Subtrajectory Clustering with Machine Learning on Quantum Computers

Wil Bishop
Department of Computer Science
University of Minnesota Duluth
Duluth, Minnesota, USA
bisho210@umn.edu

Eleazar Leal
Department of Computer Science
University of Minnesota Duluth
Duluth, Minnesota, USA
eleal@d.umn.edu

Le Gruenwald
School of Computer Science
University of Oklahoma
Norman, Oklahoma, USA
ggruenwald@ou.edu

***Abstract*—Subtrajectory clustering is vital for real-world applications such as traffic bottleneck detection, public transportation optimization, and play pattern discovery in sports analytics. This problem is NP-hard and computationally intensive for large-scale applications, and many existing classical implementations struggle with scalability and generalizability. While machine learning-based approaches allow for greater generalizability than existing rule-based methods, they still struggle with scalability. Quantum computing has shown promising performance improvements over classical computing for some machine learning tasks on regular point data. However, no research has been done on whether quantum computing can also offer improvements over classical computing for subtrajectory clustering. If that is the case, combining quantum computing with machine learning may enable solutions that are both scalable and generalizable. This paper discusses the drawbacks of current subtrajectory clustering approaches within the classical computing paradigm, the challenges associated with solving the problem using quantum computing, and a vision for the conversion of a state-of-the-art classical subtrajectory clustering algorithm to quantum.**

***Index Terms*—Quantum Computing, Subtrajectory Clustering, Quantum Machine Learning**

I. Introduction

With the pervasive use of GPS-enabled devices and location-based services, an unprecedented volume of trajectory data is being continuously generated. *Trajectories* are time-ordered sequences of spatial points that can be collected using location sensors. The unique challenges associated with mining this type of data require different solutions from those used for point data. A key task in trajectory analytics is *trajectory clustering*, where similar trajectories are grouped together. This task has numerous real-world applications, such as traffic monitoring, urban planning, and sports analytics [1]–[3]. *Subtrajectory clustering* is a technique that segments trajectories into smaller, meaningful subtrajectories and groups similar ones together. Unlike full-trajectory clustering, this technique enables the discovery of local patterns that may be obscured due to variations in trajectory length, duration, or sampling rate. Subtrajectory clustering has many applications, such as carpooling, reconstructing maps from raw data, understanding animal migration patterns, etc. [4], [5]

Despite its importance, subtrajectory clustering is computationally intensive, being NP-hard [6]. Many classical algorithms such as TRACLUS [5] rely on rule-based heuristic methods that can be sensitive to input parameters that are computationally intensive to calibrate [7], [8] and scale poorly with dataset size [9]. RLSTC [10] has proposed reinforcement learning (RL) approaches to address these limitations. It uses a Deep Q-Network (DQN) to learn policies that identify optimal segmentation points. This approach is more scalable and generalizable than rule-based methods, as it learns directly from the dataset; however, the calculation of trajectory distances is a bottleneck.

One potential strategy to address this issue of scalability of subtrajectory clustering is to use quantum computing, a paradigm that has been gaining attention from research centers, funding agencies, and companies [11]. This increasing attention is partly evidenced by the popularity of IBM's Qiskit language and the development of Microsoft's Majorana 1 chip [12], [13]. Previous research has shown that quantum computing can offer promising improvements in the execution time of machine learning algorithms for regular point data for classical computers [14]–[17]. However, no research has been done to investigate if such improvements also exist for machine learning algorithms for trajectory data. In this paper, our aim is to present a framework that could be implemented to answer this question.

This paper presents our progress towards adapting RLSTC, a state-of-the-art classical subtrajectory clustering algorithm [10] shown to outperform other state-of-the-art methods, to a quantum algorithm named Q-RLSTC (Quantum Reinforcement Learning-based Sub-Trajectory Clustering). Our aim is to present a framework to study whether quantum computing can deliver performance improvements over classical computing in the context of subtrajectory clustering. The remainder of this paper is organized as follows. Section II presents related work; Section III discusses the research challenges of quantum subtrajectory clustering; Section IV explains our proposed quantum subtrajectory clustering algorithm; finally, Section V provides conclusions and future research.

