

Quantum algorithm for estimating Olivier-Ricci curvature

Nhat A. Nghiem^{1,2,*}, Linh Nguyen^{3,†}, Tuan K. Do^{4,‡}, Tzu-Chieh Wei^{1,2,§} and Trung V. Phan^{5,||}

¹Department of Physics and Astronomy, State University of New York at Stony Brook, Stony Brook, New York 11794-3800, USA

²C. N. Yang Institute for Theoretical Physics, State University of New York at Stony Brook, Stony Brook, New York 11794-3840, USA

³Department of Mathematics, Florida A&M University, Tallahassee, Florida 32301, USA

⁴Department of Mathematics, University of California, Santa Barbara, California 93106, USA

⁵Department of Natural Sciences, Scripps and Pitzer Colleges, Claremont Colleges Consortium, Claremont, California 91711, USA



(Received 13 December 2025; accepted 3 May 2026; published 26 May 2026)

We introduce a quantum algorithm for computing the Ollivier-Ricci curvature, a discrete analog of the Ricci curvature defined via optimal transport on graphs and general metric spaces. This curvature has seen applications ranging from signaling fragility in financial networks to serving as basic quantities in combinatorial quantum gravity. For inputs given as a point cloud with pairwise distances, we show that our algorithm can achieve an exponential speedup over the best-known classical methods for two particular classes of problems. Our work is another step toward quantum algorithms for geometrical problems that are capable of delivering practical value while also informing fundamental theory.

DOI: [10.1103/hj52-lksk](https://doi.org/10.1103/hj52-lksk)

I. INTRODUCTION

The *manifold hypothesis* conjectures, with substantial empirical support, that any high-dimensional dataset can be represented by a much lower-dimensional manifold [1–3]. By stripping away redundant coordinates—often artifacts of noise or constraints [4]—one can recover the true local degrees of freedom, i.e., the intrinsic dimensionality [5]. By constructing local geometry—typically via a nearest-neighbor graph [6] and pairwise distances—we can infer the global topology of the underlying manifold and extract local geometric information such as curvature [7], whose bottlenecks or singularities flag anomalies [8]. This approach for big data analysis has many practical applications, and perhaps best known through quantitative finance, where the curvature on asset-correlation graphs is found to be a good indicator for market fragility [9,10] and potential crash magnitude [11,12]. Among discrete estimators of curvatures, the *Ollivier-Ricci curvature* (ORC) [13–15] has often emerged to be the most useful: It is defined via optimal transport between neighboring probability measures on the graph [16], associating curvature directly with how information moves across the network. For the same reason, ORC fits naturally with combinatorial quantum gravity [17–20], where distances and Markovian dynamics create emergent geometry. As ORC gains traction,

developing efficient methods to compute it becomes essential, including approaches that go beyond classical algorithms.

Quantum computation has seen significant progress since its early proposals [21,22]. By leveraging the intrinsic nature of quantum physics, including entanglement and superposition, a quantum computer can store and process information indistinguishably from a classical one. As such, it holds great promise for solving problems beyond the classical limit. Early attempts [23–26] have, to some extent, revealed such promise by showing that quantum computers can compute the property of black-box functions, search the database, and factorize integers, which can be done with considerably fewer resources than the best-known classical algorithms [27,28]. Over time, the notion of quantum computing has reached almost every corner of scientific computing. Notable examples include solving linear systems [29–31], Hamiltonian simulation [32–40], solving differential equations [34,41–44], machine learning and artificial intelligence [45–51], etc.

Recently, quantum algorithmic techniques have been applied in the context of topological data analysis (TDA), igniting a series of efforts [52–59]. In a parallel effort, the work [60] explores the potential of quantum computing for geometrical data analysis (GDA). While built on different mathematical grounds, TDA and GDA offer powerful frameworks for analyzing large-scale data, capable of revealing intrinsic properties of data in realistic scenarios where noise and sampling error often intervene. As outlined in the aforementioned works, quantum computers can offer an efficient solution to many problems in the TDA and GDA context, thus suggesting a broader potential for application, going beyond what was shown in the context of scientific computing, simulation, machine learning, etc.

In this work, following these developments—particularly, the attempt in Ref. [60]—we extend the reach of quantum computing to a closely relevant problem in the context of GDA. Specifically, we consider the problem of estimating

*Contact author: nhatanh.nghiemvu@stonybrook.edu

†Contact author: linh.nguyen@famu.edu

‡Contact author: ktdo@ucsb.edu

§Contact author: tzu-chieh.wei@stonybrook.edu

||Contact author: tphan@natsci.claremont.edu

ORC, given the input as a point cloud with pairwise connectivities and pairwise distances. This input model is the same as Ref. [60]; however, whereas Ref. [60] estimates local scalar curvature via a diffusion-based approach, in this work we adopt an optimal-transport-based estimator. It should be noted that optimal-transport-based quantities are useful for GDA because they characterize geometry at the level of probability distributions rather than relying on an underlying smooth manifold structure. In other words, this makes them applicable beyond the manifold hypothesis, including settings where data are discrete, noisy, or only partially manifold-like. By comparing neighborhood distributions via optimal transport, ORC captures mesoscopic geometric structure (e.g., connectivity, bottlenecks, and clustering) that is not accessible to purely local differential notions like Gaussian curvature [60,61].

Our work is organized as follows. In Sec. II, we provide a formal description of the key objective and related assumptions. In the same section, we describe the classical solution to the key objective and state our main result, followed by a comparison to show the advantage of our quantum method. Section III contains our main proposal for the quantum algorithm for estimating the ORC, which splits into two subsections (Secs. III A and III B) to handle two cases, corresponding to two different types of input conditions. Finally, conclusion is given in Sec. IV.

II. CLASSICAL ALGORITHM FOR CALCULATING THE OLLIVIER-RICCI CURVATURE

Consider a graph $G = (V, E)$, where V is the set of vertices and E is the set of edges. Each edge is represented by the pair of vertices it connects, $(a, b) \in E$ with $a, b \in V$, and is assigned a distance $d(a, b)$. The edges are undirected, so that $(a, b) \equiv (b, a)$. The *neighbors* of a vertex $a \in V$ are defined as the set of all vertices connected to a by an edge, i.e., $\mathcal{N}_a = \{b \in V | (a, b) \in E\}$. For any pair of vertices $a', b' \in V$, the *graph*, or *geodesic distance* between them, denoted by $d_G(a', b')$, is defined as the length of the shortest path in G connecting a' and b' . Explicitly,

$$d_G(a', b') = \min_{\substack{(v_1, v_2), (v_2, v_3), \dots, \\ (v_{k-2}, v_{k-1}), (v_{k-1}, v_k) \in E \\ v_1 = a', v_k = b'}} \sum_{i=1}^k d(v_i, v_{i+1}), \quad (2.1)$$

where the minimum is taken over all possible paths in the graph G with end points a' and b' .

An ORC value is associated with each edge $(x, y) \in E$, denoted by $\gamma(x, y)$, and can be calculated with the Earth Mover (1-Wasserstein) distance $W_1(x, y)$ as follows [62]:

$$\gamma(x, y) = 1 - \frac{W_1(x, y)}{d_G(x, y)}. \quad (2.2)$$

To calculate $W_1(x, y)$, we need to focus on the local configuration near the edge (x, y) , as illustrated in Fig. 1. Assume that the set of x neighbors *excluding* y has p points and the set of y neighbors *excluding* x has q points, i.e., $\mathcal{N}_x \setminus \{y\} = \{x_1, x_2, \dots, x_p\}$ and $\mathcal{N}_y \setminus \{x\} = \{y_1, y_2, \dots, y_q\}$. Then, the value of $W_1(x, y)$ is the minimum cost of reconfiguring mass in vertices in $\mathcal{N}_x \setminus \{y\}$, which we further assume to

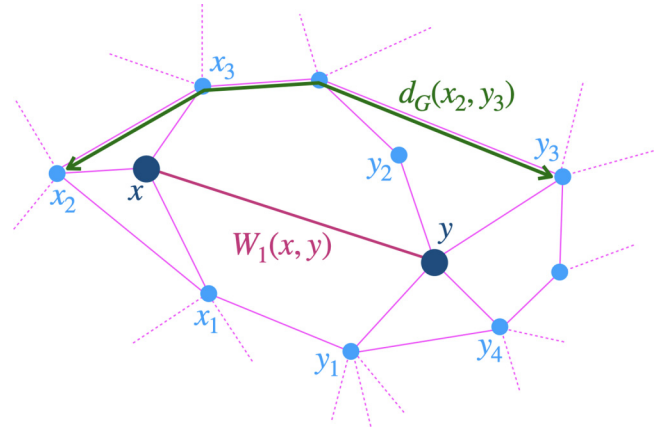


FIG. 1. Illustration of the setup used to compute the Earth Mover distance associated with an edge in a graph. The graph's vertices are in blue, and its edges are in magenta. Here, we consider the edge (x, y) , in which the neighbors of x excluding y are $\{x_1, x_2, x_3\}$ ($p = 3$) and the neighbors of y excluding x are $\{y_1, y_2, y_3, y_4\}$ ($q = 4$). We show the graph distance $d_G(x_2, y_3)$ between points x_2 and y_3 in green. By optimizing the expression given in Eq. (2.3), we arrive at the Earth Mover distance $W_1(x, y)$, which then allows us to calculate the ORC $\gamma(x, y)$ associated with the edge (x, y) using Eq. (2.2).

sum up to 1 and be uniformly distributed, to another uniform distribution of mass in vertices in $\mathcal{N}_y \setminus \{x\}$. Concretely, denote by γ_{ij} (where $i = 1, 2, \dots, p$ and $j = 1, 2, \dots, q$) the *transport variable* that represents the amount of mass transferred from x_i to y_j , which incurs a cost of $d_G(x_i, y_j)\gamma_{ij}$. We solve the following linear program with $p \times q$ variables and $p + q$ constraints:

$$W_1(x, y) = \min_{\{\gamma_{ij}\}} \sum_{i=1}^p \sum_{j=1}^q d_G(x_i, y_j) \gamma_{ij}, \quad (2.3)$$

$$\forall j : \sum_{i=1}^p \gamma_{ij} = 1/q, \quad (2.4)$$

$$\forall i : \sum_{j=1}^q \gamma_{ij} = 1/p, \quad (2.5)$$

$$\forall i, j : \gamma_{ij} \geq 0. \quad (2.6)$$

As a demonstration, in Appendix A we calculate $\gamma(x, y)$ in the graph illustrated in Fig. 1, assuming that all edges are unit length. We note that the definition of $W_1(x, y)$ varies, e.g., in Refs. [63,64], the summation above does include y as neighborhood of x (and vice versa). However, either definition is practically fine and does not lose the generality of our subsequent construction.

In general, a linear program does not admit a closed-form solution and needs to be solved algorithmically. An open question in the field of optimization that has resisted progress for several decades is whether linear programming can be solved in strongly polynomial time. All known algorithms are either worst-case exponential (in the number of variables, for fixed dimension linear programming is strongly polynomial) or algebraic/numerical and thus depend on the bit complexity of the input [65,66].

The particular category that our linear program computes $W_1(x, y)$ falls into is the structured *optimal transport* problem, for which there exist strongly polynomial algorithms. We give a brief survey of some of these and refer the reader to the references listed for more details.

(1) Orlin's strongly polynomial time combinatorial exact algorithm that runs in $O(pq(p+q)\log(p+q))$ for finding the min-cost flow on an uncapacitated complete bipartite graph $K_{p,q}$ [67].

(2) A recent breakthrough gives an $O((pq)^{1+o(1)})$ algorithm for optimal transport with polynomially bounded integral demands and supplies (into which we can easily transform our linear program, by scaling both demands and supplies by a factor of pq) [68].

(3) When $p = q$, the 1-Wasserstein optimal transport problem becomes a linear assignment (matching) problem. To see this, rescale the variables by letting $\Gamma_{ij} = p\gamma_{ij}$. Then, $\sum_{j=1}^p \Gamma_{ij} = 1$ and $\sum_{i=1}^p \Gamma_{ij} = 1$, so the matrix $\Gamma \in \mathbb{R}^{p \times p}$ is a square *doubly stochastic matrix* [69]. The objective 2.3 becomes $\frac{1}{p} \sum_{i=1}^p \sum_{j=1}^p d_G(x_i, y_j) \Gamma_{ij}$, which is equivalent (up to a constant factor, in the context of a linear program) to $\sum_{i,j} d_G(x_i, y_j) \Gamma_{ij}$.

By the classic Birkhoff theorem [70], every doubly stochastic matrix can be expressed as a convex combination of permutation matrices (matrices with exactly one 1 in each row and column). Equivalently, the permutation matrices are precisely the vertices of the Birkhoff polytope, which contains all doubly stochastic matrices. We are to optimize a linear objective function over the set of all doubly stochastic matrices and since in any linear program, if an optimal solution exists and the optimum value is finite (which is the case for our problem since the Birkhoff polytope is compact), then at least one optimal solution lies at a vertex of the feasible polytope [65,66]. Thus, it suffices to find an optimal permutation matrix. In this case, an optimal transport plan sends all the mass at each vertex to exactly one other vertex, so the problem reduces to a balanced matching problem, solvable by the Hungarian algorithm. (However, the optimal value of the objective function $W_1(x, y)$ still depends on the cost structure and does not admit a closed-form expression [71]).

We note that while there are other approaches with better asymptotic running times, the Hungarian algorithm has a small constant in the big O and works well in practice with moderately sized instances, along with plentiful off-the-shelf implementations that are highly optimized, so the Hungarian algorithm remains the standard in many applications.

(4) When the distances between the supply and demand nodes are *decomposable*, i.e., $d_{ij} = e_i + f_j$ or $d_{ij} = e_i + f_j + C$ (C is a constant), the solution is remarkably simple, as any feasible transport is optimal [16]. For instance, when G is a tree, since there is only one path from any x_i to any y_j : (x_i, x) , (x, y) , (y, y_j) , we always have $d_G(x_i, y_j) = d_G(x_i, x) + d_G(x, y) + d_G(y, y_j)$, and therefore we have

$$W_1(x, y) = \frac{1}{p} \sum_{i=1}^p d_G(x_i, x) + d_G(x, y) + \frac{1}{q} \sum_{j=1}^q d_G(y, y_j). \quad (2.7)$$

The classical algorithm in this case just needs to perform the summation, which takes $\mathcal{O}(p+q)$ time.

As can be seen above, the complexity of the classical algorithms and hence their complexities vary case by case, as linear programming in general is difficult and a closed-form solution does not exist. In this work, we pay particular attention to (1) when G forms a tree and (2) when $p = q$. In these cases, there are certain patterns in the solutions that we could exploit to construct the corresponding quantum algorithms.

