} (1 + \|\mathbf{y}_k - \mathbf{y}_z\|^2)^{-1}}. \quad (4)$$

For convenience, denote $t_{ij} = (1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}$, then $q_{ij} = \frac{t_{ij}}{\sum_k \sum_{z \neq k} t_{kz}}$. In the same way, we set $t_{ii} = 0$ ensuring that $q_{ii} = 0$.

(5) Minimize the KL divergence

$$C = \sum_i \text{KL}(P_i \parallel Q_i) = \sum_i \sum_j p_{ij} \log_2 \frac{p_{ij}}{q_{ij}}. \quad (5)$$

The gradient of the cost function for $\mathbf{y}_i$ is described as

$$\frac{\partial C}{\partial \mathbf{y}_i} = 4 \sum_j (p_{ij} - q_{ij}) t_{ij} (\mathbf{y}_i - \mathbf{y}_j). \quad (6)$$

So, as for the k th ($k = 0, 1, \dots, d-1$) element of $\mathbf{y}_i$, the corresponding gradient of y_{ik} is

$$\frac{\partial C}{\partial y_{ik}} = 4 \sum_j (p_{ij} - q_{ij}) t_{ij} (y_{ik} - y_{jk}). \quad (7)$$

(6) Update the low-dimensional data according to the gradient update rule

$$\mathbf{y}_i(l+1) = \mathbf{y}_i(l) - \eta \frac{\partial C}{\partial \mathbf{y}_i(l)}, \quad i = 1, 2, \dots, N, \quad (8)$$

where $l = 0, 1, 2, \dots$ denotes the iteration number and $\eta > 0$ is the learning rate. Repeat steps 4 and 5 until $l = L$ or the function converges. Finally, we can visualize data by plotting the resulting low-dimensional data.

As for the time consumption of the classical t-SNE algorithm, it is mainly divided into two parts. One is to calculate the similarity and probability distribution of high-dimensional data points, and the other is to calculate the probability distribution of low-dimensional data points in each iteration of the gradient descent optimization. Among them, the similarity computation primarily involves calculating the Euclidean distance, which has a complexity of $O(D)$ [1]. Therefore,

computing pairwise similarities and their corresponding probability distribution for N data points requires $O(N^2D)$ time. Similarly, the time complexity of computing the probability distribution of low-dimensional data points in each iteration is $O(N^2d)$. Since d is usually set as 2 or 3, its impact can be considered negligible. If only one iteration is considered, the time complexity of the classical t-SNE is $O(N^2D)$. To minimize the cost, we need to iteratively calculate the probability distribution of low-dimensional data to update the gradient. If the number of iterations is L , the total complexity of the t-SNE algorithm is $O(N^2(D + L))$ for a dataset with N samples [17].

It is well known that the high computational complexity of t-SNE limits its performance when handling large-scale high-dimensional data. To address this issue, van der Maaten proposed the Barnes-Hut t-SNE algorithm [18,19]. By introducing two key approximation techniques, namely, sparse approximation of high-dimensional affinities and cluster-based approximation of low-dimensional repulsive forces, the algorithm reduces the computational complexity of each gradient iteration to $O(N\log_2 N)$. Specifically, the original t-SNE gradient [i.e., Eq. (6)] is decomposed into an attraction term and a repulsion term, namely,

$$\frac{\partial C}{\partial \mathbf{y}_i} = 4(F_{\text{attr}} - F_{\text{rep}})$$

$$= 4 \left(\sum_j p_{ij} t_{ij} (\mathbf{y}_i - \mathbf{y}_j) - \sum_j q_{ij} t_{ij} (\mathbf{y}_i - \mathbf{y}_j) \right),$$

where F_{attr} and F_{rep} denote the attractive and repulsive force contributions, respectively. When computing the attractive term, Barnes-Hut t-SNE utilizes vantage-point trees [20] to construct a nearest-neighbor structure in high-dimensional space within $O(N\log_2 N)$ time. This results in a sparse similarity matrix P , thereby avoiding the computational overhead of evaluating high-dimensional similarity for all point pairs. For the repulsive term, the algorithm constructs a Barnes-Hut tree structure [21] in the low-dimensional embedding space, such as a quadtree when the embedding dimension is two, and employs the Barnes-Hut condition to determine whether a group of distant points can be approximated by a single centroid. In this manner, exact repulsive force computations over all point pairs are avoided.

By leveraging these two approximation strategies, Barnes-Hut t-SNE can handle larger datasets while maintaining visualization quality. However, it also has two inherent limitations. In particular, Barnes-Hut t-SNE does not provide any error bounds [22]. Moreover, the size of the Barnes-Hut tree grows exponentially with the embedding space dimension, which makes the method primarily practical for two- or three-dimensional embedding spaces [18]. Therefore, to further enhance t-SNE's scalability and visualization efficiency for large-scale data, researchers have attempted to integrate the unique advantages of quantum computing into classical algorithms.

Recently, Kawase *et al.* proposed two quantum t-SNE algorithms: quantum parametric t-SNE and quantum kernel t-SNE. In parametric t-SNE [14], parametrized quantum circuits with θ , similar to classical neural network, are designed

to generate a low-dimensional mapping of high-dimensional data, i.e., $\mathbf{y}_i = f(\mathbf{x}_i|\theta)$. Then, p_{ij} , q_{ij} for all pairs and the cost function C [i.e., Eq. (5)] are computed on a classical computer. Finally, the parameters in PQCs are updated until C is minimized. Although this method achieved better visualization, optimizing the design and parameters of the PQCs requires multiple accesses to a quantum computer [15]. Moreover, it also inevitably falls into barren plateaus and local optima [23], which affects the optimization results.

In order to resolve these issues and realize quantum data visualization, Kawase *et al.* proposed quantum kernel t-SNE [15]. The kernel t-SNE maps $\mathbf{x}_i$ to $\mathbf{y}_i$ via $\mathbf{y}_i = \sum_{j=0}^{N-1} \alpha_j \mathbf{k}(\mathbf{x}_i, \mathbf{x}_j)$, where $\mathbf{k}(*, *)$ is a kernel function. In the quantum algorithm, the kernel is defined by the fidelity of quantum states representing data, which is calculated by the swap test circuit on a quantum computer. Then, the parameters $\alpha_i \in \mathbb{R}^d$ are determined by minimizing the cost function C , which is performed entirely on a classical computer.

In summary, these two approaches primarily apply quantum computing to certain steps of t-SNE, but they do not fully address the overall computational challenges of the algorithm. As a result, they still encounter limitations when dealing with large-scale datasets. In the following, we propose a quantum algorithm that achieves a more comprehensive quantization of t-SNE, addressing these challenges more effectively.

III. A QUANTUM ALGORITHM FOR VISUALIZING

In this section, a quantum version of the t-SNE algorithm is proposed. According to the analysis in Sec. II, it is necessary to minimize the cost function shown in Eq. (5) to obtain the data representation in low-dimensional space. That is, Eq. (7) needs to be calculated. To accomplish this task, we design a quantum algorithm with the following steps. First, we construct the operator U_s to calculate the similarity of any two points. Then, the operators U_p and U_q are designed to calculate the probability distribution of the high-dimensional data and that of the low-dimensional data, p_{ij} and q_{ij} , respectively. Finally, the gradient can be obtained with these operators. An outline of our proposed algorithm is shown in Fig. 1, and its specific quantum implementation is described in detail in the following four sections.

In our algorithm, we assume that the classical data are loaded to quantum settings via the following quantum oracles:

$$\begin{aligned} O_X : |i\rangle|k\rangle|0\rangle &\rightarrow |i\rangle|k\rangle|x_{ik}\rangle, \\ O_Y : |i\rangle|k\rangle|0\rangle &\rightarrow |i\rangle|k\rangle|y_{ik}\rangle, \\ O_\sigma : |i\rangle|0\rangle &\rightarrow |i\rangle|\sigma_i\rangle, \end{aligned} \quad (9)$$

where $x_{ik}(y_{ik})$ means the k th element of the i th sample in the dataset $\{\mathbf{x}_i\}_{i=1}^N (\{\mathbf{y}_i\}_{i=1}^N)$ and σ_i represents the Gaussian variance corresponding to each sample. Without loss of generality, we assume that each eigenvalue x_{ik} of the data point $\mathbf{x}_i$ is normalized to $[0, 1]$ and represented using h bits of precision. The oracles can be achieved via the quantum random access memory (QRAM) [24–26], so that these three oracles can access the elements of $\mathbf{x}_i$, $\mathbf{y}_i$, and σ in times $O(\text{polylog}(ND))$, $O(\text{polylog}(Nd))$, and $O(\text{polylog}N)$, respectively.

