MACHINE
LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
14 March 2023

REVISED
4 September 2023

ACCEPTED FOR PUBLICATION
30 October 2023

PUBLISHED
16 November 2023

PAPER

# Unravelling physics beyond the standard model with classical and quantum anomaly detection

**Julian Schuhmacher**[1], **Laura Boggia**[1,2,6], **Vasilis Belis**[3], **Ema Puljak**[4,5], **Michele Grossi**[5], **Maurizio Pierini**[5], **Sofia Vallecorsa**[5], **Francesco Tacchino**[1], **Panagiotis Barkoutsos**[1,7] **and Ivano Tavernelli**[1,*]

[1] IBM Quantum, IBM Research Europe - Zurich, Rüschlikon 8803, Switzerland
[2] Institute for Theoretical Physics, ETH Zürich, 8093 Zürich, Switzerland
[3] Institute for Particle Physics and Astrophysics, ETH Zürich, 8093 Zürich, Switzerland
[4] Department of Physics, Autonomous University of Barcelona, Cerdanyola del Vallès, Spain
[5] European Organization for Nuclear Research (CERN), Geneva 1211, Switzerland
[6] Current address: LPNHE, Sorbonne Université, Université Paris Cité, CNRS/IN2P3, Paris, France and IBM Research, Orsay, France.
[7] Current address: PASQAL SAS, 2 Av. Augustin Fresnel Palaiseau, 91120, France.
[*] Author to whom any correspondence should be addressed.

**E-mail:** ita@zurich.ibm.com

**Keywords:** high energy physics, anomaly detection, quantum computing, quantum machine learning

## Abstract

Much hope for finding new physics phenomena at microscopic scale relies on the observations obtained from High Energy Physics experiments, like the ones performed at the Large Hadron Collider (LHC). However, current experiments do not indicate clear signs of new physics that could guide the development of additional Beyond Standard Model (BSM) theories. Identifying signatures of new physics out of the enormous amount of data produced at the LHC falls into the class of anomaly detection and constitutes one of the greatest computational challenges. In this article, we propose a novel strategy to perform anomaly detection in a supervised learning setting, based on the artificial creation of anomalies through a random process. For the resulting supervised learning problem, we successfully apply classical and quantum support vector classifiers (CSVC and QSVC respectively) to identify the artificial anomalies among the SM events. Even more promising, we find that employing an SVC trained to identify the artificial anomalies, it is possible to identify realistic BSM events with high accuracy. In parallel, we also explore the potential of quantum algorithms for improving the classification accuracy and provide plausible conditions for the best exploitation of this novel computational paradigm.

## 1. Introduction

Current approaches for the description of elementary particles rely on the standard model (SM) of particle physics [1–4]. Despite its experimental success, the SM is theoretically incomplete and new physics is yet to be explored [5, 6]. Since the discovery of the Higgs boson, the search for new physics, for example via experiments at the Large Hadron Collider (LHC), has eventually become one of the main focuses of research around high energy physics (HEP).

Experimental data obtained by the LHC experiments can help addressing the theoretical shortcomings of the SM. If recorded, observations showing a significant deviation from the SM would indicate the existence of new physics. However, state of the art LHC experiments do not yet show a clear indication of phenomena that could motivate and validate new theories. A key challenge in this quest is represented by the problem of storing and processing the amount of data produced by the LHC, amounting to $\mathcal{O}(10^6)$ collisions per second, each consisting of $\mathcal{O}(1)$ MB. This challenge will be amplified significantly with the start of the high luminosity-LHC program planned for 2029 [7], that will reach a larger luminosity at the price of a larger data flow.

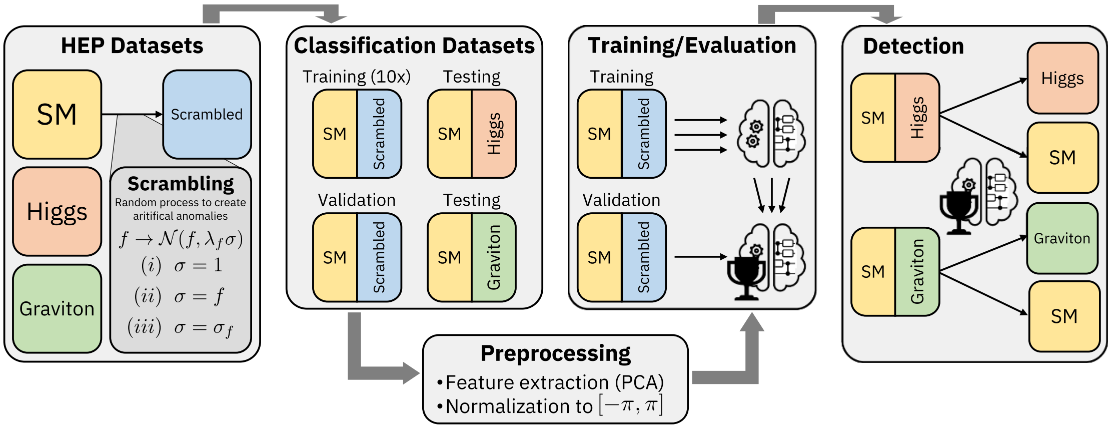

**Figure 1.** Illustration of the pipeline advised for our anomaly detection strategy. We start with HEP datasets of simulated SM, Higgs and Graviton events. The Higgs events are considered separately, because they are not included in the processes of the SM dataset. We apply a random process to the SM dataset to create artificial anomalies (see section 2). Based on the four different datasets, we create balanced two-class datasets. One of the classes is always the SM. The other class is either composed of artificial anomalies (training and validation), or Higgs or Graviton events (testing). We apply two preprocessing steps to the classification datasets: feature extraction with PCA and normalization of the extracted features to the interval $[-\pi, \pi]$. Each training dataset is then used to train multiple quantum or classical SVCs from which we select the best one based on the performance on the test dataset. The best SVC is then applied in the detection of unseen anomalies (Higgs or Graviton events).

A possible remedy to handle this huge amount of data relies on the use of machine learning (ML) models, which digest large amounts of data in order to extract underlying patterns [8–12]. The use of anomaly detection techniques has been proposed as a valuable tool to identify Beyond SM (BSM) events among the dominating number of SM background events [13–24]. Different approaches have been investigated, belonging either to the supervised or unsupervised learning regimes. In the supervised learning setting, the ML model is trained to distinguish between background and signal events using a dataset where the events were labelled with the corresponding class. However, to acquire the labels of an event some preliminary knowledge about the studied processes is required, e.g. through numerical simulations of a BSM theory. This typically limits the generalization power of a supervised algorithm. In the context of the search for new physics at the LHC, this means that this approach is typically effective whenever the considered signal is the correct one (e.g. for Higgs boson searches). In unsupervised learning setting, the ML model is trained to learn the structure of a dataset largely dominated by SM events. Without being provided any additional information, it aims at identifying an anomaly as an outlier of some typicality measure, learned during the training process. An unsupervised learning algorithm requires no labels. On one hand, this increases the generalization power of the model. On the other hand, it reduces the accuracy since less information about any specific signal is used.

In this paper, we propose a novel strategy for anomaly detection that tries to retain the best out of the supervised and unsupervised setting. We propose a supervised learning approach in which the signal sample is built by perturbing the background sample, without relying on any specific BSM theory. The background events are represented by a selection of SM processes, and signal events are generated artificially through a random process, which is denoted here as *scrambling*. This approach ensures that we introduce as little physically-inspired bias as possible into the types of signal events we are looking for, while guaranteeing that the obtained processes are compliant with the conservation laws of physics and detector-specific constraints. We solve the resulting binary classification between background and signal events with the support vector classifier (SVC) approach. An illustration of the proposed anomaly detection pipeline is shown in figure 1. The proposed approach sits between purely supervised and purely unsupervised methods, and therefore we also expected that its accuracy in detecting specific signal events and its generalizability will lie between these two approaches.

