

Quantum clustering and jet reconstruction at the LHC

Jorge J. Martínez de Lejarza^{✉,*}, Leandro Cieri^{✉,†}, and Germán Rodrigo^{✉,‡}

Instituto de Física Corpuscular, Universitat de València - Consejo Superior de Investigaciones Científicas, Parc Científic, E-46980 Paterna, Valencia, Spain

 (Received 17 May 2022; accepted 26 July 2022; published 23 August 2022)

Clustering is one of the most frequent problems in many domains, in particular, in particle physics where jet reconstruction is central in experimental analyses. Jet clustering at the CERN's Large Hadron Collider (LHC) is computationally expensive and the difficulty of this task will increase with the upcoming High-Luminosity LHC (HL-LHC). In this paper, we study the case in which quantum computing algorithms might improve jet clustering by considering two novel quantum algorithms which may speed up the classical jet clustering algorithms. The first one is a quantum subroutine to compute a Minkowski-based distance between two data points, whereas the second one consists of a quantum circuit to track the maximum into a list of unsorted data. The latter algorithm could be of value beyond particle physics, for instance in statistics. When one or both of these algorithms are implemented into the classical versions of well-known clustering algorithms (K -means, affinity propagation, and k_T -jet) we obtain efficiencies comparable to those of their classical counterparts. Even more, exponential speed-up could be achieved, in the first two algorithms, in data dimensionality and data length when the distance algorithm or the maximum searching algorithm are applied.

DOI: [10.1103/PhysRevD.106.036021](https://doi.org/10.1103/PhysRevD.106.036021)

I. INTRODUCTION

Quantum computing devices, which are based on the laws of quantum mechanics, offer the possibility to efficiently solve specific problems that become very complex or even unreachable for classical computers since they scale either exponentially or superpolynomially. Algorithms used in quantum computers [1] exploit the quantum principles of superposition and entanglement to clearly manifest a speed-up advantage over the classical counterpart algorithms. Two examples of these quantum algorithms are the well-known cases of Grover's database querying [2] and Shor's factoring of integers into primes [3]. These two quantum methods shown, for first time in the 1990s, a clear potential advantage over their corresponding classical analogs. In the past recent years, we have witnessed an impressively fast development of quantum computing algorithms going from optimization problems such as portfolios in fintech [4], applications in quantum chemistry [5], nuclear physics and Monte Carlo simulation [6–8], combinatorial optimization [9], to state diagonalization [10,11].

Very recently, quantum algorithms have started to be applied in solving problems which appear in high-energy particle physics¹ (HEP). The data already taken at the CERN's Large Hadron Collider (LHC) and its upcoming Run 3 (which is scheduled to start in the spring of 2022) demand intense data analysis routines and very precise theoretical predictions [13] which are computationally very expensive. This situation will be even more challenging in the posterior high-luminosity phase of the LHC (HL-LHC) [14] and the planned future colliders [15–17]. Recent applications of quantum algorithms to HEP cover diverse subareas such as jet clustering [18–20], jet quenching [21], determination of parton densities [22], simulation of parton showers [23–25], heavy-ion collisions [26], quantum machine learning [27–34], lattice gauge theories [35–38] and multiloop Feynman integrals [39,40].

In the present paper we address the problem of clustering and jet reconstruction from collision data, which is a non-trivial and computationally expensive task, as it often involves performing optimizations over potentially large numbers of final-state particles. To give a rough idea of how demanding this activity is, the state-of-the-art algorithm in jet clustering needs few months to clusterize all the particles generated in the data of interest that is produced at the LHC in just one year [41]. Moreover, with the

*Jorge.M.Lejarza@ific.uv.es

†lcieri@ific.uv.es

‡german.rodrigo@csic.es

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI. Funded by SCOAP³.

¹For a recent review on the applications of quantum computing to data analysis in HEP we refer the reader to Ref. [12] and references therein.

upcoming HL-LHC, the number of events will be up to an order of magnitude more than in earlier runs [42] and also the pile-up (simultaneous proton-proton collisions per bunch crossing) will increase by a factor of 5 [43]. Therefore, the state-of-the-art algorithm will require roughly 50 times the computational time needed now. So we would be talking about a few tens of years for processing the data of interest generated in just a year. This evidences the necessity of developing fast and effective jet clustering algorithms.

With this in mind, we consider the possibility of using quantum algorithms to improve the velocity in jet identification. Here we focus on three well-known classical algorithms: the K-means clustering [44,45], the affinity propagation (AP) algorithm [46], and the k_T -jet clustering method in all its variants [47–51]. We propose the corresponding quantum versions of the precedents algorithms: quantum K-means clustering, quantum AP-algorithm and quantum k_T -based algorithms.

Clustering is one of the most frequent classic problems in machine learning and computational geometry. It is a major data analysis tool used in such domains as marketing research, data mining, bioinformatics, image processing, pattern recognition and also in HEP. The popular K-means formulation [44,45], which is a method of vector quantization originally proposed for signal processing, involves the partition of n observations into K clusters in which each observation belongs to the cluster with the nearest mean (cluster center or cluster centroid), serving as a prototype of the cluster. Solving this problem exactly is NP-hard² (non-deterministic polynomial-time hardness), even with just two clusters [52]. Forty years ago, Lloyd [53] proposed a local search solution that is still very widely used today. Usually referred to simply as K-means, Lloyd’s algorithm begins with K arbitrary centers, typically chosen uniformly at random from the data points. Each point is then assigned to the nearest center, and each center is recomputed as the center of all points assigned to it. These two steps (assignment and center calculation) are repeated until the process stabilizes.

The improved version of the K-means method, the K-means++ algorithm [54], initializes the K-means algorithm by choosing random starting centers with very specific probabilities. This strategy outperforms K-means in terms of both accuracy and speed, often by a substantial margin [54]. K-means is a method of cluster analysis using a prespecified number of clusters. It requires an advance (*a priori*) knowledge of K and belongs to the group of the so-called *partitional clustering algorithms*. The classical K-means algorithm has already been used in high-energy physics in Refs. [55–58]. For example, in Ref. [55], the use of K-means led to 25% and 40%

²NP-hard problems are not solvable in polynomial time but can be verified in polynomial time.

improvement of the top quark and W boson mass resolution, respectively, compared to the k_T (Durham) algorithm, and reduced the systematic uncertainty in the measured peak positions. As a drawback, K-means was roughly three times slower than the Durham algorithm, therefore the interest to explore potential speed ups. In Ref. [56], the tagging performance of N -subjettiness for boosted top quarks was improved through minimization using a variant of K-means. The XCone jet algorithm introduced in Ref. [57] is closely related to the traditional K-means and its variants. Finally, K-means has been used in Ref. [58] to identify minijets at low p_T .

The AP algorithm, is a clustering method that identifies representative examples (exemplars) within a given dataset by exchanging messages between all data points. Points are then grouped with their most representative exemplar to give the final set of clusters. The AP algorithm has been successfully applied to a wide range of problems including face recognition, gene identification, putative exons using microarray data [59–61] and astrophysics [62]. In high-energy physics, it has been used to cluster replicas of parton densities [63]. In Ref. [46], it was shown that AP might be faster and more accurate than the K-means [44,45] clustering algorithm in solving certain problems. The AP algorithm is solid and well understood and the number of clusters is not needed to be prespecified. Among its disadvantages, the high time complexity turns out to make it not suitable for very large datasets, and the clustering result is typically sensitive to the parameters involved in the AP algorithm. Our motivation in using it for jet clustering comes from the fact that it does not need the number of clusters to be defined beforehand.

Hierarchical clustering also known as hierarchical cluster analysis (HCA) is also a method of cluster analysis that seeks to build a hierarchy of clusters without having an *a priori* fixed number of clusters. The k_T -based algorithms [64] belong to the hierarchical category, which needs a linkage function that defines the distance between any two subsets (and relies on the base distance between elements). It is the most widely used jet clustering algorithm in the LHC experiments.

The quantum K-means clustering algorithm was presented in Refs. [19,65] for HEP. An earlier study of the quantum K-means can be found in Ref. [66]. Both implementations make use of the Euclidean distance to perform the clustering of particles. In this paper, we present a version of the quantum K-means clustering algorithm which is based on the definition of a Minkowskian distance at the quantum level for the first time. Considering the case of the quantum version of the AP algorithm, it uses the invariant sum squared as a metric in the similarity matrix and calculates it through a quantum subroutine with a similar procedure as in the quantum K-means implementation. Regarding the quantum k_T -based algorithms, to our knowledge, it is the first time it has been presented in the

literature. In addition, the search for the maximum distance used in our implementation is performed with a new quantum algorithm. This new quantum method is presented in a general way, and we comment on its reach regarding other areas of interest. Beyond the specific application to jet clustering, the quantum algorithms presented in this paper are of interest to the particle physics and quantum computing communities.

This paper is organized as follows. In Sec. II we introduce our notation and we define the Euclidean and Minkowskian quantum distances. In Sec. III we present our new quantum algorithm in order to search the maximum in a set of a given number of elements. We consider the quantum version of the K-means clustering, AP and k_T -based algorithms in Sec. IV. In Sec. V we present our results considering the quantum simulations of these algorithms and a proof-of-concept implementation with Gaussian datasets as well as with simulated LHC physical events. We also compare their performance in detail. We discuss their differences and conceptual similarities and we compare them with their classical counterparts. A brief summary of our results is presented in Sec. VI.

II. QUANTUM DISTANCES

In quantum computing, it is essential to have the ability to measure quantum entanglement between two states, as in many cases it determines the possibility of obtaining a quantum advantage [67]. We rely on the *SwapTest* method [68] (see the Appendix for more details) in order to probe the entanglement between two given states. The definition of quantum distances (Euclidean distance or Minkowski invariant sum squared) presented in this section, makes use of the *SwapTest* procedure.

A. Euclidean quantum distance

We start by considering N data points or vectors in an Euclidean d -dimensional space, $\{\mathbf{x}_i\}_{i=1,\dots,N}$, which are encoded as quantum states of the form

$$|x_i\rangle = |\mathbf{x}_i|^{-1} \sum_{\mu=1}^d x_{i,\mu} |\mu\rangle, \quad (1)$$

where $|\mathbf{x}_i| = \sqrt{\sum_{\mu=1}^d (x_{i,\mu})^2}$ is the modulus of the vector \mathbf{x}_i , and $x_{i,\mu}$ are its components. Each vector requires $n \geq \log_2 d$ qubits to be encoded, i.e., for $d = 3$ we need two entangled qubits where one of its states remains free and is not used. The Euclidean distance between two vectors \mathbf{x}_i and \mathbf{x}_j is defined classically as

$$d_E^{(C)}(\mathbf{x}_i, \mathbf{x}_j) = |\mathbf{x}_i - \mathbf{x}_j|, \quad (2)$$

where the subscript E stands for Euclidean and the superscript C denotes that it corresponds to the classical version.