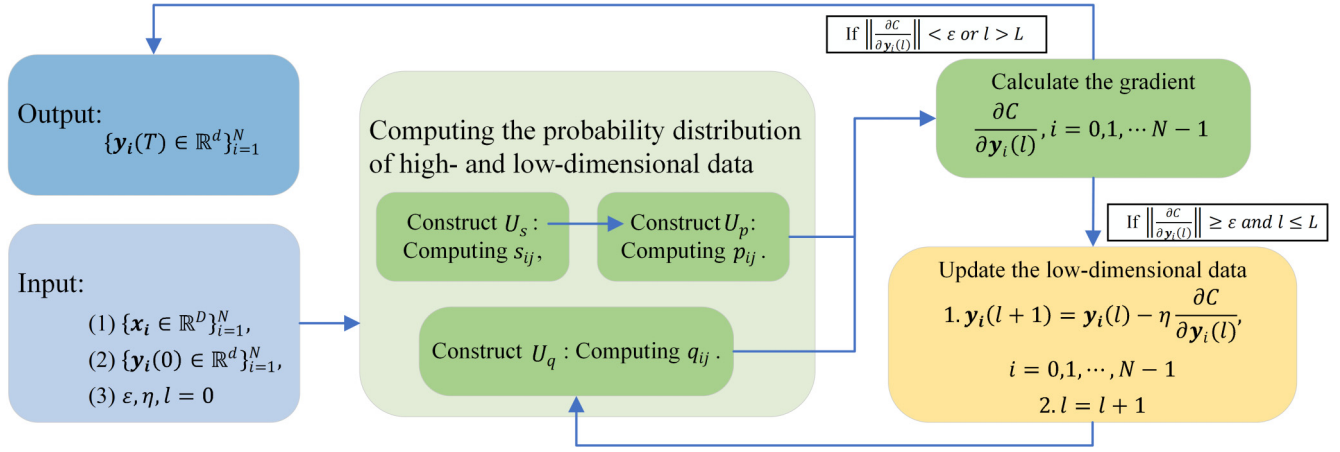


FIG. 1. The whole process of the proposed algorithm, where $\{x_i \in \mathbb{R}^D\}_{i=1}^N$ is the dataset, $\{y_i(0) \in \mathbb{R}^d\}_{i=1}^N$ is the initial low-dimensional data, ϵ is the precision, η is the learning rate, and t is the iteration number.

A. Construction of U_s

To obtain a low-dimensional representation of high-dimensional data, t-SNE approximates the probability distribution of the data in high- and low-dimensional space. Therefore, the first step in our quantum algorithm is to efficiently compute the high-dimensional probability distribution p_{ij} , which is based on pairwise similarities s_{ij} between data points. To achieve this, it is essential to construct the unitary operator that implements

$$U_s : |i\rangle|j\rangle|0\rangle \rightarrow |i\rangle|j\rangle|s_{ij}\rangle, \quad (10)$$

as depicted in Fig. 2. More details can be expressed by the following steps.

Step (1.1): Prepare the initial state $|i\rangle_1|j\rangle_2|0\rangle_3^{\otimes \log_2 D}|0\rangle_4^{\otimes h}|0\rangle_5^{\otimes h}$, where the subscript numbers represent the index of registers and h is the number of qubits required to store the data information. Then, perform the Hadamard gate on each qubit in register 3 to yield $|i\rangle_1|j\rangle_2 \sum_{k=0}^{D-1} \frac{1}{\sqrt{D}} |k\rangle_3 |0\rangle_4^{\otimes h} |0\rangle_5^{\otimes h}$. After that, O_X is executed, respectively, on registers 1, 3, 4 and registers 2, 3, 5 to access the dataset to get

$$|i\rangle_1|j\rangle_2 \sum_{k=0}^{D-1} \frac{1}{\sqrt{D}} |k\rangle_3 |x_{ik}\rangle_4 |x_{jk}\rangle_5. \quad (11)$$

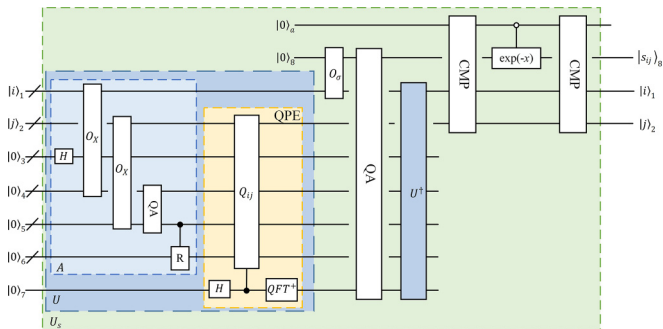


FIG. 2. The quantum circuit of computing the similarity between samples. QA denotes the quantum arithmetic, CMP is the comparator, and $\exp(-x)$ stands for the $\exp(-x)$ gate.

Step (1.2): Quantum arithmetic [27] is performed on registers 4 and 5 to get the state

$$|i\rangle_1|j\rangle_2 \sum_{k=0}^{D-1} \frac{1}{\sqrt{D}} |k\rangle_3 |x_{ik}\rangle_4 |x_{jk}\rangle_5. \quad (12)$$

Step (1.3): Add one qubit and rotate it from $|0\rangle_6$ to $c_1(x_{ik} - x_{jk})|1\rangle_6 + \sqrt{1 - c_1^2(x_{ik} - x_{jk})^2}|0\rangle_6$ controlled on $|x_{ik} - x_{jk}\rangle_5$, where $c_1 = (\max_k |x_{ik} - x_{jk}|)^{-1}$. At this point, the system state is denoted as $|i\rangle_1|j\rangle_2|\varphi_{ij}\rangle_{3-6}$, where

$$|\varphi_{ij}\rangle_{3-6} = \sum_{k=0}^{D-1} \frac{1}{\sqrt{D}} |k\rangle_3 |x_{ik}\rangle_4 |x_{jk}\rangle_5 \otimes (c_1(x_{ik} - x_{jk})|1\rangle_6 + \sqrt{1 - c_1^2(x_{ik} - x_{jk})^2}|0\rangle_6).$$

Denote the process from step (1.1) to step (1.3) as A ,

$$A : |i\rangle_1|j\rangle_2|0\rangle_{3-6} \rightarrow |i\rangle_1|j\rangle_2|\varphi_{ij}\rangle_{3-6}.$$

If we only consider the subsystem, i.e., registers 3–6, the unitary operation is denoted as A_{ij} and acts as follows:

$$A_{ij}|0\rangle_{3-6} = |\varphi_{ij}\rangle_{3-6} = \sin \theta_{ij} |\varphi_{ij}^1\rangle_{3-6} + \cos \theta_{ij} |\varphi_{ij}^0\rangle_{3-6}.$$

It is easy to verify $\sin^2 \theta_{ij} = \frac{\sum_k c_1^2(x_{ik} - x_{jk})^2}{D} = \frac{c_1^2 \|x_i - x_j\|^2}{D}$. That is, if we can estimate $\sin^2 \theta_{ij}$ successfully, we can derive the Euclidean distance between samples x_i and x_j . According to quantum amplitude estimation, the operator $Q = -AR_{3-6}AR_6$ can be defined, where the subscript denotes the register on which R acts and $R_{3-6} = I - 2|0\rangle_{3-6}\langle 0|_{3-6}$, $R_6 = I - 2|0\rangle_6\langle 0|_6$. When considering its role in registers 3–6, it is equivalent to $Q_{ij} = -A_{ij}R_{3-6}A_{ij}R_6$.