In parallel with classical SVC techniques, we also consider the application of quantum kernel based classification methods. Quantum ML (QML) has recently been proposed as a new framework offering potential speedups and performance improvements over classical ML [25–28]. Several QML algorithms, like quantum SVCs (QSVCs) [29–31], variational quantum classifiers (VQCs) [30, 31], quantum convolutional neural networks [32], or quantum autoencoders [33] have been applied to a wide range of HEP problems [34–50]. With the current methods, quantum algorithms generally achieve a performance similar to their classical counterparts. While future quantum hardware and algorithmic improvements may lead to more significant advantages, a solid and rigorous benchmark of the available quantum techniques is already important today.

In this work, we verify numerically the feasibility to distinguish SM events from artificially created anomalies. We show that such a classifier preserves its discrimination power once the scrambling anomalies are replaced by events from realistic BSM theories. Furthermore, we demonstrate the application of a kernel-based quantum classification algorithm to the problem under study.

This paper is organized as follows: In section 2 we present the scrambling method, followed by a short description of the applied classification algorithm in section 3. In section 4 we demonstrate and discuss the effectiveness of the proposed anomaly detection workflow. Finally, in section 5, we conclude with a general discussion of the proposed method and the application of quantum algorithms for the studied classification task.

## 2. Dataset scrambling

To generate a dataset for our supervised learning problem, we start from an existing collection of SM events and we vary the different features in order to introduce anomalies (artificial events). The events are represented by features like the momenta or the number of particles, extracted from the collision data obtained from experiments or simulations (see section 4.1 and appendix A for details). The variation of these features is done under certain constrains imposed by physics conservation laws or experiment-related constraints. We call this process data scrambling. Its goal is to generate events that do not conform with the SM, without relying on a specific BSM theory, and therefore introducing as little bias as possible about the type of BSM events we would like to identify. We propose to do this via a random perturbation of the SM events. The main idea of the scrambling is illustrated in figure 2(a). Starting from the SM dataset (yellow region), we generate artificial events outside the SM (blue region). Even though there is an overlap between the region of SM events and artificial events, we want to verify that a classifier, trained to separate the SM events from the artificial BSM ones, would retain its discrimination power once applied to realistic BSM events (green region).

The scrambling is done by replacing a feature in the original SM dataset with a new value chosen according to a Gaussian distribution $\mathcal{N}(\mu,\sigma)$, where $\mu = f$ is the initial value of the feature and $\sigma$ is the standard deviation. Depending on the feature, the standard deviation of the scrambling distribution is chosen according to one of the following three options; *(i)* the standard deviation is fixed to a constant, $\sigma = \lambda_f$, *(ii)* proportional to the standard deviation $\sigma_f$ of the initial feature distribution, $\sigma = \lambda_f\sigma_f$, or *(iii)* proportional to the feature value $f$, $\sigma = \lambda_f f$. The constant $\lambda_f$, in the following denoted as *scrambling factor*, determines the strength of the scrambling.

Some of the features are correlated (e.g. transverse momentum of lepton and missing transverse energy) and therefore cannot be scrambled individually. Additionally, depending on the feature we have to implement different strategies to respect conservation laws or detector-specific limitations. Therefore, the features are divided into four categories: momenta, isolations, jets and particle numbers, each with their own scrambling strategy. The scrambling strategies are presented in detail in appendix B. In figure 2(b), the scrambling is visualized for some features in the SM dataset, each belonging to one of the categories mentioned above.

Even though the scrambling does not have an explicit bias related to specific BSM theories, it contains implicit assumptions based on the chosen sampling distribution. As presented above, the scrambling is performed independently for each feature (unless the features are related to each other by e.g. conservation laws). However, in principle it is possible to use any random process to perturb the features, i.e. even introducing correlations among them. The only requirement the generated events have to fulfill is that some lie outside the 'event space' of the SM, while respecting the physical conservation laws and the constraints imposed by the detector. However, there is no guarantee that any chosen scrambling distribution will generate events resembling real BSM events. Nevertheless, the hope is that by learning to distinguish between SM and artificial events, we obtain some level of generalization on out-of-distribution samples, and therefore the possibility to detect BSM events even if they would lie outside the space of scrambled events (no overlap between the blue and green region in figure 2(a). The generalization capabilities of our approach are qualitatively tested by means of the experiments performed in section 4.

## 3. Support vector classifier

### 3.1. Classical SVC

The SVC, which belongs to the family of kernel methods, is a supervised learning model that can draw hyperplanes between two classes of data points. By embedding data points into a high dimensional feature space, where they become linearly separable, SVCs can successfully solve complex classification tasks. The

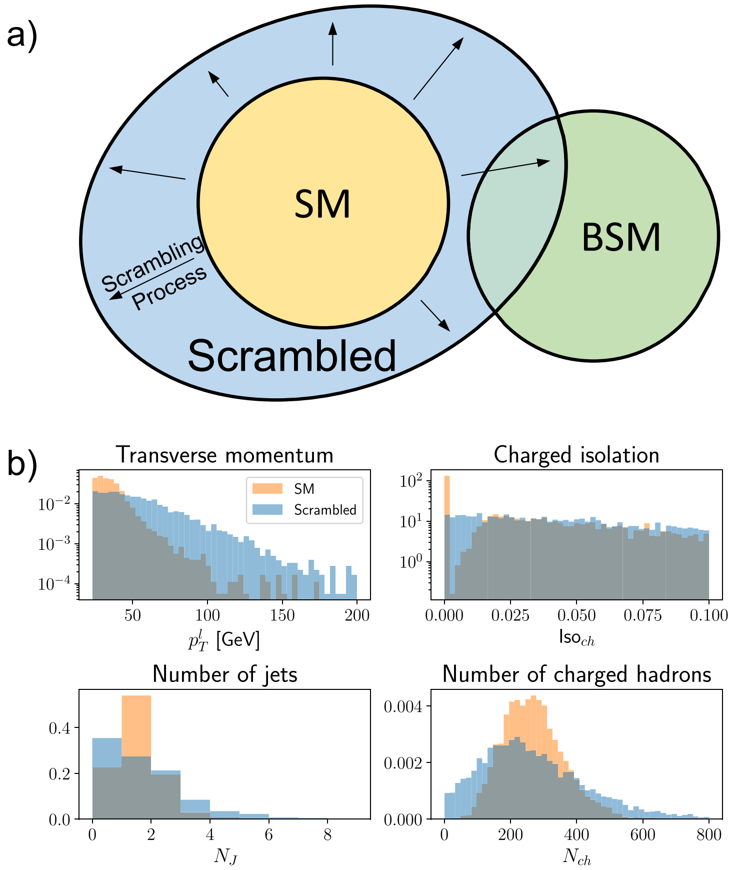

**Figure 2.** (a) Schematic illustration of the scrambling idea. The scrambling process generates random events (blue region) based on SM events (yellow region). Being able to distinguish a SM event from a scrambled event, enables us to identify events originating from physics beyond the SM (green region). (b) Comparison of initial SM data and scrambled data at medium scrambling strength, for a selection of the high-level features.

success of the SVC results from the so-called *kernel trick*, which allows to calculate a similarity measure between data points (i.e. the kernel) without explicitly performing the mapping to the high dimensional feature space. An example of such a kernel is the widely used radial basis function (RBF) kernel (also known as Gaussian kernel),

$$K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2\right) , \tag{1}$$

where $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^n$, and $\gamma$ is a hyperparameter to determine the bandwidth of the kernel function. Notice that this specific kernel function corresponds to a situation in which data points would effectively be mapped into an infinite-dimensional feature space [51].

