

Enhanced feature encoding and classification on distributed quantum hardware

R. Moretti^{1,2,3}, A. Giachero^{1,2,3}, V. Radescu⁴, M. Grossi⁵

¹ Department of Physics, University of Milano-Bicocca, Piazza della Scienza, 3, 20126, Milan, Italy

² INFN - Milano Bicocca, Piazza della Scienza, 3, 20126, Milan, Italy

³ Bicocca Quantum Technologies (BiQuTe) Centre, 3, 20126, Milan, Italy

⁴ IBM Quantum, IBM Deutschland Research & Development GmbH - Schöenaicher Str. 220, 71032 Böblingen, Germany

⁵ European Organization for Nuclear Research (CERN), Geneva, CH-1211, Switzerland

E-mail: roberto.moretti@mib.infn.it

Month year

Abstract. The steady progress of quantum hardware is motivating the search for novel quantum algorithm optimization strategies for near-term, real-world applications. In this study, we propose a novel feature map optimization strategy for Quantum Support Vector Machines (QSVMs), designed to enhance binary classification while taking into account backend-specific parameters, including qubit connectivity, native gate sets, and circuit depth, which are critical factors in noisy intermediate scale quantum (NISQ) devices. The dataset we utilised belongs to the neutrino physics domain, with applications in the search for neutrinoless double beta decay. A key contribution of this work is the parallelization of the classification task to commercially available superconducting quantum hardware to speed up the genetic search processes. The study was carried out by partitioning each quantum processing unit (QPU) into several sub-units with the same topology to implement individual QSVM instances. We conducted parallelization experiments with three IBM backends with more than 100 qubits, ranking the sub-units based on their susceptibility to noise. Data-driven simulations show how, under certain restrictions, parallelized genetic optimization can occur with the tested devices when retaining the top 20% ranked sub-units in the QPU.

Keywords: Quantum Computing, Quantum Feature Maps, Neutrino Physics, Genetic Algorithms, QSVM

1. Introduction

The ongoing advancements of quantum computing hardware, in the era of quantum utility [1, 2], are driving significant interest in leveraging intermediate-scale quantum devices available today across many research fields. Many research projects focused on

finding the potential of Variational Quantum Algorithms (VQAs) in a wide range of fields, including quantum chemistry [3], combinatorial optimization [4], and machine learning. In particular, Quantum Machine Learning (QML) held significant promise by integrating quantum computing into classical machine learning techniques with the hope of achieving better performance in tasks like classification [5], clustering [6], and generative modeling [7]. Yet, despite the high expectations surrounding QML and VQAs, solid theoretical guarantees on their performance and clear demonstrations of quantum advantage remain challenging.

Recent studies underline the need to reconsider the power of QML as a probe for quantum advantage [8–10] or overcome the limitations of early-stage quantum backends [8, 11–14], motivating the effort to provide QML algorithm design strategies tailored to the specific hardware in use.

The intrinsic ability of quantum systems to represent data within large Hilbert spaces can lead to highly expressive machine learning models [15–17] that are challenging to simulate using classical methods. For instance, quantum kernel methods, such as Quantum Support Vector Machines (QSVMs) [18], exploit the embedding of input data in a feature space whose dimensionality grows exponentially with the number of qubits in use. Although such dimensionality enhancement can expose generalizability issues [19–21], this is not necessarily true for any quantum kernel for a given dataset. Also, techniques such as projected quantum kernels [22, 23] can further help to balance expressivity and generalizability thanks to partial tracing. A remarkable advantage of quantum kernels is that, like their classical counterpart, they are positive and semi-definite functions of the input features, simplifying the training process of (Q)SVMs by making it a convex optimization problem [24].

The standard quantum kernel formulation is referred to as “fidelity quantum kernel” [22, 25], defined as

$$\mathcal{K} : \mathcal{K}(\vec{x}_i, \vec{x}_j) \rightarrow |\langle \phi(\vec{x}_i) | \phi(\vec{x}_j) \rangle|^2, \quad (1)$$

where \vec{x}_i and \vec{x}_j are feature vectors with i, j corresponding to sample indices in the dataset and ϕ is an arbitrary feature map that encodes input features in a (multi-)qubit state. We can rewrite Eq. 1 more practically in terms of a unitary ansatz that implements the transformation $U|0\rangle^{\otimes n} = |\phi\rangle$:

$$\mathcal{K} : \mathcal{K}(\vec{x}_i, \vec{x}_j) \rightarrow |\langle 0|^{\otimes n} U(\vec{x}_i)^\dagger U(\vec{x}_j) |0\rangle^{\otimes n}|^2. \quad (2)$$

The performance of the resulting QSVM depends significantly on the chosen ansatz and identifying an optimal quantum feature map for a given dataset remains an open challenge in the field. This task is often approached through manual exploration of various heuristic feature maps [26, 27], kernel alignment techniques [28], or incorporating trainable parameters to the unitary operator U [29–31]. These approaches are either time-consuming or expose to trainability hurdles such as the occurrence of barren plateaus [32, 33] which are typical of Parametrized Quantum Circuit (PQC).

An important related challenge is the phenomenon of exponential concentration [23, 34], where highly expressive kernels exhibit an exponential concentration of kernel

values due to the high dimensionality of the mapped feature vectors, distributed *uniformly* across the exponentially large Hilbert space. As a result, distinguishing between kernel values requires an exponentially large number of measurements. This issue becomes more severe when considering Noisy Intermediate-Scale Quantum (NISQ) devices [35, 36], where the precision of quantum kernel evaluations is significantly degraded by noise. While it is demonstrated that the projected quantum kernel techniques mentioned earlier help reduce exponential concentration, specific fidelity quantum kernel optimization strategies may also effectively address this problem.

Recent efforts have focused on automating the optimization of quantum feature maps at the gate level, with genetic algorithms emerging as promising tools for this purpose [37–44]. Genetic ansatz optimizations have also been carried out in the context of Quantum Approximate Optimization Algorithm (QAOA) [45]. However, current implementations have primarily demonstrated proof-of-concept results and have not fully addressed the reproducibility of quantum kernels on NISQ devices. Moreover, most of these optimization strategies [37–43] have been tested on standard, well-known datasets with minimal class overlap, achieving near-perfect accuracy with small training sets – an unrealistic scenario for many practical applications.

Our study aims to fill these gaps by designing a genetic algorithm that optimizes quantum feature maps while taking into consideration specific hardware characteristics, with their native gates and qubit connectivity map in an extended setting using multiple chips, targeting quantum utility implementation. We consider a dataset taken from the neutrino physics domain consisting of a Monte Carlo simulation of three-dimensional tracks produced by ionization of liquid argon in a Liquid Argon Time Projection Chamber (LArTPC) detector [46, 47]. The physics motivation is related to the possibility of assessing the sensitivity of next-generation experiments like DUNE to the discovery of one of the most elusive processes in low-energy physics, i.e. the neutrinoless double beta decay of ^{136}Xe [48, 49]. Crucially, we expect this dataset to have considerable class overlap, hence being a hard binary classification task. The structure of the paper follows the pipeline of the algorithm: after applying an autoencoder-assisted feature preprocessing, which is detailed in Sec. 2, we introduce a genetic optimization strategy that enhances the kernel classification accuracy on noisy hardware in Sec. 3. In Sec. 4 we describe the results obtained by the genetic optimizations with a statevector simulation. The simulation under ideal conditions represents a necessary step to effectively run the search for optimal quantum kernels on the quantum computer.

Finally, to make the most of quantum computational resources for an utility scale experiment, we assess the feasibility of parallelizing IBM quantum chips for computing multiple quantum kernels simultaneously, by partitioning three quantum processing units (QPU) into several, independent sub-units. We evaluate the impact of noise and the robustness of this parallelization approach in Sec. 5, where we show the potential for accelerating genetic optimization by computing entire generations simultaneously, with a single quantum circuit containing all quantum kernels within a generation. Fig. 1 depicts the workflow proposed in this study. This technique could demonstrate a practical,

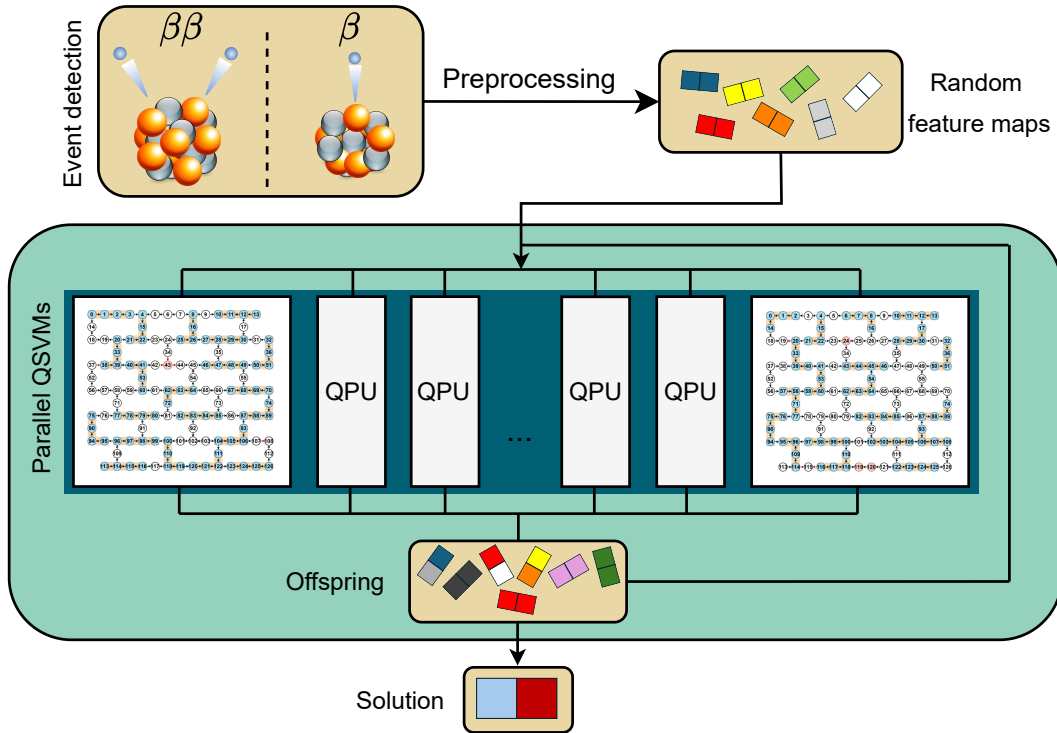


Figure 1: Schematic workflow proposal for the automatic quantum feature map optimization via genetic algorithm for classifying β and $\beta\beta$ topologies. Parallelization is achieved through QPU partitioning and leveraging each partition to compute several quantum kernels simultaneously. Additional speedup can be provided by splitting the task through multiple QPUs.

real-world utility-scale application for currently available QPUs, driving progress in data analysis across fields of study. We summarize our findings and potential future directions in Sec. 6.