As a brief overview, we discuss a few key ideas and the technical flow of our quantum algorithm. First, our work utilizes a key technical Lemma in Ref. [60]. In Ref. [60], it was shown that provided the raw distances d between data points, then there is a procedure that outputs the block encoding (see definition in Appendix B) of an operator having the corresponding geodesic distances d_G between data points on the diagonal. In particular, the index of the geodesic distance $d_G(\mathbf{x}_i, \mathbf{x}_j)$ between $\mathbf{x}_i, \mathbf{x}_j$ is $|i-1\rangle\langle i-1| \otimes |j-1\rangle\langle j-1|$. From this block encoding, our strategy is then to prepare a state of the form $\frac{1}{\sqrt{p+1}} \sum_{i=1}^p d_G(x, x_i) |i_x - 1\rangle |i - 1\rangle$, where $|i_x - 1\rangle$ refers to the vertex index, in terms of graph notation, of x , plus another state of the form $\frac{1}{\sqrt{p+1}} \sum_{i=1}^p |i_x - 1\rangle |i - 1\rangle$. In the case when G is a tree, then as pointed out above, the Earth-Mover (1-Wasserstein) distance admits an analytical form. Then, by estimating the overlaps between the two states we prepared (e.g., via Hadamard/SWAP test), we can infer the value of $W_1(x, y)$, and thus of the ORC $\gamma(x, y)$.

In the case $p = q$, as mentioned above, $W_1(x, y)$ becomes trickier as an explicit expression does not exist. Instead, it is given by

$$W_1(x, y) = \min_{\gamma_{ij}} \sum_{i=1}^p \sum_{j=1}^q \frac{1}{p} d_G(x_i, y_j) \gamma'_{ij}, \quad (2.8)$$

where $\gamma' \equiv [\gamma'_{ij}]$ is the permutation matrix. We observe the following: If we organize $\{d_G(x_i, y_j)\}_{i,j=1}^p \equiv \{d_{ij}\}_{i,j=1}^p$ into a square matrix with corresponding entries, then the minimum value of $\{d_{i_1 1} + d_{i_2 2} + \dots + d_{i_p p}\}_{i_1 \neq i_2 \neq \dots \neq i_p}$ corresponds to the desired $W_1(x, y)$ above. Our strategy then is that from the block encoding of an operator having the corresponding geodesic distances d_G between data points on the diagonal, via a sequence of Lemmas B 3 and B 2, we build a block encoding of an operator $\mathcal{D}_p$ having $\{d_{i_1 1} + d_{i_2 2} + \dots + d_{i_p p}\}_{i_1 \neq i_2 \neq \dots \neq i_p}$ on the diagonal. To find such the minimum eigenvalue of $\mathcal{D}_p$, we apply the recent result regarding finding minimum eigenvalues [72,73] of a matrix. Since the operator $\mathcal{D}_p$ is diagonal, then its minimum eigenvalue is also the minimum value among its diagonal entries. As such, we can find the desired $W_1(x, y)$. Before describing such algorithms in greater detail, we summarize the main results in the following theorem, showing how our quantum algorithm is advantageous over the classical one.

Theorem 1. Provided the raw distances $d(\mathbf{x}_i, \mathbf{x}_j)$ between a pair $(\mathbf{x}_i, \mathbf{x}_j)$ among N data points.

(1) When G is a tree, $W_1(x, y)$ can be estimated with accuracy ϵ using a quantum circuit of complexity

$$\mathcal{O}\left(\frac{1}{\epsilon} \log(N) \log^2\left(\frac{1}{\epsilon}\right) \kappa \log^2\left(\frac{\kappa}{\epsilon}\right) + \frac{1}{\epsilon} \log(pq)\right).$$

(2) When $p = q$, $W_1(x, y)$ can be estimated with accuracy ϵ using a quantum circuit of complexity

$$\mathcal{O}\left(\left(\log(N) \log^2\left(\frac{1}{\epsilon}\right) \kappa \log^2\left(\frac{\kappa}{\epsilon}\right) + \log(p \log p)\right) \times \frac{1}{\gamma} p \log\left(\frac{p}{\epsilon}\right) \left(\frac{1}{\epsilon}\right) \log^2 \frac{1}{\epsilon}\right),$$

where in the above, $\kappa = \min_{i,j,q,k} \{ \frac{d_G(\mathbf{x}_i, \mathbf{x}_j)}{d_G(\mathbf{x}_k, \mathbf{x}_q)} \}$, and γ is a randomized factor that could be of $\mathcal{O}(1)$ in the best case and of $\mathcal{O}(\exp(-p))$ in the worst case.

The complexity of classical algorithms for the case G being a tree and for the case $p = q$ is, respectively, $\mathcal{O}(p + q)$ and $\mathcal{O}(p^3)$. However, these complexities do not include the time required to compute the geodesic distances from the raw distances $d(\mathbf{x}_i, \mathbf{x}_j)$ between a pair $(\mathbf{x}_i, \mathbf{x}_j)$ among N data points. Based on Ref. [74], the time complexity for producing geodesic distances from the raw distances is $\mathcal{O}(N^3)$. So, the total classical complexity for estimating ORC is $\mathcal{O}(N^3 + p + q)$ (for graph G being a tree) and $\mathcal{O}(N^3 + p^3)$ (for $p = q$), respectively. Compared to the quantum complexities above, there is exponential speedup in the total number of points N in both cases. Regarding p, q (for the first case where G is a tree), which controls the size of local neighborhood of two points x, y of interest, there is still exponential speedup in the first case (when G is a tree). However, in the second case, the degree of speedup depends on the factor γ . As γ could be of $\mathcal{O}(1)$ in the best case and of $\mathcal{O}(\exp(-p))$ in the worst case, the best possible quantum speedup is polynomial. We remark one thing that as p (and q) controls the size of local neighborhood of x (and y), in principle, it can be chosen to be $\mathcal{O}(1)$. Thus, the scaling on p is $\mathcal{O}(1)$ in all cases, and therefore, our quantum algorithm can guarantee to provide significant speedup with respect to N —the number of data points, which is typically large in reality.

III. QUANTUM ALGORITHM

Before describing our main quantum algorithm, we provide some preliminaries. Our subsequent proposal shall be utilizing the block-encoding and quantum singular value transformation (QSVT) framework [75]. This framework was built on the qubitization and quantum signal processing technique [76,77]. In particular, it was shown in Ref. [75] that the block-encoding/QSVT framework can unify many known quantum algorithms, thus providing a universal language for this area, motivating many subsequent refinements [72,73,78]. A key component of the block-encoding/QSVT framework is the unitary block encoding of some operator. Specifically, a unitary U is said to be an exact block encoding of a matrix A (with operator norm $\|A\| \leq 1$) if $A = ((\mathbf{0} \otimes \mathbb{I})U(\mathbb{I} \otimes |\mathbf{0}\rangle))$. If $((\mathbf{0} \otimes \mathbb{I})U(\mathbb{I} \otimes |\mathbf{0}\rangle)) = \tilde{A}$ and $\|\tilde{A} - A\| \leq \epsilon$, then U is said to be an ϵ -approximated block encoding of A . Given two unitaries U_1, U_2 that block encode A_1, A_2 , then there is an efficient procedure that can construct the block encoding of many operators, including linear combination of A_1, A_2 , product of A_1, A_2 , tensor product $A_1 \otimes A_2$, etc. A fundamental result of the QSVT [75] is that, given a unitary U that block encodes A , then the unitary block encoding of $f(A)$ can be constructed efficiently, where $f(\cdot)$ is some computable function. We note

that a more complete summary of block encoding and related recipes can be found in Appendix B.

Our quantum algorithm, described below, is a continuation of the proposal initiated in Ref. [60], where the Gaussian curvature was considered. An important ingredient to our subsequent discussion is the following Lemma:

Lemma 1 (Appendixes C and D of Ref. [60]). Provided the raw distances $d(\mathbf{x}_i, \mathbf{x}_j)$ between a pair $(\mathbf{x}_i, \mathbf{x}_j)$ among N data points, then the ϵ -approximated block encoding of the operator

$$\frac{1}{\alpha} \sum_{i,j=1}^N |i-1\rangle \langle i-1| \otimes d_G^4(\mathbf{x}_i, \mathbf{x}_j) |j-1\rangle \langle j-1| \quad (3.1)$$

can be obtained by a quantum circuit of depth

$$\mathcal{O}\left(\log(N) \log^2\left(\frac{1}{\epsilon}\right)\right), \quad (3.2)$$

where $\alpha \in \mathcal{O}(1)$ is a constant that generally depends on the dataset.

Moreover, we point out the following result of Refs. [78], [75]:

Lemma 2 (Positive power exponent [75,78]). Given a block encoding of a positive matrix $\mathcal{M}/\gamma$ such that

$$\frac{\mathbb{I}}{\kappa_M} \leq \frac{\mathcal{M}}{\gamma} \leq \mathbb{I},$$

and let $c \in (0, 1)$, then we can implement an ϵ -approximated block encoding of $(\mathcal{M}/\gamma)^c/2$ in time complexity $\mathcal{O}(\kappa_M T_M \log^2(\frac{\kappa_M}{\epsilon}))$, where T_M is the complexity to obtain the block encoding of $\mathcal{M}/\gamma$.

A direct application of Lemma 1 and the above Lemma (with $c = \frac{1}{4}$) allows us to obtain the following Lemma:

Lemma 3. The ϵ -approximated block encoding of the operator

$$\frac{1}{\alpha^{1/4}} \sum_{i,j=1}^N |i-1\rangle \langle i-1| \otimes d_G(\mathbf{x}_i, \mathbf{x}_j) |j-1\rangle \langle j-1| \quad (3.3)$$

can be obtained using a quantum circuit of depth

$$\mathcal{O}\left(\log(N) \log^2\left(\frac{1}{\epsilon}\right) \kappa \log^2 \frac{\kappa}{\epsilon}\right), \quad (3.4)$$

where κ is the condition number of the operator in Eq. 3.1, which can be seen to be $\kappa = \max_{i,j,q,k} \{ \frac{d_G^4(\mathbf{x}_i, \mathbf{x}_j)}{d_G^4(\mathbf{x}_k, \mathbf{x}_q)} \}$.

We have mentioned in the previous section that the closed-form solution to Eq. 2.3 generally does not exist. Therefore, in the following, we consider case by case and outline the corresponding quantum algorithm.

A. The case when G is a tree

For simplicity, we begin with the case when G is a tree, for which a closed-form solution does exist. Treelike structures often emerge in sparse or locally hierarchical graphs, for example, through filtering procedures such as minimum spanning tree constructions [79] or information filtering of dense correlation networks [80]. More intuitively, sparse graphs are typically locally treelike in high-dimensional or large-scale settings.

As mentioned, the solution to Eq. 2.3 in this case is given by

$$W_1(x, y) = \frac{1}{p} \sum_{i=1}^p d_G(x_i, x) + d_G(x, y) + \frac{1}{q} \sum_{j=1}^q d_G(y, y_j). \quad (3.5)$$

In Sec. II, we have used the notation $\mathcal{N}_x \setminus \{y\} \equiv \{x_1, x_2, \dots, x_p\}$ to denote the set of neighbor points of x . Lemma 1 uses $\mathbf{x}_i$ to denote the i th data point in the given dataset. To make things more appropriate, without loss of generalization, we promote the following index interchange $x_i \leftrightarrow \mathbf{x}_i$. For simplicity, the set $\mathcal{N}_x \setminus \{y\} \equiv \{x_1, x_2, \dots, x_p\}$ is then labeled as

$$\{x_1, x_2, \dots, x_p\} \equiv \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p\}. \quad (3.6)$$

In a similar manner, the set $\mathcal{N}_y \setminus \{x\} = \{y_1, y_2, \dots, y_q\}$ is labeled as

$$\{y_1, y_2, \dots, y_q\} \equiv \{\mathbf{y}_{p+1}, \mathbf{y}_{p+2}, \dots, \mathbf{y}_{p+q}\}. \quad (3.7)$$

In this convention, the indexes of x_i, y_i (for $1 \leq i \leq p$) are $i, p+i$, respectively. In particular, we set x to have index i_x and y to have index i_y . To estimate $W_1(x, y)$, we assume that the indexes of x and y are classically known, as well as the neighbors of x and y , which are $\{i_1, i_2, \dots, i_p\}$ and $\{j_1, j_2, \dots, j_q\}$, respectively. Denote the index of x by i_x ; we

note that the block-encoded operator

$$\mathcal{D} \equiv \frac{1}{\alpha^{1/4}} \sum_{i=1}^N \sum_{j=1}^N |i-1\rangle \langle i-1| \otimes d_G(\mathbf{x}_i, \mathbf{x}_j) |j-1\rangle \langle j-1|$$

can be decomposed as

$$\begin{aligned} & \frac{1}{\alpha^{1/4}} \sum_{\mathbf{x}_i, \mathbf{x}_j \notin \mathcal{N}_x \setminus \mathcal{N}_y} |i-1\rangle \langle i-1| \otimes d_G(\mathbf{x}_i, \mathbf{x}_j) |j-1\rangle \langle j-1| \\ & + \frac{1}{\alpha^{1/4}} \sum_{\mathbf{x}_i \in \mathcal{N}_x \setminus \mathcal{N}_y} |i_x-1\rangle \langle i_x-1| \otimes d_G(x, \mathbf{x}_i) |i-1\rangle \langle i-1|. \end{aligned} \quad (3.8)$$

Let U_d denote the unitary block encoding of the above operator. Provided the classical knowledge of indices $\{1, 2, 3, \dots, p\}$ and also of x , the following state can be prepared using a depth-1 circuit:

$$|\phi\rangle = \frac{1}{\sqrt{p+1}} |i_x-1\rangle \otimes \sum_{i=1}^p |i-1\rangle, \quad (3.9)$$

using a quantum circuit of depth $\mathcal{O}(\log p)$ and $\mathcal{O}(1)$ ancilla, based on the state preparation method [81,82]. If we take the unitary U_d and apply it to the state $|0\rangle |\phi\rangle$, according to Definition B 1, we obtain the state

$$\begin{aligned} U_d |0\rangle |\phi\rangle &= |0\rangle \left(\frac{1}{\alpha^{1/4}} \sum_{\mathbf{x}_i, \mathbf{x}_j \notin \mathcal{N}_x \setminus \mathcal{N}_y} |i-1\rangle \langle i-1| \otimes d_G(\mathbf{x}_i, \mathbf{x}_j) |j-1\rangle \langle j-1| \right. \\ & \left. + \frac{1}{\alpha^{1/4}} \sum_{\mathbf{x}_i \in \mathcal{N}_x \setminus \mathcal{N}_y} |i_x-1\rangle \langle i_x-1| \otimes d_G(x, \mathbf{x}_i) |i-1\rangle \langle i-1| \right) \frac{1}{\sqrt{p+1}} \sum_{i=1}^p |i_x-1\rangle |i-1\rangle + |\text{Garbage}\rangle, \end{aligned} \quad (3.10)$$

where $|\text{Garbage}\rangle$ is orthogonal to the first part. It can be seen that the above state is equivalent to

$$\frac{1}{\alpha^{1/4} \sqrt{p+1}} |0\rangle \sum_{i=1}^p d_G(x, \mathbf{x}_i) |i_x-1\rangle |i-1\rangle + |\text{Garbage}\rangle. \quad (3.11)$$