II. Related Work

A. Classical Subtrajectory Clustering

Multiple classical computing algorithms for subtrajectory clustering exist [5], [6], [10], [18]–[21], aiming to uncover

local movement patterns by segmenting trajectories into sub-trajectories, using hand-crafted heuristics, and grouping those exhibiting similar spatial-temporal behavior. The effectiveness of these methods depends on the initial parameters chosen, which can be computationally expensive to calibrate [7], [8]. Furthermore, similarity metrics such as Fréchet distance [22] and Dynamic Time Warping [23] also scale poorly with trajectory length and dataset size. To address these issues, RLSTC [10] was recently introduced as a DRL approach to learn segmentation policies directly from data using a DQN for subtrajectory clustering. RLSTC circumvents the need for static rules and provides a dynamic way to identify subtrajectory boundaries [10]. However, this approach remains computationally intensive on classical hardware due to the nature of DRL, which requires exploring a large policy space.

B. Quantum Computing Background

Quantum computing stems from the principles of quantum mechanics, describing phenomena at atomic and subatomic scales. Quantum mechanics introduces concepts like superposition (a particle existing in multiple states simultaneously) and entanglement (linked quantum states of particles) that can be exploited for computing [24]. Classical computing uses bits (0 or 1). A *quantum bit (qubit)*, used in quantum computing, can be in state 0, state 1, or a superposition of both states. When measured, it yields either 0 or 1, with the probability of each outcome determined by the qubit's state prior to measurement. This allows quantum computers to process vast information via parallelism. A common approach in quantum computing for solving a problem is to construct quantum circuits using quantum gates, which are basic operations that manipulate qubits by changing their quantum states. Qubits are susceptible to environmental noise from temperature fluctuations and electromagnetic fields, which can detrimentally affect desired superposition states [25]. Quantum systems can also undergo *decoherence*, which is the degradation of superposition and entanglement as quantum information is lost [26].

III. Research Challenges in Quantum Subtrajectory Clustering

A. Challenges Common to Both Classical and Quantum Computing

(1) Data Volume: Large-scale data collection from location sensors, particularly at high sampling rates, can produce massive datasets [27]. This imposes burdens on storage and processing systems, especially when real-time analysis is required. This can be problematic for current quantum hardware which have limited qubits to encode the data with. *(2) Spatial-Temporal Context:* While point data is unordered, trajectory data requires a set ordering which puts restrictions on the ways the data can be organized and compressed. This is notable in quantum computing where it may be difficult to encode all of the spatial-temporal elements into qubits efficiently. *(3) Size Variability:* Unlike point data where each point can be compared easily to another, with trajectories there is a need to compare objects of different sizes. This issue is compounded in the quantum paradigm where objects are often converted into fixed-length vectors. *(4) Noise and Data Cleaning:* The instruments for obtaining trajectory data, such as GPS, may give inaccurate readings that can adversely affect the quality of the final clustering. This noise in the data can be compounded by the noise inherent to current quantum systems. *(5) Streaming Data:* Many applications require processing trajectories arriving as continuous streams. A limited number of qubits may exacerbate this issue, and adding more qubits to the system to encode more data may prove difficult.