The quantum analog of Eq. (2) is obtained by using the controlled *SwapTest* method. In order to define the Euclidean quantum distance between the d -dimensional vectors \mathbf{x}_i and \mathbf{x}_j , we entangle the corresponding associated quantum states $|x_i\rangle$ and $|x_j\rangle$, and define the following subsidiary states

$$\begin{aligned} |\psi_1\rangle &= \frac{1}{\sqrt{2}}(|0, x_i\rangle + |1, x_j\rangle), \\ |\psi_2\rangle &= \frac{1}{\sqrt{2Z_{ij}}}(|\mathbf{x}_i||0\rangle - |\mathbf{x}_j||1\rangle), \end{aligned} \quad (3)$$

where $Z_{ij} = |\mathbf{x}_i|^2 + |\mathbf{x}_j|^2$ is a normalization factor and $|0\rangle$ and $|1\rangle$ are the states of an ancillary qubit. It is also convenient to define the swapped state $|\psi'_1\rangle$

$$|\psi'_1\rangle = \frac{1}{\sqrt{2}}(|x_i, 0\rangle + |x_j, 1\rangle). \quad (4)$$

The inner products between the quantum states defined in Eqs. (3) and (4) are written as follows

$$\begin{aligned} \langle\psi'_1|\psi_2\rangle &= \frac{1}{\sqrt{2Z_{ij}}}(|\mathbf{x}_i|\langle x_i| - |\mathbf{x}_j|\langle x_j|), \\ \langle\psi_2|\psi_1\rangle &= \frac{1}{\sqrt{2Z_{ij}}}(|\mathbf{x}_i||x_i\rangle - |\mathbf{x}_j||x_j\rangle). \end{aligned} \quad (5)$$

From where

$$\langle\psi'_1|\psi_2\rangle\langle\psi_2|\psi_1\rangle = \frac{1}{2Z_{ij}}|\mathbf{x}_i - \mathbf{x}_j|^2. \quad (6)$$

Therefore (see Eq. (A5) in the Appendix), the Euclidean quantum distance is

$$d_E^{(Q)}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{2Z_{ij}(2P_{\Psi_3}(|0\rangle) - 1)}, \quad (7)$$

where the superscript Q refers to the *quantum* version of the distance d_E and the subscript Ψ_3 in the probability P , means that it is considered the resulting probability of measuring the ancillary qubit in the state $|0\rangle$ in the last of the three steps in the *SwapTest* procedure.

B. Quantum invariant sum squared in Minkowski space

Vectors in high-energy physics are defined in a four-dimensional space-time with Minkowski metric. They have the form $x_i = (x_{i,0}, \mathbf{x}_i)$, where $x_{i,0}$ is the temporal component and \mathbf{x}_i represent the three spatial components. In the following, we assume that the dimension of the space-time is d , where $d - 1$ is the number of spatial components. We shall define the analog of the Euclidean classical distance in the Minkowski space corresponding to the invariant sum

squared $s_{ij}^{(C)}$, which is commonly called invariant mass squared when vectors are particle four-momenta,

$$s_{ij}^{(C)} = (x_{0,i} + x_{0,j})^2 - |\mathbf{x}_i + \mathbf{x}_j|^2. \quad (8)$$

This quantity, which is Lorentz invariant, can be used as test distance to measure similarity between particle momenta. It is also equivalent to the distance used in some of the traditional jet-clustering algorithms at e^+e^- colliders [69–71]. It is necessary to apply twice the *SwapTest* subroutine (presented in the Appendix) for computing the Minkowski-type distance through a quantum algorithm. Once for the spatial and once for the temporal components.

The spatial distance is computed through the procedure explained in the previous section with a slight modification with respect to Eq. (5) (change of sign in the term proportional to qubit $|1\rangle$)

$$|\psi_2\rangle \rightarrow |\psi_2\rangle = \frac{1}{\sqrt{Z_{ij}}}(|\mathbf{x}_i|0\rangle + |\mathbf{x}_j|1\rangle), \quad (9)$$

whereas the temporal distance is computed as a result of the overlap of the following states:

$$\begin{aligned} |\varphi_1\rangle &= H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \\ |\varphi_2\rangle &= \frac{1}{\sqrt{Z_0}}(x_{0,i}|0\rangle + x_{0,j}|1\rangle), \end{aligned} \quad (10)$$

where $Z_0 = x_{0,i}^2 + x_{0,j}^2$. Then, applying the *SwapTest* to these states one gets the relation:

$$P(|0\rangle|_{\text{time}}) = \frac{1}{2} + \frac{1}{2} |\langle \varphi_1 | \varphi_2 \rangle|^2, \quad (11)$$

where the overlap $|\langle \varphi_1 | \varphi_2 \rangle|^2$ is trivially given by

$$|\langle \varphi_1 | \varphi_2 \rangle|^2 = \frac{1}{2Z_0} (x_{0,i} + x_{0,j})^2. \quad (12)$$

Therefore:

$$(x_{0,i} + x_{0,j})^2 = 2Z_0(2P_{\Psi_3}(|0\rangle|_{\text{time}}) - 1). \quad (13)$$

At this point, the quantum version of the invariant sum squared follows from the combination of results from Eq. (7) and Eq. (13):

$$s_{ij}^{(Q)} = 2(Z_0(2P_{\Psi_3}(|0\rangle|_{\text{time}}) - 1) - Z_{ij}(2P_{\Psi_3}(|0\rangle|_{\text{spatial}}) - 1)). \quad (14)$$

The quantum circuit used to implement the invariant sum-squared distance is shown in Fig. 1.

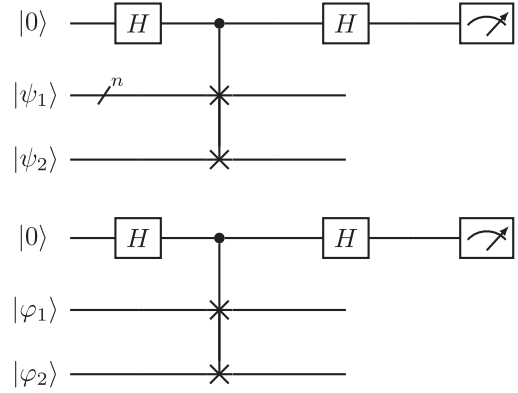


FIG. 1. Quantum circuit to obtain the invariant sum squared between two d -dimensional vectors in Minkowski space.

In the first three wires, the *SwapTest* is applied to the spatial components, where we assume that the states ψ_1, ψ_2 have been loaded from a quantum random access memory (qRAM) in $\mathcal{O}(\log(d-1))$, since the state ψ_1 is encoded in $\log_2(d-1)$ qubits. On the other hand, from the fourth wire onward, the *SwapTest* is applied to the temporal components. In this case, it takes $\mathcal{O}(1)$, since we only have 1-dimensional qubit states.

III. QUANTUM MAXIMUM SEARCH BY AMPLITUDE ENCODING

Finding a particular member belonging to a dataset is a recurring problem in data analysis. This is a computationally very expensive task. However, quantum computing offers suitable tools to solve data query in a shorter computational time. In particular, it is well known the quadratic speed up exhibited by Grover's algorithm [2]. In this paper, we present a considerably simpler algorithm that is used exclusively to find the maximum in a list of values. This algorithm, although very elementary, is sufficiently accurate for the applications that we will present in Secs. VA and VC. To our knowledge, it is the first time presented in the literature.

Let $L[0, \dots, N-1]$ be an unsorted list of N items. Solving the maximum searching problem is to find the index y such that $L[y]$ is the maximum. The quantum algorithm to solve that problem using amplitude encoding proceeds in two steps:

- (1) The list of N elements is encoded into a $\log_2(N)$ qubits state as follows:

$$|\Psi\rangle = \frac{1}{\sqrt{L_{\text{sum}}}} \sum_{j=0}^{N-1} L[j]|j\rangle, \quad (15)$$

where $L_{\text{sum}} = \sum_{j=0}^{N-1} L[j]^2$ is a normalization constant. This amplitude encoding is achieved using qRAM.

- (2) The final state is measured. This step is rerun several times to reduce the statistical uncertainty. Once done, the most repeated state gives us the maximum.

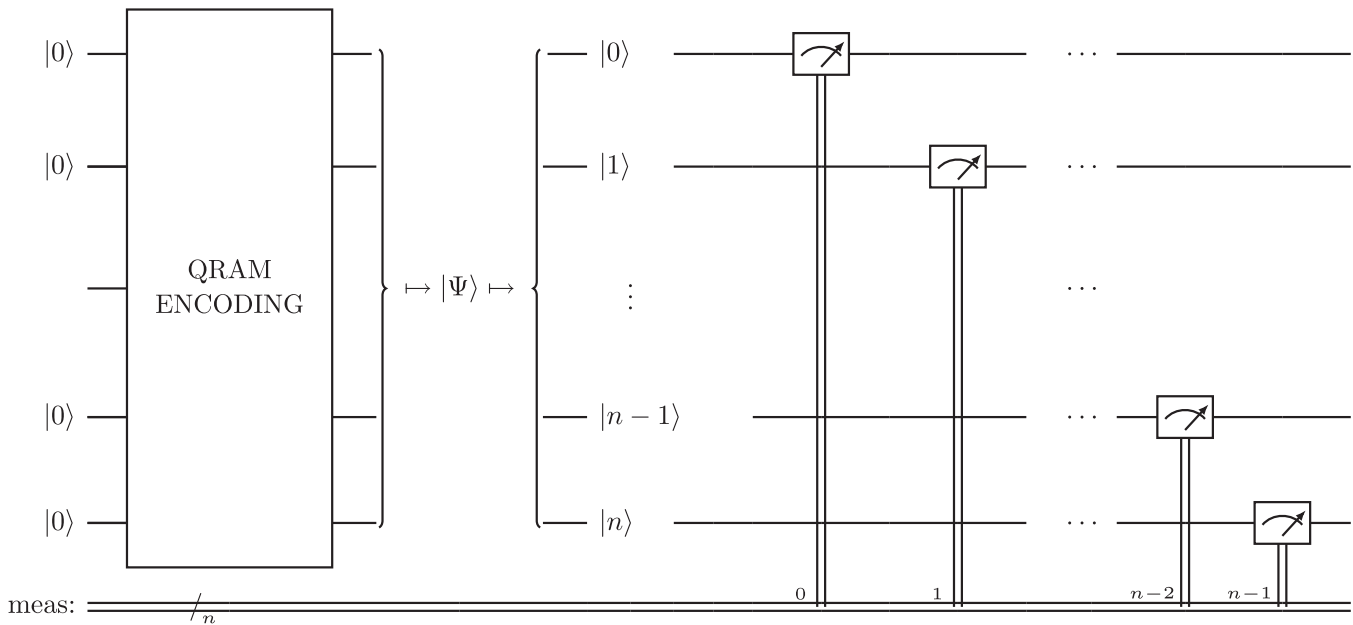


FIG. 2. Quantum circuit for quantum maximum searching by qRAM amplitude encoding.