Next, we consider the overlaps between the above state and $|0\rangle |\phi\rangle$ again:

$$\begin{aligned} \langle 0| \langle \phi| & \left(\frac{1}{\alpha^{1/4} \sqrt{p+1}} |0\rangle \sum_{i=1}^p d_G(\mathbf{x}, \mathbf{x}_i) |i_x-1\rangle |i-1\rangle + |\text{Garbage}\rangle \right) \\ &= \left(\frac{1}{\sqrt{p+1}} \sum_{i=1}^p \langle i_x-1| \langle i-1| \right) \left(\frac{1}{\alpha^{1/4} \sqrt{p+1}} |0\rangle \sum_{i=1}^p d_G(x, \mathbf{x}_i) |i_x-1\rangle |i-1\rangle + |\text{Garbage}\rangle \right) \\ &= \frac{1}{\alpha^{1/4} (p+1)} \sum_{i=1}^p d_G(x, \mathbf{x}_i) = \frac{1}{\alpha^{1/4} (p+1)} \sum_{i=1}^p d_G(x, x_i). \end{aligned} \quad (3.12)$$

These overlaps can be estimated via standard methods, such as the Hadamard test, SWAP test, or amplitude estimation. The value of the desired quantity $\frac{1}{p} \sum_{i=1}^p d_G(x_i, x)$ can be derived from the above estimation accordingly, e.g., by multiplying $\frac{1}{\alpha^{1/4} (p+1)} \sum_{i=1}^p d_G(x, x_i)$ with $\frac{\alpha^{1/4} (p+1)}{p}$.

Repeating the above procedure but interchanging $x \leftrightarrow y, p \leftrightarrow q$, and $\{x_1, x_2, \dots, x_p\} \leftrightarrow \{y_1, y_2, \dots, y_q\}$ can allow us to estimate the summation $\frac{1}{q} \sum_{j=1}^q d_G(y, y_j)$. The value $d_G(x, y)$ is, in fact, a special case of the above estimation, as we just need to set a fixed value of index x, y . Instead of preparing $|\phi\rangle$ as above, we just need to prepare the state

$|i_x - 1\rangle |i_y - 1\rangle$ and execute the same procedure. From the value of $W_1(x, y)$ and also $d_G(x, y)$, the estimation of the Ollivier-Ricci curvature is straightforward as $\kappa(x, y) = 1 - \frac{W_1(x, y)}{d_G(x, y)}$.

Complexity. To analyze the total quantum circuit depth for this part, we summarize the above procedure in the following steps.

Step 1. Obtain the ϵ -approximated unitary block encoding U_d of

$$\frac{1}{\alpha^{1/4}} \sum_{i=1}^N \sum_{j=1}^N |i-1\rangle \langle i-1| \otimes d_G(\mathbf{x}_i, \mathbf{x}_j) |j-1\rangle \langle j-1| \quad (3.13)$$

via Lemma 3, which involves a quantum circuit of depth

$$\mathcal{O}\left(\log(N) \log^2\left(\frac{1}{\epsilon}\right) \kappa \log^2\left(\frac{\kappa}{\epsilon}\right)\right). \quad (3.14)$$

Step 2. The state $|\phi\rangle$ (and hence $|\mathbf{0}\rangle |\phi\rangle$) can be prepared using a circuit of depth $\mathcal{O}(\log p)$. Therefore, the quantum circuit depth that prepares the state $U_d |\mathbf{0}\rangle |\phi\rangle$ is

$$\mathcal{O}\left(\log(N) \log^2\left(\frac{1}{\epsilon}\right) \kappa \log^2\left(\frac{\kappa}{\epsilon}\right) + \log p\right). \quad (3.15)$$

Step 3. The overlap estimation with a precision ϵ incurs a total complexity

$$\mathcal{O}\left(\frac{1}{\epsilon} \log(N) \log^2\left(\frac{1}{\epsilon}\right) \kappa \log^2\left(\frac{\kappa}{\epsilon}\right) + \frac{1}{\epsilon} \log p\right). \quad (3.16)$$

For completeness, we recall that $\kappa = \min_{i,j,q,k} \left\{ \frac{d_G(\mathbf{x}_i, \mathbf{x}_j)}{d_G(\mathbf{x}_i, \mathbf{x}_q)} \right\}$.

Step 4. Finally, repeating the above step with the interchange $x \leftrightarrow y, p \leftrightarrow q, \{x_1, x_2, \dots, x_p\} \leftrightarrow \{y_1, y_2, \dots, y_q\}$ leads to the same complexity as above, but with p replaced by q :

$$\mathcal{O}\left(\frac{1}{\epsilon} \log(N) \log^2\left(\frac{1}{\epsilon}\right) \kappa \log^2\left(\frac{\kappa}{\epsilon}\right) + \frac{1}{\epsilon} \log q\right). \quad (3.17)$$

Hence, the total complexity is

$$\mathcal{O}\left(\frac{1}{\epsilon} \log(N) \log^2\left(\frac{1}{\epsilon}\right) \kappa \log^2\left(\frac{\kappa}{\epsilon}\right) + \frac{1}{\epsilon} \log(pq)\right). \quad (3.18)$$

B. The case when $p = q$

The case $p = q$, corresponding to identical neighborhood construction and hence shared local connectivity structure, arises naturally in settings where neighborhood measures are defined from a point cloud using a common kernel or sampling rule (e.g., k -regular graph [83]). This case is considerably harder than in the previous case, as we have discussed in Sec. II. Although the Hungarian algorithm [71] can solve this case exactly in polynomial time, below we shall attempt to derive a quantum algorithm running in linear time, at the cost of achieving the approximation. We first recall that the reasoning of Ref. [71] results in the matrix $\gamma \equiv [\gamma_{ij}]_{p \times p}$ being a permutation matrix multiplied by a factor of $\frac{1}{p}$. Given that the Earth Mover (1-Wasserstein) distance $W_1(x, y)$ is defined

as

$$W_1(x, y) = \min_{\gamma_{ij}} \sum_{i=1}^p \sum_{j=1}^q \frac{1}{p} d_G(x_i, y_j) \gamma'_{ij}, \quad (3.19)$$

where $\gamma' \equiv [\gamma'_{ij}]$ is the permutation matrix. Then, we organize $\{d_G(x_i, y_j)\}_{i,j=1}^p$ as a $p \times p$ matrix, and define $d_G \equiv [d_G(x_i, y_j)]_{i,j=1}^p \equiv [d_{ij}]_{i,j=1}^p$ be the $p \times p$ matrix with entries being the geodesic distances as indicated. The minimization problem can then be translated into the following problem:

For each column index i , find an element j so that its summation $\sum_{i=1}^p d_{\pi(i),i}$ is minimized. In other words, we need to find the minimum value of

$$\{d_{i_1 1} + d_{i_2 2} + \dots + d_{i_p p}\}_{i_1 \neq i_2 \neq \dots \neq i_p}$$

How do we compute the above sum and minimize over all possible permutations? Our algorithm is built on the following observation. Suppose, for now, that $p = 3$. Then, we have

$$d_G = \begin{pmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{pmatrix}. \quad (3.20)$$

Consider the following 9×9 diagonal matrix:

$$D_G = \text{diag}(d_{11}, d_{21}, d_{31}, d_{12}, d_{22}, d_{32}, d_{13}, d_{23}, d_{33}). \quad (3.21)$$

We define three blocks D_1, D_2, D_3 as follows:

$$\begin{aligned} D_1 &= \text{diag}(d_{11}, d_{21}, d_{31}), \\ D_2 &= \text{diag}(d_{12}, d_{22}, d_{32}), \\ D_3 &= \text{diag}(d_{13}, d_{23}, d_{33}). \end{aligned} \quad (3.22)$$

Next, we consider the top-left corner submatrix of dimension 9×9 , denoted by $[\cdot]_{9 \times 9}$ of

$$\begin{aligned} [D_1 \otimes \mathbb{I}_9]_{9 \times 9} &= \text{diag}(d_{11}, d_{11}, d_{11}, d_{11}, d_{11}, d_{11}, d_{11}, d_{11}, d_{11}), \\ [\mathbb{I}_3 \otimes D_2 \otimes \mathbb{I}_3]_{9 \times 9} &= \text{diag}(d_{12}, d_{12}, d_{12}, d_{22}, d_{22}, d_{22}, d_{32}, d_{32}, d_{32}), \\ [\mathbb{I}_9 \otimes D_3]_{9 \times 9} &= \text{diag}(d_{13}, d_{23}, d_{33}, d_{13}, d_{23}, d_{33}, d_{13}, d_{23}, d_{33}). \end{aligned} \quad (3.23)$$

Their summation is as follows (again, we pay attention to the top-left corner of dimension 9×9):

$$\begin{aligned} [D_1 \otimes \mathbb{I}_9 + \mathbb{I}_3 \otimes D_2 \otimes \mathbb{I}_3 + \mathbb{I}_9 \otimes D_3]_{9 \times 9} &= \text{diag}(d_{11} + d_{12} + d_{13}, d_{11} + d_{12} + d_{23}, d_{11} + d_{12} + d_{33}, \\ &\quad d_{11} + d_{22} + d_{13}, d_{11} + d_{22} + d_{23}, d_{11} + d_{22} + d_{33}, \\ &\quad d_{11} + d_{32} + d_{13}, d_{11} + d_{32} + d_{23}, d_{11} + d_{32} + d_{33}). \end{aligned} \quad (3.24)$$

We recall that the above operator is the top-left corner submatrix of $D_1 \otimes \mathbb{I}_9 + \mathbb{I}_3 \otimes D_2 \otimes \mathbb{I}_3 + \mathbb{I}_9 \otimes D_3$ with dimension 9×9 . If we pay attention to the next block submatrix of the same dimension 9×9 , e.g., the submatrix whose row and column indices run from $9 \rightarrow 18$ of matrix $D_1 \otimes \mathbb{I}_9 + \mathbb{I}_3 \otimes D_2 \otimes \mathbb{I}_3 + \mathbb{I}_9 \otimes D_3$, then the structure is same as above, but the entry d_{11} is replaced by d_{12} . In a similar manner, the next submatrix of dimension 9×9 , or the submatrix whose row

and column indexes run from $18 \rightarrow 27$ of $D_1 \otimes \mathbb{I}_9 + \mathbb{I}_3 \otimes D_2 \otimes \mathbb{I}_3 + \mathbb{I}_9 \otimes D_3$, has the diagonal entries being similar as above, but d_{11} is replaced with d_{13} . What we can conclude from this observation is that

The matrix $D_1 \otimes \mathbb{I}_9 + \mathbb{I}_3 \otimes D_2 \otimes \mathbb{I}_3 + \mathbb{I}_9 \otimes D_3$ has the entries

$$\{(d_{i1} + d_{j2} + d_{k3})\}_{i,j,k=1}^3$$

in the diagonal.

By generalizing the above procedure to arbitrary value of p , we then have the following property, stated as a Lemma for subsequent reference:

Lemma 4. Let $d_G(x_i, y_j)$ be the geodesic distances between the points x_i and y_j from the given dataset. Define $d_G \equiv [d_G(x_i, y_j)]_{i,j=1}^p \equiv [d_{ij}]_{i,j=1}^p$ as the corresponding geodesic distance matrix of those points belonging to the neighbor of two points x, y of consideration.

Let D_G be the matrix of dimension $p \times p$ defined as $D_G \equiv \sum_{i,j=1}^p d_{ji} |i\rangle \langle i| \otimes |j\rangle \langle j|$, which contains the geodesic distances on the diagonal. Define p block submatrices as $D_i = \bigoplus_{j=1}^p d_{ji} |j\rangle \langle j|$ for $i = 1, 2, \dots, p$. Then, it holds that

$$D_p \equiv D_1 \otimes \mathbb{I}_p^{\otimes p-1} + \mathbb{I}_p \otimes D_2 \otimes \mathbb{I}_p^{\otimes p-2} + \dots + \mathbb{I}_p^{\otimes p-1} \otimes D_p$$

is diagonal and contains $\{d_{i1} + d_{i2} + \dots + d_{ip}\}_{i=1}^p$ in the diagonal.

However, we need to restrict to the cases where the indices i_k 's are all different. Namely, for the case ($p = q$) the objective reduces to finding the minimum of $\{\sum_{i=1}^p d_{\pi(i),i}\}_{i=1}^p$. The permutation requires that only a subset of $\{d_{i1} + d_{i2} + \dots + d_{ip}\}_{i=1}^p$ is of interest. In order to filter out those elements that correspond to the permutation of the indices, we shall construct a projector Π [see Eq. (3.26)] that achieves this goal. Let us first point out the following property. Within the matrix D_p , which has size $p^p \times p^p$, the entry having value $d_{i1} + d_{i2} + \dots + d_{ip}$ would have the location index (along the diagonal) as

$$k = (i_1 - 1)p^{p-1} + (i_2 - 1)p^{p-2} + \dots + i_p. \quad (3.25)$$

Because the summation $\sum_{i=1}^p d_{\pi(i),i}$ only contains the permutation for all i , it implies that the indices $i_1, i_2, \dots, i_p$ in the summation $d_{i1} + d_{i2} + \dots + d_{ip}$ must be different (no pair of indices having the same value). So, among all the diagonal entries, only those entries with the index $k = (i_1 - 1)p^{p-1} + (i_2 - 1)p^{p-2} + \dots + i_p$ with $i_1 \neq i_2 \neq \dots \neq i_p$ are of interest. To obtain these elements, we consider the projector

$$\Pi = \sum_{k: i_1 \neq i_2 \neq \dots \neq i_p} |k\rangle \langle k|, \quad (3.26)$$

whose construction is shown below right above Lemma C5. The product of this projector with D_p contains those $\{\sum_{i=1}^p d_{\pi(i),i}\}_{i=1}^p$ of interest.