### 3.2. Quantum SVC

For the QSVC, the kernel values are evaluated in a Hilbert space of quantum states. Specifically, classical features $\mathbf{x} \in \mathcal{X}$ are encoded in the quantum state space $\mathcal{F}$ via a feature map $\phi : \mathcal{X} \to \mathcal{F}$,

$$\mathbf{x} \to \rho\left(\mathbf{x}\right) = |\phi\left(\mathbf{x}\right)\rangle\langle\phi\left(\mathbf{x}\right)| \,. \tag{2}$$

Usually, the feature map is given in terms of a parameterized unitary $U(\mathbf{x})$ applied to a fixed reference state, e.g. $|\phi(\mathbf{x})\rangle = U(\mathbf{x})|0\rangle$. The kernel used in the classical SVC optimization is then calculated as the overlap between two encoded quantum states,

$$K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \mathrm{Tr}\left[\rho\left(\mathbf{x}_i\right)\rho\left(\mathbf{x}_j\right)\right] = \left|\langle\phi\left(\mathbf{x}_i\right)|\phi\left(\mathbf{x}_j\right)\rangle\right|^2 . \tag{3}$$

A kernel of this form can be evaluated on a quantum device, and could bring an advantage over classical SVCs provided that the quantum feature map is hard to simulate classically [30]. A carefully chosen quantum feature map allows to solve specific classification problems more efficiently than with any know classical counterpart [52] and it is assumed that quantum kernels will outperform classical kernels if there exists a

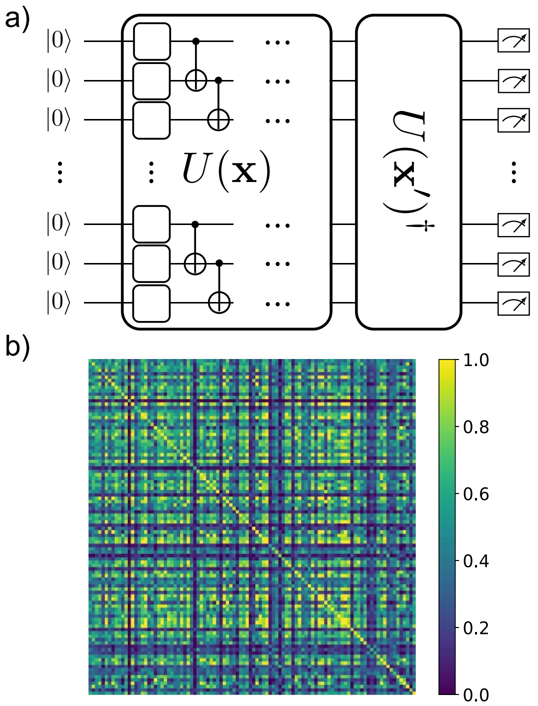

**Figure 3.** (a) Quantum circuit used by the quantum support vector classifier (QSVC) to measure the overlap between two encoded quantum states. The feature vector $\boldsymbol{x}$ is encoded into a quantum state through the parameterized unitary $U(\boldsymbol{x})$. (b) Kernel matrix for the classification between SM events and artificial anomalies calculated on the IBM quantum processor *ibm_cairo* using 6 features (corresponding to 6 qubits).

bias in the data that is hard to simulate classically [53, 54]. The quantum circuit used to evaluate the kernel in equation (3) is schematically shown in figure 3(a). The overlap between the encoded quantum states is given by the probability of measuring the all-zero state at the end of the circuit. For the encoding of the classical data into a quantum state, there exist several different strategies, e.g. the basis encoding, the amplitude encoding, or, generally, the encoding via a quantum feature map [31]. The embedding we apply is inspired by the quantum feature map introduced in [30], which is conjectured to be hard to simulate classically. The feature map is commonly applied as a "general purpose" embedding, and it is not expected that it provides the best possible results. However, the tailoring of the feature map to the studied problem [55, 56] is beyond the scope of this work. The formal definition of the applied embedding unitary is

$$\mathcal{U}_{\Phi(\boldsymbol{x})} = U_{\Phi(\boldsymbol{x})} H^{\otimes n} U_{\Phi(\boldsymbol{x})} H^{\otimes n}\,, \tag{4}$$

where $H^{\otimes n}$ is a layer of Hadamard gates and

$$U_{\Phi(\boldsymbol{x})} = \exp\left( i \prod_{S\subseteq[n]} \phi_S(\boldsymbol{x}) \prod_{i\in S} Z_i \right)\,. \tag{5}$$

The encoded state is obtained by applying the embedding unitary to the all zeros state $|0\rangle^{\otimes n}$. We choose the following coefficients $\phi_S$ for the data encoding

$$\begin{aligned} \phi_{\{i\}} &= \gamma x_i\,, \\ \phi_{\{i,j\}} &= \frac{(\gamma x_i)\,(\gamma x_j)}{\gamma\pi} = \frac{\gamma}{\pi} x_i x_j\,, \end{aligned} \tag{6}$$

where we also introduced the hyperparameter $\gamma$, which is a constant scaling factor of the input features $\boldsymbol{x}$. The rescaling of the two-index coefficients $\phi_{\{i,j\}}$ ensures that all encoded rotation angles lie in the interval $[-\gamma\pi, \gamma\pi]$, if we assume that $x_i \in [-\pi, \pi]$.

An example of a kernel matrix obtained with this quantum feature map is shown in figure 3(b). The kernel values were estimated with the IBM Quantum processor *ibm_cairo* using 6 input features (corresponding to 6 qubits).

## 4. Results

### 4.1. Datasets

In the studied anomaly detection problem, background events are represented by simulated samples of the SM processes typically observed at 13 TeV [57, 58]. The included processes are (with relative occurrences): W bosons decaying into a charged and a neutral lepton (59.2%), multi-jet production from QCD processes (33.8%), Z boson decaying into two charged leptons (6.7%), and $t\bar{t}$ production (0.3%). To test the proposed anomaly detection strategy, we use two different types of processes as signal events. The first one is the Higgs boson, which has not been included in the SM dataset, and can therefore be interpreted as a BSM particle. The dataset for the Higgs boson consists of simulated high-mass Higgs particle produced via vector boson fusion [59]. Additionally, we also use a sample of simulated Randall–Sundrum Gravitons [60] decaying to two Z bosons, forcing each Z to decay to a lepton pair [61]. These two anomaly datasets serve as proxies for new-physics searches targeted by the ATLAS and CMS experiments at the LHC.

All events are obtained with Monte Carlo simulations (PYTHIA8 [62]+DELPHES [63] for SM, PYTHIA6 [64] for Higgs and Graviton) and are described by 23 high-level features representing simulated measurement results as obtained with e.g. the CMS detector. A detail list of the features and their description in given in appendix A.

Using the scrambling process introduced in section 2 we create three different datasets with artificial anomalies, each with different scrambling intensity, denoted as *low*, *medium* and *high* scrambling. The corresponding scrambling factors $\lambda_f$ are listed in appendix B.4. Here, we limit ourselves to the scrambling of a subset of 17 out of the 23 high level features, allowing us to satisfy physical constraints like energy conservation.

Following this methodology, we create training, validation and testing datasets for the binary classification. Unless stated otherwise, the classification datasets contain 1000 samples per class, where the background is labelled as the negative class, and the signals as the positive class. We create ten training datasets for each scrambling strength, and one validation dataset, all consisting of a combination between SM data and artificial anomalies. Further, we prepare two test datasets, one consisting of a combination of SM and Higgs data, and the other of a combination of SM and Graviton data. The datasets and their composition are schematically shown in the first two panels of figure 1.

### 4.2. Numerical experiments

The training of a classifier consists of two main steps, the preprocessing of the data samples and the training itself. For the preprocessing we consider the following steps: standardization applied of the input features, feature selection or feature extraction to reduce the number of features used in the classification, and normalization applied to balance the importance between the feature before forwarding them to the classifier. In the classical case, the kernel matrix for the training of the classifier is calculated with equation (1). To calculate the kernel matrix with a quantum computer, we use a parameterized quantum circuit to encode the classical features in a quantum state, and obtain the kernel values by calculating the fidelity between two encoded data samples (equation (3)). Similarly to the classical case, and as proposed in [65, 66], we introduce a hyperparameter $\gamma$ in the quantum feature map to control its resolution in the Hilbert space. Figure 6 in the appendix shows an overview of the training workflow, and in appendix C we present a detailed hyperparameter optimization for the QSVC model.