2. Physical motivation and autoencoder processing

One of the most important physics processes under investigation within the neutrino physics community is the neutrinoless double beta decay ($0\nu\beta\beta$) [50, 51]. This process can be conceptually understood as a nucleus undergoing double beta decay, where the two (anti-)neutrinos normally emitted during beta decays annihilate each other, leaving only two electrons in the final state and the modified nucleus. Demonstrating the existence of such a process would have considerable implications for our understanding of Fundamental Physics, the matter-antimatter asymmetry in the Universe [52] and determine whether neutrinos and antineutrinos correspond to the same particle [53]. A number of studies suggest that ton-scale Liquid Argon Time Projection Chambers under certain circumstances can exhibit high discovery potential for the $0\nu\beta\beta$ decay of the ^{136}Xe [48, 54, 55], comparably to other experiments that have been specifically designed for this task, and currently pursued [56–60]. Such LArTPCs are currently

being developed within the Deep Underground Neutrino Experiment (DUNE), and the search for the $0\nu\beta\beta$ decay (together with the study of other few-MeV scale events) will be potentially taken into consideration in the realization of DUNE Phase II Far Detector LArTPCs, called Module of Opportunity (MoO) [49].

The ^{136}Xe $0\nu\beta\beta$ search with a LArTPC leverages the fact that xenon-doping of liquid argon will be pursued, because it will boost the scintillation light detection efficiency emitted by argon [61–63], improving the overall detection performance. Assuming xenon-doping occurs with xenon enriched in the ^{136}Xe isotope, we expect the decay to happen inside the active volume, with the signature consisting of two electrons ranging from the same point in space, producing a single ionization track. TPCs allow us to measure the total energy produced in the decay, and also reconstruct the full three-dimensional track information as a register of “hits”. Based on that, to boost the experimental sensitivity, we can develop a classification strategy leveraging the different topologies of signal events ($\beta\beta$ decays) and one of the most problematic background channels for this analysis, which is the β decay of ^{42}Ar [46, 48].

Considering a realistic granularity for a ton-scale LArTPC and the average track length produced by this decay in liquid argon (~ 1.2 cm) [64], most of the track information will be lost, with a detrimental effect on the classification performance. This classification task is therefore expected to be hard, making it an intriguing test bench for QSVMs, not only because it relates to a real-world application, but also because we anticipate highly overlapping (and possibly nontrivially correlated) feature distributions.

The dataset consists of an ensemble of labeled $0\nu\beta\beta$ events ($\beta\beta$ topology) and single β decay (β topology), assuming a resolution of $[2 \times 2 \times 1]$ mm³ and a hit energy threshold of 50 keV. Deep learning approaches consisting of a Convolutional Neural Network and a Transformer, when trained with more than 10^5 samples have reached around 76% accuracy for the same dataset and the same experimental settings considered in this study [46].

2.1. Autoencoder feature extraction

Each event in the dataset consists of an ensemble of hits produced inside the detector, with each hit carrying four scalar quantities (three spatial coordinates and hit energy). The number of hits varies for each different track, making the pre-processing not trivial. The adopted solution is the autoencoder, whose architecture can handle this flexibility while performing the data reduction as a label-agnostic step. The autoencoder consists of a multilayer perceptron [65] that learns an identity function of the input through one or more intermediate layers whose dimension is smaller than the input size. The number of reduced features in the “latent space” is controlled by the size of the smallest layer, from which these features are extracted. In this study, we determined the optimal compression of the original representation, reducing its size from 76 to 18, based on the autoencoder’s loss function and latent space dimensions. This compression significantly simplifies statevector QSVM simulations while minimizing information loss. The autoencoder

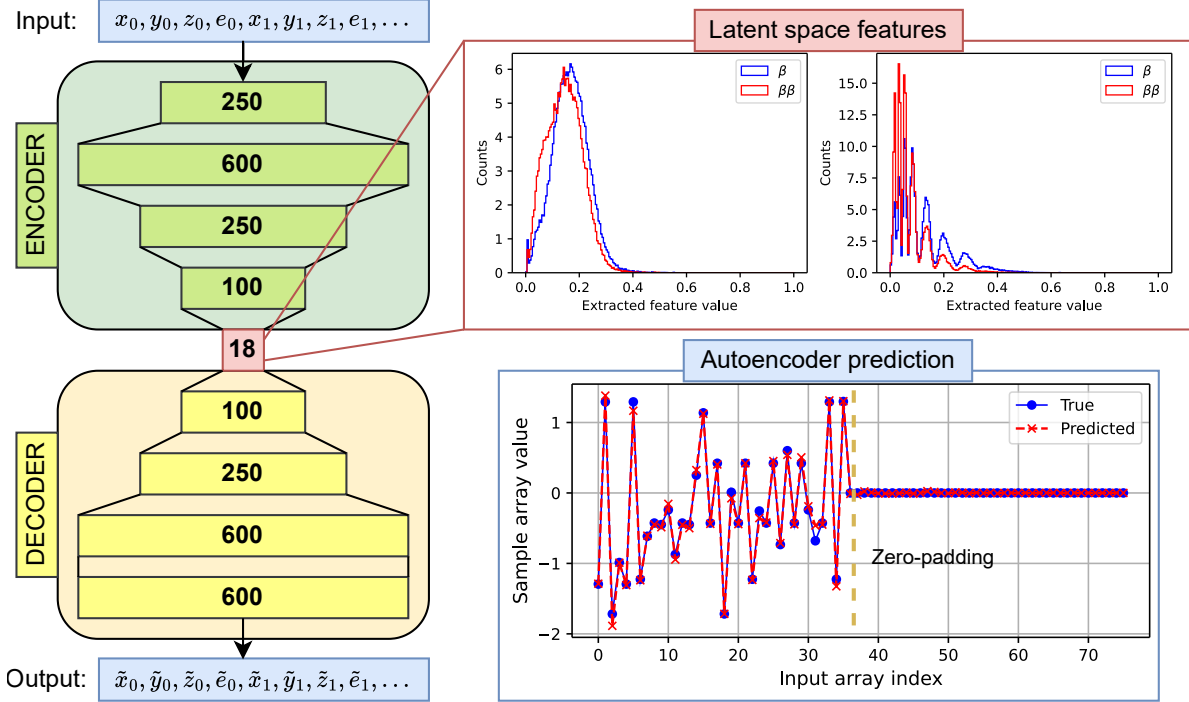


Figure 2: Autoencoder architecture used to extract a latent space representation of the input track information. The input array, as a set of concatenated hit information, is fed through several fully connected layers whose neuron number is indicated in the corresponding block in the diagram. In the top-right frame, two of the 18 latent space feature distributions are shown within classes. A comparison between the input and the autoencoder output for a single track is reported in the bottom-right frame.

architecture, feature distributions, and a sample prediction are illustrated in Fig. 2.

We designed and trained the autoencoder with the TensorFlow package [66] by partitioning the dataset into 120×10^3 training and 30×10^3 validation samples, minimizing the Mean Square Error cost function between the input data and the network prediction using the Adam stochastic gradient descent algorithm [67]. Input data is passed to the autoencoder as a single array for each sample, by concatenating hit arrays $(\vec{h}_0, \vec{h}_1, \dots, \vec{h}_n)$ where n is the hit number for a particular event, and $\vec{h}_i = (x_i, y_i, z_i, e_i)$, i.e. three spatial and one energy coordinates. The input is zero-padded so that every track is associated with an array of length 76. We observe how two example latent space feature distributions shown in Fig. 2 are irregular, either skewed or multi-peaked, with significant class overlap as we expected from the dataset. The features were rescaled in the interval $[0, 1]$ and used as input to the quantum kernels described in Sec. 4 and Sec. 5.

3. Genetic algorithm implementation

The genetic optimization implemented in this study aims to lead to the generation of quantum kernels that, at the same time, allow high classification accuracy with resource-efficient circuit embeddings [68]. To this extent, we employ a gradient-free method that optimises complex (generally non-differentiable) structures such as gate types, rotation angles, and qubit connectivities. Genetic algorithms are powerful tools that serve our purpose. The Genetic algorithm is a meta-heuristic method for solving optimization problems inspired by biological evolution processes [69]. In particular, given a fitness function that quantifies the goodness of a solution through one or more objectives, the algorithm aims to generate solutions that maximize the fitness function output (if mono-objective) or optimal solutions in terms of multi-objectives trade-off [44]. The solutions are generally referred to as “chromosomes”, and in our study, each chromosome corresponds to a quantum feature map for the kernel. Each chromosome consists of an ensemble of “genes” that encode the chromosome properties. The genetic content of chromosomes can be modified through operations called “mutation” and “crossover”. Mutation is a one-to-one operation that takes in input a chromosome and returns an offspring, in which a fraction of genes is replaced with a random value. Crossover is a two-to-one operation that takes as input two chromosomes and combines their genetic content into an offspring solution. Despite these operations being fundamental in providing the necessary exploration power throughout the genetic search, they require careful calibration so as not to destroy advantageous chromosome characteristics with too high probability. Excessive mutation and crossover rate in the genetic algorithm would lead to approximate the behaviour of a mere random search.

Our specific genetic optimization integrates the PyGAD library [70] for gene handling and the Qiskit software [71] for computing the quantum kernels. Our algorithm follows from a standard iterative procedure and can be summarized as follows:

- (i) Initialization: The algorithm begins by initializing a population of N randomly generated chromosomes. The population size remains constant across generations, setting an upper limit on the number of solutions explored in each generation. Mutation and crossover rates (μ and γ , respectively) are also fixed at this stage, as well as the definition of fitness function.
- (ii) Chromosome evaluation and parent selection: Each chromosome in the population is evaluated based on its fitness function output. A subset of the population, consisting of M chromosomes, is selected to survive into the next generation as parents without gene alteration. We based our selection criteria on a steady-state selection [72] when considering a mono-objective fitness function and on the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [73] when considering a multi-objective one. These methods assign a probability to each chromosome of being retained for the following generation. In some cases, we alternatively recur to an elitism strategy, in which the best K chromosomes in the generation are kept as parents with probability one, leaving no other parent to be selected.