Our problem then is to find the minimum diagonal entry of the matrix ΠD_p . This is trivially equivalent to the minimum (nonzero) eigenvalue of the matrix. However, before proceeding with the eigenvalue solver, we describe a procedure for constructing the block encoding of D_G , as well as that of $D_p, \Pi D_p$.

Block encoding of D_G . First, from Lemma 3, we have the block encoding of

$$\frac{1}{\alpha^{1/4}} \sum_{i=1}^N \sum_{j=1}^N |i-1\rangle \langle i-1| \otimes d_G(\mathbf{x}_i, \mathbf{x}_j) |j-1\rangle \langle j-1|, \quad (3.27)$$

which contains the geodesic distances between every pair of points in the given data. Our goal now is to first obtain the (block encoding of) operator D_G that appears in the above Lemma, which only includes certain data points of interest, before finding the minimum of $D_1 \otimes \mathbb{I}_p^{\otimes p} + \mathbb{I}_p \otimes D_2 \otimes \mathbb{I}_p^{\otimes p-1} + \dots + \mathbb{I}_p^{\otimes p} \otimes D_p$.

In the previous section, we have defined the classical indices $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p\}$ and $\{\mathbf{x}_{p+1}, \mathbf{x}_{p+2}, \dots, \mathbf{x}_{p+q}\}$ of the neighbors of x and y , respectively. Now we define the permutation operator U_p of dimension $N^2 \times N^2$ that acts as follows:

$$U_p |i\rangle |j\rangle \leftrightarrow |i\rangle |j-p\rangle, \quad \text{arbitrary for the remaining basis.} \quad (3.28)$$

An arbitrary permutation of a given dimension can be constructed with polylogarithmic resources [84,85]. That being said, the unitary U_p can be constructed using a quantum circuit of depth $\mathcal{O}(\log(N \log N))$. However, we note that we are only limiting ourselves to the space spanned by those neighbors of x and y (with the corresponding indexes $\{1, 2, 3, \dots, p, p+1, p+2, \dots, p+p\}$), which is effectively of dimension p^2 . Thus, the unitary U_p can be effectively realized with a circuit of depth $\mathcal{O}(\log(p \log p))$.

Given the unitary U_p , according to Definition B1, it block encodes itself. Then, we use Lemma B1 to construct the block encoding of

$$\begin{aligned} & U_p \frac{1}{\alpha^{1/4}} \sum_{i=1}^N \sum_{j=1}^N |i-1\rangle \langle i-1| \\ & \times \otimes d_G(\mathbf{x}_i, \mathbf{x}_j) |j-1\rangle \langle j-1| U_p^\dagger \\ & = \frac{1}{\alpha^{1/4}} \sum_{i,j=1}^p d_G(\mathbf{x}_i, \mathbf{x}_j) |i-1\rangle \langle i-1| \\ & \times \otimes |j-1\rangle \langle j-1| + (\dots) \\ & = \frac{1}{\alpha^{1/4}} \sum_{i,j=1}^p d_{ij} |i-1\rangle \langle i-1| \otimes |j-1\rangle \langle j-1| + (\dots), \end{aligned} \quad (3.29)$$

where $(\dots)$ refers to the unwanted part (which contains the distance between other pairs), and we only pay attention to the top-left submatrix of size $p \times p$.

Next, we define the SWAP operator that performs the swapping between two subsystems of dimension p , e.g., $|i-1\rangle \langle i-1| \leftrightarrow |j-1\rangle \langle j-1|$. This operator can be realized with a circuit of depth $\mathcal{O}(\log p)$ [84] (in fact, they are a specific kind of permutation). We then use Lemma B1 to

construct the block encoding of

$$\begin{aligned} \text{SWAP} &\times \frac{1}{\alpha^{1/4}} \sum_{i=1}^p \sum_{j=1}^p d_{ij} |i-1\rangle \langle i-1| \\ &\times \otimes |j-1\rangle \langle j-1| \text{SWAP}^\dagger \\ &= \frac{1}{\alpha^{1/4}} \sum_{i=1}^p \sum_{j=1}^p d_{ij} |j-1\rangle \langle j-1| \otimes |i-1\rangle \langle i-1|. \end{aligned} \quad (3.30)$$

Up to the scaling factor $\alpha^{1/4}$, the above operator contains the geodesic distances $\{d_{ij} \equiv d_G(\mathbf{x}_i, \mathbf{y}_j)\}$ on the diagonal. Defining the above operator multiplied by $\alpha^{1/4}$ as D_G , therefore, we have obtained the block encoding of $D_G/\alpha^{1/4}$.

Block encoding of $\mathcal{D}_p$. We now discuss how to construct the block encoding of $\mathcal{D}_p \equiv D_1 \otimes \mathbb{I}_p^{\otimes p} + \mathbb{I}_p \otimes D_2 \otimes \mathbb{I}_p^{\otimes p-1} + \dots + \mathbb{I}_p^{\otimes p} \otimes D_p$. First, we note that $D_G = \sum_{i=1}^p |i\rangle \langle i| \otimes D_i$. Next, we define p different permutations $U_1, \dots, U_p$ that act as follows: For any $i = 1, 2, \dots, p$,

$$U_i |i\rangle \leftrightarrow |0\rangle \quad (3.31)$$

and the remaining bases are permuted arbitrarily (for all i). Based on Refs. [84,85], these unitaries can be constructed using circuits of depth $\mathcal{O}(\log(p \log p))$. Next, for each i , we

use U_i that block encodes itself, and Lemma B 1 to construct the block encoding of

$$U_i \frac{D_G}{\alpha^{1/4}} U_i^\dagger = U_i \frac{\sum_{i=1}^p |i\rangle \langle i| \otimes D_i}{\alpha^{1/4}} U_i^\dagger \quad (3.32)$$

$$= \frac{1}{\alpha^{1/4}} |0\rangle \langle 0| \otimes D_i + (\dots), \quad (3.33)$$

where $(\dots)$ refers to the unwanted part. It can be seen that, from Definition B 1, the above operator is exactly the block encoding of $\frac{1}{\alpha^{1/4}} D_i$. Repeating the same procedure for different permutations U_i 's, we obtain the block encoding of all $\frac{1}{\alpha^{1/4}} D_i$ for $i = 1, 2, \dots, p$.

The remaining task for constructing $\mathcal{D}_p$ is straightforward as follows. As the block encoding for the identity matrix of arbitrary dimension can be constructed easily (see Definition B 1), we can employ Lemma B 2 to construct the block encoding of the following operators:

$$\frac{1}{\alpha^{1/4}} D_1 \otimes \mathbb{I}_p^{\otimes p-1}, \frac{1}{\alpha^{1/4}} \mathbb{I}_p \otimes D_2 \otimes \mathbb{I}_p^{\otimes p-2}, \dots, \frac{1}{\alpha^{1/4}} \mathbb{I}_p^{\otimes p-1} \otimes D_p. \quad (3.34)$$

Then, we use Lemma B 3 to construct the block encoding of their linear combination,

$$\frac{1}{p} \left(\frac{1}{\alpha^{1/4}} D_1 \otimes \mathbb{I}_p^{\otimes p-1} + \frac{1}{\alpha^{1/4}} \mathbb{I}_p \otimes D_2 \otimes \mathbb{I}_p^{\otimes p-2} + \dots + \frac{1}{\alpha^{1/4}} \mathbb{I}_p^{\otimes p-1} \otimes D_p \right) \quad (3.35)$$

$$= \frac{1}{p\alpha^{1/4}} (D_1 \otimes \mathbb{I}_p^{\otimes p-1} + \mathbb{I}_p \otimes D_2 \otimes \mathbb{I}_p^{\otimes p-2} + \dots + \mathbb{I}_p^{\otimes p-1} \otimes D_p) = \frac{1}{p\alpha^{1/4}} \mathcal{D}_p. \quad (3.36)$$

Block encoding of Π . Next, we show how to obtain the block encoding of the projector Π in Eq. (3.26) that allows us to restrict the indices to the permutations only, as remarked earlier. We first consider the state $\frac{1}{\sqrt{p!}} \sum_{k:i_1 \neq i_2 \neq \dots \neq i_p} |k\rangle$, which can be prepared using any of the state preparation methods in Refs. [82,86–88]. These methods generally have circuit complexity $\mathcal{O}(\log(p!))$. However, the method in Ref. [82] would require $\mathcal{O}(p!)$ ancilla qubit; meanwhile, the methods in Refs. [86,87] require $\mathcal{O}(1)$ ancilla qubits; see, e.g., Theorem 1 of Ref. [86]. In practice, the value of $p = \mathcal{O}(1)$; therefore, practically either of these methods can be used. However, we prefer the methods in Ref. [86,87] as it require less ancilla qubits. Given the preparation of such a state, then we append the ancilla initialized in $|0\rangle^{\otimes \log p^p}$, and use cnot gates to transform $\frac{1}{\sqrt{p!}} \sum_{k:i_1 \neq i_2 \neq \dots \neq i_p} |k\rangle |0\rangle^{\otimes \log p^p}$ to $\frac{1}{\sqrt{p!}} \sum_{k:i_1 \neq i_2 \neq \dots \neq i_p} |k\rangle |k\rangle$. Tracing out either of these register yields the normalized projector $\frac{1}{p!} \Pi$, which can be block encoded via the following Lemma:

Lemma 5. (Block encoding of density matrix; see, e.g., Ref. [75]). Let $\rho = \text{Tr}_A |\Phi\rangle \langle \Phi|$, where $\rho \in \mathbb{H}_B$ and $|\Phi\rangle \in \mathbb{H}_A \otimes \mathbb{H}_B$. Given a unitary U that prepares $|\Phi\rangle$ from $|\mathbf{0}\rangle_A \otimes |\mathbf{0}\rangle_B$, there exists a highly efficient procedure that constructs an exact unitary block encoding of ρ using U and $U^\dagger$ once each.

Block encoding of $\Pi \mathcal{D}_p$. From the block encoding of $\frac{1}{p!} \Pi$ and of $\frac{1}{p\alpha^{1/4}} \mathcal{D}_p$, we can use Lemma B 1 to construct the block encoding of $\frac{1}{p! p\alpha^{1/4}} \Pi \mathcal{D}_p$.

Finding the Earth Mover distance. Earlier, we have mentioned that the operator $\Pi \mathcal{D}_p$ contains the summations $\{\sum_{i=1}^p d_{\pi(i),i}\}_i^p$ of interest along the diagonal. It can be seen that for a diagonal matrix, the minimum entry along the diagonal is also the minimum eigenvalue. Therefore, our challenge now is to find the minimum eigenvalue of this block-encoded operator $\frac{1}{(p+1)! \alpha^{1/4}} \Pi \mathcal{D}_p$, which contains the desired minimum value of the Earth Mover distance $W_1(x, y)$ (up to a scaling). To this end, we point out the following result of Refs. [72,73], which is based on the classical power method for finding the largest eigenvalues/eigenvectors:

Lemma 6. Let $A \in \mathbb{R}^{n \times n}$ be a Hermitian matrix of operator norm $\|A\|_{\text{operator}} \leq 1$. Let U_A be the block encoding of A with circuit complexity T_A . Then, its largest eigenvalue $\psi_{\max}$ can be estimated up to an additive error ϵ using a quantum circuit of depth

$$\mathcal{O}\left(T_A \left(\frac{1}{\Delta \gamma \epsilon}\right) \log\left(\frac{n}{\epsilon}\right) \log\left(\frac{1}{\epsilon}\right)\right),$$

where Δ is the gap between the largest and the second largest eigenvalue. γ is defined as $|\langle \psi_0, \psi_{\max} \rangle|$, where $|\psi_{\max}\rangle$ is the

eigenstate corresponding to the largest eigenvalue, and $|\psi_0\rangle$ is some randomly initialized state.

While the above lemma aims at finding the largest eigenvalues/eigenvectors, it can be adapted to deal with the minimum ones by considering the matrix A' to be the inverse of A or the pseudoinverse of A if A is singular. In either case, the maximum nonzero eigenvalue of A' is equal to $\frac{1}{\psi_{\min}}$. An estimation of the largest eigenvalue of A' can thus be used to infer the minimum eigenvalue $\psi_{\min}$ of A . The block encoding of $\frac{A^{-1}}{\kappa_A}$ (where κ_A is the condition number of A) can be obtained from A by using matrix inversion technique of Ref. [30], or QSVT with a proper polynomial approximation to A^{-1} ; e.g., see Corollary 67 in Ref. [75] and the discussion below the corollary. As the inversion incurs a circuit complexity by $\mathcal{O}(\kappa_A \log \frac{1}{\epsilon})$, so the total complexity for using the above lemma to find the minimum eigenvalue of A is

$$\mathcal{O}\left(T_A \kappa_A \left(\frac{1}{\Delta \gamma \epsilon}\right) \log\left(\frac{n}{\epsilon}\right) \log^2\left(\frac{1}{\epsilon}\right)\right).$$

The application of the above discussion to our context is straightforward to estimate

$$\min \frac{1}{p! p \alpha^{1/4}} \left\{ \sum_{i=1}^p d_{\pi(i),i} \right\}_{i=1}^p \equiv \frac{W_1(x, y)}{p! \alpha^{1/4}}, \quad (3.37)$$

from which the value of $W_1(x, y)$ can be inferred by multiplying such estimation with $p! \alpha^{1/4}$.