The graphical representation of the algorithm is shown in Fig. 2, where $n = \log_2(N)$ qubits are needed to encode a list of N (real) elements.

The bottleneck of this procedure underlies in encoding data into a quantum state. Assuming data is stored in a qRAM, as would be the case on a true universal quantum computer, encoding takes $\mathcal{O}(\log_2(N))$ steps [72–79]. The corresponding classical algorithms typically used to obtain the minimum of an unsorted list of N items are of order $\mathcal{O}(N)$. Therefore, with the assumptions considered, the improvement introduced by this quantum algorithm is exponential.

The well-known quantum minimum searching algorithm proposed by Dürr and Høyer [80] is $\mathcal{O}(\sqrt{N})$. After their theoretical paper [80] the algorithm was studied and implemented in a quantum simulator (see Ref. [81]). In summary, previous implementations [81] of the Dürr and Høyer algorithm suggests that it could be improved, given the excessive number of qubits needed to implement the method, the unviability to hard code a different oracle for each element, the large number of *shots* required and (in some cases) the poor performance obtained. This is the aim of the new quantum maximum searching algorithm by amplitude encoding through qRAM presented here: the improvement of the previous enumerated challenges.

Nevertheless, the new algorithm presented in this paper and the corresponding Dürr and Høyer quantum method share common features that could lead to misidentification of the respective absolute maximum and minimum. These cases, in which the list typically presents a very low standard deviation (or the largest/minimum values are very close to each other) could manifest difficulties related to the

fact that the probability of measuring several candidates would be almost identical.

Regarding the practical implementation of the quantum algorithm presented in this paper, the results shown in Sec. V reveal that these potential difficulties do not manifest strongly in the context of jet clustering.

Beyond the jet clustering procedure in HEP, there are other fields where our quantum algorithm could be of value. For instance, in the so-called extreme value theory (EVT) [82]. According to Gumbell 1958 [83], this particular field studies the probability distribution of the desired data by focusing on the outliers with the ultimate goal of being able to predict them in the future. It is precisely in this estimation of the extreme values where our algorithm could be useful. Since for the predictive models historical data has to be analyzed and therefore extreme values have to be searched in large data lists. This would mean that our algorithm could be implemented successfully in statistical analysis of extreme data, including actuarial and financial sciences, meteorology, material sciences, engineering and environmental sciences climatology, geology, hydrology, and highway traffic analysis [84–86].

IV. QUANTUM CLUSTERING ALGORITHMS

A. K-means algorithm

K-means is an unsupervised machine learning algorithm that classifies the elements of a dataset into K groups called clusters [44,45]. The data points within each cluster have to be as similar (near) as possible whereas the clusters themselves have to be as different (far) as possible from each other. The input for this algorithm is a set of N data

points or vectors, in d dimensions as well as the number of clusters K , with $K \leq N$, and its output is a set of K centroids, calculated by averaging the position of the data points corresponding to each group, thus defining K clusters. The flow chart of this algorithm is the following:

- (1) K initial centroids within the data points are generated. They can be generated randomly or through a specific method such as `K-means++` [54].
- (2) Each data point is assigned to its closest centroid according to a distance that has been defined in advance, thus the K clusters are defined. The most commonly used distance is the Euclidean distance.
- (3) Each centroid is recalculated by averaging the associated data points.
- (4) Steps 2 and 3 are repeated until all centroids stabilize and convergence is achieved.

This `K-means` algorithm has a sophisticated quantum version that differs from its classical counterpart in two points [87]. First, the quantum `K-means` introduces a quantum method to calculate the distance between data points. Second, the quantum version also includes a procedure for obtaining the minimum distance of each data point with respect to the K centroids, which is achieved by Dürr and Høyer's algorithm [80].

In this paper, we focus on a new quantum version of the `K-means` algorithm, where the calculation of distances is made quantumly and the minimum distance of each data point to the centroids is obtained with the quantum maximum searching algorithm³ explained in Sec. III. Other quantum versions of the `K-means` algorithm have been studied in Refs. [19,65,66], where an Euclidean distance was used to separate the particles from each other. In this paper, we analyze for the first time an implementation of the `K-means` algorithm with a Minkowski-type quantum distance, as defined in Sec. II B.

The time complexity of this algorithm is estimated by analyzing the time complexity of its components. The distances that have to be calculated are $\mathcal{O}(N)$, the search of a minimum distance for every data point with respect to the centroids would be $\mathcal{O}(\log K)$, and the calculation of each distance itself would require $\mathcal{O}(\log(d-1))$ qubits assuming the data is stored in a qRAM. This results in a speedup from $\mathcal{O}(NKd)$ in the classical version to $\mathcal{O}(N \log K \log(d-1))$ in our quantum version. Therefore an exponential speed-up in the number of clusters and in the vector dimensionality would be achieved. A quantum simulation of the quantum `K-means` algorithm is presented in Sec. V A.

B. Affinity propagation algorithm

Although `K-means` is a successful algorithm capable of clustering data in a satisfactory manner, it needs the number

³We may apply this algorithm for finding the minimum since obtaining the minimum among the distances is equivalent to obtaining the maximum of their inverses: s_{ij}^{-1} .

of clusters K to be defined beforehand, which is not typically the case in HEP applications. The affinity propagation (AP) algorithm [46], which is an unsupervised machine learning algorithm, does not need the number of clusters as an input. AP only takes as input the data points that have to be classified. So, let x_1, \dots, x_N be a set of data points. Then, a function s to quantify the similarity between points is computed. In such a way that $s(i, j) \geq s(i, k)$ if and only if x_i is more similar to x_j than to x_k . The most common metrics to measure the similarity is the negative squared distance of the two points we are comparing: $s(i, j) = -|x_i - x_j|^2$. The diagonal $s(i, i)$ of the matrix s is especially relevant since it stores values referred as "preferences" that are related to how likely a particular instance is to become an exemplar, i.e., a cluster. Most of the metrics make the diagonal $s(i, i)$ be $s(i, i) = 0, \forall i \leq N$, although it can be different from 0. Hence, on the first iteration, every element $s(i, i)$ is set to the same certain value, which is typically the median similarity of all pairs of inputs. Next, two matrices are calculated that are related to the concept of message exchanging between data points [46]. First, there is the responsibility matrix R . This matrix contains the values $r(i, k)$ that quantify the suitability of point k to serve as the exemplar for point i , compared to other candidate exemplars for i . Then comes the availability matrix A , whose elements $a(i, k)$ reflect how appropriate it would be for point i to select point k as its exemplar, relative to the preferences of other points for k as an exemplar. As they have been described, both matrices could be viewed as log-probability ratios. Then, the AP flow chart reads:

- (1) The matrices R and A are initialized to zero.
- (2) The responsibility matrix is computed:

$$r(i, k) = s(i, k) - \max_{q \neq k} \{a(i, q) + s(i, q)\}. \quad (16)$$

- (3) The availability matrix is computed:

$$a(i, k) = \min \left(0, r(k, k) + \sum_{q \notin \{i, k\}} \max(0, r(q, i)) \right) \quad (17)$$

for $i \neq k$, and

$$a(k, k) = \sum_{q \neq k} \max(0, r(q, k)). \quad (18)$$

- (4) Steps 2 and 3 are repeated until either the cluster boundaries remain unchanged for several iterations, or a predetermined number (of iterations) is reached.

Once convergence has been reached, the exemplars i.e., the clusters, are obtained from the final matrices as those whose $r(i, i) + a(i, i) > 0$. This algorithm takes $\mathcal{O}(N^2)$ steps to fill the similarity matrix, and also computing each element takes $\mathcal{O}(d)$, since a distance between two d -dimensional points has to be calculated. Moreover, steps

2 and 3 are repeated a number T of times, so the final time complexity of this algorithm is $\mathcal{O}(N^2 T d)$.

Here, a quantum (hybrid) algorithm is presented which uses the invariant sum squared as a metric in the similarity matrix and calculates it through a quantum subroutine, as the K-means algorithm described in the Sec. IV A. Then, a speedup would be achieved, since computing the distances only requires $\mathcal{O}(\log(d-1))$ qubits. So, the quantum AP algorithm, which is as far as we know completely original, would have a time complexity of $\mathcal{O}(N^2 T \log(d-1))$.

C. Generalized k_T -jet algorithm

The inclusive variant of the generalised k_T -jet algorithm is formulated as follows [64]:

- (1) For each pair of partons i, j the following distance is computed:

$$d_{ij} = \min(p_{T,i}^{2p}, p_{T,j}^{2p}) \Delta R_{ij}^2 / R^2, \quad (19)$$

with $\Delta R_{ij}^2 = (y_i - y_j)^2 + (\phi_i - \phi_j)^2$, where $p_{T,i}$, y_i and ϕ_i are the transverse momentum (with respect to the beam direction), rapidity and azimuth of particle i . R is a jet-radius parameter usually taken of order 1.

For each particle i the beam distance is $d_{iB} = p_{T,i}^{2p}$.

- (2) Find the minimum d_{\min} among all the distances d_{ij} , d_{iB} . If d_{\min} is a d_{ij} , the particles i and j are merged into a single particle summing their four-momenta (this is the E-scheme recombination); if d_{\min} is a d_{iB} then the particle i is declared as a final jet and it is removed from the list.
- (3) Repeat from step 1 until there are no particles left.

It is noticeable that for specific values of p in Eq. (19), the generalized k_T algorithm is reduced to the algorithms: k_T ($p = 1$), Cambridge/Aachen ($p = 0$) and anti- k_T ($p = -1$). As it is claimed in Ref. [88], this classical version of the k_T -jet algorithm is $\mathcal{O}(N^3)$, since the bottleneck of the algorithm is scanning the $\mathcal{O}(N^2)$ table with all the distances d_{ij} , d_{iB} , and it has to be done N times. Nevertheless, the `FastJet` algorithm is able to reduce the complexity to $\mathcal{O}(N^2)$. It is achieved by identifying each particle's geometrical nearest neighbor, thereby it is not necessary to construct a size- N^2 table of d_{ij} , but only the size- N array, $d_{i\mathcal{G}_i}$, where \mathcal{G}_i is i 's geometrical nearest neighbor. Furthermore, this `FastJet` algorithm can be optimized further using the so-called Voronoi diagrams achieving a reduction in the time complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log N)$.