B. Challenges Specific to Quantum Computing

(1) Data Loading and Encoding: Classical trajectory data, stored as bit strings, must be efficiently encoded into quantum states while minimizing the number of qubits and circuit depth [28]. Choosing an appropriate encoding technique that preserves spatial and temporal relationships is vital. Angle encoding [29], for example, maps data values to rotation angles of qubits, allowing for a direct, intuitive representation of continuous features like coordinates and timestamps. This is vital for trajectory clustering because maintaining the relative order and magnitudes of spatial and temporal differences between points is crucial for accurate similarity calculations between subtrajectories. Without an encoding method that respects this inherent order and structure, the quantum algorithm would lose the very information it needs to distinguish and cluster distinct movement patterns. *(2) Optimal Circuit Design:* The design of effective, trainable, and hardware-efficient quantum circuits is a formidable challenge. Selecting a circuit depth that does not match the problem may result in slow training known as a barren plateau [30]. If the circuit is too deep, it may accumulate decoherence as mentioned in Section II.B and become untrainable due to noise. There is an inherent trade-off between a quantum circuit's ability to represent the complex states required for a given computation and its efficiency which necessitates intimate domain knowledge for optimal circuit design. *(3) Hybrid Algorithm Design:* Due to limitations in the number of qubits and in the decoherence time available in quantum computers today, it is often advisable to adapt only those components of an algorithm that see the greatest improvements from a quantum adaptation, leaving the other elements as classical. Determining the optimal division of labor between classical and quantum processors is a key design choice. Efficiently managing the communication overhead in terms of the frequency of exchanges, the volume of data transferred per exchange, and the speed of data transfer and encoding/decoding between classical and quantum components is vital to ensure that any potential quantum speedup is not nullified by classical bottlenecks. *(4) Quantum Hardware Restraints:* Noisy Intermediate-Scale Quantum (NISQ) [31] devices are the most accessible and cost-effective pieces of quantum hardware available today for research. They are characterized by a limited number of qubits, short coherence times in which each qubit can maintain its quantum properties like superposition and entanglement, high gate error rates which are the probability that an error occurs while a quantum

gate is applied to a qubit, and restricted qubit connectivity which means that not all qubits can directly interact with each other. These hardware limitations severely constrain the size and complexity of quantum algorithms that can be reliably executed, making the practical implementation of ambitious Quantum Machine Learning (QML) models for subtrajectory clustering a long-term goal. *(5) Decoding Quantum Outputs:* Quantum algorithms typically yield probabilistic outcomes through measurements. Translating these measurement back into classical information (e.g., cluster assignments, Q-values, distances) that can be used by the classical components of the hybrid algorithm requires robust decoding strategies such as expectation-value estimation [32]. This is particularly challenging with large or noisy quantum output spaces, as it can be with subtrajectory clustering due to its high dimensionality.

IV. Proposed Quantum Subtrajectory Clustering Algorithm

We describe RLSTC [10], a state-of-the-art subtrajectory clustering algorithm for classical computers, shown to outperform other state-of-the-art methods, and our approach for adapting it to quantum computing (Quantum RLSTC).

A. Description of RLSTC

RLSTC clusters subtrajectories with the following steps:

(1) Preprocessing: Trajectories are simplified by keeping only significant points with Minimum Description Length (MDL).

(2) Computation of Initial Cluster Centers: Cluster centers are initially derived using k-means++ [33]. A *cluster center* is a representative trajectory capturing the collective movement pattern of sub-trajectories within that cluster, generated by calculating the average coordinate at that timestamp. If the number of trajectories within a specific timestamp meets a threshold $MinNum$, the average coordinate for that timestamp is computed, using linear interpolation for trajectories not having a point at that timestamp.