Complexity. To analyze the complexity, we recapitulate the algorithm above as follows:

Step 1. Obtain the block encoding of

$$\frac{1}{\alpha^{1/4}} \sum_{i,j=1}^N |i-1\rangle \langle i-1| \otimes d_G(\mathbf{x}_i, \mathbf{x}_j) |j-1\rangle \langle j-1| \quad (3.38)$$

via Lemma 3, which involves a quantum circuit of depth

$$\mathcal{O}\left(\log(N) \log^2\left(\frac{1}{\epsilon}\right) \kappa \log^2 \frac{\kappa}{\epsilon}\right). \quad (3.39)$$

Step 2. From the block encoding of the above operator, construct the block encoding of

$$\frac{1}{\alpha^{1/4}} \sum_{m,k=1}^p d_{mk} |m-1\rangle \langle m-1| \otimes |k-1\rangle \langle k-1| + (\dots),$$

which uses further a permutation operator and a SWAP operator of dimension $p \times p$, with circuit depth $\mathcal{O}(\log(p \log p))$. So, the total complexity is

$$\mathcal{O}\left(\log(N) \log^2\left(\frac{1}{\epsilon}\right) \kappa \log^2 \frac{\kappa}{\epsilon} + \log(p \log p)\right). \quad (3.40)$$

Step 3. Use permutation operator of dimension $p \times p$ again to construct the block encoding of $\frac{D_i}{\alpha^{1/4}}$ for all $i = 1, 2, \dots, p$. The total complexity at this point is

$$\mathcal{O}\left(\log(N) \log^2\left(\frac{1}{\epsilon}\right) \kappa \log^2 \frac{\kappa}{\epsilon} + \log(p \log p)\right). \quad (3.41)$$

Step 4. Use Lemma B 3 to construct the block encoding of

$$A \equiv \frac{(D_1 \otimes \mathbb{I}_p^{\otimes p-1} + \mathbb{I}_p \otimes D_2 \otimes \mathbb{I}_p^{\otimes p-2} + \dots + \mathbb{I}_p^{\otimes p-1} \otimes D_p)}{p \alpha^{1/4}},$$

which incurs a total quantum circuit depth

$$\mathcal{O}\left(p \log(N) \log^2\left(\frac{1}{\epsilon}\right) \kappa \log^2 \frac{\kappa}{\epsilon} + p \log(p \log p)\right). \quad (3.42)$$

Step 5. Use the modified version of Lemma 6, with A being defined as above to find the minimum eigenvalue, which can be used to infer the desired value $W_1(x, y)$ (up to the scaling $p! \alpha^{1/4}$), resulting in the overall complexity:

$$\mathcal{O}\left(\left(\log(N) \log^2\left(\frac{1}{\epsilon}\right) \kappa \log^2\left(\frac{\kappa}{\epsilon}\right) + \log(p \log p)\right) p \log\left(\frac{p}{\epsilon}\right) \times \left(\frac{1}{\Delta \gamma \epsilon}\right) \log^2 \frac{1}{\epsilon}\right), \quad (3.43)$$

where in this case Δ is defined as the gap between the smallest and the second smallest eigenvalue of the operator A , and we recall that $\kappa = \min_{i,j,q,k} \left\{ \frac{d_G(\mathbf{x}_i, \mathbf{x}_j)}{d_G(\mathbf{x}_k, \mathbf{x}_q)} \right\}$.

Discussion. As indicated above, the complexity depends on a parameter γ , which is defined as $|\langle \psi_0, \psi_{\min} \rangle|$, where $|\psi_0\rangle$ is the initially randomized vector and $|\psi_{\min}\rangle$ is the eigenstate corresponding to the minimum eigenvalue of A above. As it involves a randomization, the value of γ also behaves probabilistically. If we are lucky to have large overlaps, then γ is of $\mathcal{O}(1)$. Otherwise, in the worst-case scenario, γ can be exponentially small as the dimension of matrix, and thus the complexity could scale as $\mathcal{O}(p^p)$, which is exponential in p . Therefore, in order for our method to maintain its efficiency, the value of p should be small, of order $\mathcal{O}(1)$. This can be achieved by choosing a small neighborhood around x and y .

Earlier, we have noted that our definition of Earth Mover distance does not include the points x, y , and that adding these points (according to Refs. [63,64]) would not change the procedure as well as the complexity of our algorithm.

IV. CONCLUSION

In this work, we have constructed a quantum algorithm for estimating the Ollivier-Ricci curvature in two specific classes of problems, an important quantity with practical applications. Our work incorporated a wide range of methods, including optimal transport theory (specifically its discrete version), linear programming, and, notably, the block-encoding/QSVT framework introduced in Ref. [75]. By leveraging the recent result in Ref. [60], we are able to obtain the block encoding of an operator that contains the geodesic distances between data points as diagonal entries. From such a (block-encoded) operator, we showed that a series of QSVT techniques can be applied appropriately, resulting in the estimation of the Earth Mover (1-Wasserstein) distance $W_1(x, y)$ for two classes of the problem: (1) when G forms a tree and (2) the case $p = q$; i.e., number of the respective neighbors of two points on an edge (x, y) is the same. Estimating the Ollivier-Ricci curvature in these cases can be straightforwardly done by quantum methods. We then provide an analysis showing that our quantum algorithm admits logarithmic running time in the number of data points, and also the (local) size of the neighborhood of interest. Thus, our quantum algorithm offers an exponential speedup compared to the classical algorithm. Our result has strongly highlighted the potential of quantum computation for geometrical data analysis, adding another

example to our previous attempt [60]. What other problems in which our technique can be proved useful is thus of high interest. For example, the arbitrary case of the Ollivier-Ricci curvature remains open and is left for future exploration.

ACKNOWLEDGMENTS

This work was supported by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers, Co-design Center for Quantum Advantage (C2QA) under Contract No. DE-SC0012704. N.A.N. and T.-C.W. also acknowledge support from the Center for Distributed Quantum Processing at Stony Brook Uni-

versity. N.A.N. thanks the hospitality of Harvard University where he has an academic visit during the completion of this project.

DATA AVAILABILITY

There are no publicly available research data or software supporting this manuscript. Requests for further information or data should be sent to the authors.

APPENDIX A: CALCULATING ORC

Here, we show the concrete steps required to estimate the ORC associated with edge (x, y) of the graph in Fig. 1. The geodesic distances between pairs of relevant points for this calculation are

$$\begin{aligned} d_G(x_1, y_1) &= 1, & d_G(x_1, y_2) &= 3, & d_G(x_1, y_3) &= 3 & d_G(x_1, y_4) &= 2, \\ d_G(x_2, y_1) &= 2, & d_G(x_2, y_2) &= 3, & d_G(x_2, y_3) &= 3 & d_G(x_2, y_4) &= 3, \\ d_G(x_3, y_1) &= 3, & d_G(x_3, y_2) &= 2, & d_G(x_3, y_3) &= 2 & d_G(x_3, y_4) &= 3. \end{aligned} \quad (\text{A1})$$

We need to find the non-negative $3 \times 4 = 12$ values $\{\gamma_{ij}\}_{i=1, j=1}^{3,4}$ that minimize the following linear combination:

$$\begin{aligned} \mathcal{W}(\{\gamma_{ij}\}) &= d_G(x_1, y_1)\gamma_{11} + d_G(x_1, y_2)\gamma_{12} + d_G(x_1, y_3)\gamma_{13} + d_G(x_1, y_4)\gamma_{14} \\ &+ d_G(x_2, y_1)\gamma_{21} + d_G(x_2, y_2)\gamma_{22} + d_G(x_2, y_3)\gamma_{23} + d_G(x_2, y_4)\gamma_{24} \\ &+ d_G(x_3, y_1)\gamma_{31} + d_G(x_3, y_2)\gamma_{32} + d_G(x_3, y_3)\gamma_{33} + d_G(x_3, y_4)\gamma_{34} \\ &= \gamma_{11} + 3\gamma_{12} + 3\gamma_{13} + 2\gamma_{14} + 2\gamma_{21} + 3\gamma_{22} + 3\gamma_{23} + 3\gamma_{24} + 3\gamma_{31} + 2\gamma_{32} + 2\gamma_{33} + 3\gamma_{34}, \end{aligned} \quad (\text{A2})$$

subjected to the constraints

$$\begin{aligned} \gamma_{11} + \gamma_{12} + \gamma_{13} + \gamma_{14} &= \gamma_{21} + \gamma_{22} + \gamma_{23} + \gamma_{24} \\ &= \gamma_{31} + \gamma_{32} + \gamma_{33} + \gamma_{34} = 1/3 \end{aligned} \quad (\text{A3})$$

and

$$\begin{aligned} \gamma_{11} + \gamma_{21} + \gamma_{31} &= \gamma_{12} + \gamma_{22} + \gamma_{32} \\ &= \gamma_{13} + \gamma_{23} + \gamma_{33} \\ &= \gamma_{14} + \gamma_{24} + \gamma_{34} = 1/4. \end{aligned} \quad (\text{A4})$$

The minimum value of $\mathcal{W}$ can be found to be $25/12$, located at

$$\begin{aligned} \gamma_{11} &= 1/4, & \gamma_{14} &= 1/12, & \gamma_{23} &= 1/6, \\ \gamma_{24} &= 1/6, & \gamma_{32} &= 1/4, & \gamma_{33} &= 1/12, \end{aligned}$$

and other γ values are 0. From this result, we can use Eq. (2.3) to estimate the Earth Mover distance

$$W_1(x, y) = \min_{\{\gamma_{ij}\}} \mathcal{W}(\{\gamma_{ij}\}) = 25/12.$$

The geodesic distance between point x and point y is of unit length, i.e., $d_G(x, y) = 1$; therefore we can obtain the estimated ORC value from Eq. (2.2):

$$\gamma(x, y) = 1 - \frac{W_1(x, y)}{d_G(x, y)} = 1 - \frac{25/12}{1} = -\frac{13}{12}. \quad (\text{A5})$$

APPENDIX B: BLOCK ENCODING AND QUANTUM SINGULAR VALUE TRANSFORMATION

We briefly summarize the essential quantum tools used in our algorithm. For conciseness, we highlight only the main results and omit technical details, which are thoroughly covered in Ref. [75]. An identical summary is also presented in Ref. [89].

Definition B1 (Block-encoding unitary; see, e.g., Refs. [75–77]). Let A be a Hermitian matrix of size $N \times N$ with operator norm $\|A\| < 1$. A unitary matrix U is said to be an *exact block encoding* of A if

$$U = \begin{pmatrix} A & * \\ * & * \end{pmatrix}, \quad (\text{B1})$$

where the top-left block of U corresponds to A . Equivalently, one can write

$$U = |\mathbf{0}\rangle \langle \mathbf{0}| \otimes A + (\cdots), \quad (\text{B2})$$

where $|\mathbf{0}\rangle$ denotes an ancillary state used for block encoding and $(\cdots)$ represents the remaining components orthogonal to $|\mathbf{0}\rangle \langle \mathbf{0}| \otimes A$. If instead U satisfies

$$U = |\mathbf{0}\rangle \langle \mathbf{0}| \otimes \tilde{A} + (\cdots), \quad (\text{B3})$$

for some $\tilde{A}$ such that $\|\tilde{A} - A\| \leq \epsilon$, then U is called an ϵ -approximate block encoding of A . Furthermore, the action of U on a state $|\mathbf{0}\rangle |\phi\rangle$ is given by

$$U |\mathbf{0}\rangle |\phi\rangle = |\mathbf{0}\rangle A |\phi\rangle + |\text{Garbage}\rangle, \quad (\text{B4})$$

where $|\text{Garbage}\rangle$ is a state orthogonal to $|\mathbf{0}\rangle A |\phi\rangle$. The circuit complexity (e.g., depth) of U is referred to as the complexity of block encoding A .

Based on Definition B 1, several properties, though immediate, are of particular importance and are listed below.

Remark B 1 (Properties of block-encoding unitary). The block-encoding framework has the following immediate consequences:

(1) Any unitary U is trivially an exact block encoding of itself.

(2) If U is a block encoding of A , then so is $\mathbb{I}_m \otimes U$ for any $m \geq 1$.

(3) The identity matrix $\mathbb{I}_m$ can be trivially block encoded, for example, by $\sigma_z \otimes \mathbb{I}_m$.

Given a set of block-encoded operators, various arithmetic operations can be done with them. Here, we simply introduce some key operations that are especially relevant to our algorithm, focusing on how they are implemented and their time complexity, without going into proofs. For more detailed explanations, see Refs. [75,90].

Lemma B 1 (Informal, product of block-encoded operators; see, e.g., Ref. [75]). Given unitary block encodings of two matrices A_1 and A_2 , with respective implementation complexities T_1 and T_2 , there exists an efficient procedure for constructing a unitary block encoding of the product $A_1 A_2$ with complexity $T_1 + T_2$.

Lemma B 2 (Informal, tensor product of block-encoded operators; see, e.g., (Theorem 1 of Ref. [90]). Given unitary block encodings $\{U_i\}_{i=1}^m$ of multiple operators $\{M_i\}_{i=1}^m$ (assumed to be exact), there exists a procedure that constructs a unitary block encoding of $\bigotimes_{i=1}^m M_i$ using a single application of each U_i and $\mathcal{O}(1)$ swap gates.

Lemma B 3 (Informal, linear combination of block-encoded operators; see, e.g., (Theorem 52 of Ref. [75]). Given the unitary block encoding of multiple operators $\{A_i\}_{i=1}^m$. Then, there is a procedure that produces a unitary block encoding operator of $\sum_{i=1}^m \pm(A_i/m)$ in time complexity $\mathcal{O}(m)$, e.g., using the block encoding of each operator A_i a single time.

Lemma B 4 (Informal, scaling multiplication of block-encoded operators). Given a block encoding of some matrix A , as in Definition B 1, the block encoding of A/p where $p > 1$ can be prepared with an extra $\mathcal{O}(1)$ cost.

Lemma B 5 (Matrix inversion; see, e.g., Refs. [30,75]). Given a block encoding of some matrix A with operator norm $\|A\| \leq 1$ and block-encoding complexity T_A , then there is a quantum circuit producing an ϵ -approximated block encoding of A^{-1}/κ , where κ is the conditional number of A . The complexity of this quantum circuit is $\mathcal{O}(\kappa T_A \log(1/\epsilon))$.

APPENDIX C: PROOF OF LEMMA 1

Here we directly quote the proof of Lemma 1, which was worked out in Ref. [60]. An identical proof can be found in the same work.

Provided the raw distance $\{d(\mathbf{x}_i, \mathbf{x}_j)\}$ among data points, the kernel matrix K is defined as

$$K_{ij} = \exp\left(-\frac{d(\mathbf{x}_i, \mathbf{x}_j)^2}{\sigma^2}\right), \quad (\text{C1})$$

where σ is a chosen parameter.

1. Quantum algorithm for block encoding the kernel matrix K

We first mention the following technique introduced in Ref. [91]:

Lemma C 1. Let U (assumed to have depth T_x) be some unitary of dimension $> N \times N$ that contains a vector $\mathbf{x} = \sum_{i=1}^N x_i |i-1\rangle$ as the first column. Then, there is a quantum circuit of depth $\mathcal{O}(T_x + \log(N))$, which is a block encoding of a matrix that contains $\sum_{i=1}^N x_i^2 |i-1\rangle$ as the first column.

For completeness, we directly quote their proof as follows.