Regarding the quantum version of this algorithm, the distance ΔR_{ij}^2 will be computed classically whereas the minimum will be obtained through a quantum algorithm. This is due to the fact that the speed up achieved by obtaining the minimum here with a quantum subroutine will be dominant. Thereby, what is to be used here is

the new algorithm to obtain the maximum of a list of values (see Sec. III). So obtaining the minimum among all the distances d_{ij} , d_{iB} will turn out to be obtaining the maximum of its inverses: d_{ij}^{-1} , d_{iB}^{-1} . Actually, these inverse distances are what will be computed directly for each pair i, j . Since computing the distances and thereafter computing its inverses would require traversing a vector of size N , so it would have a complexity $\mathcal{O}(N^2)$. With that in mind one may also directly compute d_{ij}^{-a} , d_{iB}^{-a} , with $a \in \mathbb{N}$, to increase the separation among the data, which makes the maximum more likely when measuring. And this will not increase the overall time complexity of the algorithm either. In Sec. V we compare the results obtained when applying the algorithm with different a values.

The quantum maximum searching algorithm presented above could be applied to the k_T -jet algorithm successfully because accuracy is not critical. Even if our quantum algorithm fails to obtain the absolute maximum in one of the multiples iterations, this could end up not affecting the overall jet clustering process. Since an error in finding the maximum will provoke a flip in the order in which two particles merge, and the final result will in many cases be independent of this permutation.

As a final remark, notice that the k_T -jet quantum algorithm would be $\mathcal{O}(N^2 \log(N))$, since computing all the distances takes $\mathcal{O}(N^2)$ and finding the minimum would be $\mathcal{O}(\log(N))$, in comparison with the $\mathcal{O}(N^3)$ that requires its classical analog [88]. Furthermore, the quantum minimum searching could also be implemented in the `FastJet` algorithm of complexity $\mathcal{O}(N^2)$. In this case, the resulting quantum algorithm would be $\mathcal{O}(N \log(N))$, which is of the same order as the `FastJet` algorithm version with Voronoi diagrams, which is the most efficient clustering algorithm known to date. This quantum `FastJet` algorithm has been tested in Sec. IV C with LHC physical datasets.

V. QUANTUM SIMULATIONS

The implementation of the quantum algorithms has been performed through the open-source IBMQ software. In particular, the Python module `Qiskit` developed by IBMQ has been used to build the quantum circuit to calculate the invariant sum squared as described in Sec. II B for the K-means and the AP algorithm, as well as to build the quantum circuit for finding the minimum distance in the K-means and the k_T -jet algorithm. Afterward, these quantum subroutines have been introduced into their respective classical algorithm substituting the classical part they are speeding up. The `Qiskit` module serves for executing circuits on real quantum devices. Nevertheless, in previous studies such as [66,89], it has been found that the experimental error associated with the quantum devices provided by IBMQ

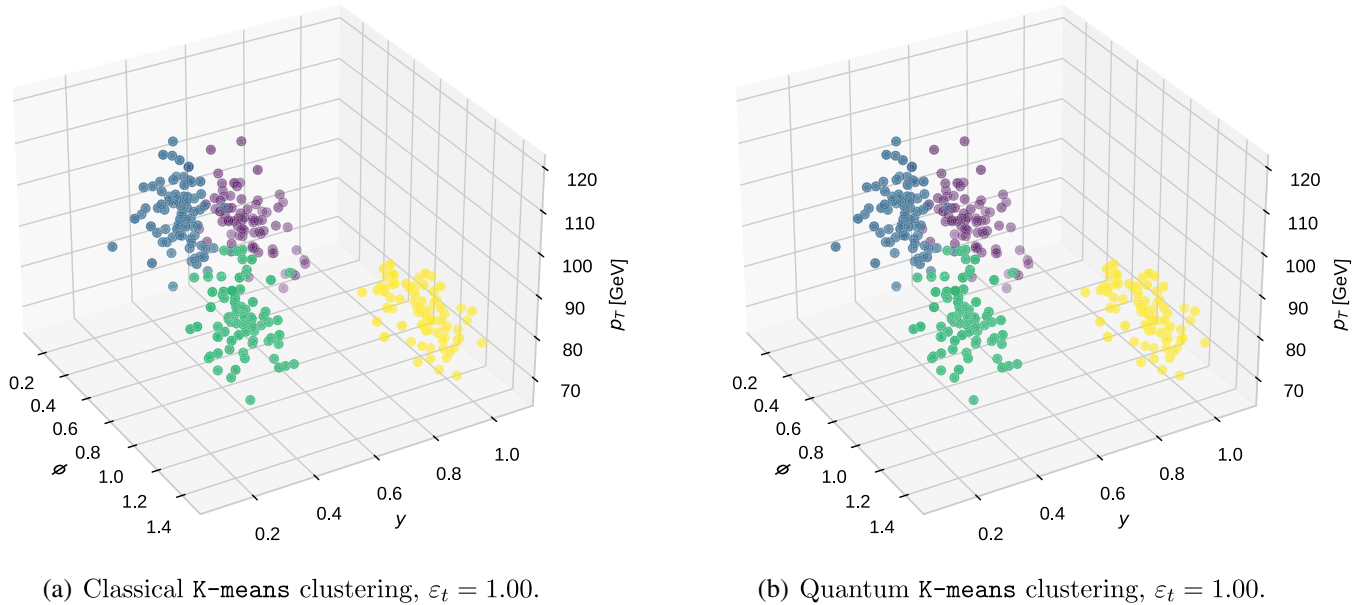


FIG. 3. In different colors, clusters identified after 5 iterations by the classical and quantum versions of the \mathcal{K} -means algorithm in a Gaussian dataset generated with a random seed and a standard deviation of 2.0 from the cluster centroids. Note that clusterization has been performed using a Minkowski-type distance assuming that all particles are massless and on shell and the efficiencies of both algorithms are $\varepsilon_t = 1.00$. (a) Classical \mathcal{K} -means clustering, $\varepsilon_t = 1.00$. (b) Quantum \mathcal{K} -means clustering, $\varepsilon_t = 1.00$.

is not yet sufficiently small to extract significant results. Hence, the algorithms presented here have been executed on a quantum simulator that offers an unrestricted and noise-free environment. A quantum implementation in an existing quantum device taking advantage of the claimed maximum speed-up is also not possible, as a qRAM architecture does not exist yet. Nonetheless, the quantum simulations in IBMQ presented in this section show a satisfactory performance and clustering efficiencies comparable to those of their classical counterparts.

A. Quantum \mathcal{K} -means with Minkowski-type distance

At this point we present our implementation of the \mathcal{K} -means algorithm with the invariant sum squared as a distance as well as a maximum searching algorithm, and compare its performance with its classical analog. To this end, we have generated 15 Gaussian clustered datasets of $N = 300$ three-dimensional vectors⁴ with different levels of noise and clustering using the *Scikit-learn* function *make_blobs*, which gives us the *true labels*⁵

⁴In general, it is possible to relate this generated set of three-dimensional vectors, to a physical event at the LHC. It is enough to consider the set of n three-dimensional vectors as massless partons recoiling against a small number of tagged particles.

⁵The data generator function preassigns each data point to a particular cluster, so by analyzing these *true labels* one may know which is the correct way to cluster the data.

of the generated data. These *true labels* of the data points are used to calculate the true efficiencies, ε_t , of the algorithms when analyzing Gaussian datasets. The efficiency ε_t is obtained as the ratio of the number of particles classified by the algorithm in the same way as the *true labels* to the total number of particles. We then applied the hybrid and classical versions of the \mathcal{K} -means algorithm to each dataset. Note that the data we are analyzing represent the particle four-momenta in such a way that the three-dimensional vectors correspond to the spatial components, while the temporal components are calculated assuming that all particles are massless and on shell. Results are shown in Figs. 3 and 4.

Regarding Fig. 3 one can see at a glance that both classical and quantum versions perform the clustering in the same way in the three-dimensional space of transverse momentum (p_T), rapidity (y), and azimuth (ϕ).

Figure 4 shows the efficiency in the reconstruction of the clusters as a function of the standard deviations used to generate the data, namely we check whether clustering occurs as expected. It is evident that for small values of the standard deviation both algorithms perform really well, with efficiencies close to one, while for larger values of the standard deviation (i.e., highly noisy data) both efficiencies drop. Furthermore, we can compare the performances of the \mathcal{K} -means algorithm when the seed of the centroids is chosen randomly [see Fig. 4(a)], with

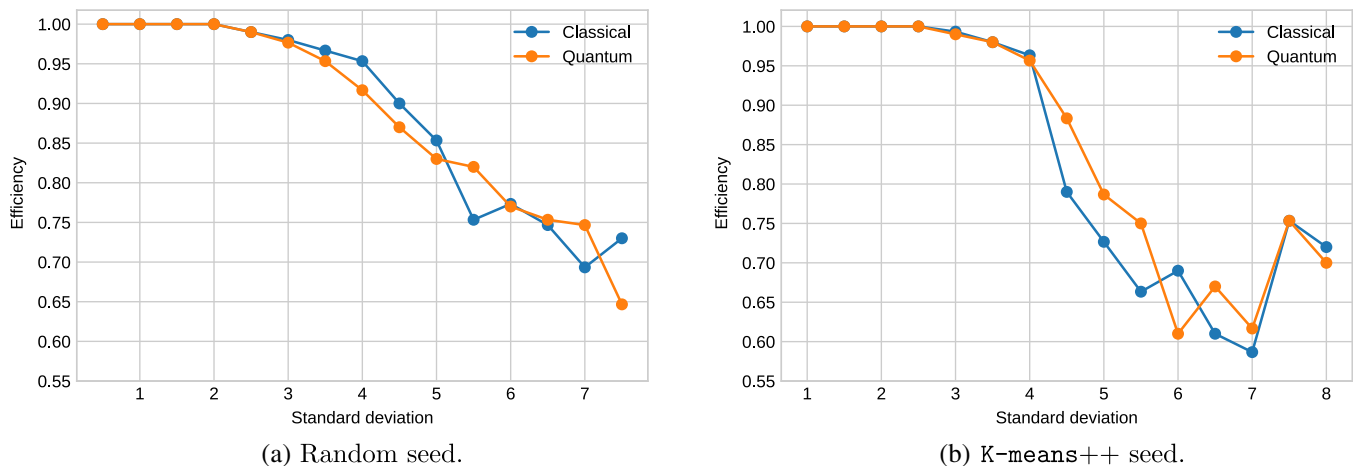


FIG. 4. Cluster efficiency of the K-means algorithm versus standard deviations of the data with respect to centroids. Both the classical and quantum versions have been run on 15 datasets with standard deviations ranging from 0.5 to 7.5. (a) Random seed. (b) K-means++ seed.

respect to the case when the seed centroids are carefully selected to be as far as possible from each other, according to the K-means++ prescription [see Fig. 4(b)]. The random seed variant in Fig. 4(a), has a linear decrease with respect to the standard deviation, and the performances of classical and quantum versions are very similar. On the other hand, the K-means++ variant, Fig. 4(b), presents a different behavior. The quantum version outperforms, in the majority of the cases, the classical one from a standard deviation of 4 onward. Furthermore, in this variant both performances show a dropoff from 4 standard deviations to 7, and then a slight rise from 7 to 8. Finally, comparing both variants it is observed that the K-means++ method outperforms the random seed case for small values of the standard deviation (< 4). However, for larger values of the standard deviation the random seed prescription presents higher efficiencies.