- (iii) Offspring generation: The remaining $N - M$ chromosomes in the new generation (or $N - K$ when elitism is applied) are generated as offspring. These are produced by combining the genetic information from the M selected parents through a two-point crossover [74] and subsequent mutation operations. For each offspring, a random pair of parents is selected, with crossover occurring with a probability γ . If crossover does not occur, the offspring is a direct copy of the first parent. Each offspring chromosome then undergoes mutation (regardless crossover happened or not), where a fraction μ of its genes are randomly replaced.
- (iv) Steps (ii) and (iii) are repeated until a stop criterion is met. This criterion could be a predefined maximum number of generations or a convergence condition on the fitness function values within generations.

3.1. Gene representation

To allow the genetic algorithm to explore a wide range of feature maps throughout generations, mutation and crossover operations should be able to affect the quantum feature map circuit in the most general way possible. To achieve this, we translate the quantum circuit structure into a convenient array of integer numbers (genes) that encode specific pieces of information according to their value and position in the array, such as gate type and order. In this gene representation, rotation angles for variational gates are included, as well as specific gene-dependent transformations of one or two dataset features.

We associate a sub-array of six integers for each gate composing the circuit. The position of each sub-array inside the whole gene array determines the order in which it is applied, as well as the specific qubit(s) involved in the gate. The gate-gene correspondence is summarized in Table 1, and the gene-dependent gate arguments are collected in Table 2. The gate type gene defines whether the other genes will have an effect on the quantum circuit or not. For instance, if the gate type is non-parametric, the other genes for that gate referring to features and transformations will be ignored.

To roughly set a circuit size for the quantum kernels, we define an initial circuit size parameter S , such that the gene array length is $6 \times Q \times S$. This represents the total number of single-qubit gates acting on one qubit, plus the number of two-qubit gates where that qubit serves as the control qubit. We note that the gate number of a feature map can be smaller than $Q \times S$ when the identity gate is allowed in the circuit, making it possible for quantum feature maps to evolve into simpler circuits when needed.

3.2. Fitness function choices

By considering quantum kernels evaluated through statevector simulations, we identify three QSVM metrics to compose one mono-objective and one multi-objective fitness function:

- a - k -fold cross-validation accuracy of the QSVM's predictions, where the accuracy

Gene	Position	Range	Description
Gate type	0	$[0, N)$	Identifies a gate type among the allowed ones.
Transformation (T)	1	$[0, 2]$	Determines a transformation type.
Multi-feature (MF)	2	$[0, 1]$	Either using one or two features in the gate argument.
First feature index	3	$[0, F]$	Feature index to be used as first rotation argument.
Second feature index	4	$[0, F]$	Feature index to be used as second rotation argument
Second qubit index	5	$[0, Q) - \{q\}$	Target qubit for the two-qubit gate.

Table 1: Genes expressing a single gate in the feature map circuit acting on a specific qubit q in the circuit. If the gate type gene is associated with a two-qubit gate, q corresponds to the control qubit, while the target qubit is specified by the second qubit index gene. Each gene is an integer number bound in specific intervals, depending on the number of allowed gates N , total features in the dataset F , and qubits in the quantum circuit Q .

T \ MF	0	1
	0	$2\pi(x_i - 0.5)$
1	$2\pi(x_i - 0.5)(1 - x_i) - \pi$	$(2\pi(x_i - 0.5)(1 - x_j) - \pi) \times (2\pi(x_j - 0.5)(1 - x_i) - \pi) / \pi$
2	$2 \arcsin(2x_i - 1) - \pi$	$2 \arcsin((2x_i - 1)(2x_j - 1)) - \pi$

Table 2: Mathematical expressions for gate arguments determined by the Transformation (T) and Multi-feature (MF) genes. Each expression is bound between either $[-\pi, \pi]$ or $[0, -2\pi]$ for $x_i, x_j \in [0, 1]$. i and j correspond to the value of the first and second feature index genes respectively, as referred to in Table 1.

is evaluated as the fraction of correctly predicted samples over the total.

- σ - standard deviation of the off-diagonal kernel matrix entries.
- d - feature map circuit depth, which can be defined by the maximum path length (number of gates) connecting one qubit in the initial state and one in the final state, moving in the forward direction only. An equivalent definition in terms of complexity time steps is provided in [75].

While promoting QSVMs with the highest a is of primary importance, this metric alone is insufficient in NISQ setting where the quantum kernel matrix cannot be evaluated with arbitrary precision. The second and third metrics under consideration for this study allow us to estimate the feasibility of implementing a feature map on quantum hardware, without depending on a specific noise model. σ is an intuitive – yet powerful – metric that is correlated to the precision required by quantum hardware in estimating the kernel matrix. The ability of quantum hardware to distinguish between two different entries of the kernel matrix depends on both the number of measurement shots and the level of noise. An increased σ corresponds to a lower precision requirement, as the difference between kernel entries will generally be higher. In addition, promoting feature maps leading to high σ will also mitigate the exponential concentration problem, which manifests in a rapid decrease in σ as the number of qubits and circuit depth grow, without relying on projected quantum kernels. The circuit depth d straightforwardly addresses the quantum kernel estimation feasibility on noisy quantum hardware, as a higher depth implies more gates to be implemented sequentially, increasing the computation time and, therefore, the coherence time requirements for the qubits in the device. The circuit depth metric, being dependent on particular native gate sets and qubit connectivities, allows us to promote quantum circuits that are tailored to a specific QPU design.

We consider relying on statevector simulation and the control metrics mentioned above as the most efficient way to rapidly evaluate as many quantum kernels as required for the genetic search. Other approaches may involve leveraging noise models in the quantum kernel estimation, or real QPUs directly. This would partially replace the need for σ and d . However, these methods would be either computationally expensive or require excessive runtime for modern QPU availabilities, at this preliminary stage of the investigation. QPU parallelization, as motivated in Sec. 1 and investigated in Sec. 5, would overcome the second obstacle.

4. Statevector genetic optimization

We carried out several genetic optimizations for several initial settings, classically simulating from 4 to 12 qubits quantum kernels on a statevector backend. We randomly extracted a subset of 500 samples from the original dataset and employed it for all the analyses described in this section. Every run was initialized with $N = 50$ feature maps by generating chromosomes whose gene content was uniformly sampled from the intervals

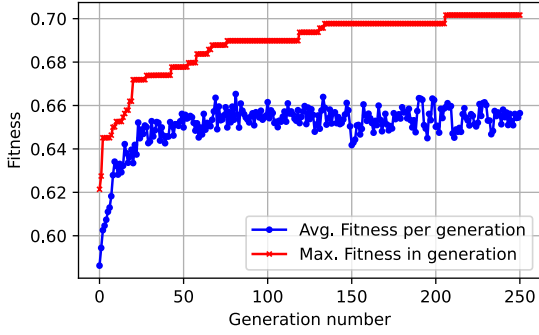
reported in Table 1. We let the feature maps evolve until completing 250 generations, which always resulted in reaching a fitness function convergence, and resorted to elitism by setting $K = 5$, meaning that the best 5 QSVMs in each generation (according to either the steady state selection or the NSGA-II algorithm) were propagated to the following one without any change in their genes. Running a genetic algorithm with these settings requires computing a total of $N + 250(N - 5) = 11300$ distinct quantum kernels.

Considering the fitness function metrics discussed in Sec. 3.2, we opted to design two distinct fitness function categories. The first one was mono-objective and consisted of a weighted sum of five-fold cross-validated accuracy a and the standard deviation of the kernel matrix off-diagonal elements σ :

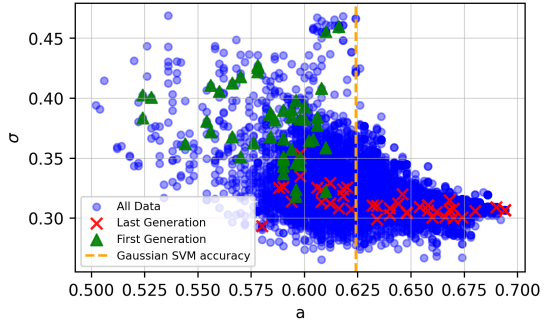
$$f = a + \eta\sigma. \quad (3)$$

The underlying idea was to keep η low enough to mainly promote QSVM with high accuracy, while preferring the kernel matrix with the highest variability only when the accuracies were equal or very similar. The value of $\eta = 0.025$ was determined experimentally. In Fig. 3 we report the two extreme cases 4 and 12 qubits in which we can clearly observe a positive trend in the fitness function throughout the generations, both for the optimal kernel in each generation and the average population fitness. This trend is also represented as the difference between the first generation kernels – that were randomly selected – and the last generation in the (a, σ) plane. In Fig. 3b and 3d, we marked the highest cross-validated accuracy achieved with our attempts to train classical SVM, using the scikit-learn package [76], represented by a Gaussian kernel. The Gaussian SVM hyperparameters have been optimized through a grid-search technique, leading to $a = 62.5\%$. This SVM is outperformed on the same dataset by many QSVMs, peaking at 69.5% for 4 qubits, and 67.5% for 12 qubits. Interestingly, the 4-qubits search leads to higher fitness values than in the 12-qubits case when considering the best kernel in each generation, while the contrary can be observed for the population average. We explain these effects by considering two key differences arising from this kind of optimization strategy. First, the number of genes increases linearly with the number of qubits, exponentially increasing the space of possible kernels and slowing down the genetic search. Second, the number of features (18) is comparable with the number of variational quantum gates composing the quantum circuits. With the specific parameters used in this study, the average number of variational gates in the feature map is 18 for the 4-qubit case and 52 for the 12-qubit case. Therefore, the probability of missing an informative feature for classification in a 12-qubit kernel circuit is lower, leading to higher accuracy on average. It is important to note that these considerations depend on the initial genetic settings and can be mitigated by increasing the parameter S and varying the allowed gates in the circuits.