Proof. First, we consider the matrix of size $N \times N$ on the top-left corner of U . Denote this matrix as A_x , and the first column of this matrix is $\mathbf{x}$. It can also be seen that U_x is the block encoding of A_x . Then, we can use Lemma B 2 to construct the block encoding of $A_x \otimes A_x$. The first column of this matrix is $\mathbf{x} \otimes \mathbf{x}$, which is

$$\mathbf{x} \otimes \mathbf{x} = \sum_{i=1}^n x_i |i-1\rangle \otimes \sum_{i=1}^n x_i |i-1\rangle \quad (\text{C2})$$

$$= \sum_{i,j=1}^n x_i x_j |i-1\rangle |j-1\rangle \quad (\text{C3})$$

$$= \sum_{i=1}^n x_i^2 |i-1\rangle |i-1\rangle + \sum_{i \neq j} x_i x_j |i-1\rangle |j-1\rangle. \quad (\text{C4})$$

Next, we use the permutation:

Lemma C 2 (Refs. [84,85]). Let $\mathcal{H}$ be some N -dimensional Hilbert space and $\{|0\rangle, |1\rangle, |2\rangle, \dots, |N\rangle\}$ are basis. Then, for a known permutation of basis $\{|i-1\rangle \leftrightarrow |j\rangle\}$, there exists a permutation unitary circuit $U_{\text{permutation}}$ of depth $\mathcal{O}(\log(N \log N))$.

We achieve the following permutation:

$$|0\rangle |0\rangle \leftrightarrow |0\rangle |0\rangle, \quad (\text{C5})$$

$$|0\rangle |1\rangle \leftrightarrow |1\rangle |1\rangle, \quad (\text{C6})$$

$$\vdots \quad (\text{C7})$$

$$|0\rangle |n-1\rangle \leftrightarrow |n-1\rangle |n-1\rangle, \quad (\text{C8})$$

and the remaining basis permutation can be arbitrary. We use such permutation unitary U_{permute} and Lemma B 1 to construct the block encoding of $U_{\text{permute}}(A_x \otimes A_x)$. The first column of this matrix is

$$U_{\text{permute}}(\mathbf{x} \otimes \mathbf{x}) = U_{\text{permute}} \sum_{i=1}^n x_i^2 |i-1\rangle |i-1\rangle + \sum_{i \neq j} x_i x_j |i-1\rangle |j-1\rangle \quad (\text{C9})$$

$$= \sum_{i=1}^n x_i^2 |0\rangle |i-1\rangle + (\dots), \quad (\text{C10})$$

where $(\dots)$ refers to the redundant part. The first part is $\sum_{i=1}^n x_i^2 |0\rangle |i-1\rangle$. Therefore, if we restrict to the top-left corner matrix of dimension $n \times n$, then the first column is exactly $\sum_{i=1}^n x_i^2 |i-1\rangle$. ■

The above procedure can be modified to yield the following more general lemma:

Lemma C3. Let U (assumed to have depth T_x) be some unitary of dimension $> N \times N$ that contains a vector $\mathbf{x} = \sum_{i=1}^N x_i |i-1\rangle$ as the first column. For some $p \in \mathbb{Z}$, there is a quantum circuit of depth $\mathcal{O}(T_x + p \log(N))$, which is a block encoding of a matrix that contains $\sum_{i=1}^N x_i^p |i-1\rangle$ as the first column.

The proof of the above lemma, or the modification of the procedure, is as follows. Earlier, we consider the state $\mathbf{x} \otimes \mathbf{x}$ and then use permutation to move those entries $\{x_i^2\}_{i=1}^N$. Now we simply need to consider the state $\mathbf{x}^{\otimes p}$ and consider the following permutation:

$$|0\rangle^{\otimes p-1} |0\rangle \leftrightarrow |0\rangle^{\otimes p}, \quad (\text{C11})$$

$$|0\rangle^{\otimes p-1} |1\rangle \leftrightarrow |1\rangle^{\otimes p}, \quad (\text{C12})$$

$$\vdots \quad (\text{C13})$$

$$|0\rangle^{\otimes p-1} |n-1\rangle \leftrightarrow |n-1\rangle^{\otimes p}, \quad (\text{C14})$$

with the rest of the basis state permuted arbitrarily. According to Refs. [84,85], the above permutation can be obtained with a circuit of complexity $\mathcal{O}(p \log N)$. ■

Recall that the kernel matrix K is defined as

$$K_{ij} = \exp\left(-\frac{d(\mathbf{x}_i, \mathbf{x}_j)^2}{\sigma^2}\right), \quad (\text{C15})$$

where $d(\mathbf{x}_i, \mathbf{x}_j)$ is the pairwise distance between two data points $\mathbf{x}_i, \mathbf{x}_j$. To proceed, first we need the following recipe, which shall be proved in Appendix C3:

Lemma C4 (Efficient state preparation; Appendix C3).

An n -dimensional quantum state $|\Phi\rangle$ with known entries (assuming they are normalized to 1) can be prepared with fidelity $1 - \epsilon$ with a circuit of depth $\mathcal{O}(\log n + \log(1 - \epsilon))$, using $\mathcal{O}(1)$ ancilla qubits.

The classical knowledge of pairwise distance $d_{ij} \equiv d(\mathbf{x}_i, \mathbf{x}_j)$ and the above lemma allow us to obtain the unitary U_d that prepares the following state:

$$|\phi\rangle = \frac{1}{\|D\|} \sum_{i,j=1}^N |i-1\rangle d_{ij} |j\rangle, \quad (\text{C16})$$

where $\|D\| = \sqrt{\sum_{i,j=1}^N d_{ij}^2}$. It is straightforward to see that the first column of the unitary U_d is $|\phi\rangle$. Our goal is to obtain the state (encoded in some matrix) $\propto \sum_{i,j=1}^N |i-1\rangle \exp(-\frac{d_{ij}^2}{\sigma^2}) |j\rangle$ from the unitary U_d . To achieve this, we first use Lemmas C3 and C1 to obtain the block encodings of matrices having the first columns as

$$\begin{aligned} & \frac{1}{\|D\|^p} \sum_{i,j=1}^N |i-1\rangle d_{ij}^p |j\rangle, \\ & \frac{1}{\|D\|^{p-1}} \sum_{i,j=1}^N |i-1\rangle d_{ij}^{p-1} |j\rangle, \dots, \frac{1}{\|D\|} \sum_{i,j=1}^N |i-1\rangle d_{ij} |j\rangle. \end{aligned} \quad (\text{C17})$$

Next, we point out the following approximation from Ref. [92]:

Proposition C1 (Ref. [92]). On any compact interval $[-a, a]$, we have that the Gaussian function $\exp(-x^2)$ is infinitely differentiable and analytic, and that

$$\sup_{x \in [-a, a]} |\exp(-x^2) - P_p(x)| \leq C \exp(-\alpha p), \quad (\text{C18})$$

where $p_n(x)$ is the Chebyshev polynomial, which satisfies the following recurrence relation:

$$P_0(x) = 1, \quad P_1(x) = x, \quad (\text{C19})$$

$$P_{p+1}(x) = 2xP_p(x) - P_{p-1}(x). \quad (\text{C20})$$

Fact. The values of C and α in the above depend on the value of a . For $a = 1$, then as analyzed in Ref. [92], $C \approx 0.1$, $\alpha \approx 1.09$, which are both $\mathcal{O}(1)$.

To apply the above result to our problem, we first need to figure out the value of p , which is the degree of the polynomial for approximation. By setting $C \exp(-\alpha p) = \epsilon$, we have that

$$p = \frac{1}{\alpha} \log\left(\frac{C}{\epsilon}\right) = \mathcal{O}\left(\log \frac{1}{\epsilon}\right). \quad (\text{C21})$$

For convenience, let the polynomial $P_p(x) = \sum_{k=1}^p \alpha_k x^k$. We then use the block encodings of matrices containing states in Eq. (C17) and Lemma B3 to construct the unitary block encoding, denoted as U_D , of a matrix, which has the following vector as the first column:

$$\begin{aligned} & \frac{1}{\alpha} \sum_{i,j=1}^N |i-1\rangle P_p\left(\frac{d_{ij}}{\|D\|}\right) |j\rangle \\ & \approx \frac{1}{\alpha} \sum_{i,j=1}^N |i-1\rangle \exp\left(-\frac{d_{ij}^2}{\|D\|^2}\right) |j\rangle, \end{aligned} \quad (\text{C22})$$

where $\alpha = \sqrt{\sum_{k=1}^p \alpha_k^2}$. We note that, if we choose $\sigma = \|D\|$, then the entry $\exp(-\frac{d_{ij}^2}{\|D\|^2})$ is exactly the entry of kernel K defined earlier. On the other hand, if we wish to choose $\sigma \neq \|D\|$, then we can slightly modify the above procedure as follows. We use Lemma B4 to transform the block-encoded columns in Eq. (C17):

$$\frac{1}{\alpha \|D\|^p} \sum_{i,j=1}^N |i-1\rangle d_{ij}^p |j\rangle \longrightarrow \frac{1}{\alpha \|D\|^p} \sum_{i,j=1}^N |i-1\rangle \frac{d_{ij}^p}{\sigma^p} |j\rangle, \quad (\text{C23})$$

$$\begin{aligned} & \frac{1}{\alpha \|D\|^{d-1}} \sum_{i,j=1}^N |i-1\rangle d_{ij}^{p-1} |j\rangle \longrightarrow \frac{1}{\alpha \|D\|^p} \\ & \times \sum_{i,j=1}^N |i-1\rangle \frac{d_{ij}^{p-1}}{\sigma^{p-1}} |j\rangle, \end{aligned} \quad (\text{C24})$$

$$\vdots \quad (\text{C25})$$

$$\frac{1}{\alpha \|D\|} \sum_{i,j=1}^N |i-1\rangle d_{ij} |j\rangle \longrightarrow \frac{1}{\alpha \|D\|^p} \sum_{i,j=1}^N |i-1\rangle \frac{d_{ij}}{\sigma} |j\rangle. \quad (\text{C26})$$

Then, we use Lemma B 3 to construct the block encoding of a matrix that has the following vector as the first column:

$$\begin{aligned} & \frac{1}{\alpha \|D\|^p} \sum_{i,j=1}^N |i-1\rangle P_p\left(\frac{d_{ij}}{\sigma}\right) |j\rangle \\ & \approx \frac{1}{\alpha \|D\|^p} \sum_{i,j=1}^N |i-1\rangle \exp\left(-\frac{d_{ij}^2}{\sigma^2}\right) |j\rangle. \end{aligned} \quad (\text{C27})$$

To obtain the block encoding of K , we point out the following result from Ref. [75]:

Lemma C 5 (Block encoding density matrix [75]). Let $\rho = \text{Tr}_A |\Phi\rangle\langle\Phi|$, where $\rho \in \mathbb{H}_B$, $|\Phi\rangle \in \mathbb{H}_A \otimes \mathbb{H}_B$. Given unitary U that generates $|\Phi\rangle$ from $|0\rangle_A \otimes |0\rangle_B$, then there exists a highly efficient procedure that constructs an exact unitary block encoding of ρ using U and $U^\dagger$ a single time, respectively.

Now we take the unitary U_D and apply it to the state $|0\rangle|0\rangle_{N^2}$ (where $|0\rangle_{N^2}$ denotes the first computational basis of an N^2 -dimensional Hilbert space); according to Definition B 1, we have

$$\begin{aligned} U_D |0\rangle|0\rangle_{N^2} &= \frac{1}{\alpha \|D\|^p} |0\rangle \sum_{i,j=1}^N |i-1\rangle \exp\left(-\frac{d_{ij}^2}{\|\sigma\|^2}\right) |j\rangle \\ &+ \sum_{k \neq 0} |k\rangle |\text{Garbage}_k\rangle. \end{aligned} \quad (\text{C28})$$

By tracing out the third register, we obtain the density state

$$\begin{aligned} \rho &= |0\rangle\langle 0| \otimes \frac{1}{\alpha^2 \|D\|^{2p}} K^\dagger K \\ &+ \sum_{k \neq 0} |k\rangle\langle k| \otimes |\text{Garbage}_k\rangle\langle \text{Garbage}_k|. \end{aligned} \quad (\text{C29})$$

The above density state is again a block encoding of $\alpha K^\dagger K$, and can be block encoded via Lemma C 5.

2. Quantum algorithm for obtaining geodesic distances

We recall that the geodesic distance between two data points $\mathbf{x}_i, \mathbf{x}_j$ is approximated as

$$d_G(\mathbf{x}_i, \mathbf{x}_j) \approx \left[\sum_{k=1}^N \lambda_k^{2t} (\psi_{ik} - \psi_{jk})^2 \right]^{1/2}, \quad (\text{C30})$$

where $|\psi_k\rangle_i$ refers to the i th component of the k th eigenvector $|\psi_k\rangle$ of K and λ_k is the corresponding eigenvalue of K . For convenience, we set $t = 1$, so that the geodesic is further simplified as

$$d_G(\mathbf{x}_i, \mathbf{x}_j) \approx \left[\sum_{k=1}^N \lambda_k^2 (\psi_{ik} - \psi_{jk})^2 \right]^{1/2}. \quad (\text{C31})$$

We remark that from the previous section, we have the block encoding of $\alpha K^\dagger K$, which is also $\propto \sum_{k=1}^N \lambda_k^2 |\psi_k\rangle\langle\psi_k|$. To proceed, we mention the following QSVT recipe:

Lemma C 6 (Negative power exponent [75,78]). Given a block encoding of a positive matrix $\frac{\mathcal{M}}{\gamma}$ such that

$$\frac{\mathbb{I}}{\kappa_M} \leq \frac{\mathcal{M}}{\gamma} \leq \mathbb{I}.$$

Then, we can implement an ϵ -approximated block encoding of $\mathcal{M}^{-c}/(2\kappa_M^c)$ in complexity $\mathcal{O}(\kappa_M T_M (1+c) \log^2(\frac{\gamma \kappa_M^{1+c}}{\epsilon}))$, where T_M is the complexity to obtain the block encoding of $\mathcal{M}$.