Step (1.4): Add a multiqubit register $|0\rangle_7$ and perform amplitude estimation [28,29] to yield

$$|i\rangle_1|j\rangle_2|\varphi_{ij}\rangle_{3-6} \|\mathbf{x}_i - \mathbf{x}_j\|^2_{7-8}. \quad (13)$$

It is important to note that $\|\mathbf{x}_i - \mathbf{x}_j\|^2 = 0$ when $i = j$.

Step (1.5): A multiqubit register $|0\rangle_8$ is added, then O_σ is performed on registers 1 and 8 to prepare σ . After that, quantum arithmetic [30] is performed on registers 7 and 8 to

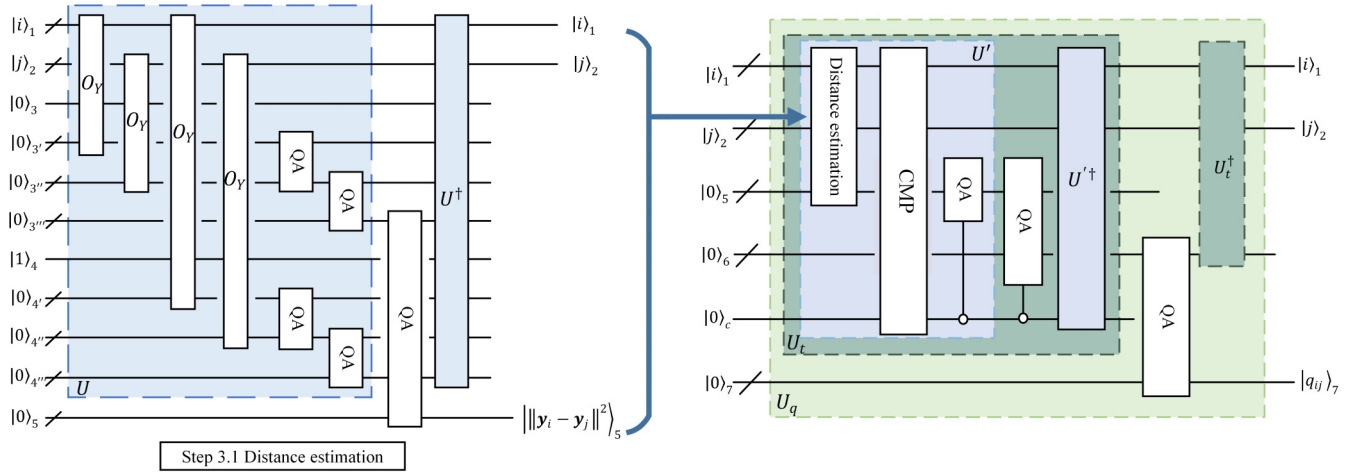


FIG. 4. The circuit for unitary operator U_q . The circuit diagram on the left side of the figure is used to estimate distance $\|\mathbf{y}_i - \mathbf{y}_j\|^2$ for $d = 2$.

Perform the inverse processes of steps (2.1) and (2.2), and discard registers 2, 3, b , giving the state $|i\rangle_1|s_i\rangle_4$. That is, we have achieved the unitary operation U_{s+} .

Step (2.4): Initialize the state as $|i\rangle_1|j\rangle_2|0\rangle_3|0\rangle_4|0\rangle_5|0\rangle_6$, and then perform the operator U_s and $\text{swap}_{1,2}U_s\text{swap}_{1,2}$ to prepare $|s_{ij}\rangle$ and $|s_{ji}\rangle$, respectively. Here, $\text{swap}_{1,2}$ means to exchange the contents of registers 1 and 2. Next, U_{s+} are performed to prepare $|s_i\rangle$ and $|s_j\rangle$. The system state becomes

$$|i\rangle_1|j\rangle_2|s_{ij}\rangle_3|s_i\rangle_4|s_{ji}\rangle_5|s_j\rangle_6. \quad (22)$$

Step (2.5): Perform quantum arithmetic [30]. First, quantum division is performed on registers 3 and 4 and registers 5 and 6, respectively, and then the data in registers 4 and 6 are added together. After that, the quantum state becomes

$$|i\rangle_1|j\rangle_2|s_{ij}\rangle_3|s_i\rangle_4|s_{ji}\rangle_5|p_{ij}\rangle_6, \quad (23)$$

where $p_{ij} = \frac{1}{2N}(\frac{s_{ij}}{s_i} + \frac{s_{ji}}{s_j})$. Finally, perform the inverse operations of step (2.4), with the registers 3–5 rolling back to the initial state and discarded. The final state is

$$|i\rangle_1|j\rangle_2|p_{ij}\rangle_6. \quad (24)$$

That is, we have achieved the unitary operation U_p .

C. Construction of U_q

The key of the t-SNE algorithm is to make the probability distribution q_{ij} of the data in the low-dimensional space approach to the probability distribution p_{ij} of the high-dimensional data. In this section, we will construct the unitary operator U_q that achieves

$$U_q : |i\rangle|j\rangle|0\rangle \rightarrow |i\rangle|j\rangle|q_{ij}\rangle, \quad (25)$$

as depicted in Fig. 4. The specific operation is described below.

Step (3.1): Estimate Euclidean distances in low-dimensional data. Unlike high-dimensional data, the dimensions of low-dimensional data are 2 or 3, which can be easily estimated using quantum addition and multiplication operations [33]. As shown in Fig. 4, as long as the low-dimensional data are accessed via O_Y , it is

not difficult to obtain the quantum state

$$|i\rangle_1|j\rangle_2\|\mathbf{y}_i - \mathbf{y}_j\|^2\rangle_5. \quad (26)$$

Step (3.2): Add an ancilla qubit $|0\rangle_c$, and perform the comparator [31] to generate the state

$$|i\rangle_1|j\rangle_2\|\mathbf{y}_i - \mathbf{y}_j\|^2\rangle_5|\delta_{ij}\rangle_c, \quad (27)$$

where δ_{ij} is the Kronecker delta. Then, perform quantum addition controlled by δ_{ij} to obtain t_{ij} . Specifically, if the ancilla qubit is $|0\rangle_c$, add the data in register 5 by 1; otherwise, leave it unchanged. That is, the state becomes

$$\begin{aligned} |i\rangle_1|j\rangle_2|1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2\rangle_5|0\rangle_c, & \quad i \neq j, \\ |i\rangle_1|j\rangle_2|0\rangle_5|1\rangle_c, & \quad i = j. \end{aligned} \quad (28)$$

Step (3.3): Append a multiqubit register $|0\rangle_6$ and perform controlled quantum arithmetic to obtain t_{ij} . When $\delta_{ij} = 0$, quantum arithmetic [34] is performed in registers 5 and 6; otherwise, no operation is applied. Then, the state becomes

$$\begin{aligned} |i\rangle_1|j\rangle_2|1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2\rangle_5|0\rangle_c|t_{ij}\rangle_6, & \quad i \neq j, \\ |i\rangle_1|j\rangle_2|0\rangle_5|1\rangle_c|0\rangle_6, & \quad i = j. \end{aligned} \quad (29)$$

Then, the inverse operators of steps (3.1) and (3.2) are performed to get $|i\rangle_1|j\rangle_2|t_{ij}\rangle_6$.

Steps (3.1)–(3.3) amount to an unitary operation that acts as

$$U_t : |i\rangle_1|j\rangle_2|0\rangle_6 \rightarrow |i\rangle_1|j\rangle_2|t_{ij}\rangle_6. \quad (30)$$

Step (3.4): Prepare the state as $\frac{1}{N} \sum_{i,j=0}^{N-1} |i\rangle_1|j\rangle_2|t_{ij}\rangle_6|0\rangle_d$ with the operator U_t , and then rotate $|0\rangle_d$ to $\sqrt{t_{ij}}|1\rangle_d + \sqrt{1-t_{ij}}|0\rangle_d$. The state becomes

$$\frac{1}{N} \sum_{i,j=0}^{N-1} |i\rangle_1|j\rangle_2|t_{ij}\rangle_6(\sqrt{t_{ij}}|1\rangle_d + \sqrt{1-t_{ij}}|0\rangle_d). \quad (31)$$

Then, measure the ancillary qubit to yield $|1\rangle_d$; the probability is $\frac{\sum_{i,j=0}^{N-1} t_{ij}}{N^2}$. Namely, by this step we can estimate $T = \sum_{i,j=0}^{N-1} t_{ij}$ for the subsequent calculation.