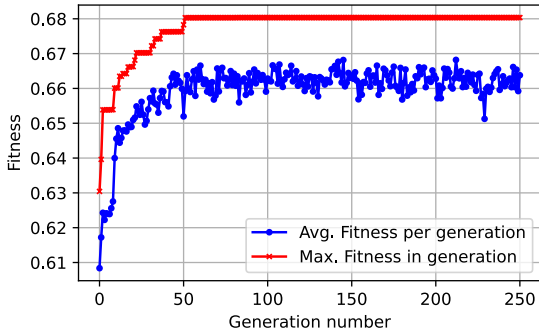
The second type of fitness function under investigation was multi-objective and takes



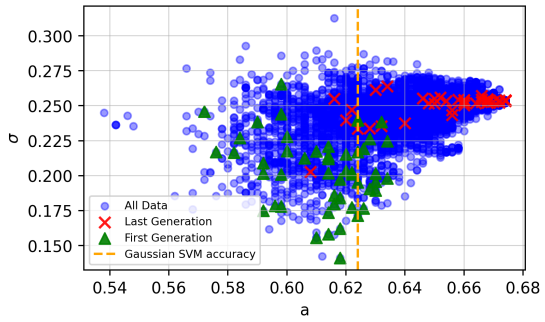
(a) 4-qubits fitness function through generations via steady-state selection.



(b) Distribution of 4-qubits QSVM performances in the (a, σ) plane.



(c) 12-qubits fitness function through generations via steady-state selection.



(d) Distribution of 12-qubits QSVM performances in the (a, σ) plane.

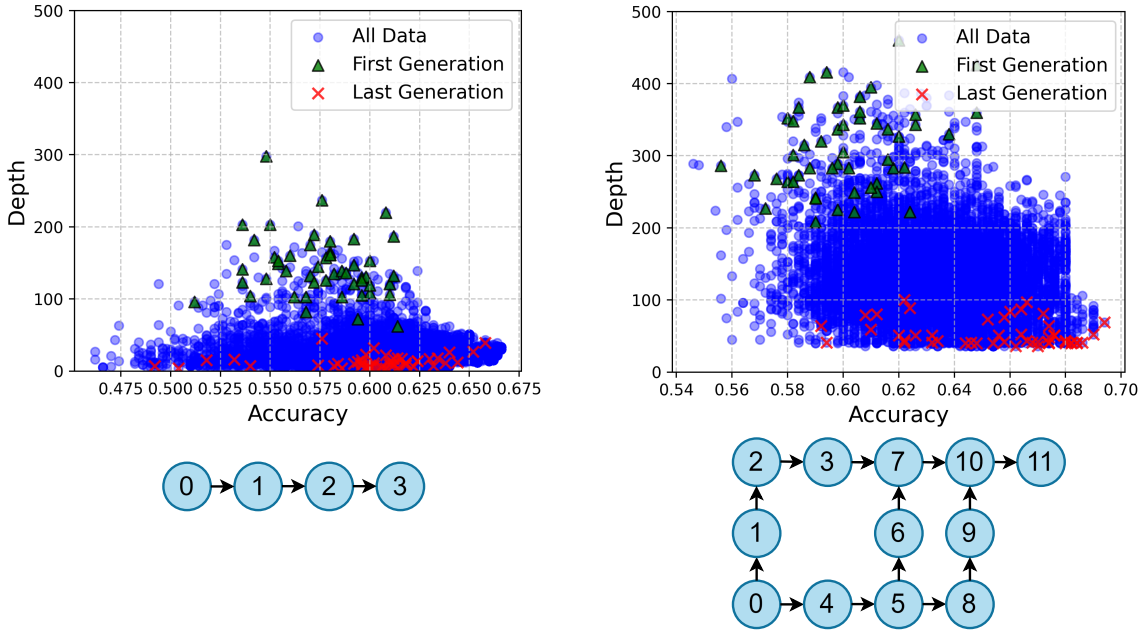
Figure 3: Optimized genetic feature maps for β and $\beta\beta$ classification are simulated with a statevector backend. Fitness function values, defined in Eq. 3, show improvement across generations for average (blue line) and best (red line) in (a) and (c). Cross-validation accuracy and kernel matrix standard deviation distributions in (b) and (d) compare the initial (green) and final (red) QVSMS, also benchmarked against a classical out-of-the-box SVM (vertical orange dashed line). Allowed gates: I, H, X, SX, RX, RY, RZ, CX, CRX, CRY, CRZ. Circuit size: $S = 8$.

the form:

$$f = \begin{pmatrix} a \\ -d \end{pmatrix}. \quad (4)$$

Hence we performed the genetic search through the Non-dominated Sorting Genetic Algorithm II (NSGA-II) to optimize the trade-off between accuracy and circuit depth. The results are shown in Fig. 4. We simulated a 4-qubit system with a spin-chain connectivity map and a 12-qubit system with a connectivity map as depicted in Fig. 4b. The circuit depths are calculated on a transpiled circuit considering the respective connectivity maps and a basis gate set consisting of I, X, SX, RZ, ECR.

In this case, we observe how the feature maps evolve towards optimal configurations



(a) 4-qubits fitness function objectives optimization via NSGA-II. (b) 12-qubits fitness function objectives optimization via NSGA-II.

Figure 4: Optimized genetic feature maps for β and $\beta\beta$ classification are simulated via a statevector backend without noise. (a) and (b) show the distributions of transpiled circuit depth and cross-validation accuracy, comparing the initial (green) and final (red) QVSMs under specific qubit connectivity assumptions. The fitness function is defined in Eq. 4. Allowed gates: I, H, X, SX, RX, RY, RZ, CX, ECR, CRX, CRY, CRZ. Circuit size: $S = 8$. Arrows in the connectivity maps represent the operation direction (control \rightarrow target) for the ECR gate.

in terms of high accuracy and low depth. In the 4-qubits case, we observe how accuracy in the population is generally lower than in the mono-objective optimization case, possibly suggesting that under a certain level of complexity of the quantum circuit, the QSVMS expressivity starts to decrease. The same observation does not apply to the 12-qubit case, suggesting that the Hilbert space accessed by 12 qubits is enough, on average, to prevent expressivity loss.

5. Parallelization of intermediate-scale quantum hardware

Despite the superconducting platform being significantly faster than other quantum computing platforms, such as trapped ions or neutral atoms [77, 78], computing a full genetic optimization run as in Sec. 4 using cloud-based access to commercially available superconducting quantum backends would require several days of continuous run time. Computing one single kernel matrix element requires repeated executions (on the order of thousands) of the quantum kernel circuit, for just one element in the

population. To mitigate this technological bottleneck we propose a partition of the quantum processing unit to parallelize the algorithm execution taking advantage of the large number of available qubits, executing the calculation for all kernels within the same calibration window (intra-chip execution). Moreover, we propose a multi-device parallelization, exploiting modular execution for chips of the same hardware release (intra-platform execution). This allows us to simultaneously extract the measurement outcome from each independent sub-unit. In this section, we describe in detail the parallelization routine for 4-qubits quantum kernels evaluation, assessing the impact of noise for different backends.

To assess the performance of intra vs extra platform execution we ran the experiment on three different superconducting backends provided by IBM: “ibm_nazca” (127 qubits), “ibm_brisbane” (127 qubits) and “ibm_torino” (133 qubits), considering a subset of 100 samples extracted from the original dataset. In the comparisons across the different backends (extra-platform), it was of fundamental importance to consider the different native gate sets for each QPU in choosing the circuit to run and identifying the qubit partitions. `ibm_nazca` and `ibm_brisbane` share the same basis gates (I, RZ, SX, X, ECR), using the echoed cross-resonance (ECR) gate as a native two-qubit interaction. For these devices, the ECR gate acts in one direction, i.e. for each qubit pair in the QPU, only one qubit can be designated as control, and only the other one as target for the gate. Despite sharing similar connectivity maps, `ibm_nazca` and `ibm_brisbane` have different target-control ordering, requiring different partitioning to compare their performances. The QPU partitioning for `ibm_nazca` and `ibm_brisbane` is depicted in Fig. 5, in which we carefully selected 21 sites for `ibm_nazca` and 20 for `ibm_brisbane`, sharing the same connectivities, i.e. a “directional chain” ($\rightarrow\rightarrow\leftarrow$), where each arrow starts from a control qubit and points to a target qubit. In choosing the partition schemes, we also avoided qubits with higher readout error rates and gate infidelity, which would negatively affect the kernel output.

Backend `ibm_torino` required a separate analysis, due to the different set of basis gates (I, RZ, SX, X, CZ). The controlled-Z (CZ) interaction, contrarily to ECR for the previous QPUs in our study, can be applied in both directions for any pair of connected qubits. This simplified the partitioning into non-directional spin chain sub-units.

Running the same quantum kernel evaluation across all QPU sub-units (or sites) for multiple devices allowed us to simultaneously evaluate the impact of noise in terms of output spread, i.e. the average standard deviation between the partitions output and the true kernel value. We also ranked the sub-units based on their performance, by evaluating the Frobenius norm [79] of the difference between the kernel matrix computed on each sub-unit and the one computed with a statevector simulation. In Fig. 6 we show an example of kernel entry estimation from all the QPU sub-units and the associated Frobenius norm value for `ibm_nazca` and `ibm_brisbane`.