(3) Learning the Optimal Policy: The reinforcement learning model is formulated as a Markov Decision Process (MDP). *States:* Each state consists of five features: $s_t(OD_s)$, the overall distance between all points in a subtrajectory and the nearest cluster center to that subtrajectory if the trajectory is segmented at the current point p_t; $s_t(OD_n)$, the overall distance if a trajectory isn't segmented at the current point p_t; OD_b, the expert knowledge estimate of the overall distance calculated by TRACLUS, used to make sure the RLSTC algorithm doesn't make a premature partition that fails to minimize overall distance; $s_t(L_b)$ the relative length of the generated subtrajectory; and $s_t(L_f)$, the relative length of the remaining subtrajectory. *Actions:* Whether or not to segment the trajectory at the current point p_t. *Rewards:* $s_t(OD) - s_{t+1}(OD)$ for the immediate reward, and $s_1(OD) - s_{|T|}(OD)$ for the cumulative reward, where both represent the difference in overall distance between states. *Initialization:* The DQN algorithm [34] initializes a main Q-network which takes as input a pair (s_t, a_t) and returns $Q(s_t, a_t; \theta)$, and a target network, along with a replay memory M storing all the experience vectors (s_t, a_t, r_t, s_{t+1}). The goal of the DQN is to learn a function Q and construct a policy for selecting an action a_t given a state s_t. Training is episodic. For each trajectory, points are processed sequentially. *(3.1) Distance Calculation:* The Overall Distance (OD) measures are based on the Euclidean distance using the Trapezoid approximation [35], which is identified as a bottleneck in the paper. *(3.2) Action Selection:* At each point, an action is chosen using an ϵ-greedy strategy based on the main network [36]. The action is executed, leading to a new state s_{t+1} and reward r_t. This experience (s_t, a_t, r_t, s_{t+1}) is stored in the replay memory M. *(3.3) Loss Function:* A minibatch of experiences is randomly sampled from M to train the main network by minimizing the Mean Squared Error (MSE) loss function using Stochastic Gradient Descent (SGD). The target Q-value for the loss calculation is derived from the target network and the immediate reward. The target network's parameters are periodically updated. The learned optimal policy is to select the action that maximizes $Q(s_t, a_t; \theta)$ for a given state s_t because Q is the expected long-term reward starting from state s_t after taking action a_t.

(4) Classical Segmentation and Clustering Loop: Use the cluster centers from Step 2 and rerun segmentation using the learned policy. Update the cluster centers by assigning each newly generated subtrajectory to its nearest cluster. The maximum distance $maxdist$ between the newly updated cluster centers and the previous cluster centers is calculated; if $maxdist$ is below a threshold τ, the algorithm converges and the k clusters are returned, which represent the most common shared subtrajectories.

B. Quantum RLSTC (Q-RLSTC): Our Vision to Transform RLSTC to a Quantum Approach

Utilizing a hybrid quantum-classical approach, we present quantum alternatives to existing sub-algorithms of RLSTC, with comments on both their near-term feasibility and utility over classical methods. Figure 1 illustrates our proposed algorithm, the steps of which are discussed below.

(1) Preprocessing with MDL: Given NISQ hardware limitations, the classical MDL preprocessing step is retained for our framework. We will then use angle encoding to represent the trajectories with qubits for Step 2.

(2) Quantum Initial Clustering: RLSTC utilizes k-means++ [33] for the initialization of clusters. There have been multiple quantum implementations of the k-means algorithm [17], [37]–[40], so it seems like the most promising area to hybridize this algorithm. In particular, q-means [39] was found to scale polylogarithmically with the number of data points, suggesting an exponential speedup with respect to the size of the dataset, whereas the simplest version of the classical k-means scales is linear. This is important with trajectory data which often has many data points. Our vision uses q-means++ [39], which replicates the superior initial clustering of k-means++ in a quantum environment.

In addition to k-means [41] and k-means++ [33], RLSTC can work with other density clustering algorithms, such as

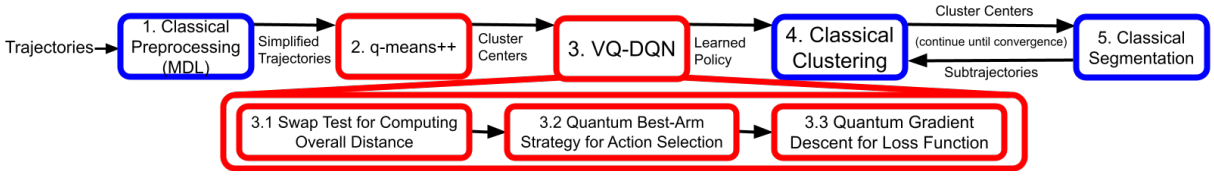

Fig. 1. Our vision for Q-RLSTC. Boxes with blue borders are classical and boxes with red borders are quantum.