Step (3.5): Prepare the state as $|i\rangle_1|j\rangle_2|t_{ij}\rangle_3|0\rangle_4$ with the operator U_t , and then perform quantum arithmetic [30] on

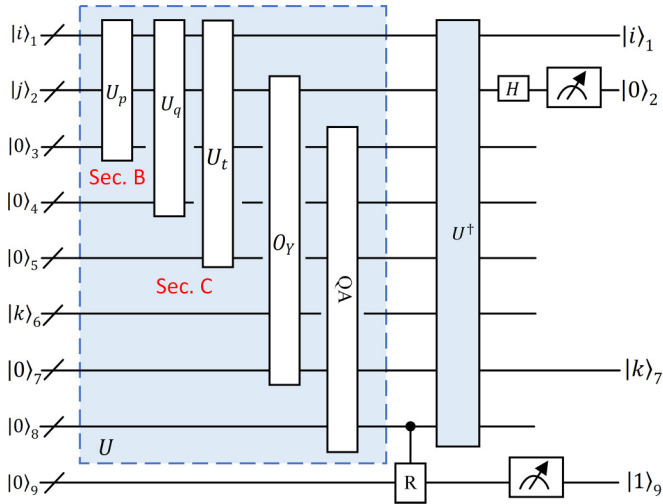


FIG. 5. Compute the gradient. U_p , U_q , and U_t are the unitary operators designed in Secs. III B and III C, respectively.

registers 3 and 4, which yields the quantum state

$$|i\rangle_1 |j\rangle_2 |t_{ij}\rangle_3 |q_{ij}\rangle_4, \quad (32)$$

where $q_{ij} = t_{ij}/T$ is the probability distribution of the data in low-dimensional space. Finally, perform $U_t^\dagger$ to roll back register 3 as $|0\rangle$ and discard it to obtain the state as Eq. (25). In other words, the operation U_q has been successfully implemented.

D. Computation of gradients

After calculating the probability distributions for the high- and low-dimensional data, we will give the procedure for calculating the gradient. It is worth noting that in order to determine the next direction of optimization for low-dimensional data, we need to know the sign of each gradient component. To achieve this, we utilize the technique of amplitude product sign determination [35], which is summarized in the following lemma.

Lemma 1 (Amplitude product sign determination [35]). Given a normalized single-qubit state $|\xi\rangle = a|0\rangle + b|1\rangle$, where $a > 0$, $ab \neq 0$ and $a^2 + b^2 = 1$. There is a procedure to determine the sign of b with a probability greater than or equal to $1 - \tau$ using $\frac{1}{\tau}(\frac{1}{a^2} + \frac{1}{b^2})$ copies of $|\xi\rangle$.

The full steps to calculate the gradient are shown in Fig. 5. In the following, we provide a detailed description.

Step (4.1): Prepare the state $|i\rangle_1 \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle_2 |0\rangle_3 |0\rangle_4 |0\rangle_5$, and then perform the unitary operators U_p , U_q , and U_t to generate

$$|i\rangle_1 \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle_2 |p_{ij}\rangle_3 |q_{ij}\rangle_4 |t_{ij}\rangle_5. \quad (33)$$

Step (4.2): Append additional multiqubit registers $|k\rangle_6 |0\rangle_7 |0\rangle_8$, where $k = 0, \dots, d-1$ is the index of dimension d . Then, perform O_Y on registers 2, 6, and 7 to get $|y_{jk}\rangle_7$. Finally, quantum arithmetic [36] is performed

on register 7 to obtain the quantum state

$$|i\rangle_1 \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle_2 |p_{ij}\rangle_3 |q_{ij}\rangle_4 |t_{ij}\rangle_5 |k\rangle_6 |y_{ik} - y_{jk}\rangle_7 |0\rangle_8. \quad (34)$$

Step (4.3): Perform quantum addition and multiplication [33] to obtain the quantum state

$$|i\rangle_1 |k\rangle_6 \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle_2 |p_{ij}\rangle_3 |q_{ij}\rangle_4 |t_{ij}\rangle_5 |y_{ik} - y_{jk}\rangle_7 |r_{ijk}\rangle_8, \quad (35)$$

where $r_{ijk} = (p_{ij} - q_{ij})t_{ij}(y_{ik} - y_{jk})$. Obviously, $\frac{\partial C}{\partial y_{ik}} = 4 \sum_j r_{ijk}$.

Step (4.4): Add an auxiliary qubit $|0\rangle_9$ and rotate it to $c_2 r_{ijk} |1\rangle_9 + \sqrt{1 - c_2^2 r_{ijk}^2} |0\rangle_9$ controlled on register 8, where $c_2 = (\max_{ijk} |r_{ijk}|)^{-1}$. Then, undoing the processes of steps (4.1)–(4.3) makes the registers 3–5, 7, and 8 return to their initial states and discards them. We obtain the following state:

$$|i\rangle_1 |k\rangle_6 \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle_2 (c_2 r_{ijk} |1\rangle_9 + \sqrt{1 - c_2^2 r_{ijk}^2} |0\rangle_9). \quad (36)$$

Step (4.5): Perform the gate $H^{\otimes \log N}$ on register 2 to derive

$$|i\rangle_1 |k\rangle_6 \frac{1}{\sqrt{N}} \sum_{l=0}^{N-1} \sum_{j=0}^{N-1} \frac{(-1)^{j \cdot l}}{\sqrt{N}} |l\rangle_2 \otimes (c_2 r_{ijk} |1\rangle_9 + \sqrt{1 - c_2^2 r_{ijk}^2} |0\rangle_9). \quad (37)$$

For any i and k , measure register 2 to get $|0\rangle_2$ with probability $P(0) = \frac{N_r}{N^2}$, where $N_r = (\sum_{j=0}^{N-1} c_2 r_{ijk})^2 + (\sum_{j=0}^{N-1} \sqrt{1 - c_2^2 r_{ijk}^2})^2$. Then, the final state of the last register is

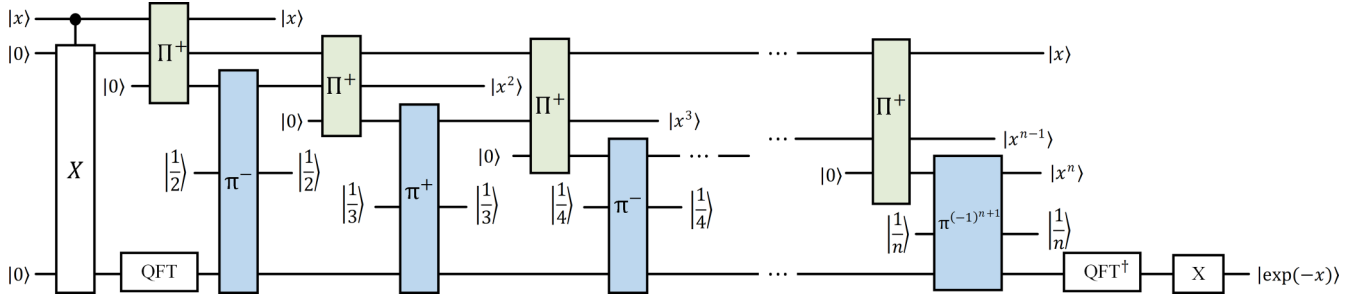
$$\frac{1}{\sqrt{N_r}} \left(\sum_{j=0}^{N-1} c_2 r_{ijk} |1\rangle_9 + \sum_{j=0}^{N-1} \sqrt{1 - c_2^2 r_{ijk}^2} |0\rangle_9 \right). \quad (38)$$

At this time, measure the last register to get $|1\rangle_9$; the probability is $P(1) = \frac{(\sum_{j=0}^{N-1} c_2 r_{ijk})^2}{N_r}$. Obviously, an approximation of the absolute value of the gradient is obtained, i.e., $|\frac{\partial C}{\partial y_{ik}}| = 4 \sum_j r_{ijk} = 4 \frac{\sqrt{N_r}}{c_2} \times \sqrt{P(1)}$. Furthermore, from Eq. (38), we observe that the amplitude of $|0\rangle_9$ is strictly positive. That is, the sign of the amplitude of $|1\rangle_9$ (i.e., the sign of the gradient) can be inferred by the sign of the product of these amplitudes. Therefore, according to Lemma 1, the sign of the gradient can be estimated.