For the simulations we have fixed the steps of the training workflow in the following way. We use no standardization transformation prior to the feature extraction with PCA, since the tested standardization algorithms all lead to a reduced performance (see appendix C.1). In the normalization step we scale all features to the interval $[-\pi, \pi]$. For the quantum classifier, we encode the data with a feature map similar to the one introduced in [30] with a hyperparameter fixed to $\gamma = 0.5$. A detailed description of the applied feature map and a justification for fixing its hyperparameter are given in section 3.2 and appendix C.4, respectively. In the case of the classical SVC, the hyperparameter of the RBF kernel (equation (1)) is optimized for each classification individually, by selecting the one that achieves the best validation score on an independent validation dataset.

Using this workflow, we train (Q)SVCs for each scrambling strength and training dataset, and evaluate/validate them on the corresponding validation/test datasets. Since the events in the test dataset are not visible to the classifier during its training, this methodology can be used to verify that the proposed anomaly detection approach generalizes to unknown BSM events. All quantum computations were done with Qiskit [67]. The results are shown in figure 4.

In figure 4, we compare the average AUC score for identifying artificial anomalies (blue lines), Higgs events (red lines) and Graviton events (green lines) with a quantum (solid lines) and a classical (dashed lines) classifier at different scrambling strength and for different number of features. The average is taken

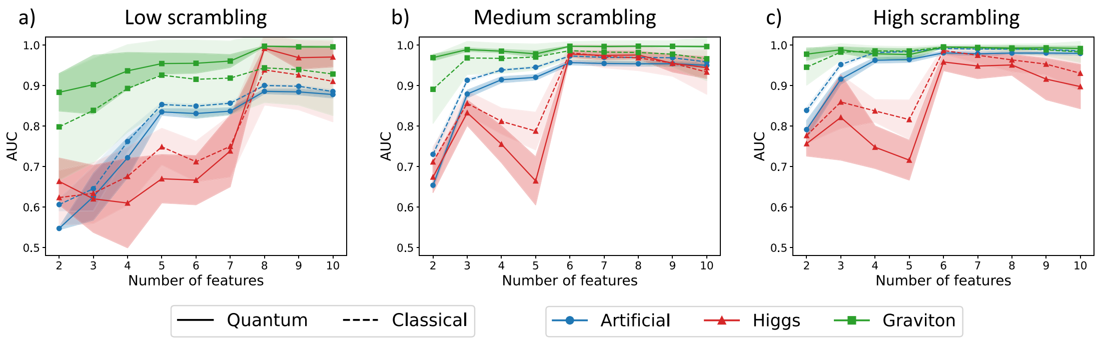

**Figure 4.** AUC for classification of BSM events among SM events with quantum and classical SVCs for different scrambling strength and different number of features. (a) AUC for identifying artificial anomaly (blue), Higgs (red) or Graviton (green) events among SM events for a QSVC (solid lines) and a classical SVC (dashed lines). The classifiers were trained on artificial anomalies generated with the low scrambling strength. (b) and (c) Equivalent results, but for medium and high scrambling intensity, respectively.

**Table 1.** Highest validation AUC for classification between artificial anomalies and SM events with quantum and classical SVCs for different scrambling strengths (fourth column). The third column lists the number of features for which this validation AUC is achieved. The last two columns hold the corresponding detection AUC of the Higgs and Graviton events.

| Algorithm | Scrambling strength | Number of features | Anomalies AUC | Higgs AUC | Graviton AUC |
|---|---|---|---|---|---|
| SVC | low | 8 | 0.900 ± 0.003 | 0.939 ± 0.087 | 0.944 ± 0.086 |
| | medium | 6 | 0.971 ± 0.003 | 0.981 ± 0.021 | 0.986 ± 0.022 |
| | high | 6 | 0.992 ± 0.001 | 0.987 ± 0.010 | 0.995 ± 0.006 |
| QSVC | low | 8 | 0.886 ± 0.006 | 0.992 ± 0.003 | 0.997 ± 0.001 |
| | medium | 6 | 0.957 ± 0.006 | 0.978 ± 0.008 | 0.997 ± 0.001 |
| | high | 6 | 0.981 ± 0.004 | 0.958 ± 0.021 | 0.996 ± 0.002 |
| QSVC[a] | medium | 6 | 0.870 ± 0.052 | | |
| sSVC-H[b] | | 6 | | 0.9997 ± 0.0001 | 0.989 ± 0.005 |
| sSVC-G[c] | | 6 | | 0.9995 ± 0.0003 | 0.99999 ± 0.00001 |

[a] Quantum kernel estimation executed on IBM Quantum device *ibm_cairo*.

[b] Reference calculation: supervised training for SM/Higgs classification.

[c] Reference calculation: supervised training for SM/Graviton classification.

over the AUC scores of the classifiers resulting from the 10 training dataset instances. The uncertainty bands are given by the standard deviation of the AUC scores.

A few observations are in order: confirming our expectations, the validation score for identifying the artificial anomalies (blue lines) increases with the number of features, and the scrambling strength. We can also confidently conclude that the proposed anomaly detection strategy is successful, since it generalizes to Higgs and Graviton events, even though the classifiers were trained to identify the artificial anomalies. For the low scrambling strength, using 8 features leads to the highest identification AUC of artificial anomalies and the highest detection AUC of the Higgs and Graviton events. For the medium and high scrambling strengths the highest AUC values are reached for 6 features. A possible reason why the number of features with the highest AUC changes for the different scrambling factors is that inducing more extreme anomalies produces, on average, events which are easier to distinguish from SM ones. In general, this leads to both an increased validation and detection AUC for the same number of features. In other words, for an increased scrambling strength, a lower number of features is required to achieve the same performance. However, the highest detection AUC value is reached for the low scrambling.

In most cases the classifiers are even better at detecting the Higgs and the Graviton events than identifying the artificial ones. The numerical values of the highest validation AUC and the corresponding detection AUC are listed in table 1.

The performance of the classical and QSVCs is very similar. Looking only at the classification between SM and artificial anomalies (blue curves), the classical SVC outperforms the quantum SVC in all cases. However, the gap gets smaller for increasing number of features, and increasing scrambling strength. Focusing on the detection we observe the contrary behaviour. For low scrambling strength (where the detection AUC is the highest) the QSVC is better at detecting the Higgs and Graviton events. However, increasing the scrambling strength closes the gap between the classical and QSVC when detecting Graviton events, and the order gets reversed for the detection of Higgs events.

Overall, the results suggest that, although a QSVC can be better than a classical one in terms of detection ability, in general the two methods exhibit essentially comparable performances.

A possible explanation for the improved detection score of the QSVC for Higgs and Graviton events—which are not explicitly present in the dataset of anomalies—could lie in lower overfitting on the classification task and, hence, a better generalization power. However, we have observed that introducing a bias to the classical SVC by fixing the hyperparameter of the classical kernel to $\gamma = 0.5$, only leads to a minimal drop in its classification accuracy, but to a significant gain in the detection accuracy. The corresponding figures are shown in appendix D.

For completeness, we also compare the obtained AUC scores in table 1 with SVCs that are trained in a purely supervised setting. The sSVC-H classifier (see table 1) is trained to distinguish Higgs events from SM events, while the sSVC-G classifier does the same for the Graviton events. For this evaluation, we additionally create ten training datasets and one validation dataset for both types of BSM events, each having 1000 randomly selected samples per class. To validate the classifiers, we use the same test datasets as above. As expected in a supervised setting, the SVC trained on SM/Higgs data achieves very high AUC for the detection of the Higgs, and a slightly lower AUC for detecting Graviton events. The same is also true the other way around when the classifier was trained on SM/Graviton data (see last row in table 1). However, the specific generalizability to a different type of events is highly dataset dependent and has to be evaluated on a case-by-case basis. Nevertheless, the proposed anomaly detection strategy applying the scrambling results in comparable AUC scores, while having no physically motivated knowledge about the anomalies at the stage of the training.

We find similar results for the detection of other types of BSM processes in appendix G.