Fig. 7a depicts the distribution of site output spread (standard deviations) for each off-diagonal kernel entry. We then examined how the distributions are affected when progressively discarding the worst sites in terms of the Frobenius norm. In Fig. 7b we

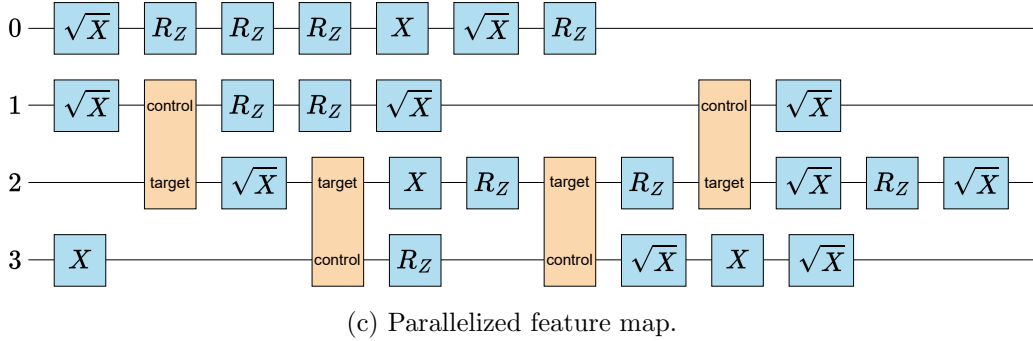
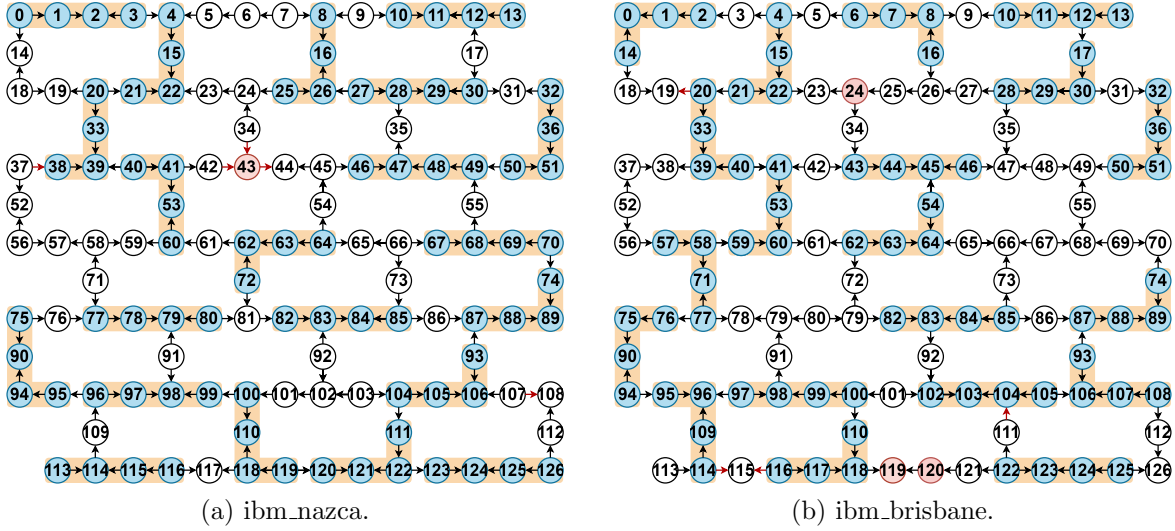
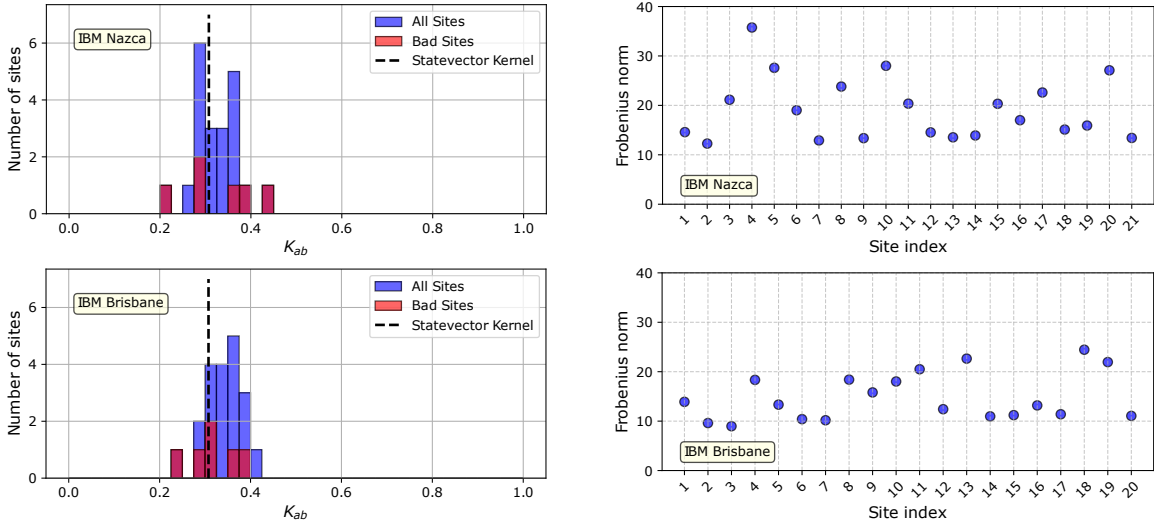


Figure 5: Partitioning scheme for the `ibm_nazca` (a) and `ibm_brisbane` (b) 127-qubits QPUs. Arrows indicate the direction of operation of the native ECR gate. Sub-units are highlighted in orange and the light-blue coloring identifies the qubits used. Red colouring indicates that either qubit readout or ECR gate fidelities were too low to be used. (c) The transpiled, random-generated feature map used in the parallelization experiment, with four ECR gates and a depth of 12.

show how averaging all kernel entries, the backend site spreads decrease proportionally to the number of excluded sites. By construction, when excluding all sites except one, the site spread is zero.

To carry out the extra-platform comparison, we opted to exclude from the analysis one of `ibm_brisbane` sites that presented several outliers in the kernel matrix entries, unaligned with the other site outputs, despite exhibiting a low Frobenius score. Moreover, to keep the same number of sites to compare between the 2 backends as a function of the discarded sites number, we also excluded from the analysis two average-performing sites for `ibm_nazca` (ranked 11th and 12th out of 21), so that we could keep a total of 19 sites both for both QPUs.

From this analysis we observe that the Brisbane output spread is lower, and the spread distribution widths (i.e. the error bars in Fig. 7b) are significantly lower for all



(a) Output spread across sites for `ibm_nazca` and `ibm_brisbane`. (b) Frobenius norm-based evaluation of `ibm_nazca` and `ibm_brisbane` sub-units.

Figure 6: (a) Output spread across sites for `ibm_nazca` and `ibm_brisbane` for an example kernel entry. The worst 6 sites (in terms of Frobenius norm) distributions are coloured in red. The black dashed lines mark the exact value computed with a statevector simulation. (b) Frobenius norm of the difference between the kernel matrix computed in each site and the statevector result.

the configurations. This result is aligned with our choice since `ibm_brisbane` exhibits higher single and two-qubit gate fidelities overall. The backend was characterized by a slightly lower Error Per Layered Gate (EPLG), considering the maximum-length qubit chain [80]: 3.0% against the 3.3% of `ibm_nazca`.

We carried out an intra-platform site-spread characterization also for `ibm_torino`. For a comparable analysis in terms of feature encoding circuits depth and entangling gates utilized for `ibm_nazca` and `ibm_brisbane`, we modified the kernel according to the basis gates set of `ibm_torino`. Fig. 8 shows the output spread analysis for the 21 designated partitions in the QPU. Despite this platform having the highest two-qubit gate fidelity and lowest EPLG for maximum-length qubit chain (0.8) among the QPUs analyzed in this study, the Frobenius distances are comparable with the ones of `ibm_brisbane`. In addition, the site spread as a function of the worst excluded site number is slightly higher than both `ibm_nazca` and `ibm_brisbane`. This result underlines the impact of noise in different quantum kernel estimations. Our data-driven approach allowed us to estimate the noise impact order of magnitude, which was consistent across the IBM devices under testing. Despite the attempt to compute quantum circuits with similar structures, the possibility of precisely quantifying site spread for generic quantum feature maps remains challenging.

We conclude our study by integrating the site spread information analyzed in this

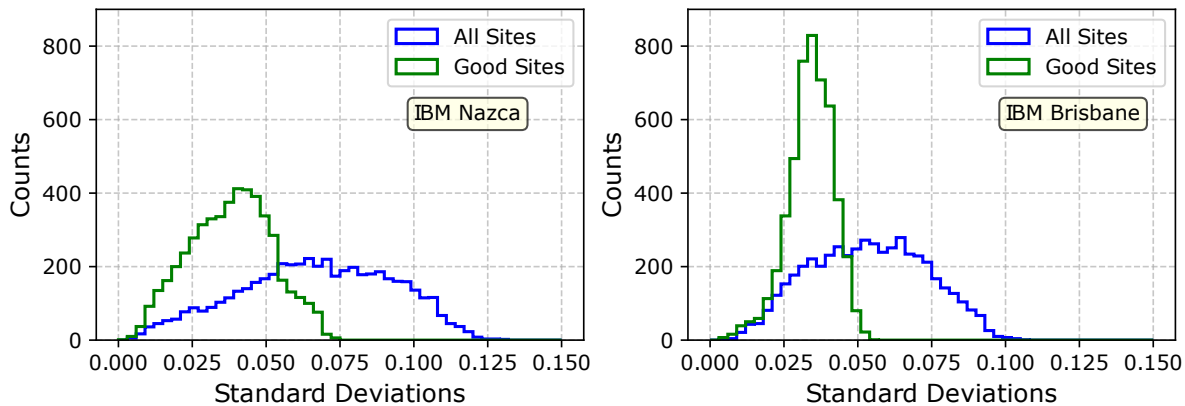
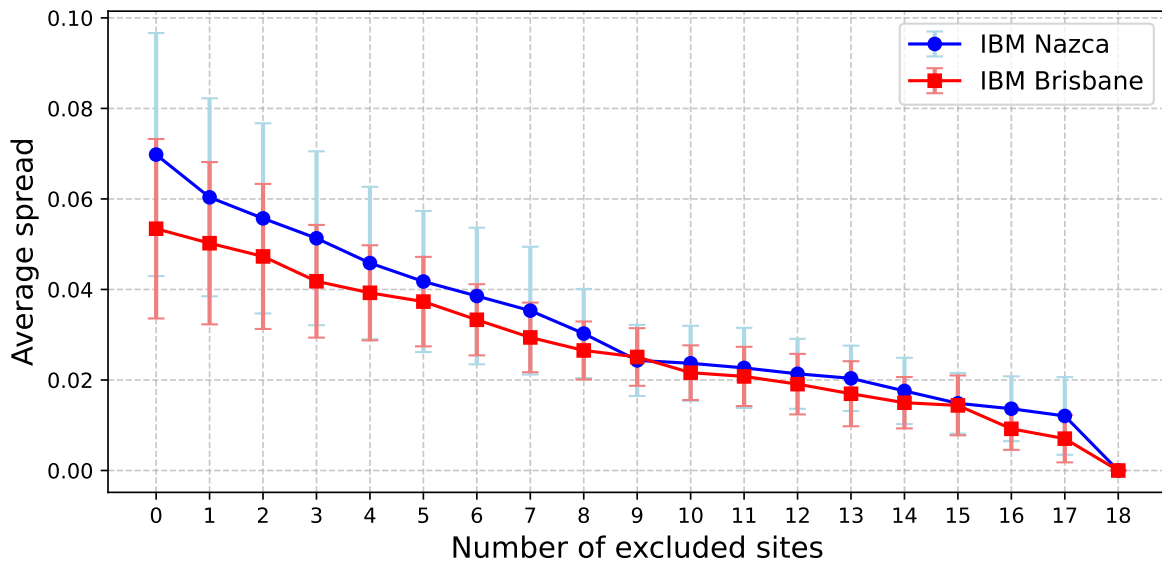
(a) Site spread distributions for `ibm_nazca` and `ibm_brisbane`.(b) Average spread distribution as a function of the number of sites excluded for `ibm_nazca` and `ibm_brisbane`.