In conclusion, the proposed approach can implement the key steps of t-SNE on a quantum computer, yielding a low-dimensional mapping that preserves the essential characteristics of the original data through optimization. To illustrate the advantages of our proposed method, a detailed time complexity analysis will be given in Sec. IV.

IV. COMPLEXITY ANALYSIS

In this section, we present a detailed analysis of the proposed algorithm. First, each component of the algorithm is

FIG. 6. The circuit of $f(x) = \exp(-x)$.

analyzed to derive the overall time complexity. Then, a comparative analysis is conducted against the classical t-SNE algorithm and two existing quantum approaches. The results demonstrate the advantages of our method in several aspects.

A. Analysis for each step of the proposed algorithm

The first step is to evaluate the time complexity of the operator U_s . In step (1.1), the complexity is $O(\text{polylog}(ND))$ that comes from preparing the required data by Hadamard gates and O_X . Here, assume that $O(h)$ qubits are sufficient to accurately encode $(x_{ik} - x_{jk})$, and the time complexities of quantum arithmetic [36] and controlled rotation [37] are $O(\text{poly}(h))$. Moreover, when the precision of quantum arithmetic is $\epsilon_h = 2^{-h}$, the time complexity of step (1.2) is $O(\text{polylog}(\frac{1}{\epsilon_h}))$. Then, we consider the complexity of step (1.4). According to Ref. [34], the time complexity of the amplitude estimation is related to the Grover operator Q_{ij} and the accuracy ϵ_a . Specifically, the complexity of building the Grover operator Q_{ij} consists mainly of steps (1.1)–(1.3), namely, $O(\text{polylog}(ND) + \text{poly}(h))$. Let ϵ_a be the error in estimating the amplitude. Since the controlled operator Q_{ij} needs to perform $1/\epsilon_a$ times, the complexity of step (1.4) will be $O(\frac{\text{polylog}(ND) + \text{poly}(h)}{\epsilon_a})$. The error in estimating the Euclidean distance is

$$\begin{aligned}
 \Delta \|x_i - x_j\|^2 &= |\widetilde{\|x_i - x_j\|^2} - \|x_i - x_j\|^2| \\
 &= |\widetilde{\|x_i - x_j\|^2} - \widehat{\|x_i - x_j\|^2} + \widehat{\|x_i - x_j\|^2} - \|x_i - x_j\|^2| \\
 &= \left| \frac{D}{c_1^2} (\sin^2 \tilde{\theta}_{ij} - \sin^2 \hat{\theta}_{ij}) + \sum_k (\widehat{x_{ik}} - \widehat{x_{jk}})^2 - \sum_k (x_{ik} - x_{jk})^2 \right| \\
 &\leq \frac{D}{c_1^2} |\sin 2\zeta_{ij} (\tilde{\theta}_{ij} - \hat{\theta}_{ij})| + \sum_k |\widehat{(x_{ik} - x_{jk})^2} - (x_{ik} - x_{jk})^2| \\
 &\leq \frac{D}{c_1^2} \epsilon_a + 2D\epsilon_h,
 \end{aligned} \tag{39}$$

where $\Delta*$ means the error of $*$, and $\widetilde{\|x_i - x_j\|^2}$ and $\widehat{\|x_i - x_j\|^2}$ denote the estimates of $\|x_i - x_j\|^2$ obtained via amplitude estimation and quantum arithmetic, respectively. ζ_{ij} between $\tilde{\theta}_{ij}$

and θ_{ij} and the third equation follow from Lagrangian median theorem.

In step (1.5), the complexity of performing O_σ is $O(\text{polylog } N)$, the time complexity of quantum arithmetic is $O(\text{poly}(h))$, and the time complexity of the inverse operation is the same as in steps (1.1)–(1.4). The comparator in step (1.6) and its inverse operation are executed in $O(\text{polylog } N)$. In step (1.7), based on the analysis in Ref. [36] and the quantum circuit for the exp gate shown in Fig. 6, the time complexity of the exp gate can be determined as $O(\text{poly } h)$. Then, we analyze that the error in calculating s_{ij} is

$$\begin{aligned}
 \Delta s_{ij} &= |\tilde{s}_{ij} - s_{ij}| \\
 &= \left| \exp\left(-\frac{\widetilde{\|x_i - x_j\|^2}}{2\sigma_i^2}\right) - \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma_i^2}\right) \right| \\
 &= |\exp(-\xi_{ij})| \times \frac{1}{2\sigma_i^2} \times \Delta \|x_i - x_j\|^2 \\
 &\leq \frac{1}{2\sigma_{\min}^2} \times \left(\frac{D}{c_1^2} \epsilon_a + 2D\epsilon_h \right),
 \end{aligned} \tag{40}$$

where ξ_{ij} between $\frac{\widetilde{\|x_i - x_j\|^2}}{2\sigma_i^2}$ and $\frac{\|x_i - x_j\|^2}{2\sigma_i^2}$, and $\sigma_{\min}$ denotes the minimum value among all $\{\sigma_i\}$.

Therefore, the total time complexity to construct U_s is $O(\frac{\text{polylog}(ND) + \text{poly}(h)}{\epsilon_a})$. To ensure that the overall error in computing s_{ij} is bounded by ϵ (i.e., $\Delta s_{ij} = \epsilon$), and if the error introduced by amplitude estimation and quantum arithmetic are $\epsilon/2$, we can get $\epsilon_a = \frac{\sigma_{\min}^2 c_1^2}{D} \epsilon = O(\epsilon/D)$ and $\epsilon_h = \frac{\sigma_{\min}^2}{2D} \epsilon = O(\epsilon/D)$. In this case, the cost of quantum arithmetic and controlled rotation is $O(\text{poly}(h)) = O(\text{polylog}(D/\epsilon))$. Consequently, the time complexity of the unitary operation to calculate the similarity between samples is $O(\frac{D(\text{polylog}(ND) + \text{polylog}(D/\epsilon))}{\epsilon})$.

The second step is to analyze the time complexity of preparing the state $|p_{ij}\rangle$. As described in the above analysis, the time complexity of step (2.1) is $O(\frac{D(\text{polylog}(ND) + \text{poly}(h))}{\epsilon})$. The time complexity of the controlled rotation in step (2.2) is $O(h)$. In step (2.3), there are two manners to compute $|s_i\rangle$. In the first approach, assuming that the measurement error is ϵ (i.e., $|\sum_j \tilde{s}_{ij} - \sum_j s_{ij}| = \epsilon$, $\sum_j \tilde{s}_{ij}$ is obtained by the measurement), then $O(1/\epsilon^2)$ measurements are required to obtain the classical data. So, we can load the data $|s_i\rangle$ into the quantum state in the following step within time $O(\log N)$. The second is to use the amplitude estimation. Similarly to

the previous analysis of step (1.4), the time complexity of this step is primarily determined by the time required to prepare the quantum state given in Eq. (20). Let the precision of estimating θ_i be ϵ (i.e., $|\hat{\theta}_i - \theta_i| = \epsilon$, $\hat{\theta}_i$ is obtained by phase estimation), then the time complexity of this step is $O(\frac{D(\text{polylog}(ND) + \text{poly}(h))}{\epsilon^2})$. Regardless of which method is used, the error of s_i can be calculated:

$$\begin{aligned} \Delta \frac{s_i}{N} &= \frac{1}{N} \times \left| \sum_j \widetilde{s}_{ij} - \sum_j s_{ij} \right| \\ &= \frac{1}{N} \times \left| \left(\sum_j \widetilde{s}_{ij} - \sum_j \widetilde{s}_{ij} \right) + \left(\sum_j \widetilde{s}_{ij} - \sum_j s_{ij} \right) \right| \\ &\leq \frac{1}{N} \times \left(N |\sin 2\theta_i \times (\hat{\theta}_i - \theta_i)| + \sum_j |\widetilde{s}_{ij} - s_{ij}| \right) \\ &\leq \frac{1}{N} \times (N\epsilon + N\epsilon) = 2\epsilon. \end{aligned} \quad (41)$$

Then, we get that the time of U_{s+} is $O(\frac{D(\text{polylog}(ND) + \text{poly}(h))}{\epsilon^2})$. Therefore, the time complexity of step (2.4) is the time to perform U_s and U_{s+} . In step (2.5), the time for quantum arithmetic is less than step (2.4), which can be ignored. Finally, we know that the time complexity of the unitary operator U_p is $O(\frac{D(\text{polylog}(ND) + \text{poly}(h))}{\epsilon^2})$.