DBSCAN [42], BIRCH [43], and OPTICS [44]. Notably, quantum and quantum-inspired DBSCAN versions have also been developed [45]–[47]. While this would require further alteration of the RLSTC framework, a DBSCAN-inspired algorithm may be a better fit for trajectory clustering than k-means or k-means++, as it can find arbitrarily-shaped clusters based on density rather than finding a set number of clusters [48], but the necessary alterations would be substantial due to the move away from centroid-based updates.

(3) Quantum Policy Learning: RLSTC uses a DQN with an ϵ-greedy strategy to learn a policy. There is a growing corpus of research dedicated to studying the use of Variational Quantum Circuits (VQCs) to accomplish this objective [49]–[52]. Lokes et al. [50] mention the $O(n)$ linear complexity where n is the number of parameters needed in a Variational Quantum Deep Q-Network (VQ-DQN) compared to the $O(n^3)$ parameter complexity of classical Q-Learning and $O(n^2)$ complexity of a classical DQN. It is also notable that Chen et al.'s algorithm [49] is quite robust against the noise present in current-day NISQ devices because their action selection mechanism does not need to find the exact expectation value of each qubit, but rather only identifying the qubit with the largest expectation value. Due to the probabilistic nature of quantum measurements, it would require a large number of measurements, or *shots*, to converge on an exact expectation value. These results are promising for more near-term improvements as noise will continue to be a major obstacle in quantum computing in the near future. **(3.1) Quantum Overall Distance Calculation:** [35] mentions the continued calculation of distances as a bottleneck in RLSTC. When a point is scanned, the distance is computed between the cluster center and the current sub-trajectory [10]. Multiple quantum approaches are available for distance calculation [53]–[55], including the *swap test* [56], which estimates how much two quantum states differ. It scales linearly with the number of qubits whereas the classical method scales exponentially [57]. **(3.2) Quantum Action Selection:** The ϵ-greedy strategy can be replaced with Grover's algorithm [58]. Instead of random exploration, Grover's algorithm probabilistically selects an action that is beneficial by increasing the amplitude of the desired items in the superposition of all items. With N being the database size, the worst-case time complexity of Grover's algorithm is $O(\sqrt{N})$, improving over unstructured classic search's $O(N)$ complexity [59].

Another approach for action selection is Variable-Time Amplitude Amplification (VTAA), which extends Grover's algorithm. The algorithm given by Wang et al. cites a quadratic speedup compared to the best possible classical results [60]. The multi-armed bandit problem [36] that the algorithm is based on has enough differences from the ϵ-greedy strategy that it may require additional changes to make it feasible. However, because there are only two possible actions per state, it may still be a good fit. **(3.3) Loss Function:** Q-RLSTC can use quantum gradient descent (QGD) [61] instead of SGD to calculate the loss. QGD has a complexity of $O(1)$ compared with SGD's $O(N)$, with N the number of parameters. We can decode the optimal policy back to classical.

(4) Classical Segmentation and (5) Clustering: RLSTC uses a classical segmentation and clustering loop because of the limited number of qubits on NISQ hardware.

V. Conclusions and Future Research

Subtrajectory clustering is a core challenge in spatial data mining, limited by the scalability of classical methods like RLSTC. This paper introduced our work in progress regarding Q-RLSTC, a hybrid quantum-classical framework designed to study whether it is possible to leverage quantum computing to overcome these limitations by selectively integrating quantum algorithms for distance estimation, policy learning, and clustering while retaining classical components where quantum advantages are expected to be marginal or cost-prohibitive on current NISQ devices. We also identified and discussed the research challenges of quantum trajectory clustering. For future work, we intend to formalize and implement the proposed quantum algorithm using an open-source quantum software development framework such as IBM's Qiskit [12]. We will then compare its performance against that of its classical counterpart.