Figure 7: (a) Site spread distributions, where each count in a histogram corresponds to the standard deviation of the kernel matrix entries evaluated by each site taken into consideration. The blue histograms collect the results for all partitions, while for the green ones the 6 worst sites in terms of Frobenius norm were rejected. (b) Average spread, corresponding to the mean value of the distributions in (a), for a different number of excluded sites. Sites were excluded in ascending order of the Frobenius score.

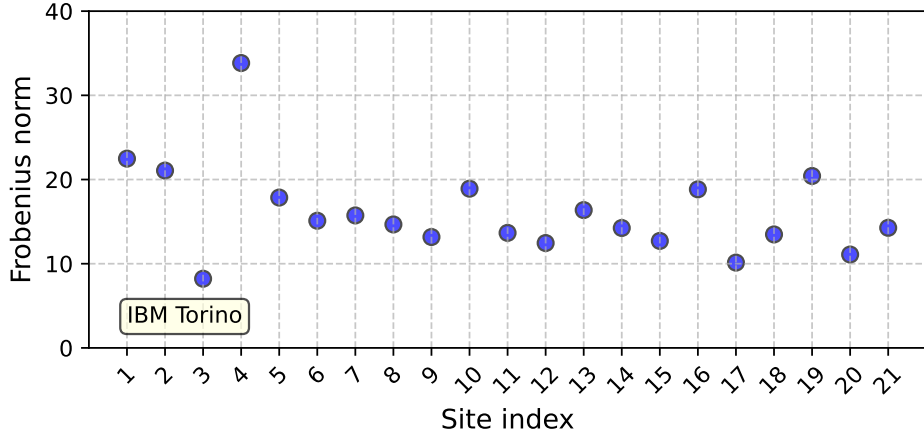
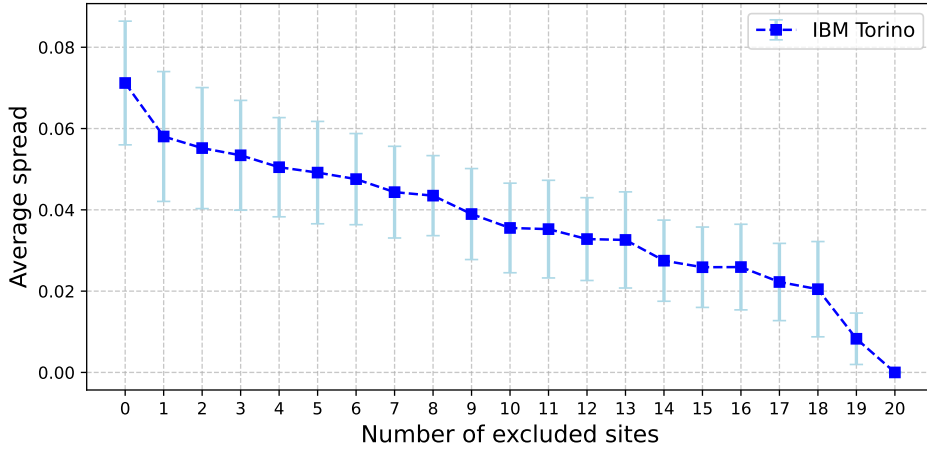
(a) Frobenius norm-based evaluation of `ibm_torino` partitions.(b) Average spread distribution as a function of the number of sites excluded for `ibm_torino`.

Figure 8: (a) Frobenius norm of the difference between the kernel matrix computed in each site and the statevector result. (b) Average site spread as a function of excluded sites number. Sites were excluded in ascending order of the Frobenius score shown in (a).

section into statevector genetic optimizations. By simulating the effect of noise directly on the kernel matrix at levels comparable to those observed in NISQ devices, we can assess whether genetic optimization can still be carried out under noisy conditions.

We applied a Gaussian distributed noise with zero mean and different standard deviations to each of the kernel entries. For these genetic optimizations, we considered 4-qubits quantum kernels with the basis gates set of `ibm_nazca` and `ibm_brisbane`, and we constrained the two-qubit gate to only apply to qubits that are directly connected in the $\rightarrow\rightarrow\leftarrow$ scheme of sub-units considered in the experiment. The dataset in use was the same as the one analysed in Sec. 4, and elitism was not applied ($K = 0$). The results are shown in Fig. 9, exhibiting a positive trend in the average fitness function throughout

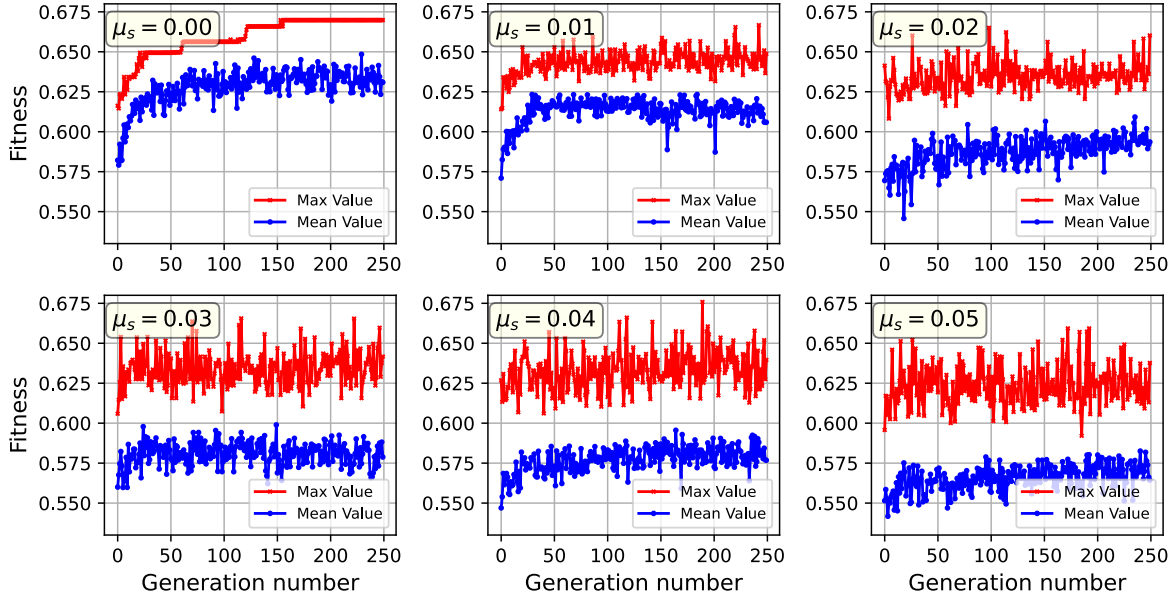


Figure 9: 4-qubits fitness function through generations via steady-state selection, applying a Gaussian noise with standard deviation μ_s , corresponding to the average site spread. Allowed gates: I, RZ, SX, X, ECR.

generations for average site spreads below $\mu_s = 0.02$, which is a level within reach of the NISQ devices tested in this study when retaining the best 20% of partitions. The impact of noise on the fittest quantum kernel curve seems more relevant, as a positive trend can be observed only for $\mu_s \leq 0.01$. In the $\mu_s = 0.01$ case, the best fitness function per generation reaches a peak of $f = 0.667$, translating into a cross-validated accuracy of $a = 65.8\%$ (following from eq. 3, $\eta = 0.025$ and $\sigma = 0.35$). In the ideal case, the fitness function plateaus at $f = 0.670$ and $a = 66.2\%$.

We expect the fitness function variability to derive both from noise and feature map variability throughout generations. This suggests that noise will impact differently in proportion to the overall variability for different genetic settings such as mutation and crossover rates. The effect of noise is also related to the circuit depth and the overall noise dependence of the other metrics composing the fitness function. We also expect the dataset size to play a role in the noisy optimization, as the intrinsic cross-validated accuracy fluctuations would be more relevant when considering small-sized datasets.

6. Conclusions

In this study, we introduced a novel feature map optimization strategy utilizing both mono-objective and multi-objective genetic algorithms to enhance the performance of QSVMs for binary classification tasks, with a specific focus on applications in neutrino physics. Our approach significantly improved the classification accuracy for single and double beta decay topologies (β and $\beta\beta$), as the genetic algorithm consistently converged

to feature maps yielding higher accuracies than the randomly initialized ones. Notably, our optimized QSVMs systematically outperformed classical Gaussian kernel SVMs, suggesting that the problem can be addressed more efficiently with quantum algorithms or quantum-inspired methods, under the same dataset dimensionality. Furthermore, despite the performance of deep learning methods for this task [46] remains unmatched for orders of magnitude larger datasets, the rarity of double beta decay events motivates the development of classification strategies capable of achieving high performance with limited training data.

From a pure algorithmic perspective, we explored a novel and potential approach for parallelizing genetic searches directly on quantum hardware, by executing identical quantum kernel circuits simultaneously on dedicated 4-qubits spin chain-like QPU partitions. These experiments allowed us to evaluate the noise-driven output spread on the kernel function values across such partitions. Incorporating this data into statevector simulations allowed us to estimate an output spread threshold above which the fitness function stops improving throughout generations. Our findings suggest that, after careful selection of the best-performing partition, parallelized genetic optimization on quantum hardware is within reach of the tested devices.

While this parallelization effort has shown potential, several near-term challenges remain. These include upscaling the qubit number in QPU partitions and exploring more complex connectivity schemes to optimize the distribution to multiple chips (extra-platform execution). For instance, significant benefits could be gained by including an automated transpilation process of the overall circuit, consisting of all sub-units, into an equivalent one with optimized depth, also taking into account readout error and gate fidelities of individual qubits. In this work we did not include and discuss quantum error mitigation strategies into the algorithm, which is left for future work.