### 4.3. Hardware experiments

For all the results presented in the previous sections, the quantum kernel values were computed via the simulation of a perfect quantum computer, without errors due to finite measurement statistics or hardware noise. While the former can be included in numerical simulations without much effort [68], the latter is harder to capture, and it is therefore important to benchmark quantum algorithms directly on existing quantum processors.

We perform all hardware calculations on the IBM Quantum superconducting device *ibm_cairo*, using the same experimental setup as above (same training workflow and hyperparameters). We use $10^4$ repetitions (shots) for the estimation of the kernel values, and we apply a depolarization error mitigation method to the obtained kernel matrices (see appendix F.1). To ease the requirements on the quantum device, we only use 50 events per class and only look at the classification between SM and artificial events. Additionally, we also just consider the 6-feature case for medium scrambling strength. In appendix F.2, we have included an estimation of the required quantum resources.

One instance of a kernel matrix calculated with the quantum device is displayed in figure 3(b). In figure 5, we also report the average ROC-AUC curve of the QSVC trained on noisy kernel matrices. Performances are compared with the simulated QSVC and the classical SVC. The test AUC of the QSVC is listed in table 1.

The ROC-AUC curve evaluated on hardware lies consistently below the curves obtained with the classical SVC and the simulated QSVC (see figure 5). The most probable reasons for this discrepancy are the different sources of error inherent to quantum computations. These include shot noise and various sources of hardware errors, e.g. due to depolarization, decoherence, read out missassignment, or leakage. To improve the results with respect to the measurement noise, we could apply a larger number of shots in the estimation of the kernel values. Similarly, to reduce the effects of the hardware noise, we could either use a quantum processor with better noise characteristics, or apply more elaborate error mitigation strategies. However, a comparison between different quantum processors or different error mitigation techniques is beyond the scope of this work.

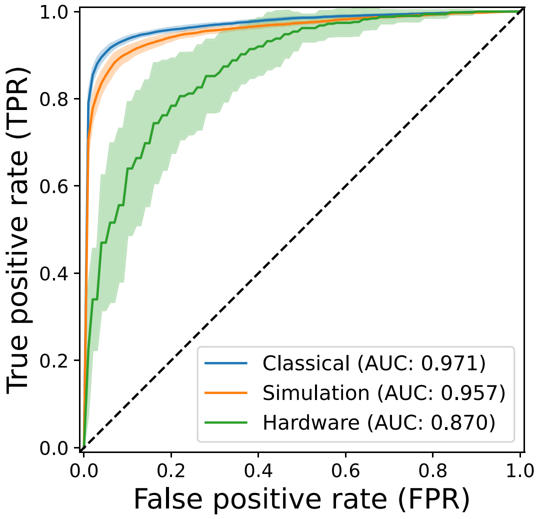

**Figure 5.** ROC-AUC curve of the classification between SM events and artificial anomalies. The kernel matrices for the classification were provided by a classical kernel function (blue), a simulated quantum kernel (orange), and a quantum kernel estimated using the quantum device *ibm_cairo* (green).

## 5. Conclusions

In this work, we proposed and successfully tested a supervised learning strategy for anomaly detection. Instead of generating anomalies provided by a BSM theory we randomly generate them with a scrambling process that is based on an initial dataset containing SM processes. We verify the feasibility of the proposed anomaly detection strategy by identifying HEP processes that were not included in the SM dataset during the training of a classifier. The success of the strategy can be confirmed with both classical and QSVCs approaches.

Currently, the scrambling process generates events which have a high overlap with the initial data samples, and reducing this overlap could improve the classification between SM and artificial events. However, this does not guarantee that the detection of unknown events will also improve. A possible strategy to reduce the overlap would be to use a scrambling distribution that is different from the Gaussian distribution. The only constraints that the generated events have to respect are physical conservation laws and requirements by the detection. Therefore, it would in principle be possible to construct a scrambling process that generates events with less overlap to the initial data samples, or even generate events that lie in a desired region of the 'event space'. The proposed scrambling strategy can also be applied for anomaly detection approaches applying different ML models, e.g. neural networks, which are typically employed in the context of HEP when the number of events is very large (up to millions or even billions). Additionally, we would like to stress that the proposed scrambling strategy is not limited to HEP datasets, and could in principle be applied for any anomaly detection problem.

While our results establish empirically a successful application of quantum kernel methods to a HEP anomaly detection task, we could not yet observe a generalized promise for quantum advantage. However, it is not possible to rule out individual problem instances where a quantum classifier could outperform a classical one, e.g. for different scrambling distributions, or a higher number of features. In fact, a similar study on anomaly detection in HEP has found evidence for scenarios where a quantum model can outperform the classical counterpart [69]. Such results emerge from a combination of unsupervised learning approaches applied to a dataset where the collision events are described by the 4-momenta of the involved particles, hence using a representation closer to the physical 'raw data' compared to the dataset used in this work. Generally, there is mounting evidence that quantum advantage on classical datasets can only be strictly guaranteed when specific structure is present [52, 53, 70, 71]. An analysis of our classification problem via the methods proposed in [54] is also not conclusive (see appendix E).

As the data constituting the target of our work originates from a quantum HEP process, the idea of using a quantum technique for its classification seems rather natural. However, the features currently used to describe the processes, collected from detector measurements, are fully classical. This loss of 'quantumness' could in fact represent an important limitation to the use of more sophisticated QML techniques for the analysis and classification of quantum states [32, 72–74]. We therefore believe that a different setup bypassing the extraction of classical features could be a promising road towards quantum advantage also in the context of HEP. While for collision events at LHC such a setting is currently not possible, there already exist experiments at CERN where quantum sensors are studied for information extraction [75]. In the future, it could certainly be interesting to couple a quantum processor to quantum sensors embedded in detectors, hence enabling direct manipulation and classification of quantum amplitudes produced in an experiment.

## Data availability statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

## Acknowledgments

J S and I T acknowledge financial support from the Swiss National Science Foundation (SNF) through the Grant No. 200021-179312. V B is supported by an ETH Research Grant (Grant No. ETH C-04 21-2). M G and S V are supported by the CERN Quantum Technology Initiative. M P and E P are supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant Agreement No. 772369).

We acknowledge the use of IBM Quantum services for this work. IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. The current list of IBM trademarks is available at https://www.ibm.com/legal/copytrade.

**Table 2.** High-level features used as description of the events in the HEP datasets [57, 58]. The additional column indicates if the feature is considered in the scrambling process. The abbreviation PF stands for Particle Flow, and is related to the event reconstruction algorithm used to process the raw collision data. The output of the algorithm are the so-called PF candidates [58].

| Feature | Description | Scrambled? |
|---|---|---|
| $H_T$ | The scalar sum of the transverse momenta $p_T$ of all jets having $p_T > 30$ GeV and $\lvert\eta\rvert < 2.4$. | x |
| $M_J$ | The invariant mass of all jets entering the $H_T$ sum. | |
| $N_J$ | The number of jets entering the $H_T$ sum. | x |
| $N_B$ | The number of jets identified as originating from a $b$ quark. | x |
| $p^{\mu}_{T,\mathrm{TOT}}$ | The vector sum of the $\boldsymbol{p}_T$ of all PF muons in the event having $p_T > 0.5$ GeV. | x |
| $M_\mu$ | The combined invariant mass of all muons entering the sum in $p^{\mu}_{T,\mathrm{TOT}}$. | |
| $N_\mu$ | The number of muons entering the sum in $p^{\mu}_{T,\mathrm{TOT}}$. | x |
| $p^{e}_{T,\mathrm{TOT}}$ | The vector sum of the $\boldsymbol{p}_T$ of all PF electrons in the event having $p_T > 0.5$ GeV. | x |
| $M_e$ | The combined invariant mass of all electrons entering the sum in $p^{e}_{T,\mathrm{TOT}}$. | |
| $N_e$ | The number of electrons entering the sum in $p^{e}_{T,\mathrm{TOT}}$. | x |
| $N_{neu}$ | The number of all neutral hadron PF-candidates. | x |
| $N_{ch}$ | The number of all charged hadron PF-candidates. | x |
| $N_\gamma$ | The number of all photon PF-candidates. | x |
| $p^l_T$ | The transverse momentum of the highest $p_T$ lepton in the event. | x |
| $\eta_l$ | The lepton pseudorapidity. | x |
| $q_l$ | The lepton charge (either $-1$ or $+1$). | |
| $\mathrm{Iso}^l_{ch}$ | The lepton isolation related to all other charged hadron PF-candidates. | x |
| $\mathrm{Iso}^l_{neu}$ | The lepton isolation related to all neutral hadron PF-candidates. | x |
| $\mathrm{Iso}^l_\gamma$ | The lepton isolation related to all photons. | x |
| $\mathrm{MET}_{\parallel}$ | The parallel component of the missing transverse energy with respect to the lepton. | x |
| $\mathrm{MET}_{\perp}$ | The orthogonal component of the missing transverse energy with respect to the lepton. | x |
| $M_T$ | The combined transverse mass of the lepton and the missing transverse energy system. | |
| IsEle | A flat set to 1 if the lepton is an electron, 0 if it is a muon. | |