Acknowledgment

This work is supported in part by the National Science Foundation under Grant No. 2425838.

References

[1] H. Luo, Z. Bao, G. Cong, J. Culpepper, and K. Nguyen, "Let Trajectories Speak Out the Traffic Bottlenecks," *TIST*, vol. 13, 2022.
[2] Y. Wang, Y. Tian, B. Yang, J. Wang, X. Hu, and S. An, "Planning Flexible Bus Service as an Alternative to Suspended Bicycle-Sharing Service: A Data-Driven Approach," *JAT*, 2023.
[3] Z. Wang, C. Long, G. Cong, and C. Ju, "Effective and Efficient Sports Play Retrieval with Deep Representation Learning," in *SIGKDD*, 2019.
[4] K. Buchin, M. Buchin, J. Gudmundsson, J. Hendriks, E. H. Sereshgi, V. Sacristán, R. I. Silveira, J. Sleijster, F. Staals, and C. Wenk, "Improved Map Construction using Subtrajectory Clustering," in *SIGSPATIAL*, 2020.
[5] J.-G. Lee, J. Han, and K.-Y. Whang, "Trajectory clustering: a partition-and-group framework," in *SIGMOD*, 2007.

[6] P. K. Agarwal, K. Fox, K. Munagala, A. Nath, J. Pan, and E. Taylor, "Subtrajectory Clustering: Models and Algorithms," in *PODS*, 2018.

[7] Y. Mao, H. Zhong, H. Qi, P. Ping, and X. Li, "An Adaptive Trajectory Clustering Method Based on Grid and Density in Mobile Pattern Analysis," *Sensors*, vol. 17, no. 9, 2017.

[8] C. Jiashun, "A new trajectory clustering algorithm based on TRACLUS," in *Intl. Conf. on Comp. Sci. and Network Tech.*, 2012.

[9] H. Mustafa, C. Barrus, E. Leal, and L. Gruenwald, "GTraclus: A Local Trajectory Clustering Algorithm for GPUs," in *ICDEW*, 2021.

[10] A. Liang, B. Yao, B. Wang, Y. Liu, Z. Chen, J. Xie, and F. Li, "Sub-trajectory clustering with deep reinforcement learning," *The VLDB Journal*, vol. 33, no. 3, 2024.

[11] (2025) National Quantum Coordination Office (NQCO). National Quantum Initiative. [Online]. Available: https://www.quantum.gov/

[12] "Qiskit IBM Quantum Computing." [Online]. Available: https://www.ibm.com/quantum/qiskit

[13] C. Nayak, "Microsoft unveils Majorana 1, the world's first quantum processor powered by topological qubits," 2025.

[14] Z. Abohashima, M. Elhosen, E. H. Houssein, and W. M. Mohamed, "Classification with Quantum Machine Learning: A Survey," *arXiv*, 2020.

[15] C. Ciliberto, M. Herbster, A. D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossnig, "Quantum machine learning: a classical perspective," *Proc. of the Royal Society A: Math., Phys. and Eng. Sci.*, vol. 474, 2018.

[16] S. DiAdamo, C. O'Meara, G. Cortiana, and J. Bernabé-Moreno, "Practical Quantum K-Means Clustering: Performance Analysis and Applications in Energy Grid Classification," *IEEE Trans. Quantum Eng.*, 2022.

[17] A. Poggiali, A. Berti, A. Bernasconi, G. M. Del Corso, and R. Guidotti, "Quantum clustering with k-Means: A hybrid approach," *Theor. Comp. Sci.*, vol. 992, 2024.