Our work highlights the need for further exploration of quantum parallelization strategies and deeper integration with hardware-aware compilation techniques in the utility scale regime and provides useful insights for future studies aiming to enhance the practical utility of QML in real-world applications.

Acknowledgements This work is supported by PNRR MUR projects PE0000023-NQSTI and CN00000013-ICSC and by QUART&T, a project funded by the Italian Institute of Nuclear Physics (INFN) within the Technological and Interdisciplinary Research Commission (CSN5) and Theoretical Physics Commission (CSN4). MG is supported by CERN through CERN Quantum Technology Initiative. We gratefully acknowledge Matteo Biassoni for generating the dataset used throughout the study. We also thank Daniele Guffanti and Francesco Terranova for many insightful discussions on LArTPCs and the DUNE low-energy physics. We acknowledge the use of IBM Quantum services for this work. The views expressed are those of the authors, and do not reflect the official policy or position of IBM or the IBM Quantum team.

Data availability statement The dataset generated and analysed during this study is available upon reasonable request from the author. The manuscript has associated data in a repository: <https://github.com/rmoretti9/MaQIP.git>

References

- [1] Y. Kim, A. Eddins, S. Anand, et al. Evidence for the utility of quantum computing before fault tolerance. *Nature*, 618:500–505, 2023.
- [2] Nils Herrmann, Daanish Arya, Marcus W. Doherty, Angus Mingare, Jason C. Pillay, Florian Preis, and Stefan Prestel. Quantum utility – definition and assessment of a practical quantum advantage. In *2023 IEEE International Conference on Quantum Software (QSW)*. IEEE, July 2023.
- [3] Jules Tilly, Hongxiang Chen, Shuxiang Cao, Dario Picozzi, Kanav Setia, Ying Li, Edward Grant, Leonard Wossnig, Ivan Rungger, George H. Booth, and Jonathan Tennyson. The variational quantum eigensolver: A review of methods and best practices. *Physics Reports*, 986:1–128, 2022. The Variational Quantum Eigensolver: a review of methods and best practices.
- [4] Amira Abbas, Andris Ambainis, Brandon Augustino, Andreas Bärtzchi, Harry Buhrman, Carleton Coffrin, Giorgio Cortiana, Vedran Dunjko, Daniel J. Egger, Bruce G. Elmegreen, Nicola Franco, Filippo Fratini, Bryce Fuller, Julien Gacon, Constantin Gocciulea, Sander Gribling, Swati Gupta, Stuart Hadfield, Raoul Heese, Gerhard Kircher, Thomas Kleinert, Thorsten Koch, Georgios Korpas, Steve Lenk, Jakub Marecek, Vanio Markov, Guglielmo Mazzola, Stefano Mensa, Naeimeh Mohseni, Giacomo Nannicini, Corey O’Meara, Elena Peña Tapia, Sebastian Pokutta, Manuel Proissl, Patrick Rebentrost, Emre Sahin, Benjamin C. B. Symons, Sabine Törnøw, Victor Valls, Stefan Woerner, Mira L. Wolf-Bauwens, Jon Yard, Sheir Yarkoni, Dirk Zechiel, Sergiy Zhuk, and Christa Zoufal. Quantum optimization: Potential, challenges, and the path forward, 2024.
- [5] David Peral-García, Juan Cruz-Benito, and Francisco José García-Peñalvo. Systematic literature review: Quantum machine learning and its applications. *Computer Science Review*, 51:100619, 2024.
- [6] V. Belis, K. A. Woźniak, E. Puljak, and et al. Quantum anomaly detection in the latent space of proton collision events at the lhc. *Communications Physics*, 7:334, 2024.
- [7] Jinkai Tian, Xiaoyu Sun, Yuxuan Du, Shanshan Zhao, Qing Liu, Kaining Zhang, Wei Yi, Wanrong Huang, Chaoyue Wang, Xingyao Wu, Min-Hsiu Hsieh, Tongliang Liu, Wenjing Yang, and Dacheng Tao. Recent advances for quantum neural networks in generative learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):12321–12340, 2023.
- [8] Yang Qian, Xinbiao Wang, Yuxuan Du, Xingyao Wu, and Dacheng Tao. The dilemma of quantum neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 35(4):5603–5615, 2024.
- [9] Maria Schuld and Nathan Killoran. Is quantum advantage the right goal for quantum machine learning? *PRX Quantum*, 3:030101, Jul 2022.
- [10] Jonas Kübler, Simon Buchholz, and Bernhard Schölkopf. The inductive bias of quantum kernels. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 12661–12673. Curran Associates, Inc., 2021.
- [11] Joseph Bowles, Shahnawaz Ahmed, and Maria Schuld. Better than classical? the subtle art of benchmarking quantum machine learning models, 2024.
- [12] Xinbiao Wang, Yuxuan Du, Yong Luo, and Dacheng Tao. Towards understanding the power of quantum kernels in the nisq era. *Quantum*, 5:531, August 2021.
- [13] M. Alam and S. Ghosh. Qnet: A scalable and noise-resilient quantum neural network architecture for noisy intermediate-scale quantum computers. *Frontiers in Physics*, 9:755139, 2022.

- [14] S. Chen, J. Cotler, H.Y. Huang, and et al. The complexity of nisy. *Nature Communications*, 14(6001), 2023.
- [15] E. Peters, J. Caldeira, A. Ho, and et al. Machine learning of high dimensional data on a noisy quantum processor. *npj Quantum Information*, 7(161), 2021.
- [16] Maria Schuld and Nathan Killoran. Quantum machine learning in feature hilbert spaces. *Phys. Rev. Lett.*, 122:040504, Feb 2019.
- [17] Yadong Wu, Juan Yao, Pengfei Zhang, and Hui Zhai. Expressivity of quantum neural networks. *Phys. Rev. Res.*, 3:L032049, Aug 2021.
- [18] Vojtěch Havlíček, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, March 2019.
- [19] S. Jerbi, L.J. Fiderer, H. Poulsen Nautrup, and et al. Quantum machine learning beyond kernel methods. *Nature Communications*, 14(517), 2023.
- [20] Evan Peters and Maria Schuld. Generalization despite overfitting in quantum machine learning models. *Quantum*, 7:1210, December 2023.
- [21] Gabriele Agliardi, Giorgio Cortiana, Anton Dekusar, Kumar Ghosh, Naeimeh Mohseni, Corey O’Meara, Víctor Valls, Kavitha Yogaraj, and Sergiy Zhuk. Mitigating exponential concentration in covariant quantum kernels for subspace and real-world data, 2024.
- [22] H.Y. Huang, M. Broughton, M. Mohseni, and et al. Power of data in quantum machine learning. *Nature Communications*, 12(2631), 2021.
- [23] S. Thanasilp, S. Wang, M. Cerezo, and et al. Exponential concentration in quantum kernel methods. *Nature Communications*, 15(5200), 2024.
- [24] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, September 1995.
- [25] Maria Schuld. Supervised quantum machine learning models are kernel methods, 2021.
- [26] X. Vasques, H. Paik, and L. Cif. Application of quantum machine learning using quantum kernel algorithms on multiclass neuron m-type classification. *Scientific Reports*, 13:11541, 2023.
- [27] Teng Li, Zhipeng Yao, Xingtao Huang, Jiaheng Zou, Tao Lin, and Weidong Li. Application of the quantum kernel algorithm on the particle identification at the besiii experiment. *Journal of Physics: Conference Series*, 2438:012071, 2023.
- [28] N. Cristianini, J. Kandola, A. Elisseeff, and J. Shawe-Taylor. On kernel target alignment. In D.E. Holmes and L.C. Jain, editors, *Innovations in Machine Learning*, volume 194 of *Studies in Fuzziness and Soft Computing*. Springer, Berlin, Heidelberg, 2006.
- [29] Thomas Hubregtsen, David Wierichs, Elies Gil-Fuster, Peter-Jan H. S. Derks, Paul K. Faehrmann, and Johannes Jakob Meyer. Training quantum embedding kernels on near-term quantum computers. *Physical Review A*, 106(4), October 2022.
- [30] Li Xu, Xiao-yu Zhang, Ming Li, and Shu-qian Shen. Quantum classifiers with a trainable kernel. *Phys. Rev. Appl.*, 21:054056, May 2024.
- [31] Pablo Rodriguez-Grasa, Yue Ban, and Mikel Sanz. Training embedding quantum kernels with data re-uploading quantum neural networks, 2024.
- [32] J.R. McClean, S. Boixo, V.N. Smelyanskiy, and et al. Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9(4812), 2018.
- [33] M. Cerezo, A. Sone, T. Volkoff, and et al. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nature Communications*, 12(1791), 2021.
- [34] Lucas Slattery, Ruslan Shaydulín, Shouvanik Chakrabarti, Marco Pistoia, Sami Khairy, and Stefan M. Wild. Numerical evidence against advantage with quantum fidelity kernels on classical data. *Physical Review A*, 107(6), June 2023.
- [35] John Preskill. Quantum computing in the nisy era and beyond. *Quantum*, 2:79, August 2018.
- [36] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S. Kottmann, Tim Menke, Wai-Keong Mok, Sukin Sim, Leong-Chuan Kwek, and Alán Aspuru-Guzik. Noisy intermediate-scale