Here, we analyze the error of estimating p_{ij} . According to the error propagation theorem [38], the error of p_{ij} can be calculated by

$$\begin{aligned} \Delta p_{ij} &\approx \frac{1}{2N} \sqrt{(\Delta p_{j|i})^2 + (\Delta p_{i|j})^2} \\ &\leq \frac{1}{2N} (\Delta p_{j|i} + \Delta p_{i|j}), \end{aligned} \quad (42)$$

where $\Delta p_{j|i}$ can be gained via

$$\begin{aligned} \Delta p_{j|i} &\approx \frac{s_{ij}}{s_i} \times \left(\frac{\Delta s_{ij}}{s_{ij}} + \frac{\Delta s_i}{s_i} \right) \\ &= \frac{\Delta s_{ij}}{s_i} + \frac{s_{ij} \Delta s_i}{s_i^2}. \end{aligned} \quad (43)$$

Similarly, we can obtain the error of $\Delta p_{i|j}$. From Eq. (1), we know $s_{ij} \in (0, 1]$, $i \neq j$ and $s_{ii} = 0$. For the sake of analysis, we introduce a parameter $\delta_s \in (0, 1]$ such that

$$\frac{1}{N-1} \sum_j s_{ij} \geq \delta_s, \quad \forall i \in \{0, \dots, N-1\},$$

which characterizes a lower bound on the average similarity between $\mathbf{x}_i$ and the remaining data points. That is, $s_i \geq (N-1)\delta_s$. Combining with Eqs. (41)–(43), we can get

$$\Delta p_{j|i} \leq \frac{\epsilon}{(N-1)\delta_s} + \frac{2N\epsilon}{(N-1)^2\delta_s^2}$$

and

$$\Delta p_{ij} \leq O\left(\frac{\epsilon}{(N-1)^2\delta_s^2}\right).$$

Moreover, we can get that the relative error of p_{ij} is $O(\epsilon/\delta_s)$, independent of N .

The third step is to discuss the complexity of the unitary operator U_q . The main operations in step (3.1) are loading the dataset $\{y_i\}_{i=0}^{N-1}$ and performing quantum arithmetic, with time complexities $O(\text{polylog}(Nd))$ and $O(\text{poly}(h))$, respectively. The comparator in step (3.2) requires $O(\log N)$ time, while the quantum arithmetic operations in steps (3.2), (3.3), and (3.5) are executed in time $O(\text{poly}(h))$. Step (3.4) is used to obtain $T = \sum_{i,j=0}^{N-1} t_{ij}$, if the measurement error is ϵ , that is, $\Delta \frac{T}{N^2} = \epsilon$, then $O(1/\epsilon^2)$ measurements are required. Consequently, the overall time complexity of computing the probability distribution of the low-dimensional data via U_q is $O(\frac{\text{polylog}(Nd) + \text{poly}(h)}{\epsilon^2})$. Then, we analyze the error of q_{ij} . Similarly, we introduce the lower bound on the average similarity between $\mathbf{y}_i$ and the remaining data points $\delta_t \in (0, 1]$,

$$\frac{1}{N(N-1)} \sum_{i,j} t_{ij} \geq \delta_t,$$

that is, $T \geq N(N-1)\delta_t$. Thus, the error of q_{ij} satisfies

$$\Delta q_{ij} = \frac{t_{ij}}{T^2} \times \Delta T \leq \frac{\epsilon}{(N-1)^2\delta_t^2}. \quad (44)$$

Then, we can get the relative error of q_{ij} is $O(\epsilon/\delta_t)$, which is independent of N .

Finally, let us analyze the time complexity of computing the gradient in one iteration of the quantum t-SNE algorithm. As shown in Sec. III D, the computational cost is primarily dominated by step (4.1), which involves executing U_p and U_q . From the preceding analysis, this step requires $O(\frac{D(\text{polylog}(ND) + \text{poly}(h))}{\epsilon^2})$. Since there are Nd gradient components, the time complexity of computing the gradient in one iteration becomes $O(\frac{ND(\text{polylog}(ND) + \text{poly}(h))}{\epsilon^2})$. The corresponding error in calculating the gradient satisfies

$$\begin{aligned} \Delta &= 4 \left| \sum_j \widetilde{r}_{ijk} - \sum_j r_{ijk} \right| \\ &\leq 4 \sum_j |\widetilde{r}_{ijk} - r_{ijk}| \\ &= 4 \sum_j |(y_{ik} - y_{jk})t_{ij}| \times (\Delta p_{ij} + \Delta q_{ij}) \\ &\leq 4 \sum_j (\Delta p_{ij} + \Delta q_{ij}) \\ &\leq O\left(\frac{\epsilon}{N-1} \times \left(\frac{1}{\delta_s^2} + \frac{1}{\delta_t^2}\right)\right), \end{aligned} \quad (45)$$

where the second-to-last equation is because of $|(y_{ik} - y_{jk})t_{ij}| \leq 1$. Combining the relative error of p_{ij} and q_{ij} , we can conclude that the relative error of the gradient is also independent of N .

The time complexity of each step is summarized in Table I. Notably, the initialization of the low-dimensional data requires $O(Nd)$ time, which matches that of the classical algorithm. Furthermore, since the structure of QRAM does not change with specific data, the time required to update a memory of $\mathbf{y}_i$ is $O(\log_2(Nd))$ [39]. That is, the time required to update $\mathbf{y}$ in each iteration is $O(Nd \log_2(Nd))$. In summary, the time complexity of our algorithm in each iteration is

TABLE I. The time complexity of each step of the whole algorithm in one iteration.

Function	Corresponding step	Classical t-SNE	Proposed quantum t-SNE
Calculation of s_{ij}	Step 1	$O(N^2D)$	$O(\frac{D(\text{polylog}(ND)+\text{poly}(h))}{\epsilon})$
Calculation of p_{ij}	Step 2	$O(N^2D)$	$O(\frac{D(\text{polylog}(ND)+\text{poly}(h))}{\epsilon^2})$
Initialization	Step 3	$O(Nd)$	$O(Nd)$
Calculation of q_{ij}	Step 4	$O(N^2d)$	$O(\frac{\text{polylog}(Nd)+\text{poly}(h)}{\epsilon^2})$
Calculation of gradient	Step 5	$O(N^2d)$	$O(\frac{ND(\text{polylog}(ND)+\text{poly}(h))}{\epsilon^2})$
Total		$O(N^2D)$	$O(\frac{ND(\text{polylog}(ND)+\text{poly}(h))}{\epsilon^2})$

$O(\frac{ND(\text{polylog}(ND)+\text{poly}(h))}{\epsilon^2})$. In particular, ϵ is a parameter that trades off speedup and precision. From the above analysis, the amplitude estimation precision controls the relative error of the gradient, rather than an absolute error that shrinks with N . When the parameter satisfies $\epsilon^{-1} = \text{polylog}(ND)$ [8,13], the expected time complexity reduces to $O(ND \text{polylog}(ND))$.