In the following, we will apply our quantum K-means method to LHC physical events. To do so we first have processed the data to avoid the following problem: a negative vector $-\mathbf{x}$ represents the same quantum state $|x\rangle$ as its positive analog \mathbf{x} up to a global phase. This data processing consists of rescaling the data to be analyzed in the interval $\{1, 10\}$.⁶ This means every component of every data point will be rescaled in the desired interval. Thus, all the data points are positive now. Moreover, when analyzing LHC physical events, we no longer have the *true labels*, so we cannot calculate ε_t . Instead, we define the efficiency ε_c , which is defined as the quotient

⁶Note that the value 0 is not included to avoid numerical and statistical fluctuations

of the number of particles clustered in the same way as their classical counterpart and the total number of particles to be classified.

We consider the generation of a physical n -particle event produced at the LHC. We use a private implementation of an n -particle (n can be of the order of tens of thousands) phase-space event generator. This C++ code, which is based on ROOT [90], generates n -particle events, in which the final-state particles can be massive or massless in any combination of each other (combination chosen by the user). This allows the user to generate final states in which all the particles are massless QCD partons, massless QCD partons associated with photons, massive vector bosons, top-quarks, etc.

The precision in the generation of the final-state event is verified on an event-by-event basis by computing the kinematical constraint between the initial and the n -particle final state. The required precision⁷ is always better than 10^{-2} . Each generated event is then analyzed with the classical versions of the k_T -jet algorithms (as implemented in FastJet [64]) and with our quantum version of the corresponding jet algorithms.

In this paper we consider the n -particle massless final-state production in proton-proton⁸ collisions at a

⁷If we consider all momenta of the event outgoing, the kinematical constraint is evaluated over the resulting three-momentum space vector. The test in the accuracy of the kinematical constraint is performed at the highest multiplicity in the final state. This constitutes the lowest limit for the precision, since reducing the particle number in the final state, the precision improves.

⁸Since we are considering unweighted events, our study is not only valid for proton-proton colliders, but also for e^+e^- colliders.

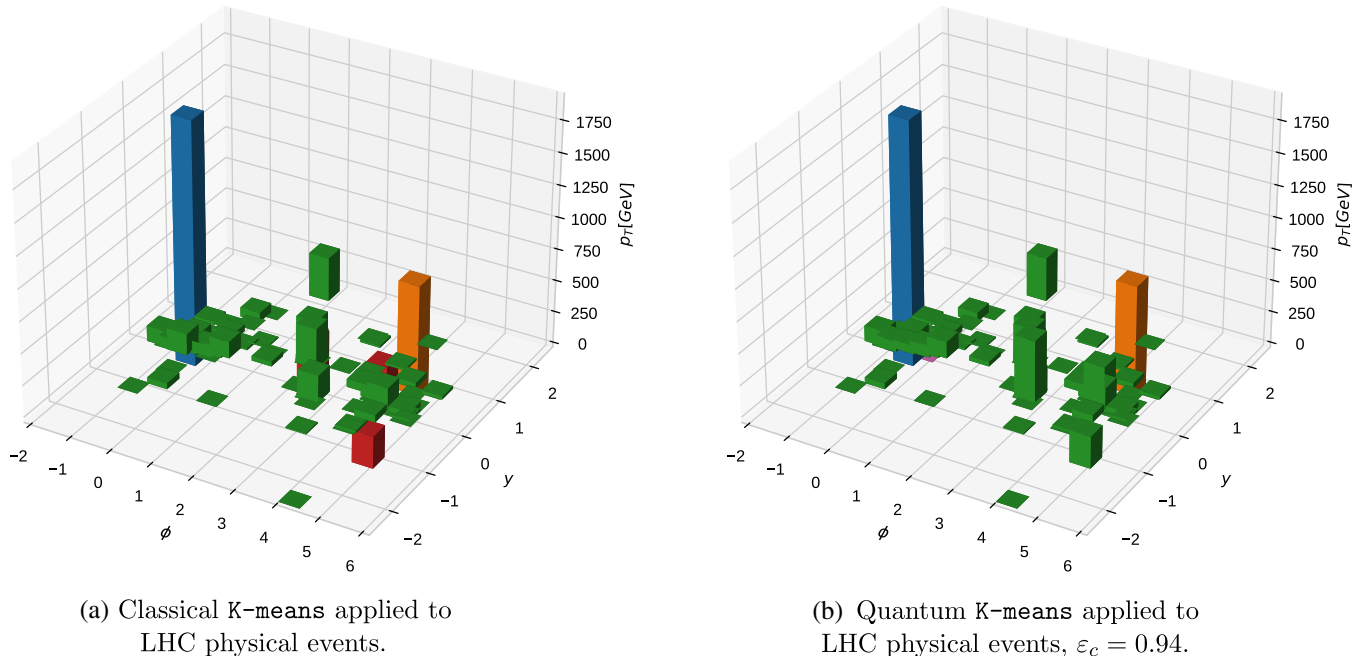


FIG. 5. A sample parton-level event generated as described in the text and clustered with the classical and the quantum version of the K -means++ algorithm, taking $K = 8$. (a) Classical K -means applied to LHC physical events. (b) Quantum K -means applied to LHC physical events, $\varepsilon_c = 0.94$.

centre-of-mass energy of $\sqrt{s} = 14$ TeV. We apply the following final-state selection cuts. We select jets with the k_T -jet algorithms according to the following parameters: the minimum transverse momentum of the resulting jets is required to be $p_{T\min} \geq 10$ GeV and with a radius $R = 1$. For our study, we consider n massless particles in the final state with $n = 128$.

The application of the quantum K -means++ method to LHC physical events is displayed in Fig. 5. Notice that even if we choose $K = 8$ beforehand, one may see in Fig. 5 that the algorithms clearly distinguish only 3 or 4 clusters (jets). There is actually a simple explanation. Although the algorithm starts with K centroids, the algorithm may converge to a local minimum when the number of clusters is less than K , leaving the remaining clusters completely empty.

TABLE I. Efficiencies of AP algorithms for Gaussian datasets with different number of clusters.

Number of clusters K	Efficiency classical AP (ε_l)	Efficiency quantum AP (ε_l)
4	1.00	0.99
5	1.00	1.00
6	0.99	0.98
7	1.00	0.98
8	0.98	0.94

In Fig. 5 one can observe graphically that both algorithms classify the data in much the same way, and also the efficiency shown by the quantum algorithm is close to one. Therefore, the results of this quantum version using physical data may be considered satisfactory.

B. Quantum affinity propagation algorithm

In this subsection, a simulation of the quantum AP algorithm is presented. First, we apply this algorithm to Gaussian datasets with different numbers of clusters, generated with a standard deviation of 0.6. That value of the standard deviation has been chosen arbitrarily by convenience. The efficiencies resulted for the classical and the quantum versions are shown in Table I. Table I depicts that the AP classical algorithm and its quantum counterpart clustered the low-noise Gaussian datasets successfully.

In the following, we apply this algorithm to the physical dataset described in Section VA, which was preprocessed for the reasons explained in the same section. The results obtained are shown in Fig. 6. In Fig. 6(b) exactly the same clustering is performed as in Fig. 6(a) (notice that the efficiency of the quantum version is $\varepsilon_c = 1.00$). Nonetheless, this algorithm only finds 2 clusters, which differs with respect to the 3 or 4 clusters found by the K -means algorithm (see Fig. 5). Even more, both algorithms identify correctly the most energetic jets of the event (the blue and the orange ones)

while the majority of the remaining particles are not classified in the same way, probably because they are soft particles.

C. Quantum k_T jet algorithm

In this section, we apply the quantum version of the k_T -jet algorithm to the same LHC physical events as described in Sec. VA in order to compare the three clustering algorithms.

In Fig. 7 we show the performance of classical and quantum k_T jet algorithms. It depicts the jet clustering process carried out by each one of the k_T algorithm versions, i.e., anti- k_T , k_T and Cambridge/Aachen. The classical and quantum versions perform the same jet clustering. When comparing Figs. 5, 6, and 7, one can observe that the latter performs a cleaner clusterization with a larger number of jets. This is a visual effect because jet clusterization is represented graphically in 3-dimensions, which coincides with the dimensionality of the k_T metrics, while the K -means and AP use a 4-dimensional Minkowski distance.

To conclude this section we also analyze the efficiencies and the number of *shots* required for all the quantum versions as a function of the a parameter (see Sec. IV C). These are shown in Table II. Table II displays that the efficiencies of the quantum algorithms are close to one,

i.e., they classify particles almost identically to their classical counterparts. Furthermore, it may be observed that the larger the parameter a , the smaller the number of *shots* required to achieve a successful efficiency. In this case, we only need to increase the parameter a to the number 5 to achieve the desired efficiencies with at most 10 *shots*. However, in other problems (with a larger dataset) a parameter greater than $a = 5$ can be used to separate the data points and achieve the highest possible efficiency with the smallest number of *shots*.

VI. CONCLUSIONS

In this paper, we have considered the quantum versions of the well-known K -means, affinity propagation, and k_T -jet clustering algorithms. These quantum versions are based on two novel quantum procedures. The first one is a quantum subroutine which serves to compute distances satisfying Minkowski metric, whereas the second one consists of a quantum circuit to track the maximum into a list of unsorted data.