## Appendix A. HEP datasets

The events in the HEP datasets are described by 23 high-level features [57, 58]. The features are listed in table 2, including a column which indicates if the feature is considered in the scrambling process.

## Appendix B. Scrambling

With the scrambling process we randomly generate data samples that do not conform with the SM, without relying on any BSM theory. The scrambling is done by replacing the feature values in the original dataset with a new value chosen according to a Gaussian distribution $\mathcal{N}(\mu,\sigma)$, where $\mu = f$ is the initial value of the feature and $\sigma$ is the standard deviation. Depending on the feature we apply different strategies on how to choose the standard deviation, and also different post-processing methods to comply with conservation laws and detector limitations. Therefore, the features are divided into four categories: momenta, isolations, jets and particle numbers, each with their own scrambling strategy.

### B.1. Momenta

There are two different types of transverse momenta we can scramble, the transverse momenta related to the leptons and the transverse momenta related to the jets. In theory, due to conservation of momenta, these two types of momenta would be related. However, for the jets we only have the scalar sum of the transverse momenta as a feature in the dataset ($H_T$). We therefore scramble the $H_T$ independently of the transverse momenta of the leptons, and assume that the change in $H_T$ can be absorbed in an appropriate change in the directions of the transverse momenta of the jets. The $H_T$ is updated according to the following equation,

$$(H_T)' \sim \lvert\mathcal{N}(H_T, \lambda_{H_T} H_T)\rvert \, . \tag{B1}$$

For the transverse momenta related to the leptons we only scramble the transverse momentum $\boldsymbol{p}^l_T$ of the chosen lepton $l$ with the highest transverse momentum. The features describing $\boldsymbol{p}^l_T$, are the transverse momentum $p^l_T$ and the pseudorapidity $\eta^l$, where $l \in \{e, \mu\}$ is either an electron or a muon. Changing these two features has an effect on other features, which have to be adapted accordingly. Specifically, the parallel and orthogonal component of the missing transverse momentum ($\mathrm{MET}_{\|}$, $\mathrm{MET}_{\perp}$), and the sum of the transverse momenta of the leptons $p^{l}_{T,\mathrm{TOT}}$. The total momentum of the lepton $l$ can be written as

$$\boldsymbol{p}^{l} = |\boldsymbol{p}^{l}| \begin{pmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ \cos\theta \end{pmatrix} = \begin{pmatrix} p^{l}_{T}\cos\phi \\ p^{l}_{T}\sin\phi \\ p^{l}_{L} \end{pmatrix} , \tag{B2}$$

where the coordinate system is chosen such that the $z$-axis is along the beam line, and the transverse momenta lies in the $x$–$y$ plane. The polar angle $\theta$ is the angle between $\boldsymbol{p}$ and the $z$-axis, the azimuthal angle $\phi$ is the angle between $x$-axis and the transverse momentum, $p^{l}_{T}$ is the transverse momentum, and $p^{l}_{L}$ is the longitudinal momentum. The pseudorapidity $\eta^{l}$ is defined as a function of the polar angle, $\eta^{l} = -\log(\tan\theta/2)$. As an initial point we set $\phi = 0$ as it is possible to align the $x$-axis with the transverse momentum $\boldsymbol{p}^{l}_{T}$. In this case, the momentum $\boldsymbol{p}^{l}$ is fully determined by the transverse momentum $p^{l}_{T}$ and the pseudorapidity $\eta^{l}$,

$$\boldsymbol{p}^{l} = p^{l}_{T} \begin{pmatrix} 1 \\ 0 \\ \frac{1}{\tan(\theta(\eta^{l}))} \end{pmatrix} , \tag{B3}$$

where the polar angle $\theta$ is determined by the pseudorapidity $\eta^{l}$. To sample a new momentum vector for lepton $l$ we randomly generate values for the following three quantities,

$$\begin{aligned} \left(p^{l}_{T}\right)' &\sim \mathcal{N}\left(p^{l}_{T}, \lambda_{p} p^{l}_{T}\right) , \\ \phi' &\sim \mathcal{N}\left(0, \lambda_{\phi}\right) , \\ \left(\eta^{l}\right)' &\sim \mathcal{N}\left(\eta^{l}, \lambda_{\eta}\sigma_{\eta}\right) . \end{aligned} \tag{B4}$$

The sampled transverse momentum $p^{l}_{T}$ has to fulfil the requirement $(p^{l}_{T})' > 23$ GeV, and the sampling is therefore repeated until this constraint is respected.

Assigning a new value to the momentum of lepton $l$ has an effect on other quantities, which we have to adapt accordingly:

- *Missing transverse energy*: The missing transverse energy MET is specified by a vector with two components

$$\begin{pmatrix} \mathrm{MET}_{\|} \\ \mathrm{MET}_{\perp} \end{pmatrix} = \begin{pmatrix} -\sum_{q} p^{q}_{T,\|} \\ -\sum_{q} p^{q}_{T,\perp} \end{pmatrix} , \tag{B5}$$

  where the parallel and perpendicular direction are with respect to lepton $l$. Sampling a new transverse momentum $\boldsymbol{p}^{l}_{T}$ for lepton $l$ also changes the definition of the parallel and perpendicular direction. To update the components of the MET, we therefore first add $p^{l}_{T}$ to $\mathrm{MET}_{\|}$, rotate the MET by the sampled azimuthal angle $\phi'$ and add $(p^{l}_{T})'$ to the new parallel component, which results in the following equations,

$$\begin{aligned} \left(\mathrm{MET}_{\|}\right)' &= \left(\mathrm{MET}_{\|} + p^{l}_{T}\right)\cos\phi' + \mathrm{MET}_{\perp}\sin\phi' - \left(p^{l}_{T}\right)' , \\ \left(\mathrm{MET}_{\perp}\right)' &= -\left(\mathrm{MET}_{\|} + p^{l}_{T}\right)\sin\phi' + \mathrm{MET}_{\perp}\cos\phi' . \end{aligned} \tag{B6}$$

- *Sum of transverse momenta of leptons*: In the SM dataset, the vector sum of the transverse momenta of all leptons is only characterized by its absolute value. Therefore, we miss the directional information, required for an accurate compensation of the change in $p^{l}_{T}$, and we update the sum of transverse momenta in the following way

$$\left(p^{l}_{T,\mathrm{TOT}}\right)' = p^{l}_{T,\mathrm{TOT}} - p^{l}_{T} + \left(p^{l}_{T}\right)' . \tag{B7}$$

  This does not consider the direction of the momenta, however, usually $\boldsymbol{p}^{l}_{T}$ is the dominant component of $\boldsymbol{p}^{l}_{T,\mathrm{TOT}}$ and equation (B7) is a good approximation of the actual change in $p^{l}_{T,\mathrm{TOT}}$.