In the above analysis, we consider the general case. Here, δ_s and δ_t are introduced as lower bounds on the average similarity in the high-dimensional and low-dimensional spaces, respectively. When applying the classical t-SNE algorithm in practice, the perplexity is usually set between 5 and 50. Under this setting, the high-dimensional similarity distribution is highly localized, meaning that each data point only has a constant number of significant neighbors. As a result, $\sum_j s_{ij}$ is typically $O(1)$, which implies that δ_s may scale as $O(1/N)$. Similarly, in the low-dimensional embedding, to achieve a successful visualization, δ_t may also decrease as N increases. In such cases, the relative error bounds $O(\epsilon/\delta_s)$ and $O(\epsilon/\delta_t)$ indicate that the required precision ϵ may scale with $O(1/N)$ to maintain a desired approximation accuracy. This dependence may affect the overall computational advantage in practice. This phenomenon reflects the intrinsic locality constraint imposed by t-SNE and highlights how such structural properties influence the performance of the proposed quantum algorithm.

B. Comparison and discussion

To highlight the advantages of our proposed algorithm, we compare it with the classical t-SNE, Barnes-Hut t-SNE, and existing quantum methods. First, compared to the classical algorithm with time complexity $O(N^2D)$, the proposed algorithm has a quadratic speedup with respect to the number of samples N . As shown in Table I, the time complexity of each quantum subroutine in our framework is superior to that of its classical counterpart. In particular, steps 1, 2, and 4 achieve exponential improvements in terms of the number of samples. More importantly, each step designed in this paper can be applied as a subroutine to other problems, resulting in algorithmic speedup.

Furthermore, we compare our method with the Barnes-Hut variant [18,19], as shown in Table II. As described in Sec. II, Barnes-Hut t-SNE reduces the computational cost of gradient evaluation to $O(N \log_2 N)$ by approximating both high-dimensional affinities and low-dimensional repulsive forces. In contrast, our method designs three unitary operators and leverages quantum parallelism to accelerate the gradient evaluation, achieving an overall complexity of $O(\frac{ND(\text{polylog}(ND)+\text{poly}(h))}{\epsilon^2})$. Hence, these two approaches improve the efficiency of t-SNE through fundamentally different mechanisms. Owing to its reliance on the approximation strategy, Barnes-Hut t-SNE does not provide explicit error bounds [22], and is primarily practical for two- or three-dimensional embeddings [18]. In contrast, our algorithm, similar to the original classical t-SNE, does not rely on such approximations and is therefore free from these limitations.

Finally, we compare the proposed algorithm with two quantum t-SNE approaches [14,15] in terms of degree of quantization, main techniques, time complexity, and the capability of processing quantum data, as summarized in Table III. As discussed in Sec. II, one of the previous methods employs parametrized quantum circuits to construct a variational model, while the other uses the swap test to evaluate sample similarities. In both cases, quantization is only partially achieved, being limited to either data initialization or fidelity estimation. In contrast, our method designs unitary operators U_s , U_p , U_q to realize additional steps, resulting in a more complete quantum framework and efficient gradient computation. In terms of time complexity, both the quantum t-SNE with PQC and the quantum kernel t-SNE do not achieve a quantum speedup due to their limited quantization scope, while our approach demonstrates a potential acceleration. Moreover, both quantum t-SNE methods can be applied to handle quantum data. However, their primary computational processes still rely on classical computers, particularly the calculation of high-dimensional similarities. That is, the quantum advantage is limited. In contrast, our algorithm not only supports quantum data processing, but also leverages quantum parallelism to accelerate the core computational tasks.

TABLE II. Comparison between Barnes-Hut t-SNE and the proposed quantum t-SNE.

	Barnes-Hut t-SNE [18]	Proposed quantum t-SNE
Time complexity	$O(N \log_2 N)$	$O(\frac{ND(\text{polylog}(ND)+\text{poly}(h))}{\epsilon^2})$
Error bounds	No	$O(\epsilon)$
Low-dimensional space	2 or 3	Any dimension (typically 2 or 3)

TABLE III. The comparison of our method and existing quantum t-SNE algorithms.

	Parametric t-SNE with QNN [14]	Quantum kernel t-SNE [15]	Proposed quantum t-SNE
Quantization steps	Initialization of data (step 3)	Calculation of data state fidelity	Other steps (steps 1, 2, 4, and 5)
Main techniques	Parametrized quantum circuits	Swap test	U_s, U_p, U_q , quantum arithmetic
Time complexity	$O(N^2D)$	$O(N^2D)$	$O(ND \text{ polylog}(ND))$
Process quantum data	Yes	Yes	Yes

In short, the advantages of our proposed scheme is evident in Tables I–III. Compared to the classical algorithms and two previous quantum approaches, the present method achieves quadratic acceleration.

V. CONCLUSION

In this study, we present a quantum t-SNE algorithm with a time complexity of $O(ND \text{ polylog}(ND))$, quadratically improving the dependence on N over the classical and existing quantum methods. Our algorithm aims at the core task of t-SNE, i.e., calculating the gradient of KL divergence between the high- and low-dimensional data probability distributions. To do this, three unitary operators U_s , U_p , and U_q are constructed. These operators enable efficient computation of sample similarities and probability distributions, significantly improving overall algorithmic efficiency. Moreover, the proposed subroutines are modular and can be applied to other quantum machine learning frameworks. For instance, integrating these procedures into quantum parametric t-SNE or quantum kernel t-SNE [14,15] could potentially lead to more efficient hybrid schemes. We anticipate that this algorithm will inspire further developments in quantum machine learning. Furthermore, implementing our quantum algorithm with real-world datasets is a promising direction for future research. We also note that in practical t-SNE settings, the locality induced by fixed perplexity may reduce the potential quantum advantage. Addressing such locality effects remains an important direction for future research.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (Grants No. 62171131 and No. 62006105), the China Postdoctoral Science Foundation (Grant No. 2023M731429), the Jiangxi Provincial Natural Science Foundation (Grant No. 20242BAB25113), and the Fujian Provincial Natural Science Foundation (Grants No. 2022J01186 and No. 2023J01533).

DATA AVAILABILITY

The data that support the findings of this article are openly available Refs. [14,15,18].

APPENDIX: QUANTUM CIRCUIT OF $\exp(-x)$

Given $x \in [0, 1)$ with its binary presentation $x = x_1x_2 \cdots x_h$, the $\exp$ gate computes the function $f(x) = \exp(-x)$. Specifically,

$$|x\rangle|0^{\otimes h}\rangle|0^{\otimes h}\rangle \mapsto |x\rangle|\exp(-x)\rangle|\psi_{\text{ancilla}}\rangle \quad (\text{A1})$$

$$\mapsto |x\rangle|\exp(-x)\rangle|0^{\otimes h}\rangle. \quad (\text{A2})$$

The conversion of formula (A1) can be implemented using the quantum circuit shown in Fig. 6, while the conversion of formula (A2) can be achieved by performing the corresponding inverse operation at the appropriate position. Inspired by the method proposed by Cong *et al.* [10], the quantum circuit diagram of $\exp(-x)$ can be constructed. The Taylor series of $\exp(-x)$ at around $x = 0$ is

$$\begin{aligned} \exp(-x) &= 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \cdots + \frac{(-1)^n x^n}{n!} \\ &+ \frac{(-1)^{n+1} x^{n+1}}{(n+1)!} + \cdots. \end{aligned} \quad (\text{A3})$$

According to Taylor's theorem [40], the reminder term for the n th term in the expansion is $\frac{f^{(n+1)}(\xi)}{(n+1)!} x^{n+1}$, where $\xi \in (0, x)$. So, in Eq. (A3), the reminder term (error) is $\frac{x^{n+1}}{(n+1)!}$ and is obviously bounded by $O(2^{-h})$ for $n = O(h/\log_2 h)$. In addition, a circuit of $f(x)$ can be constructed by the QFT Multiply-Adder (QMA) gate proposed in Ref. [36]. The specific circuit is shown in Fig. 6. The $\Pi_{m,n}^\pm$ in the figure is the QMA gate, realizing the following transformation $\Pi_{m,n}^\pm |a\rangle|b\rangle|c\rangle = |a\rangle|b\rangle|c \pm ab\rangle$, where m and n denote the number of digits of a and b , respectively. Moreover, it can be decomposed into the form $\Pi_{m,n}^\pm = (I \otimes I \otimes \text{QFT}^\dagger) \pi_{m,n}^\pm (I \otimes I \otimes \text{QFT})$, $\pi_{m,n}^\pm |a\rangle|b\rangle\text{QFT}|c\rangle = |a\rangle|b\rangle\text{QFT}|c \pm ab\rangle$. Therefore, the complexity of the $\exp$ gate is $(\text{poly } h)$ one- or two-qubit gates for accuracy 2^{-h} .