In the case of the K -means clustering algorithm, the quantum version is based on the standard classical algorithm with a quantum procedure to compute distances in Minkowski space and an additional quantum procedure to assign each particle to the nearest centroid. We found that the K -means quantum algorithm has a

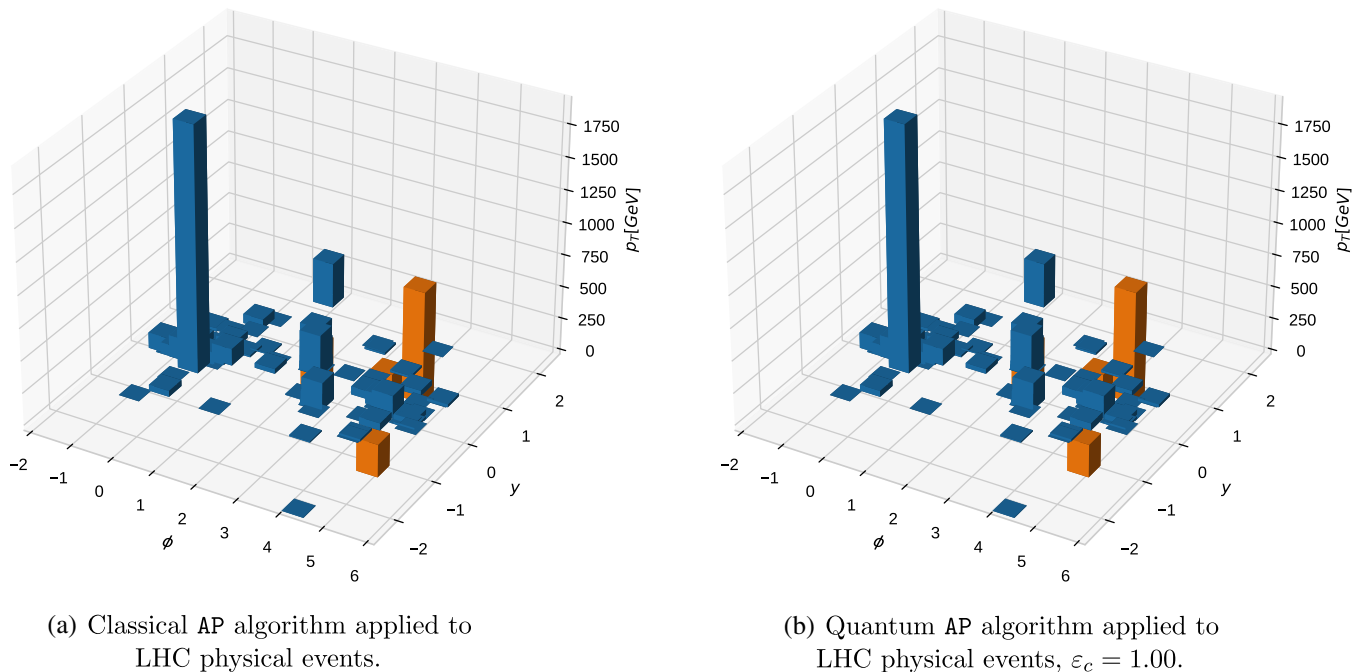


FIG. 6. A sample parton-level event generated as described in the text and clustered in $K = 2$ different clusters with the classical and the quantum version of the AP algorithm. (a) Classical AP algorithm applied to LHC physical events. (b) Quantum AP algorithm applied to LHC physical events, $\epsilon_c = 1.00$.

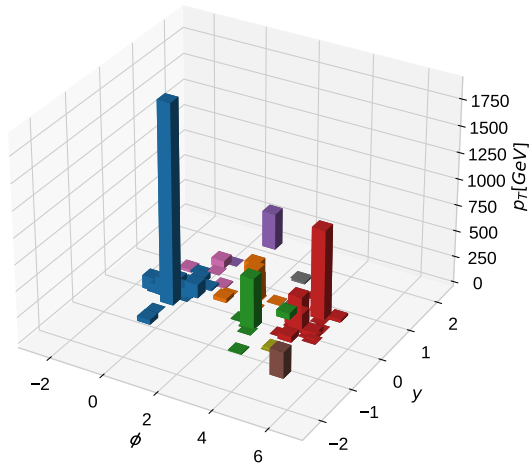
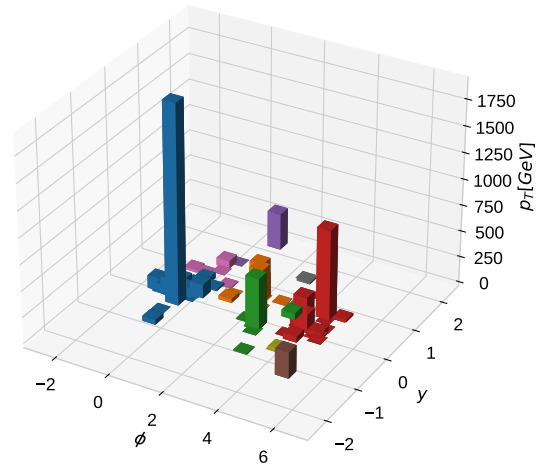
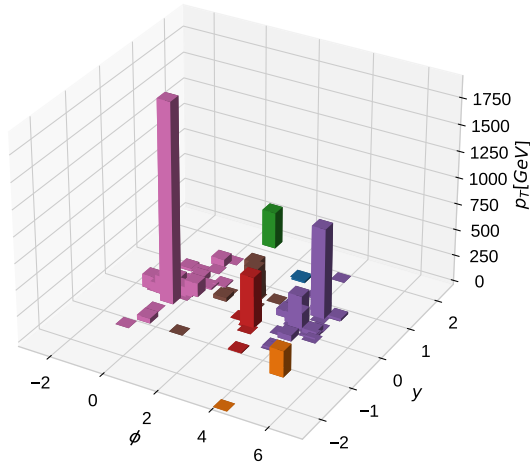
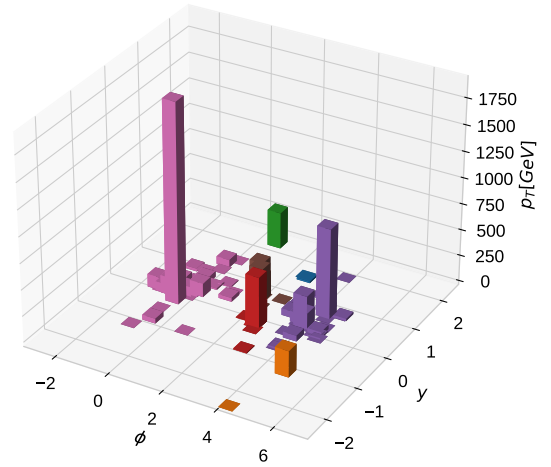
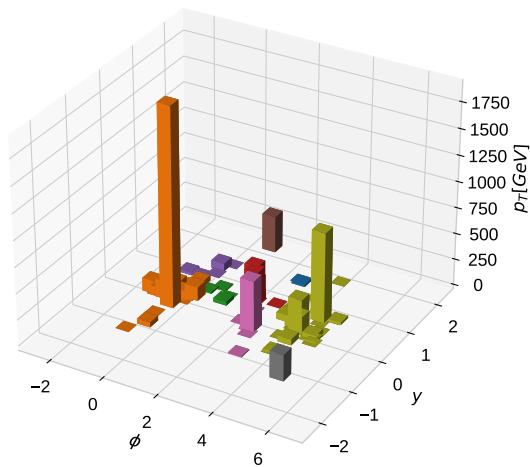
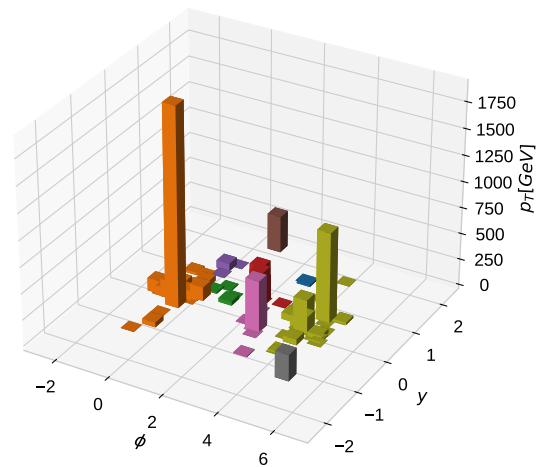
(a) Classical anti- k_T , $p = -1$, $R = 1$.(b) Quantum anti- k_T , $p = -1$, $R = 1$, $\epsilon_c = 0.99$.(c) Classical k_T , $p = 1$, $R = 1$.(d) Quantum k_T , $p = 1$, $R = 1$, $\epsilon_c = 0.98$.(e) Classical Cam/Aachen, $p = 0$, $R = 1$.(f) Quantum Cam/Aachen, $p = 0$, $R = 1$, $\epsilon_c = 0.98$.

FIG. 7. A sample parton-level event generated as described in the text and clustered with three different k_T -jets algorithms as well as three quantum versions. (a) Classical anti- k_T , $p = -1$, $R = 1$. (b) Quantum anti- k_T , $p = -1$, $R = 1$, $\epsilon_c = 0.99$. (c) Classical k_T , $p = 1$, $R = 1$. (d) Quantum k_T , $p = 1$, $R = 1$, $\epsilon_c = 0.98$. (e) Classical Cam/Aachen, $p = 0$, $R = 1$. (f) Quantum Cam/Aachen, $p = 0$, $R = 1$, $\epsilon_c = 0.98$.

TABLE II. Efficiencies and number of *shots* of the different quantum k_T -jet algorithms as a function of parameter a .

a	Efficiency anti- k_T	<i>Shots</i> anti- k_T	Efficiency k_T	<i>Shots</i> k_T	Efficiency Cam/Aachen	<i>Shots</i> Cam/Aachen
1	0.96	50	0.98	50	0.96	70
2	0.99	40	0.99	45	0.98	60
3	1.00	25	0.98	20	0.97	40
4	1.00	15	0.95	15	1.00	20
5	0.99	5	1.00	8	0.98	10

clustering efficiency as good as its classical counterpart while it would show an exponential speed-up in computational time in the vector dimensionality d , as well as in the number of clusters K on a quantum device with qRAM.

In the second place, we have considered a quantum version of the affinity propagation method, which is an unsupervised machine learning algorithm, where the similarity is computed with the same quantum procedure as in the K-means case. Thus, it would lead to an exponential speed-up regarding its classical counterpart in the vector dimensionality d while maintaining the clustering efficiency.

Finally, we have presented the quantum versions of the well-known k_T -jet clustering algorithms. On a true universal quantum device, the implementation of these algorithms would exhibit an exponential speed-up in finding the minimum distance. Therefore, while the classical version requires $\mathcal{O}(N^3)$ in computational cost, where N is the number of particles to cluster, the quantum counterpart would only require $\mathcal{O}(N^2 \log(N))$. Notice that this comparison is performed between the classical *nonoptimal* and not optimized version and its quantum analog. Further improvements can be obtained by applying to the quantum algorithm the geometrical nearest neighbor optimization procedure that is also applied to `FastJet`. In this way, we would obtain a quantum version of order $\mathcal{O}(N \log(N))$, which is of the same order as the fully optimal version of `FastJet`.

For all the clustering algorithms considered, the quantum simulations presented in this paper show an excellent performance and clustering efficiencies. Furthermore, the comparison with their classical counterparts displays that both classifications of the LHC simulated data are quite in agreement.

ACKNOWLEDGMENTS

We thank Abhijat Sharma and Guillermo Alonso for very helpful conversations and suggestions. This work is supported by the Spanish Government (Agencia Estatal de Investigación MCIN/AEI/10.13039/501100011033) Grant No. PID2020-114473GB-I00, and Generalitat Valenciana Grant No. PROMETEO/2021/071. L. C. is

supported by Generalitat Valenciana GenT Excellence Programme (CIDEAGENT/2020/011). G. R. is currently on leave at the European Research Council Executive Agency, European Commission, BE-1049 Brussels, Belgium. The views expressed are purely those of the writer and may not in any circumstances be regarded as stating an official position of the European Commission.