**Table 3.** Factors determining the scrambling strength for low/medium/high scrambling.

| Factor | Low | Medium | High |
|---|---|---|---|
| $\lambda_p$ | 0.5 | 1.0 | 2.0 |
| $\lambda_\phi$ | 0.05 | 0.1 | 0.2 |
| $\lambda_\eta$ | 0.05 | 0.1 | 0.2 |
| $\lambda_{H_T}$ | 0.5 | 1.0 | 2.0 |
| $\lambda_{\text{Iso}}$ | 0.25 | 0.5 | 1.0 |
| $\lambda_J$ | 0.5 | 1.0 | 2.0 |
| $\lambda_b$ | 0.5 | 1.0 | 2.0 |
| $\lambda_{N,\text{charged}}$ | 0.25 | 0.5 | 1.0 |
| $\lambda_{N,\text{neutral}}$ | 0.25 | 0.5 | 1.0 |
| $\lambda_{N,\text{photons}}$ | 0.25 | 0.5 | 1.0 |
| $\lambda_{N,\text{electrons}}$ | 0.5 | 1.0 | 2.0 |
| $\lambda_{N,\text{muons}}$ | 0.5 | 1.0 | 2.0 |

#### *B.1.1. Isolations*

The isolation Iso of the leptons, photons and neutral atoms is randomly assigned a new value according to

$$\text{Iso}' \sim \left|\mathcal{N}\left(\text{Iso}, \lambda_{\text{Iso}}\,\sigma_{\text{Iso}}\right)\right|. \tag{B8}$$

The absolute value is taken, because the isolation is always positive. Additionally, as a requirement of the reconstruction process of an CMS event, the isolation has to be smaller than 0.45. Therefore, the sampling is repeated until this constraint is respected.

### B.2. Jets

The total number of jets $N_J$ and the number of jets involving a b-quark $N_B$ are randomly assigned a new value according to

$$N'_{J,B} \sim \left|\left[\mathcal{N}\left(N_{J,B}, \lambda_{J,B} N_{J,B}\right)\right]\right|. \tag{B9}$$

The number of jets is a non-negative integer, and therefore we round to the nearest integer (denoted by $[\cdot]$) and take the absolute value. Additionally, the number of b-jets cannot exceed the total number of jets. Therefore, the sampling is repeated until $N'_b \leqslant N'_J$.

### B.3. Particle number

The particle number $N$ for the neutral and charged hadrons, photons, electrons and muons are assigned a new value according to

$$N' \sim \left|\left[\mathcal{N}\left(N, \lambda_N N\right)\right]\right|, \tag{B10}$$

where the standard deviation is proportional to the original value. The particle number is a non-negative integer, and therefore we round to the nearest integer and take the absolute value.

### B.4. Scrambling factors

We create three different anomaly datasets each with a different strength of the scrambling. The strength of the scrambling is controlled by the scrambling factors introduced in section 2. The three scrambling strength are denoted as low, medium and high. The specific values of the scrambling factors are listed in table 3.

## Appendix C. Evaluation of training workflow

In the following we present a detailed evaluation of the different steps in the training workflow of a QSVC. The steps are shown as a flowchart in figure 6. A similar study can also be done for the classical SVC. However, some steps of the preprocessing of the data for the classical SVC coincide with the steps presented below, therefore we will only focus on the QSVC in the following.

We are going to evaluate the different options at each step of the training workflow. To reduce the simulation times, all classifications are done with datasets containing 50 samples per class. Additionally, we only consider the scrambled dataset with medium scrambling strength. In all evaluations, a QSVC is trained on 10 different training datasets and validated on one validation dataset.

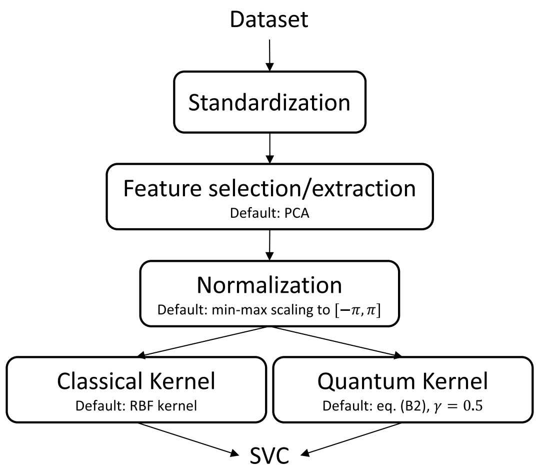
Dataset
Standardization
Feature selection/extraction
Default: PCA
Normalization
Default: min-max scaling to $[-\pi, \pi]$
Classical Kernel
Default: RBF kernel
Quantum Kernel
Default: eq. (B2), $\gamma = 0.5$
SVC

**Figure 6.** Flowchart of the pipeline used for the training of a quantum or classical SVC.

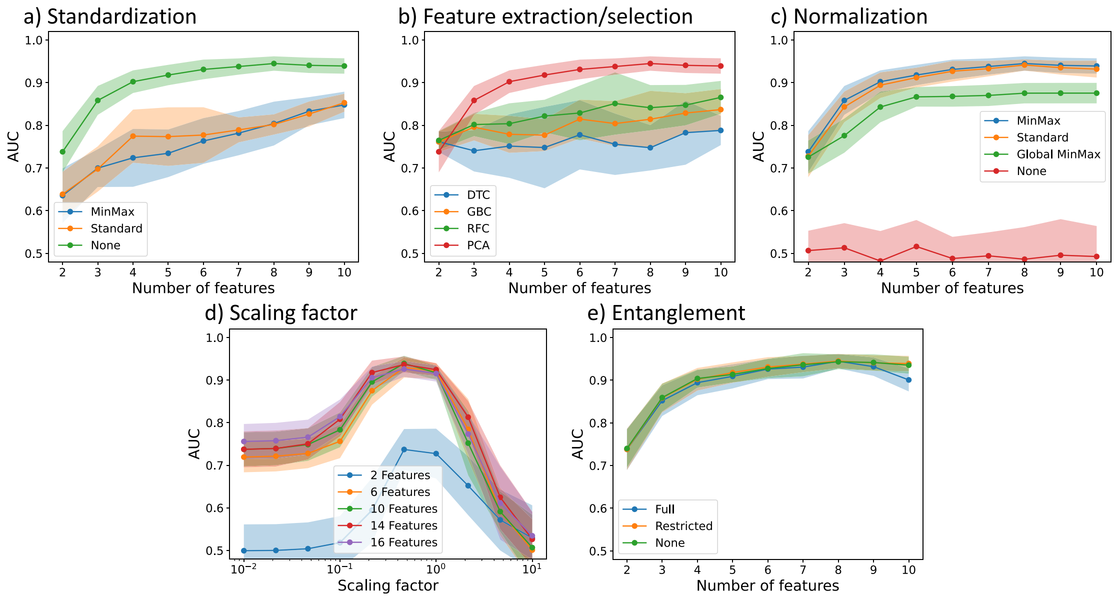

**Figure 7.** Hyperparameter analysis for the training of a QSVC. Comparing different standardization transformations (a), feature selection and feature extraction methods (b), and normalization transformations (c) for an increasing number of features, searching for the optimal scaling factor (d), and comparing different entanglement strategies of the feature map (e). More detailed information about the specific algorithms and evaluation settings can be found in the text of the corresponding sections.

### C.1. Standardization

The standardization is applied to a dataset prior to the feature selection/extraction, and has the purpose to balance the importance of the different features. We consider two different standardization transformations, the transformation to mean zero and unit variance, and the transformation to the fixed interval $[-1, 1]$, denoted as standard scaling and min-max scaling, respectively. As a reference, we also consider the training without any standardization transformation. For all other steps of the training workflow we choose the defaults given in figure 6. The results are shown in figure 7(a).

As expected, the AUC increases with the number of features for all considered standardization options. Surprisingly, however, we get the best performance when we use no standardization transformation (green curve). Therefore, we will use no standardization transformation prior to the feature selection/extraction in the following experiments.