- [1] C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, New York, NY, 2006).
- [2] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, Visualizing the loss landscape of neural nets, in *Advances in Neural Information Processing Systems* (Curran Associates Inc., Montréal, Canada, 2018).
- [3] D. Erhan, A. Courville, Y. Bengio, and P. Vincent, Why does unsupervised pre-training help deep learning? in

Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (JMLR Workshop and Conference Proceedings) (JMLR.org, 2010), pp. 201–208.

- [4] L. van der Maaten and G. Hinton, Visualizing data using t-SNE, *J. Mach. Learn. Res.* **9**, 2579 (2008).
- [5] P. W. Shor, Algorithms for quantum computation: Discrete logarithms and factoring, in *Proceedings of the 35th Annual*

- Symposium on Foundations of Computer Science* (IEEE, New York, NY, 1994), pp. 124–134.
- [6] L. K. Grover, A fast quantum mechanical algorithm for database search, in *Proceedings of the 28th Annual ACM Symposium on Theory of Computing* (ACM, New York, NY, 1996), pp. 212–219.
 - [7] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum algorithm for linear systems of equations, *Phys. Rev. Lett.* **103**, 150502 (2009).
 - [8] C. H. Yu, F. Gao, and Q. Y. Wen, An improved quantum algorithm for ridge regression, *IEEE Trans. Knowl. Data Eng.* **33**, 858 (2019).
 - [9] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum principal component analysis, *Nat. Phys.* **10**, 631 (2014).
 - [10] I. Cong and L. Duan, Quantum discriminant analysis for dimensionality reduction and classification, *New J. Phys.* **18**, 073011 (2016).
 - [11] C. H. Yu, F. Gao, S. Lin, and J. B. Wang, Quantum data compression by principal component analysis, *Quantum Inf. Process* **18**, 249 (2019).
 - [12] W. Feng, G. Guo, S. Lin, and Y. Xu, Quantum Isomap algorithm for manifold learning, *Phys. Rev. Appl.* **22**, 014049 (2024).
 - [13] H.-L. Liu, L.-C. Wan, C.-H. Yu, S.-J. Pan, S.-J. Qin, F. Gao, and Q.-Y. Wen, A quantum algorithm for solving eigenproblem of the Laplacian matrix of a fully connected weighted graph, *Adv. Quantum Technol.* **6**, 2300031 (2023).
 - [14] Y. Kawase, K. Mitarai, and K. Fujii, Parametric t-stochastic neighbor embedding with quantum neural network, *Phys. Rev. Res.* **4**, 043199 (2022).
 - [15] Y. Kawase, K. Mitarai, and K. Fujii, Quantum kernel t-distributed stochastic neighbor embedding, *Phys. Rev. Res.* **6**, 043234 (2024).
 - [16] Z. Ye, K. Yu, G.-D. Guo, and S. Lin, Quantum self-organizing feature mapping neural network algorithm based on Grover search algorithm, *Phys. A: Stat. Mech. Appl.* **639**, 129690 (2024).
 - [17] H. Strange and R. Zwigelaar, *Open Problems in Spectral Dimensionality Reduction* (Springer, Berlin, Heidelberg, 2014).
 - [18] L. Van Der Maaten, Barnes-Hut-SNE, in *Proceedings of the International Conference on Learning Representations* (ICLR, 2013).
 - [19] L. Van Der Maaten, Accelerating t-SNE using tree-based algorithms, *J. Mach. Learn. Res.* **15**, 3221 (2014).
 - [20] P. Yianilos, Data structures and algorithms for nearest neighbor search in general metric spaces, in *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms* (SIAM, Philadelphia, PA, 1993), pp. 311–321.
 - [21] J. Barnes and P. Hut, A hierarchical $O(N \log N)$ force-calculation algorithm, *Nature (London)* **324**, 446 (1986).
 - [22] J. K. Salmon and M. S. Warren, Skeletons from the treecode closet, *J. Comput. Phys.* **111**, 136 (1994).
 - [23] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, *et al.*, Variational quantum algorithms, *Nat. Rev. Phys.* **3**, 625 (2021).
 - [24] V. Giovannetti, S. Lloyd, and L. Maccone, Quantum random access memory, *Phys. Rev. Lett.* **100**, 160501 (2008).
 - [25] V. Giovannetti, S. Lloyd, and L. Maccone, Architectures for a quantum random access memory, *Phys. Rev. A* **78**, 052310 (2008).
 - [26] C. T. Hann, G. Lee, S. Girvin, and L. Jiang, Resilience of quantum random access memory to generic noise, *PRX Quantum* **2**, 020311 (2021).
 - [27] L. Ruiz-Perez and J. C. Garcia-Escartin, Quantum arithmetic with the quantum Fourier transform, *Quantum Inf. Process* **16**, 152 (2017).
 - [28] C. P. Shao, Y. Li, and H. B. Li, Quantum algorithm design: Techniques and applications, *J. Syst. Sci. Complex.* **32**, 375 (2019).
 - [29] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, Quantum amplitude amplification and estimation, *Contemp. Math.* **305**, 53 (2002).
 - [30] H. S. Li, P. Fan, H. Y. Xia, and G. L. Long, The circuit design and optimization of quantum multiplier and divider, *Sci. China Phys. Mech. Astron.* **65**, 260311 (2022).
 - [31] D. Wang, Z. H. Liu, W. N. Zhu, and S. Z. Li, Design of quantum comparator based on extended general Toffoli gates with multiple targets, *Comput. Sci.* **39**, 302 (2012).
 - [32] I. Kerenidis, J. Landman, A. Luongo, and A. Prakash, q-means: A quantum algorithm for unsupervised machine learning, in *Proceedings of the 33rd International Conference on Neural Information Processing Systems* (Curran Associates, Red Hook, NY, 2019), pp. 4134–4144.
 - [33] A. Pavlidis and E. Floratos, Quantum-Fourier-transform-based quantum arithmetic with qudits, *Phys. Rev. A* **103**, 032417 (2021).
 - [34] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, Quantum algorithms revisited, *Proc. R. Soc. London Ser. A* **454**, 339 (1998).
 - [35] C. H. Yu, H. M. Rao, Y. P. Wu, D. X. Liu, X. P. Liu, and L. C. Wan, Quantum-assisted anomaly detection with multivariate Gaussian distribution, [arXiv:2505.02316v2](https://arxiv.org/abs/2505.02316v2).
 - [36] S. S. Zhou, T. Loke, J. A. Izaac, and J. B. Wang, Quantum Fourier transform in computational basis, *Quantum Inf. Process* **16**, 82 (2017).
 - [37] D. Dervovic, M. Herbster, P. Mountney, S. Severini, N. Usher, and L. Wossnig, Quantum linear systems algorithms: A primer, [arXiv:1802.08227v1](https://arxiv.org/abs/1802.08227v1).
 - [38] J. R. Taylor, *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements* (MIT Press, Cambridge, MA, 2022).
 - [39] I. Kerenidis and A. Prakash, Quantum recommendation systems, in *Proceedings of the 8th Innovations in Theoretical Computer Science Conference* (Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Wadern, Germany, 2017), pp. 49:1–49:21.
 - [40] M. Kline, *Calculus: An Intuitive and Physical Approach* (Courier Corporation, 1998).