APPENDIX: CONTROLLED *SwapTest*

A well-known procedure for determining the entanglement between two quantum states is the controlled *SwapTest* [68] method. This method allows us to quantify the overlap between $|\psi_1\rangle$ and $|\psi_2\rangle$, which are two input general quantum states of n and m qubits respectively such that $n \geq m$ (otherwise we exchange the labels 1 and 2), by measuring an ancillary qubit. The controlled *SwapTest* proceeds in three steps starting from the initial state

$$|\Psi_0\rangle = |0, \psi_1, \psi_2\rangle, \quad (\text{A1})$$

where the ancillary qubit has been initialized to $|0\rangle$. In the first step, a Hadamard (H) gate is applied to the ancillary qubit, while the states to be probed are left unchanged, resulting in the new state

$$|\Psi_1\rangle = (H \otimes I^{\otimes n+m})|\Psi_0\rangle = \frac{1}{\sqrt{2}}(|0, \psi_1, \psi_2\rangle + |1, \psi_1, \psi_2\rangle), \quad (\text{A2})$$

where the identity $I^{\otimes n+m}$ acts over the $|\psi_1\rangle$ and $|\psi_2\rangle$ states and the tensor product \otimes is omitted in the composed states (e.g., $|0\rangle \otimes |\psi_1\rangle \otimes |\psi_2\rangle = |0, \psi_1, \psi_2\rangle$). A controlled swap gate (CSWAP) is then applied to $|\Psi_1\rangle$ where all the m qubits of $|\psi_2\rangle$ are swapped with the m first qubits of $|\psi_1\rangle$, leading to

$$|\Psi_2\rangle = \text{CSWAP}|\Psi_1\rangle = \frac{1}{\sqrt{2}}(|0, \psi_1, \psi_2\rangle + |1, \psi_2, \psi'_1\rangle), \quad (\text{A3})$$

where ψ'_i is the swapped state of ψ_i , i.e., a state where the m first qubits of ψ_1 have been swapped with the rest $n - m$ qubits. The final step consist of applying again a Hadamard gate to the ancillary qubit

$$\begin{aligned} |\Psi_3\rangle &= (H \otimes I^{\otimes n+m})|\Psi_2\rangle \\ &= \frac{1}{2}(|0\rangle \otimes (|\psi_1, \psi_2\rangle + |\psi_2, \psi'_1\rangle) + |1\rangle \\ &\quad \otimes (|\psi_1, \psi_2\rangle - |\psi_2, \psi'_1\rangle)). \end{aligned} \quad (\text{A4})$$

The resulting probability of measuring the ancillary qubit in the state $|0\rangle$ is given by

$$\begin{aligned}
P_{\Psi_3}(|0\rangle) &= |\langle 0|\Psi_3\rangle|^2 = \frac{1}{4} ||\psi_1, \psi_2\rangle + |\psi_2, \psi_1'\rangle|^2 \\
&= \frac{1}{2} + \frac{1}{2} \text{Re}[\langle \psi_2, \psi_1' | \psi_1, \psi_2 \rangle] \\
&= \frac{1}{2} + \frac{1}{2} \langle \psi_1' | \psi_2 \rangle \langle \psi_2 | \psi_1 \rangle,
\end{aligned} \tag{A5}$$

which turns out to be as follows if $m = n$, thus $|\psi_1'\rangle = |\psi_1\rangle$

$$P_{\Psi_3}(|0\rangle) = \frac{1}{2} + \frac{1}{2} |\langle \psi_1 | \psi_2 \rangle|^2, \tag{A6}$$

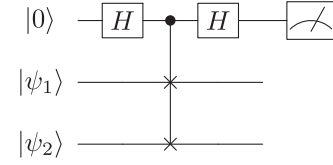


FIG. 8. Quantum circuit *SwapTest*.

and this provides the squared inner product between the two states with an uncertainty of $\mathcal{O}(\epsilon)$ after $\mathcal{O}(\epsilon^{-2})$ shots. The corresponding quantum circuit associated to the *SwapTest* method is shown in Fig. 8.

-
- [1] R. P. Feynman, Simulating physics with computers, *Int. J. Theor. Phys.* **21**, 467 (1982).
- [2] L. K. Grover, Quantum Mechanics Helps in Searching for a Needle in a Haystack, *Phys. Rev. Lett.* **79**, 325 (1997).
- [3] P. W. Shor, Polynomial time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM J. Sci. Stat. Comput.* **26**, 1484 (1997).
- [4] R. Orús, S. Mugel, and E. Lizaso, Quantum computing for finance: Overview and prospects, *Rev. Phys.* **4**, 100028 (2019).
- [5] J. Liu and Y. Xin, Quantum simulation of quantum field theories as quantum chemistry, *J. High Energy Phys.* **12** (2020) 011.
- [6] E. T. Holland, K. A. Wendt, K. Kravvaris, X. Wu, W. Erich Ormand, J. L. DuBois, S. Quaglioni, and F. Pederiva, Optimal control for the quantum simulation of nuclear dynamics, *Phys. Rev. A* **101**, 062307 (2020).
- [7] J. E. Lynn, I. Tews, S. Gandolfi, and A. Lovato, Quantum Monte Carlo methods in nuclear physics: Recent advances, *Annu. Rev. Nucl. Part. Sci.* **69**, 279 (2019).
- [8] A. Montanaro, Quantum speedup of Monte Carlo methods, *Proc. R. Soc. A* **471**, 20150301 (2015).
- [9] C. Kokail *et al.*, Self-verifying variational quantum simulation of lattice models, *Nature (London)* **569**, 355 (2019).
- [10] R. LaRose, A. Tikku, É. O’Neel-Judy, L. Cincio, and P. J. Coles, Variational quantum state diagonalization, *npj Quantum Inf.* **5**, 57 (2019).
- [11] C. Bravo-Prieto, D. García-Martín, and J. I. Latorre, Quantum singular value decomposer, *Phys. Rev. A* **101**, 062310 (2020).
- [12] A. Delgado *et al.*, Quantum computing for data analysis in high-energy physics, in 2022 Snowmass Summer Study (2022), [arXiv:2203.08805](https://arxiv.org/abs/2203.08805).
- [13] R. K. Ellis *et al.*, Physics briefing book: Input for the european strategy for particle physics update 2020, [arXiv:1910.11775](https://arxiv.org/abs/1910.11775).
- [14] F. Gianotti *et al.*, Physics potential and experimental challenges of the LHC luminosity upgrade, *Eur. Phys. J. C* **39**, 293 (2005).
- [15] FCC Collaboration, FCC physics opportunities: Future circular collider conceptual design report volume 1, *Eur. Phys. J. C* **79**, 474 (2019).
- [16] P. Roloff, R. Franceschini, U. Schnoor and A. Wulzer (CLIC and CLICdp Collaborations), The compact linear e^+e^- collider (clic): Physics potential, [arXiv:1812.07986](https://arxiv.org/abs/1812.07986).
- [17] CEPC Study Group Collaboration, CEPC conceptual design report: Volume 2—physics & detector, [arXiv:1811.10545](https://arxiv.org/abs/1811.10545).
- [18] A. Y. Wei, P. Naik, A. W. Harrow, and J. Thaler, Quantum algorithms for jet clustering, *Phys. Rev. D* **101**, 094015 (2020).
- [19] D. Pires, P. Bargassa, J. a. Seixas, and Y. Omar, A digital quantum algorithm for jet clustering in high-energy physics, [arXiv:2101.05618](https://arxiv.org/abs/2101.05618).
- [20] D. Pires, Y. Omar, and J. a. Seixas, Adiabatic quantum algorithm for multijet clustering in high energy physics, [arXiv:2012.14514](https://arxiv.org/abs/2012.14514).
- [21] J. a. Barata and C. A. Salgado, A quantum strategy to compute the jet quenching parameter \hat{q} , *Eur. Phys. J. C* **81**, 862 (2021).
- [22] A. Pérez-Salinas, J. Cruz-Martinez, A. A. Alhajri, and S. Carrazza, Determining the proton content with a quantum computer, *Phys. Rev. D* **103**, 034027 (2021).
- [23] C. W. Bauer, W. A. de Jong, B. Nachman, and D. Provasoli, Quantum Algorithm for High Energy Physics Simulations, *Phys. Rev. Lett.* **126**, 062001 (2021).
- [24] S. Williams, S. Malik, M. Spannowsky, and K. Bepari, A quantum walk approach to simulating parton showers, [arXiv:2109.13975](https://arxiv.org/abs/2109.13975).
- [25] K. Bepari, S. Malik, M. Spannowsky, and S. Williams, Towards a quantum computing algorithm for helicity amplitudes and parton showers, *Phys. Rev. D* **103**, 076020 (2021).
- [26] W. A. De Jong, M. Metcalf, J. Mulligan, M. Płoskoń, F. Ringer, and X. Yao, Quantum simulation of open quantum systems in heavy-ion collisions, *Phys. Rev. D* **104**, L051501 (2021).
- [27] W. Guan, G. Perdue, A. Pesah, M. Schuld, K. Terashi, S. Vallecorsa *et al.*, Quantum machine learning in high energy physics, *Mach. Learn.: Sci. Technol.* **2**, 011003 (2011).