[18] N. Pelekis, P. Tampakis, M. Vodas, C. Panagiotakis, and Y. Theodoridis, "In-DBMS Sampling-based Sub-trajectory Clustering," *EDBT*, 2017.

[19] Y. Xia and L. Zhou, "Improved clustering algorithm based on hypercube," *MLCR*, 2022.

[20] D. Qiao, X. Yang, Y. Liang, and X. Hao, "Rapid trajectory clustering based on neighbor spatial analysis," *Patt. Recog. Letters*, vol. 156, 2022.

[21] S. Dutta, A. Das, and B. K. Patra, "CLUSTMOSA: Clustering for GPS trajectory data based on multi-objective simulated annealing to develop mobility application," *Applied Soft Computing*, vol. 130, 2022.

[22] J. Gudmundsson, M. Horton, J. Pfeifer, and M. P. Seybold, "A Practical Index Structure Supporting Fréchet Proximity Queries among Trajectories," *ACM Trans. Spatial Algorithms Syst.*, vol. 7, no. 3, 2021.

[23] Y. Luo, X. Fang, W. Ke, C.-T. Lam, S.-K. Im, and L. Paquete, "An algorithm for improving lower bounds in dynamic time warping," *Expert Systems with Applications*, vol. 279, 2025.

[24] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[25] E. Chae, J. Choi, and J. Kim, "An elementary review on basic principles and developments of qubits for quantum computing," *Nano Convergence*, vol. 11, 2024.

[26] M. Schlosshauer, "Quantum decoherence," *Phys. Rep.*, vol. 831, 2019.

[27] Y. Zheng and X. Zhou, Eds., *Computing with Spatial Trajectories*. New York, NY: Springer, 2011.

[28] L. Gruenwald, T. Winker, U. Çalıkyılmaz, J. Groppe, and S. Groppe, "Index Tuning with Machine Learning on Quantum Computers for Large-Scale Database Applications," *QDSM*, 2023.

[29] M. Rath and H. Date, "Quantum data encoding: a comparative analysis of classical-to-quantum mapping techniques and their impact on machine learning accuracy," *EPJ Quantum Technology*, vol. 11, no. 1, 2024.

[30] M. Larocca, S. Thanasilp, S. Wang, K. Sharma, J. Biamonte, P. J. Coles, L. Cincio, J. R. McClean, Z. Holmes, and M. Cerezo, "Barren plateaus in variational quantum computing," *Nat. Rev. Phys.*, vol. 7, no. 4, 2025.

[31] J. Preskill, "Quantum Computing in the NISQ era and beyond," *Quantum*, vol. 2, 2018.

[32] M. Kohda, R. Imai, K. Kanno, K. Mitarai, W. Mizukami, and Y. O. Nakagawa, "Quantum expectation-value estimation by computational basis sampling," *Phys. Rev. Res.*, vol. 4, 2022.

[33] D. Arthur and S. Vassilvitskii, "k-means++: The Advantages of Careful Seeding," *SODA*, 2007.

[34] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, 2015.

[35] E. Frentzos, K. Gratsias, and Y. Theodoridis, "Index-based Most Similar Trajectory Search," in *ICDE 2007*.

[36] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. A Bradford Book, 2018.

[37] H. Ohno, "A quantum algorithm of K-means toward practical use," *Quantum Information Processing*, vol. 21, no. 4, 2022.

[38] S. DiAdamo, C. O'Meara, G. Cortiana, and J. Bernabé-Moreno, "Practical Quantum K-Means Clustering: Performance Analysis and Applications in Energy Grid Classification," *Trans. Quant. Eng.*, vol. 3, 2022.

[39] I. Kerenidis, J. Landman, A. Luongo, and A. Prakash, "q-means: a quantum algorithm for unsupervised machine learning," in *NEURIPS*, 2019.

[40] Z. Wu, T. Song, and Y. Zhang, "Quantum k-means algorithm based on Manhattan distance," *Quantum Inf. Proc.*, vol. 21, no. 1, 2021.