- quantum algorithms. *Rev. Mod. Phys.*, 94:015004, Feb 2022.
- [37] Sergio Altares-López, Angela Ribeiro, and Juan José García-Ripoll. Automatic design of quantum feature maps. *Quantum Science and Technology*, 6(4):045015, 2021. © 2021 The Author(s). Published by IOP Publishing Ltd.
- [38] Sergio Altares-López, Juan José García-Ripoll, and Angela Ribeiro. Autoqml: Automatic generation and training of robust quantum-inspired classifiers by using evolutionary algorithms on grayscale images. *Expert Systems with Applications*, 244:122984, 2024.
- [39] R. Pellow-Jarman, A. Pillay, I. Sinayskiy, and et al. Hybrid genetic optimization for quantum feature map design. *Quantum Machine Intelligence*, 6(45), 2024.
- [40] Elham Torabian and Roman V. Krems. Compositional optimization of quantum circuits for quantum kernels of support vector machines. *Phys. Rev. Res.*, 5:013211, Mar 2023.
- [41] Haiyan Wang. A novel feature selection method based on quantum support vector machine. *Physica Scripta*, 99(5):056006, 2024. © 2024 IOP Publishing Ltd.
- [42] Haiyan Wang. Several fitness functions and entanglement gates in quantum kernel generation, 2023.
- [43] Floyd M. Creevey, Jamie A. Heredge, Martin E. Sevier, and Lloyd C. L. Hollenberg. Kernel alignment for quantum support vector machines using genetic algorithms, 2023.
- [44] Massimiliano Incudini, Daniele Lizzio Bosco, Francesco Martini, Michele Grossi, Giuseppe Serra, and Alessandra Di Pierro. Automatic and effective discovery of quantum kernels, 2023.
- [45] Francesca Schiavello, Edoardo Altamura, Ivano Tavernelli, Stefano Mensa, and Benjamin Symons. Evolving a multi-population evolutionary-qaoa on distributed qpus, 2024.
- [46] R. Moretti, M. Rossi, M. Biassoni, and et al. Assessment of few-hits machine learning classification algorithms for low-energy physics in liquid argon detectors. *European Physical Journal Plus*, 139(723), 2024.
- [47] W. M. Bonivento and F. Terranova. The science and technology of liquid argon detectors. *Rev. Mod. Phys.*, 96(4):045001, 2024.
- [48] A. Mastbaum, F. Psihas, and J. Zennaro. Xenon-doped liquid argon tpcs as a neutrinoless double beta decay platform. *Phys. Rev. D*, 106:092002, Nov 2022.
- [49] DUNE Collaboration. Dune phase ii: Scientific opportunities, detector concepts, technological solutions, 2024.
- [50] Michelle J. Dolinski, Alan W.P. Poon, and Werner Rodejohann. Neutrinoless double-beta decay: Status and prospects. *Annual Review of Nuclear and Particle Science*, 69(Volume 69, 2019):219–251, 2019.
- [51] Matteo Agostini, Giovanni Benato, Jason A. Detwiler, Javier Menéndez, and Francesco Vissani. Toward the discovery of matter creation with neutrinoless $\beta\beta$ decay. *Reviews of Modern Physics*, 95(2), May 2023.
- [52] Arnab Dasgupta, P. S. Bhupal Dev, Sin Kyu Kang, and Yongchao Zhang. New mechanism for matter-antimatter asymmetry and connection with dark matter. *Phys. Rev. D*, 102:055009, Sep 2020.
- [53] YeolLin ChoeJo, Yechan Kim, and Hye-Sung Lee. Dirac-majorana neutrino type oscillation induced by a wave dark matter. *Phys. Rev. D*, 108:095028, Nov 2023.
- [54] S. Andringa, J. Asaadi, J.T.C. Bezerra, and et al. Low-energy physics in neutrino lartpcs. *Journal of Physics G: Nuclear and Particle Physics*, 50(3):033001, 2023. © 2023 IOP Publishing Ltd.
- [55] T. Bezerra, A. Borkum, E. Church, and et al. Large low background kton-scale liquid argon time projection chambers. *Journal of Physics G: Nuclear and Particle Physics*, 50(6):060502, 2023. © 2023 IOP Publishing Ltd.
- [56] A. Laing and on behalf of the NEXT Collaboration. The next double beta decay experiment. *Journal of Physics: Conference Series*, 718(6):062033, 2016.
- [57] A. Gando, Y. Gando, T. Hachiya, A. Hayashi, S. Hayashida, H. Ikeda, K. Inoue, K. Ishidoshiro, Y. Karino, M. Koga, S. Matsuda, T. Mitsui, K. Nakamura, S. Obara, T. Oura, H. Ozaki, I. Shimizu, Y. Shirahata, J. Shirai, A. Suzuki, T. Takai, K. Tamae, Y. Teraoka, K. Ueshima,

- H. Watanabe, A. Kozlov, Y. Takemoto, S. Yoshida, K. Fushimi, T. I. Banks, B. E. Berger, B. K. Fujikawa, T. O'Donnell, L. A. Winslow, Y. Efremenko, H. J. Karwowski, D. M. Markoff, W. Tornow, J. A. Detwiler, S. Enomoto, and M. P. Decowski. Search for majorana neutrinos near the inverted mass hierarchy region with kamland-zen. *Phys. Rev. Lett.*, 117:082503, Aug 2016.
- [58] Lin Si, Zhaokan Cheng, Abdusalam Abdukerim, Zihao Bo, Wei Chen, Xun Chen, Yunhua Chen, Chen Cheng, Yunshan Cheng, Xiangyi Cui, Yingjie Fan, Deqing Fang, Changbo Fu, Mengting Fu, Lisheng Geng, Karl Giboni, Linhui Gu, Xuyuan Guo, Ke Han, Changda He, Jinrong He, Di Huang, Yanlin Huang, Zhou Huang, Ruquan Hou, Xiangdong Ji, Yonglin Ju, Chenxiang Li, Jiafu Li, Mingchuan Li, Shu Li, Shuaijie Li, Qing Lin, Jianglai Liu, Xiaoying Lu, Lingyin Luo, Yunyang Luo, Wenbo Ma, Yugang Ma, Yajun Mao, Yue Meng, Nasir Shaheed, Xiaofeng Shang, Xuyang Ning, Ningchun Qi, Zhicheng Qian, Xiangxiang Ren, Changsong Shang, Guofang Shen, Wenliang Sun, Andi Tan, Yi Tao, Anqing Wang, Meng Wang, Qiuhong Wang, Shaobo Wang, Siguang Wang, Wei Wang, Xiuli Wang, Zhou Wang, Yuehuan Wei, Mengmeng Wu, Weihao Wu, Jingkai Xia, Mengjiao Xiao, Xiang Xiao, Pengwei Xie, Binbin Yan, Xiyu Yan, Jijun Yang, Yong Yang, Chunxu Yu, Jumin Yuan, Ying Yuan, Zhe Yuan, Dan Zhang, Minzhen Zhang, Peng Zhang, Shibo Zhang, Shu Zhang, Tao Zhang, Li Zhao, Qibin Zheng, Jifang Zhou, Ning Zhou, Xiaopeng Zhou, and Yong Zhou. Determination of double beta decay half-life of ^{136}Xe with the pandax-4t natural xenon detector. *Research*, 2022:9798721, 2022.
- [59] F. Agostini, S.E.M.A. Maouloud, L. Althueser, and et al. Sensitivity of the darwin observatory to the neutrinoless double beta decay of xe. *European Physical Journal C*, 80(808), 2020.
- [60] G. Adhikari, S. Al Kharusi, E. Angelico, and et al. nexo: neutrinoless double beta decay search beyond 10^{28} year half-life sensitivity. *Journal of Physics G: Nuclear and Particle Physics*, 49(1):015104, 2022. © 2021 IOP Publishing Ltd.
- [61] J. Soto-Oton and on behalf of the DUNE Collaboration. Impact of xenon doping in the scintillation light in a large liquid-argon tpc. *Journal of Physics: Conference Series*, 2374:012164, 2022. © 2022 IOP Publishing Ltd.
- [62] Dune Collaboration. The dune far detector vertical drift technology. technical design report. *Journal of Instrumentation*, 19:T08004, 2024. © 2024 CERN.
- [63] DUNE Collaboration. Doping liquid argon with xenon in protodune single-phase: Effects on scintillation light, 2024.
- [64] M.J. Berger, J.S. Coursey, M.A. Zucker, and J. Chang. Stopping-power & range tables for electrons, protons, and helium ions. NIST Standard Reference Database 124, 2017. Last Update to Data Content: July 2017.
- [65] Fionn Murtagh. Multilayer perceptrons for classification and regression. *Neurocomputing*, 2(5):183–197, 1991.
- [66] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [67] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, 2015.
- [68] Yabo Wang, Bo Qi, Xin Wang, Tongliang Liu, and Daoyi Dong. Power characterization of noisy quantum kernels, 2024.
- [69] S. Katoch, S.S. Chauhan, and V. Kumar. A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, 80:8091–8126, 2021. Issue Date: February 2021.

- [70] A.F. Gad. Pygad: an intuitive genetic algorithm python library. *Multimedia Tools and Applications*, 83:58029–58042, 2024. Issue Date: June 2024.
- [71] Ali Javadi-Abhari, Matthew Treinish, Kevin Krsulich, Christopher J. Wood, Jake Lishman, Julien Gacon, Simon Martiel, Paul D. Nation, Lev S. Bishop, Andrew W. Cross, Blake R. Johnson, and Jay M. Gambetta. Quantum computing with Qiskit, 2024.
- [72] A. Agapie and A.H. Wright. Theoretical analysis of steady state genetic algorithms. *Applied Mathematics*, 59:509–525, 2014. Issue Date: October 2014.
- [73] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
- [74] A.J. Umbarkar and P.D. Sheth. Crossover operators in genetic algorithms: A review. *ICTACT Journal on Soft Computing*, 6(1):1083–1092, October 2015.
- [75] L. Gyongyosi and S. Imre. Circuit depth reduction for gate-model quantum computers. *Scientific Reports*, 10:11229, 2020.
- [76] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [77] Norbert M. Linke, Dmitri Maslov, Martin Roetteler, Shantanu Debnath, Caroline Figgatt, Kevin A. Landsman, Kenneth Wright, and Christopher Monroe. Experimental comparison of two quantum computing architectures. *Proceedings of the National Academy of Sciences*, 114(13):3305–3310, 2017.
- [78] Karen Wintersperger, Florian Dommert, Thomas Ehmer, Andrey HOURSANOV, Johannes Klepsch, Wolfgang Mauerer, Georg Reuber, Thomas Strohm, Ming Yin, and Sebastian Luber. Neutral atom quantum computing hardware: performance and end-user perspective. *EPJ Quantum Technology*, 10(1), August 2023.
- [79] Gérard Meurant. *Computer Solution of Large Linear Systems*, volume 28 of *Studies in Mathematics and Its Applications*. Elsevier, 1999.
- [80] David C. McKay, Ian Hincks, Emily J. Pritchett, Malcolm Carroll, Luke C. G. Govia, and Seth T. Merkel. Benchmarking quantum processor performance at scale, 2023.