- [28] S. L. Wu *et al.*, Application of quantum machine learning using the quantum variational classifier method to high energy physics analysis at the LHC on IBM quantum computer simulator and hardware with 10 qubits, *J. Phys. G* **48**, 125003 (2021).
- [29] T. Felser, M. Trenti, L. Sestini, A. Gianelle, D. Zuliani, D. Lucchesi, and S. Montangero, Quantum-inspired machine learning on high-energy physics data, *npj Quantum Inf.* **7**, 111 (2021).
- [30] S. Abel, J. C. Criado, and M. Spannowsky, Completely quantum neural networks, [arXiv:2202.11727](https://arxiv.org/abs/2202.11727).
- [31] J. Y. Araz and M. Spannowsky, Classical versus quantum: Comparing tensor network-based quantum circuits on LHC data, [arXiv:2202.10471](https://arxiv.org/abs/2202.10471).
- [32] V. S. Ngairangbam, M. Spannowsky, and M. Takeuchi, Anomaly detection in high-energy physics using a quantum autoencoder, *Phys. Rev. D* **105**, 095004 (2022).
- [33] J. Y. Araz and M. Spannowsky, Quantum-inspired event reconstruction with tensor networks: Matrix product states, *J. High Energy Phys.* **08** (2021) 112.
- [34] A. Blance and M. Spannowsky, Quantum machine learning for particle physics using a variational quantum classifier, *J. High Energy Phys.* **02** (2021) 212.
- [35] S. P. Jordan, K. S. M. Lee, and J. Preskill, Quantum algorithms for quantum field theories, *Science* **336**, 1130 (2012).
- [36] M. C. Bañuls *et al.*, Simulating lattice gauge theories within quantum technologies, *Eur. Phys. J. D* **74**, 165 (2020).
- [37] E. Zohar, J. I. Cirac, and B. Reznik, Quantum simulations of lattice gauge theories using ultracold atoms in optical lattices, *Rep. Prog. Phys.* **79**, 014401 (2016).
- [38] T. Byrnes and Y. Yamamoto, Simulating lattice gauge theories on a quantum computer, *Phys. Rev. A* **73**, 022328 (2006).
- [39] S. Ramírez-Uribe, A. E. Rentería-Olivo, G. Rodrigo, G. F. R. Sborlini, and L. Vale Silva, Quantum algorithm for Feynman loop integrals, *J. High Energy Phys.* **05** (2022) 100.
- [40] S. Ramírez-Uribe, Four-loop scattering amplitudes through the loop-tree duality, *Rev. Mex. Fis. Suppl.* **3**, 020720 (2022).
- [41] L. R. Evans, *The Large Hadron Collider: A Marvel of Technology*, 2nd ed. (EPFL Press, Lausanne, Switzerland, 2018).
- [42] HEP Software Foundation Collaboration, A roadmap for HEP software and computing R&D for the 2020s, *Comput. Softw. Big Sci.* **3**, 7 (2019).
- [43] ATLAS Collaboration, ATLAS software and computing HL-LHC roadmap, Technical Reports No. CERN-LHCC-2022-005, No. LHCC-G-182, CERN, Geneva, 2022, <https://cds.cern.ch/record/2802918>.
- [44] J. MacQueen *et al.*, Some methods for classification and analysis of multivariate observations, in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* (University of California Press, Oakland, CA, 1967).
- [45] G. H. Ball and D. J. Hall, A clustering technique for summarizing multivariate data, *Behav. Sci.* **12**, 153 (1967).
- [46] B. J. Frey and D. Dueck, Clustering by passing messages between data points, *Science* **315**, 972 (2007).
- [47] S. Catani, Y. L. Dokshitzer, M. Olsson, G. Turnock, and B. R. Webber, New clustering algorithm for multi-jet cross-sections in e^+e^- annihilation, *Phys. Lett. B* **269**, 432 (1991).
- [48] S. Catani, Y. L. Dokshitzer, M. H. Seymour, and B. R. Webber, Longitudinally invariant K_T clustering algorithms for hadron hadron collisions, *Nucl. Phys.* **B406**, 187 (1993).
- [49] S. D. Ellis and D. E. Soper, Successive combination jet algorithm for hadron collisions, *Phys. Rev. D* **48**, 3160 (1993).
- [50] M. Cacciari, G. P. Salam, and G. Soyez, The anti- k_T jet clustering algorithm, *J. High Energy Phys.* **04** (2008) 063.
- [51] Y. L. Dokshitzer, G. D. Leder, S. Moretti, and B. R. Webber, Better jet clustering algorithms, *J. High Energy Phys.* **08** (1997) 001.
- [52] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay, Clustering large graphs via the singular value decomposition: Theoretical advances in data clustering (guest editors: Nina Mishra and Rajeev Motwani), *Mach. Learn.* **56**, 9 (2004).
- [53] S. Lloyd, Least squares quantization in PCM, *IEEE Trans. Inf. Theory* **28**, 129 (1982).
- [54] D. Arthur and S. Vassilvitskii, K-means++: The advantages of careful seeding, in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, Symposium on Discrete Algorithms, New Orleans, Louisiana, 2007.
- [55] S. Chekanov, A new jet algorithm based on the k-means clustering for the reconstruction of heavy states from jets, *Eur. Phys. J. C* **47**, 611 (2006).
- [56] J. Thaler and K. Van Tilburg, Maximizing boosted top identification by minimizing N-subjettiness, *J. High Energy Phys.* **02** (2012) 093.
- [57] I. W. Stewart, F. J. Tackmann, J. Thaler, C. K. Vermilion, and T. F. Wilkason, X Cone: N-jettiness as an exclusive cone jet algorithm, *J. High Energy Phys.* **11** (2015) 072.
- [58] C.-Y. Wong, H. Jiang, N. Yao, L. Wen, G. Wang, and H. Zhong Huang, Clustering properties of produced particles in high-energy pp collisions, *Phys. Rev. D* **102**, 054007 (2020).
- [59] M. Leone, Sumedha, and M. Weigt, Clustering by soft-constraint affinity propagation: Applications to gene-expression data, *Bioinformatics* **23**, 2708 (2007).
- [60] M. L. Sumedha and M. Weigt, Unsupervised and semi-supervised clustering by message passing: Soft-constraint affinity propagation, *Eur. Phys. J. B* **66**, 125 (2008).
- [61] M. Bailly-Bechet, S. Bradde, A. Braunstein, A. Flaxman, L. Foini, and R. Zecchina, Clustering with shallow trees, *J. Stat. Mech.* (2009) P12010.
- [62] O. González-Martín, J. Masegosa, A. Hernán-Caballero, I. Márquez, C. R. Almeida, A. Alonso-Herrero *et al.*, Hints on the gradual re-sizing of the torus in agn by decomposing irs/spitzer spectra, *Astrophys. J.* **841**, 37 (2017).
- [63] S. Carrazza and J. I. Latorre, Towards the compression of parton densities through machine learning algorithms, in *51st Rencontres de Moriond on QCD and High Energy Interactions* (2016), pp. 235–238, [arXiv:1605.04345](https://arxiv.org/abs/1605.04345).
- [64] M. Cacciari, G. P. Salam, and G. Soyez, FastJet user manual, *Eur. Phys. J. C* **72**, 1896 (2012).

- [65] A. Blance and M. Spannowsky, Unsupervised event classification with graphs on classical and photonic quantum computers, *J. High Energy Phys.* **21** (2020) 170.
- [66] A. Sarma, R. Chatterjee, K. Gili, and T. Yu, Quantum unsupervised and supervised learning on superconducting processors, *Quantum Inf. Comput.* **20**, 541 (2020).
- [67] S. Foulds, V. Kendon, and T. Spiller, The controlled SWAP test for determining quantum entanglement, *Quantum Sci. Technol.* **6**, 035002 (2021).
- [68] H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf, Quantum Fingerprinting, *Phys. Rev. Lett.* **87**, 167902 (2001).
- [69] JADE Collaboration, Experimental evidence for differences in p_T between quark jets and gluon jets, *Phys. Lett.* **123B**, 460 (1983).
- [70] S. Bethke, Z. Kunszt, D. E. Soper, and W. J. Stirling, New jet cluster algorithms: Next-to-leading order QCD and hadronization corrections, *Nucl. Phys.* **B370**, 310 (1992).
- [71] G. Rodrigo, M. S. Bilenky, and A. Santamaria, Quark mass effects for jet production in e^+e^- collisions at the next-to-leading order: Results and applications, *Nucl. Phys.* **B554**, 257 (1999).
- [72] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum algorithms for supervised and unsupervised machine learning, [arXiv:1307.0411](https://arxiv.org/abs/1307.0411).
- [73] V. Giovannetti, S. Lloyd, and L. Maccone, Quantum Random Access Memory, *Phys. Rev. Lett.* **100**, 160501 (2008).
- [74] V. Giovannetti, S. Lloyd, and L. Maccone, Architectures for a quantum random access memory, *Phys. Rev. A* **78**, 052310 (2008).
- [75] F. DeMartini, V. Giovannetti, S. Lloyd, L. Maccone, E. Nagali, L. Sansoni, and F. Sciarrino, Experimental quantum private queries with linear optics, *Phys. Rev. A* **80**, 010302 (2009).
- [76] I. Chiorescu, N. Groll, S. Bertaina, T. Mori, and S. Miyashita, Magnetic strong coupling in a spin-photon system and transition to classical regime, *Phys. Rev. B* **82**, 024413 (2010).
- [77] D. I. Schuster, A. P. Sears, E. Ginossar, L. DiCarlo, L. Frunzio, J. J. L. Morton *et al.*, High-Cooperativity Coupling of Electron-Spin Ensembles to Superconducting Cavities, *Phys. Rev. Lett.* **105**, 140501 (2010).
- [78] Y. Kubo, F. R. Ong, P. Bertet, D. Vion, V. Jacques, D. Zheng *et al.*, Strong Coupling of a Spin Ensemble to a Superconducting Resonator, *Phys. Rev. Lett.* **105**, 140502 (2010).
- [79] H. Wu, R. E. George, J. H. Wesenberg, K. Mølmer, D. I. Schuster, R. J. Schoelkopf, K. M. Itoh, A. Ardavan, J. J. L. Morton, and G. A. D. Briggs, Storage of Multiple Coherent Microwave Excitations in an Electron Spin Ensemble, *Phys. Rev. Lett.* **105**, 140503 (2010).
- [80] C. Dürr and P. Høyer, A quantum algorithm for finding the minimum, [arXiv:quant-ph/9607014](https://arxiv.org/abs/quant-ph/9607014).
- [81] P. J. Coles, S. J. Eidenbenz, S. Pakin, A. Adedoyin, J. Ambrosiano, P. M. Anisimov *et al.*, Quantum algorithm implementations for beginners, *ACM Trans. Quantum Comput.* **3**, 1 (2022).
- [82] R. L. Smith, Extreme value theory, *Handbook of Applicable Mathematics* (Springer, New York, US, 1990), Vol. 7.
- [83] E. J. Gumbel, *Statistics of Extremes* (Columbia University Press, New York, US, 1958).
- [84] T. M. Reiss R-D., *Statistical Analysis of Extreme Values With Applications to Insurance, Finance, Hidrology and Other Fields* (Birkhauser-Verlag, Basel, Switzerland, 2007).
- [85] S. Coles, *An Introduction to Statistical Modeling of Extreme Values* (Springer-Verlag, Berlin, 2001).
- [86] Enrique Castillo, A. S. Hadi, N. Balakrishnan, and J.-M. Sarabia, *Extreme Value and Related Models with Applications in Engineering and Science* (Wiley, New York, 2004).
- [87] D. Kocprzyk, Quantum machine learning for data scientists, [arXiv:1804.10068](https://arxiv.org/abs/1804.10068).
- [88] M. Cacciari and G. P. Salam, Dispelling the N^3 myth for the k_T jet-finder, *Phys. Lett. B* **641**, 57 (2006).
- [89] M. Fanizza, A. Mari, and V. Giovannetti, Optimal universal learning machines for quantum state discrimination, *IEEE Trans. Inf. Theory* **65**, 5931 (2019).
- [90] R. Brun and F. Rademakers, ROOT: An object oriented data analysis framework, *Nucl. Instrum. Methods Phys. Res., Sect. A* **389**, 81 (1997).