We point out the following property:

$$|\psi_k\rangle_i - |\psi_k\rangle_j = e_{ij}^T |\psi_k\rangle, \quad (\text{C32})$$

where e_{ij} is the vector (of dimension N) that has entry 1 at i th position, -1 at j th position, and 0 otherwise. Let E_i be the matrix having j th row being e_{ij} (for $j \neq i$), and for $i = j$, the whole row is zero. Then, we have

$$\begin{aligned} & E_i \sum_{k=1}^N \lambda_k^2 |\psi_k\rangle\langle\psi_k| E_i \\ &= \sum_{k=1}^N \lambda_k^2 \begin{pmatrix} |\psi_k\rangle_i - |\psi_k\rangle_1 \\ |\psi_k\rangle_i - |\psi_k\rangle_2 \\ \vdots \\ |\psi_k\rangle_i - |\psi_k\rangle_N \end{pmatrix} \\ &\quad \times (|\psi_k\rangle_i - |\psi_k\rangle_1, |\psi_k\rangle_i - |\psi_k\rangle_2, \dots, |\psi_k\rangle_i - |\psi_k\rangle_N). \end{aligned} \quad (\text{C33})$$

The geodesic distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is the j th diagonal entry of the above matrix. More generally, we consider the matrices $\sum_{i=1}^N |i-1\rangle\langle i-1| \otimes E_i$, $\sum_{i=1}^N |i-1\rangle\langle i-1| \otimes \sum_{k=1}^N \frac{1}{\lambda_k} |\psi_k\rangle\langle\psi_k|$ and their product

$$\begin{aligned} & \left(\sum_{i=1}^N |i-1\rangle\langle i-1| \otimes E_i \right) \\ & \times \left(\sum_{i=1}^N |i-1\rangle\langle i-1| \otimes \sum_{k=1}^N \lambda_k^2 |\psi_k\rangle\langle\psi_k| \right) \\ & \times \left(\sum_{i=1}^N |i-1\rangle\langle i-1| \otimes E_i \right). \end{aligned} \quad (\text{C34})$$

The $(i \times j)$ th diagonal entry of the above matrix is exactly the geodesic distance $d_G(\mathbf{x}_i, \mathbf{x}_j)$ between the points $\mathbf{x}_i$ and $\mathbf{x}_j$.

From the block encoding of $\propto \sum_{k=1}^N \lambda_k^2 |\psi_k\rangle\langle\psi_k|$ (as obtained from above), to obtain the block encoding of E_i , we need the following result from Ref. [91]:

Lemma C 7. Suppose that A is a matrix of size $N \times N$ with condition number κ_A , and that we are provided with classical knowledge/description of entries of A . Then, the ϵ -approximated block encoding of $\frac{A}{\|A\|_F}$ can be obtained with a quantum circuit of complexity $\mathcal{O}(\log N \log^2 \frac{\kappa_A}{\epsilon})$ and a classical preprocessing of complexity $\mathcal{O}(\log N)$ (where $\|A\|_F$ is the Frobenius norm of A).

The application of the above lemma is straightforward to obtain the block encoding of $\frac{\sum_{i=1}^N |i-1\rangle\langle i-1| \otimes E_i}{\|E\|_F}$, where $\|E\|_F$ is the Frobenius norm of the numerator. Using Lemma B 1, we

can obtain the block encoding of

$$\begin{aligned} & \frac{\sum_{i=1}^N |i-1\rangle \langle i-1| \otimes E_i}{\|E\|_F} \\ & \times \left(\sum_{i=1}^N |i-1\rangle \langle i-1| \otimes \sum_{k=1}^N \frac{\lambda_k^2}{\alpha \|D\|^p} |\psi_k\rangle \langle \psi_k| \right) \\ & \times \frac{\sum_{i=1}^N |i-1\rangle \langle i-1| \otimes E_i}{\|E\|_F} \quad (C35) \\ & = \sum_{i=1}^N |i-1\rangle \langle i-1| \otimes \frac{(\lambda_k^2)}{\alpha \|D\|^p \|E\|_F^2} E_i |\lambda_k\rangle \langle \lambda_k| E_i, \quad (C36) \end{aligned}$$

which contains the square of geodesic distance (up to a scaling of Frobenius norm $\|E\|_F^2 d_G^2$ on the diagonal. Because each row of matrix E_i has two nonzero entries being 1 and -1 , so the Frobenius norm $\|E_i\|_F$ is $\sqrt{2N}$. The Frobenius norm of E is $\sqrt{2N^2} = \sqrt{2}N$. For a reason that would be clear later, we only want to keep the diagonal entry. To “filter” out those off-diagonal entries, we can use the following procedure:

Lemma C8. Let U be a unitary block encoding of some matrix M of size $n \times n$. Let T_U be the circuit complexity of U , then the block encoding of $\sum_{i=1}^n \frac{1}{n} M_{ii}^2 |i\rangle \langle i|$ can be obtained with a circuit of depth $\mathcal{O}(T_U + \log n)$.

Proof. By applying U to the state $|0\rangle \frac{1}{\sqrt{n}} \sum_{i=1}^n |i-1\rangle$, we obtain the following state:

$$\begin{aligned} & \frac{1}{\sqrt{n}} |0\rangle \sum_{i=1}^n |i-1\rangle M^i + \sum_{k \neq 0} |k\rangle |\text{Garbage}\rangle \quad (C37) \\ & = \frac{1}{\sqrt{n}} |0\rangle \sum_{i=1}^n \sum_{j=1}^N M_{ij} |i-1\rangle |j-1\rangle \\ & + \sum_{k \neq 0} |k\rangle |\text{Garbage}\rangle, \quad (C38) \end{aligned}$$

where M^i is the i th column of M and M_{ij} its j th entry. Now we append another ancilla initialized in $|0\rangle^{\otimes \log n}$, and use the cnot gates to obtain the following state:

$$\frac{1}{\sqrt{n}} |0\rangle \sum_{i=1}^n \sum_{j=1}^n M_{ij} |i-1\rangle |j-1\rangle + \sum_{k \neq 0} |k\rangle |\text{Garbage}\rangle, \quad (C39)$$

where the $|\text{Garbage}\rangle$ contains a slight abuse of notation. If we trace out the ancilla, we obtain the following density state:

$$|0\rangle \langle 0| \otimes \frac{1}{n} \sum_{i=1}^n (M_{ii})^2 |i-1\rangle \langle i-1| + \rho_{\text{Garbage}}. \quad (C40)$$

The above density state can be block encoded via Lemma C5, and in fact, the above density state is also the block encoding of $\frac{1}{n} \sum_{i=1}^n (M_{ii})^2 |i-1\rangle \langle i-1|$, which contains the diagonal entries only. ■

The application of the above lemma to our case is straightforward. Denote the operator in Eq. (C36) as M , and its unitary block encoding is U . Then, using the above lemma enables us to obtain the block encoding of

$\frac{1}{N^2} \sum_{p=1}^{N^2} (M_{pp})^2 |p-1\rangle \langle p-1|$. We note that the diagonal entries of the matrix M are the geodesic distances (divided by the Frobenius norm $\|E\|_F$), i.e.,

$$\begin{aligned} & \frac{1}{N^2} \sum_{p=1}^{N^2} (E_{pp})^2 |p-1\rangle \langle p-1| \\ & = \frac{1}{\alpha \|D\|^p \|E\|_F^4 N^2} \sum_{i,j=1}^N |i-1\rangle \langle i-1| \\ & \times \otimes d_G^4(\mathbf{x}_i, \mathbf{x}_j) |j-1\rangle \langle j-1|, \quad (C41) \end{aligned}$$

where, in the above, we have decomposed $|p\rangle = |i\rangle |j\rangle$.

Let U_{p_j} denote the N -dimensional permutation unitary such that

$$|0\rangle \leftrightarrow |i-1\rangle \quad (C42)$$

and the remaining basis permuted arbitrarily. This unitary can be constructed via Lemma C2 with a depth $\mathcal{O}(\log N)$. We then use Lemma B2 to obtain the unitary $U_{p_j} \otimes \mathbb{I}_N$, and then Lemma B1 to construct the block encoding of

$$\begin{aligned} & (U_{p_j} \otimes \mathbb{I}_N) \times \frac{1}{\alpha \|D\|^p \|E\|_F^2 N^2} |i-1\rangle \langle i-1| \\ & \times \otimes d_G^2(\mathbf{x}_i, \mathbf{x}_j) |j-1\rangle \langle j-1| \quad (C43) \\ & = \frac{1}{\alpha \|D\|^p \|E\|_F^2 N^2} \sum_{j=1}^N |0\rangle \langle 0| \otimes d_G^4(\mathbf{x}_i, \mathbf{x}_j) |j-1\rangle \\ & \times \langle j-1| + \frac{1}{\alpha \|D\|^p \|E\|_F^2 N^2} \sum_{i,j=2}^N |i-1\rangle \\ & \times \langle i-1| \otimes d_G^4(\mathbf{x}_i, \mathbf{x}_j) |j-1\rangle \langle j-1|, \quad (C44) \end{aligned}$$

which is also the block encoding of the operator:

$$\frac{1}{\alpha \|D\|^p \|E\|_F^4 N^2} \sum_{j=1}^N d_G^4(\mathbf{x}_i, \mathbf{x}_j) |j-1\rangle \langle j-1|, \quad (C45)$$

which essentially contains $\propto d_G^4(\mathbf{x}_i, \mathbf{x}_j)$ on the diagonal.

3. Proof of Lemma C4

State preparation has been proposed in a variety of works [82,86,87,93,94]. Although the method in Ref. [82] is universal and can prepare an exact state, the number of ancilla qubits is proportional to the number of nonzero entries. Therefore, it is only effective, in terms of both depth and space, when the target state is sparse.

Here, we build on the method proposed in Ref. [86], which is more qubit-efficient while achieving the same depth (asymptotically) compared to Ref. [82]. To proceed, we first recapitulate their main result:

Lemma C9 (Theorem 1 in Ref. [86]). Let $f : [0, 1] \rightarrow \mathbb{R}^+$ be a positive integrable function in $L^2([0, 1])$ and $\eta = \sup_{x \in [0, 1]} |\partial_x^2 \log f^2(x)|$. Define $|f\rangle = \sum_{i=1}^n f(\frac{i}{n}) |i\rangle$. Then, it

is possible to prepare a state $|\psi_f\rangle$ such that $\| |\psi_f\rangle - |f\rangle \|_2 \leq \epsilon$ using a quantum circuit of gate complexity $2^{k_0(\epsilon)} - 1$, where

$$k_0(\epsilon) = \max \left\{ \left\lceil -\frac{1}{2} \log_2 \left(4^{-\log n} - \frac{96}{\eta^2} \log(1 - \epsilon) \right) \right\rceil, 2 \right\}. \quad (\text{C46})$$

As proved in Ref. [86], the value of $k_0(\epsilon)$ is asymptotically independent of n . Recall that in our case, we need to prepare a state of the form $|\phi\rangle = \sum_{i=1}^n a_i |i\rangle$ (assuming normalization condition). For convenience in applying the above result, we assume that all a_i 's are positive for now, and we will extend to the negative case later. We consider the $\mathbb{R}^2$ plane, and promote each amplitude a_i to a point having the following coordinates: $(\frac{i}{n}, a_i)$. Then, we connect consecutive points by a line, which results in a piecewise-linear function f on $\mathbb{R}^2$. Existing works [95–97] have shown how to approximate piecewise-linear functions by polynomials or smooth functions. In principle, any of these can be used to replace or approximate f . Because $f(\frac{i}{n}) = a_i$, which is positive and square integrable, we can apply the above lemma with this choice of function f .

Now we extend the above procedure to the case where $|\phi\rangle$ contains negative amplitudes. We first break the amplitudes $\{a_i\}_{i=1}^n$ into positive and negative parts, denoted as $\{a_i\}_+$, $\{a_i\}_-$, respectively. Then, we consider the

decomposition:

$$|\phi\rangle = \sum_{i, a_i \in \{a_i\}_+} a_i |i\rangle + \sum_{i, a_i \in \{a_i\}_-} a_i |i\rangle \quad (\text{C47})$$

$$= |\phi\rangle_+ + |\phi\rangle_- . \quad (\text{C48})$$

The first part $|\phi\rangle_+$ contains positive amplitudes, so the procedure from the previous paragraph can be used to construct the unitary U_+ that prepares $|\phi\rangle_+$. The second part $|\phi\rangle_-$ contains negative amplitudes, so the state $-|\phi\rangle_-$ contains positive amplitudes. As such, we can use the procedure earlier to construct the unitary U_- that prepares the state $-|\phi\rangle_-$. Next, we can use the linear combination of unitaries, or Lemma B 3, to construct the block encoding of $\frac{1}{2}(U_+ - U_-)$. Taking this block encoding and applying it to the state $|0\rangle |0\rangle^n$, according to Definition B 1, we obtain the state

$$\begin{aligned} & \frac{1}{2} |0\rangle (|\phi\rangle_+ - (-|\phi\rangle_-)) + |\text{Garbage}\rangle \\ &= \frac{1}{2} (|\phi\rangle_+ + |\phi\rangle_-) + |\text{Garbage}\rangle, \end{aligned} \quad (\text{C49})$$

where $|\text{Garbage}\rangle$ is some unwanted state. Performing measurement on the ancilla and postselecting on $|0\rangle$, we obtain the desired state $|\phi\rangle_+ + |\phi\rangle_- = |\phi\rangle$.

Besides, we remark that in Refs. [86,87], the authors also proposed variational strategies for state preparation, which use a quantum circuit of logarithmical complexity (in terms of both qubits and depth). In principle, any of these can be applied to our case.

-
- [1] C. Fefferman, S. Mitter, and H. Narayanan, Testing the manifold hypothesis, *J. Am. Math. Soc.* **29**, 983 (2016).
 - [2] A. N. Gorban and I. Y. Tyukin, Blessing of dimensionality: Mathematical foundations of the statistical physics of data, *Philos. Trans. R. Soc. A* **376**, 20170237 (2018).
 - [3] B. C. A. Brown, A. L. Caterini, B. L. Ross, J. C. Cresswell, and G. Loaiza-Ganem, Verifying the union of manifolds hypothesis for image data, *arXiv:2207.02862*.
 - [4] D. Sternad, It's not (only) the mean that matters: Variability, noise and exploration in skill learning, *Current Opin. Behav. Sci.* **20**, 183 (2018).
 - [5] L. Amsaleg, O. Chelly, T. Furon, S. Girard, M. E. Houle, K.-I. Kawarabayashi, and M. Nett, Estimating local intrinsic dimensionality, in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Association for Computing Machinery, New York, 2015), pp. 29–38.
 - [6] D. Eppstein, M. S. Paterson, and F. F. Yao, On nearest-neighbor graphs, *Discrete Comput. Geo.* **17**, 263 (1997).
 - [7] J. M. Lee, *Riemannian Manifolds: An Introduction to Curvature* (Springer Science & Business Media, 2006), Vol. 176.
 - [8] K. Grover, G. J. Gordon, and C. Faloutsos, CurvGAd: Leveraging curvature for enhanced graph anomaly detection, *arXiv:2502.08605*.
 - [9] R. S. Sandhu, T. T. Georgiou, and A. R. Tannenbaum, Ricci curvature: An economic indicator for market fragility and systemic risk, *Sci. Adv.* **2**, e1501495 (2016).
 - [10] A. Samal, H. K. Pharasi, S. J. Ramaia, H. Kannan, E. Saucan, J. Jost, and A. Chakraborti, Network geometry and market instability, *R. Soc. Open Sci.* **8**, 201734 (2021).
 - [11] V. Boginski, S. Butenko, and P. M. Pardalos, Mining market data: A network approach, *Comput. Operat. Res.* **33**, 3171 (2006).
 - [12] K. T. Chi, J. Liu, and F. C. M. Lau, A network perspective of the stock market, *J. Empir. Finance* **17**, 659 (2010).
 - [13] Y. Ollivier, Ricci curvature of metric spaces, *C. R. Math.* **345**, 643 (2007).
 - [14] Y. Ollivier, Ricci curvature of Markov chains on metric spaces, *J. Funct. Anal.* **256**, 810 (2009).
 - [15] J. S. Garcia and S. Gherghe, On the Ollivier-Ricci curvature as fragility indicator of the stock markets, *arXiv:2405.07134*.
 - [16] F. Santambrogio, Optimal transport for applied mathematicians (2015).
 - [17] C. A. Trugenberger, Combinatorial quantum gravity: Geometry from random bits, *J. High Energy Phys.* **09** (2017) 045.
 - [18] Pim Van Der Hoorn, W. J. Cunningham, G. Lippner, C. Trugenberger, and D. Krioukov, Ollivier-Ricci curvature convergence in random geometric graphs, *Phys. Rev. Res.* **3**, 013211 (2021).
 - [19] C. Kelly, F. Biancalana, and C. Trugenberger, Convergence of combinatorial gravity, *Phys. Rev. D* **105**, 124002 (2022).
 - [20] C. A. Trugenberger, Combinatorial quantum gravity and emergent 3D quantum behaviour, *Universe* **9**, 499 (2023).
 - [21] R. P. Feynman, Simulating physics with computers, in *Feynman and Computation* (CRC Press, 2018), pp. 133–153.
 - [22] P. Benioff, The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines, *J. Stat. Phys.* **22**, 563 (1980).