[41] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, 1982.

[42] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *KDD*, 1996.

[43] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: an efficient data clustering method for very large databases," *SIGMOD Rec.*, 1996.

[44] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: ordering points to identify the clustering structure," *SIGMOD Rec.*, vol. 28, no. 2, 1999.

[45] X. Xie, L. Duan, T. Qiu, and J. Li, "Quantum algorithm for MMNG-based DBSCAN," *Scientific Reports*, vol. 11, no. 1, 2021.

[46] K. Shiba, C.-C. Chen, M. Sogabe, K. Sakamoto, and T. Sogabe, "Quantum-Inspired Classification Algorithm from DBSCAN–Deutsch–Jozsa Support Vectors and Ising Prediction Model," *Applied Sciences*, vol. 11, no. 23, 2021.

[47] A. Banerjee, "Robust Density-Based Data Clustering Using a Quantum-Inspired Genetic Algorithm," in *CEC*, 2023.

[48] Z. Zhang, G. Ni, and Y. Xu, "Comparison of Trajectory Clustering Methods based on K-means and DBSCAN," in *ICIBA*, 2020.

[49] S. Y.-C. Chen, C.-H. H. Yang, J. Qi, P.-Y. Chen, X. Ma, and H.-S. Goan, "Variational Quantum Circuits for Deep Reinforcement Learning," *IEEE Access*, vol. 8, 2020.

[50] S. Lokes, C. S. J. Mahenthar, S. P. Kumaran, P. Sathyaprakash, and V. Jayakumar, "Implementation of Quantum Deep Reinforcement Learning Using Variational Quantum Circuits," in *TQCEBT*, 2022.

[51] N. F. Bar, H. Yetis, and M. Karakose, "An efficient and scalable variational quantum circuits approach for deep reinforcement learning," *Quantum Information Processing*, vol. 22, no. 8, 2023.

[52] A. Sequeira, L. P. Santos, and L. S. Barbosa, "Policy gradients using variational quantum circuits," *Quantum Mach. Intel.*, vol. 5, no. 1, 2023.

[53] K. Yu, G.-D. Guo, J. Li, and S. Lin, "Quantum Algorithms for Similarity Measurement Based on Euclidean Distance," *Intl. Jour. Theor. Phys.*, vol. 59, no. 10, 2020.

[54] A. A. Zecchi, C. Sanavio, S. Perotto, and S. Succi, "Improved amplitude amplification strategies for the quantum simulation of classical transport problems," *Quantum Science and Technology*, 2025.

[55] A. Berti, A. Bernasconi, G. M. Del Corso, and R. Guidotti, "The role of encodings and distance metrics for the quantum nearest neighbor," *Quantum Machine Intelligence*, vol. 6, no. 2, 2024.

[56] A. Barenco, A. Berthiaume, D. Deutsch, A. Ekert, R. Jozsa, and C. Macchiavello, "Stabilization of Quantum Computations by Symmetrization," *SIAM J. Comput.*, vol. 26, no. 5, 1997.

[57] X. Gitiaux, I. Morris, M. Emelianenko, and M. Tian, "SWAP test for an arbitrary number of quantum states," *Quantum Information Processing*, vol. 21, no. 10, 2022.

[58] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *STOC*, 1996.

[59] D. Barbosa, L. Gruenwald, L. D'Orazio, and J. Bernardino, "QRLIT: Quantum Reinforcement Learning for Database Index Tuning," *Future Internet*, vol. 16, no. 12, 2024.

[60] D. Wang, X. You, T. Li, and A. M. Childs, "Quantum Exploration Algorithms for Multi-Armed Bandits," *AAAI*, 2021.

[61] R. Chen, Z. Guang, C. Guo, G. Feng, and S.-Y. Hou, "Pure quantum gradient descent algorithm and full quantum variational eigensolver," *Frontiers of Physics*, vol. 19, no. 2, 2023.