- [23] D. Deutsch, Quantum theory, the Church–Turing principle and the universal quantum computer, *Proc. R. Soc. Lond. A* **400**, 97 (1985).
- [24] D. Deutsch and R. Jozsa, Rapid solution of problems by quantum computation, *Proc. R. Soc. Lond. Ser. A* **439**, 553 (1992).
- [25] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM Rev.* **41**, 303 (1999).
- [26] L. K. Grover, A fast quantum mechanical algorithm for database search, in *Proceedings of the 28th Annual ACM Symposium on Theory of Computing* (Association for Computing Machinery, New York, 1996), pp. 212–219.
- [27] B. Bauer, S. Bravyi, M. Motta, and G. K.-L. Chan, Quantum algorithms for quantum chemistry and quantum materials science, *Chem. Rev.* **120**, 12685 (2020).
- [28] 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).
- [29] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum algorithm for linear systems of equations, *Phys. Rev. Lett.* **103**, 150502 (2009).
- [30] A. M. Childs, R. Kothari, and R. D. Somma, Quantum algorithm for systems of linear equations with exponentially improved dependence on precision, *SIAM J. Comput.* **46**, 1920 (2017).
- [31] L. Wossnig, Z. Zhao, and A. Prakash, Quantum linear system algorithm for dense matrices, *Phys. Rev. Lett.* **120**, 050502 (2018).
- [32] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, Efficient quantum algorithms for simulating sparse Hamiltonians, *Commun. Math. Phys.* **270**, 359 (2007).
- [33] D. W. Berry and A. M. Childs, Black-box Hamiltonian simulation and unitary implementation, *Quantum Inf. Comput.* **12**, 29 (2009).
- [34] D. W. Berry, High-order quantum algorithm for solving linear differential equations, *J. Phys. A: Math. Theor.* **47**, 105301 (2014).
- [35] D. W. Berry, A. M. Childs, and R. Kothari, Hamiltonian simulation with nearly optimal dependence on all parameters, in *2015 IEEE 56th Annual Symposium on Foundations of Computer Science* (IEEE, 2015), pp. 792–809.
- [36] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Simulating Hamiltonian dynamics with a truncated Taylor series, *Phys. Rev. Lett.* **114**, 090502 (2015).
- [37] S. Lloyd, Universal quantum simulators, *Science* **273**, 1073 (1996).
- [38] A. M. Childs, D. Maslov, Y. Nam, N. J. Ross, and Y. Su, Toward the first quantum simulation with quantum speedup, *Proc. Natl. Acad. Sci. USA* **115**, 9456 (2018).
- [39] R. Gerritsma, G. Kirchmair, F. Zähringer, E. Solano, R. Blatt, and C. F. Roos, Quantum simulation of the Dirac equation, *Nature (London)* **463**, 68 (2010).
- [40] R. Babbush, N. Wiebe, J. McClean, J. McClain, H. Neven, and Garnet Kin-Lic Chan, Low-depth quantum simulation of materials, *Phys. Rev. X* **8**, 011044 (2018).
- [41] Juan Miguel Arrazola, T. Kalajdzievski, C. Weedbrook, and S. Lloyd, Quantum algorithm for nonhomogeneous linear partial differential equations, *Phys. Rev. A* **100**, 032306 (2019).
- [42] A. M. Childs, J.-P. Liu, and A. Ostrander, High-precision quantum algorithms for partial differential equations, *Quantum* **5**, 574 (2021).
- [43] J.-P. Liu, H. Ø. Kolden, H. K. Krovi, N. F. Loureiro, K. Trivisa, and A. M. Childs, Efficient quantum algorithm for dissipative nonlinear differential equations, *Proc. Natl. Acad. Sci. USA* **118**, e2026805118 (2021).
- [44] H. Krovi, Improved quantum algorithms for linear and nonlinear differential equations, *Quantum* **7**, 913 (2023).
- [45] M. Schuld and F. Petruccione, *Supervised Learning with Quantum Computers* (Springer, 2018), Vol. 17.
- [46] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, Evaluating analytic gradients on quantum hardware, *Phys. Rev. A* **99**, 032331 (2019).
- [47] M. Schuld, Machine learning in quantum spaces, *Nature (London)* **567**, 179 (2019).
- [48] M. Schuld and N. Killoran, Quantum machine learning in feature Hilbert spaces, *Phys. Rev. Lett.* **122**, 040504 (2019).
- [49] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, Circuit-centric quantum classifiers, *Phys. Rev. A* **101**, 032308 (2020).
- [50] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum algorithms for supervised and unsupervised machine learning, *arXiv:1307.0411*.
- [51] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, Supervised learning with quantum-enhanced feature spaces, *Nature (London)* **567**, 209 (2019).
- [52] S. Lloyd, S. Garnerone, and P. Zanardi, Quantum algorithms for topological and geometric analysis of data, *Nat. Commun.* **7**, 10138 (2016).
- [53] D. W. Berry, Y. Su, C. Gyurik, R. King, J. Basso, A. D. Toro Barba, A. Rajput, N. Wiebe, V. Dunjko, and R. Babbush, Analyzing prospects for quantum advantage in topological data analysis, *PRX Quantum* **5**, 010319 (2024).
- [54] A. Schmidhuber and S. Lloyd, Complexity-theoretic limitations on quantum algorithms for topological data analysis, *PRX Quantum* **4**, 040349 (2023).
- [55] N. A. Nghiem, X. D. Gu, and T.-C. Wei, Quantum algorithm for estimating Betti numbers using a cohomology approach, *Quantum* **9**, 1955 (2025).
- [56] N. A. Nghiem, New aspects of quantum topological data analysis: Betti number estimation, and testing and tracking of homology and cohomology classes, *arXiv:2506.01432*.
- [57] R. Hayakawa, Quantum algorithm for persistent Betti numbers and topological data analysis, *Quantum* **6**, 873 (2022).
- [58] C. Gyurik, A. Schmidhuber, R. King, V. Dunjko, and R. Hayakawa, Quantum computing and persistence in topological data analysis, *arXiv:2410.21258*.
- [59] S. Ubaru, I. Y. Akhalwaya, M. S. Squillante, K. L. Clarkson, and L. Horesh, Quantum topological data analysis with linear depth and exponential speedup, *arXiv:2108.02811*.
- [60] N. A. Nghiem, T. K. Do, T.-C. Wei, and T. V. Phan, Quantum algorithm for estimating intrinsic geometry, *arXiv:2508.06355*.
- [61] For example, in a dataset consisting of two dense clusters connected by a thin bridge, ORC detects the bottleneck through increased optimal transport cost between neighborhood distributions across the bridge, whereas Gaussian curvature estimated from point clouds depends on local manifold reconstruction and typically reflects only local within-cluster geometry without directly encoding intercluster connectivity.

- [62] T. L. Saidi, A. Hickok, and A. J. Blumberg, Recovering manifold structure using Ollivier-Ricci curvature, [arXiv:2410.01149](#).
- [63] E. Saucan, R. P. Sreejith, R. P. Vivek-Ananth, J. Jost, and A. Samal, Discrete Ricci curvatures for directed networks, *Chaos Solitons Fract.* **118**, 347 (2019).
- [64] N. Azarhooshang, P. Sengupta, and B. DasGupta, A review of and some results for Ollivier-Ricci network curvature, *Mathematics* **8**, 1416 (2020).
- [65] V. Chvátal, *Linear Programming* (Macmillan, 1983).
- [66] G. B. Dantzig, Linear programming, *Oper. Res.* **50**, 42 (2002).
- [67] J. Orlin, A faster strongly polynomial minimum cost flow algorithm, in *Proceedings of the 20th Annual ACM Symposium on Theory of Computing* (Association for Computing Machinery, New York, 1988), pp. 377–387.
- [68] L. Chen, R. Kyng, Y. Liu, and R. Peng, M. P. Gutenberg, and S. Sachdeva, Maximum flow and minimum-cost flow in almost-linear time, *J. ACM* **72**, 1 (2025).
- [69] A. W. Marshall and I. Olkin, *Inequalities: Theory of Majorization and its Applications* (Academic Press, 1979).
- [70] G. Birkhoff, Three observations on linear algebra, *Univ. Nac. Tacuman Rev. A* **5**, 147 (1946).
- [71] H. W. Kuhn, The Hungarian method for the assignment problem, *Nav. Res. Logist. Q.* **2**, 83 (1955).
- [72] N. A. Nghiem and T.-C. Wei, Improved quantum algorithms for eigenvalues finding and gradient descent, [arXiv:2312.14786](#).
- [73] N. A. Nghiem, H. Sukeno, S. Zhang, and T.-C. Wei, Improved quantum power method and numerical integration using a quantum singular-value transformation, *Phys. Rev. A* **111**, 012434 (2025).
- [74] A. Hickok and A. J. Blumberg, An intrinsic approach to scalar-curvature estimation for point clouds, [arXiv:2308.02615](#).
- [75] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetics, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing* (Association for Computing Machinery, New York, 2019), pp. 193–204.
- [76] G. H. Low and I. L. Chuang, Optimal Hamiltonian simulation by quantum signal processing, *Phys. Rev. Lett.* **118**, 010501 (2017).
- [77] G. H. Low, and I. L. Chuang, Hamiltonian simulation by qubitization, *Quantum* **3**, 163 (2019).
- [78] S. Chakraborty, A. Gilyén, and S. Jeffery, The power of block-encoded matrix powers: Improved regression techniques via faster Hamiltonian simulation, [arXiv:1804.01973](#).
- [79] R. N. Mantegna, Hierarchical structure in financial markets, *Eur. Phys. J. B* **11**, 193 (1999).
- [80] M. Tumminello, T. Aste, T. D. Matteo, and R. N. Mantegna, A tool for filtering information in complex systems, *Proc. Natl. Acad. Sci. USA* **102**, 10421 (2005).
- [81] S. McArdle, A. Gilyén, and M. Berta, Quantum state preparation without coherent arithmetic, [arXiv:2210.14892](#).
- [82] X.-M. Zhang, T. Li, and X. Yuan, Quantum state preparation with optimal circuit depth: Implementations and applications, *Phys. Rev. Lett.* **129**, 230504 (2022).
- [83] B. Bollobás, Random graphs, in *Modern Graph Theory* (Springer, 2011), pp. 215–252.
- [84] V. V. Shende, S. S. Bullock, and I. L. Markov, Synthesis of quantum logic circuits, in *Proceedings of the 2005 Asia and South Pacific Design Automation Conference* (Association for Computing Machinery, New York, 2005), pp. 272–275.
- [85] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, Elementary gates for quantum computation, *Phys. Rev. A* **52**, 3457 (1995).
- [86] G. Marin-Sanchez, J. Gonzalez-Conde, and M. Sanz, Quantum algorithms for approximate function loading, *Phys. Rev. Res.* **5**, 033114 (2023).
- [87] K. Nakaji, S. Uno, Y. Suzuki, R. Raymond, T. Onodera, T. Tanaka, H. Tezuka, N. Mitsuda, and N. Yamamoto, Approximate amplitude encoding in shallow parameterized quantum circuits and its application to financial market indicators, *Phys. Rev. Res.* **4**, 023136 (2022).
- [88] C. Zoufal, Aurélien Lucchi, and S. Woerner, Quantum generative adversarial networks for learning and loading random distributions, *npj Quantum Inf.* **5**, 103 (2019).
- [89] J. Lee and N. A. Nghiem, New aspects of quantum topological data analysis: Betti number estimation, and testing and tracking of homology and cohomology classes, [arXiv:2506.01432](#).
- [90] D. Camps and R. Van Beeumen, Approximate quantum circuit synthesis using block encodings, *Phys. Rev. A* **102**, 052411 (2020).
- [91] N. A. Nghiem, Refined quantum algorithms for principal component analysis and solving linear system, [arXiv:2504.00833](#).
- [92] L. N. Trefethen, *Approximation Theory and Approximation Practice* (SIAM, 2019).
- [93] M. Plesch and Č. Brukner, Quantum-state preparation with universal gate decompositions, *Phys. Rev. A* **83**, 032302 (2011).
- [94] X.-M. Zhang, M.-H. Yung, and X. Yuan, Low-depth quantum state preparation, *Phys. Rev. Res.* **3**, 043200 (2021).
- [95] F. Arandiga, A. Cohen, R. Donat, and N. Dyn, Interpolation and approximation of piecewise smooth functions, *SIAM J. Numer. Anal.* **43**, 41 (2005).
- [96] Y. Lipman and D. Levin, Approximating piecewise-smooth functions, *IMA J. Numer. Anal.* **30**, 1159 (2010).
- [97] V. M. Jimenez-Fernandez, M. Jimenez-Fernandez, H. Vazquez-Leal, E. Muñoz-Aguirre, H. H. Cerecedo-Núñez, U. A. Filobello-Niño, and F. J. Castro-Gonzalez, Transforming the canonical piecewise-linear model into a smooth-piecewise representation, *SpringerPlus* **5**, 1612 (2016).