### C.2. Feature selection/extraction

The principal component analysis (PCA) algorithm is a widely used algorithm for feature extraction. For comparison, we also consider three feature selection strategies, the decision tree classifier, the gradient boosting classifier, and the random forest classifier. We use no standardization transformation prior to the feature selection/extraction, and for the remaining steps in the training workflow we choose the defaults given in figure 6. The results are shown in figure 7(b).

All feature selection strategies have about the same performance. However, all are out-performed by the feature extraction with PCA. Therefore, we will use PCA to extract the features in the following experiments.

### C.3. Normalization

The normalization transformation is applied after the feature selection, with a similar purpose as the standardization transformation. We want to balance the importance of the features before inputting them to the classification algorithm. We again consider the standard scaling and min-max scaling to the interval $[-\pi,\pi]$. Additionally, we also consider a global min-max scaling, where the features are transformed to a fixed interval with a joint transformation, instead of individual transformations for each feature. The intuition behind the global min-max scaling is that the features retain their relative structure obtained through the feature selection/extraction algorithm. As a reference, we again consider the training without any normalization transformation. We use no standardization transformation, and PCA for the feature extraction. For the remaining steps of the training workflow we take the defaults given in figure 6. The results are shown in figure 7(c).

Using a normalization transformation is clearly beneficial. Without the normalization the trained classifier has the same performance as a random classifier (AUC is 0.5). All other normalization transformations have a similar performance, with the standard and min-max scaling having the highest AUC values. In the following we will use the min-max scaling to the interval $[-\pi,\pi]$, in order to have some control over the range of values the features will take after the normalization.

### C.4. Scaling factor

The scaling factor, as introduced in [65, 66], is interpreted as a hyperparameter of the applied quantum kernel. This parameter should therefore be optimized. Here, we show an example of such an optimization, and how the optimal scaling factor is chosen. For the preprocessing we apply the optimal steps found in the previous sections, and for the feature map we use the default given in figure 6. The results are shown in figure 7(d).

The figure shows a sweep over the scaling factor, for different number of features used for the classification. Clearly, there is some region of scaling factor values, where the resulting classifier has the best performance. Additionally, this region is similar for different number of features. Of the considered scaling factor values, $\gamma = 0.46$ results in the highest performance for all considered number of features. For this learning task and this specific choice of feature map a scaling factor of $\gamma \approx 0.5$ therefore leads to a good performance. This values of the scaling factor ensures that all angles entering the feature map will effectively be in the interval $[-\frac{\pi}{2},\frac{\pi}{2}]$.

For a different dataset and/or a different feature map the optimal scaling factor may be different.

### C.5. Feature map

We consider three different options for the entanglement layout of the feature map introduced in section 3.2. We have *full* entanglement if we apply a two-qubit gates for each pair of qubits. We also consider *linear* entanglement is we apply a 'ladder' of two-qubit gates only for neighbouring qubits (without periodic boundary). For the $R_{ZZ}$ rotation applied in the feature map, the linear entanglement is equivalent to a more depth efficient layout, where at most two layers of $R_{ZZ}$ are required (even and odd connections between neighbouring qubits). We call this layout *restricted* entanglement, and apply it here instead of the linear entanglement, which is favourable when running the circuits on hardware. The last type of layout we consider is the *separable* encoding, without any entanglement between the qubits. For the preprocessing of the input data, we use the optimized steps presented in the previous sections. The results are shown in figure 7(e).

The figure shows the test AUC for an increasing number of features for different entangling strategies. Against our expectations, the entangling strategy does not have a significant influence on the AUC of the classification. Generally, adding entanglement between the qubits, is expected to reveal correlations among the features, and we therefore would expect an improvement in the AUC for increasing entanglement. Possible explanations why this can not be observe here, could be that the classification task is too easy, or during the preprocessing step (especially the PCA) all correlations between the extracted features are removed, and adding correlations in form of entanglement is therefore not improving the classification. It could also be, that the dataset does not fall into the class of problems where using a quantum model could lead to an advantage over classical models. We present a corresponding investigation in appendix E.

Although we do not see a practical advantage of using entanglement for the specific datasets used in the evaluation, we still use the restricted entanglement for the experiments in the main text in order to keep the anomaly detection scheme as general as possible, also across different datasets.

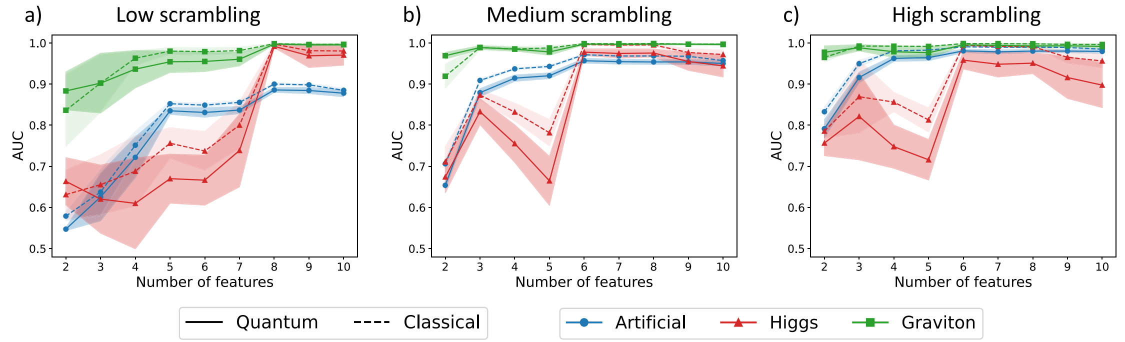

**Figure 8.** Validation AUC for classification of BSM events among SM events with quantum and classical SVCs for different scrambling strength and different number of features. Same experimental setup as in figure 4, except that the hyperparameter of the classical RBF kernel (equation (1)) is fixed to $\gamma = 0.5$. Note: only the curves of the classical SVC changed (dashed lines) compared to figure 4.

## Appendix D. Restricting classical model

In the main text the hyperparameter of the classical RBF kernel (equation (1)) was optimized for each combination of training and validation dataset. Here, the hyperparameter will be fixed to $\gamma = 0.5$. The resulting classification and detection score are shown in figure 8. Compared to the results in figure 4 the drop in the classification AUC is minimal, but the detection AUC is significantly improved. The classical SVC now also outperforms the QSVC in the detection. However, this improvement is expected to be very specific to the classification and detection problem at hand, and cannot be expected in general (especially when the real anomalies are unknown).

## Appendix E. 'Power of data' metrics

In [54] the authors introduce a strategy to check if a dataset falls into the class of problems where quantum ML models may perform better than classical models. In the following, we will follow this strategy to check the class of the studied HEP dataset. The metrics introduced in [54] were calculated using the code provided in the software package QuASK [76]. First, we evaluate the geometric difference $g$, defined as a similarity measure between two kernel matrices,

$$g_{\text{gen}} = \sqrt{\left\| \sqrt{K_1}\sqrt{K_2}\left(K_2 + \lambda I\right)^{-2}\sqrt{K_2}\sqrt{K_1} \right\|_\infty}, \tag{E1}$$

where $\lambda$ is a regularization parameter. The geometric difference $g_{\text{gen}}$ is also related to a training error, which is upper bounded by

$$g_{\text{tra}} = \lambda\sqrt{\left\| \sqrt{K_1}\left(K_2 + \lambda I\right)^{-2} K_1 \right\|_\infty}. \tag{E2}$$

Similarly to [54] we report the geometric difference $g_{\text{gen}}$ for a $\lambda$ such that the training error $g_{\text{tra}} \approx 0.0045$.

In figure 9(a) we show the geometric difference, calculated from kernel matrices obtained with the workflow described in the previous sections, for the classification between the SM and anomalies (resulting from the medium scrambling) for different numbers of features and different number of samples $N$ in the training and validation datasets. For each problem instance (specific number of features and samples), the geometric difference is calculated between the corresponding classical and quantum kernel matrix. In the figure, we additionally fit a function proportional to $\sqrt{N}$ to the geometric differences (dashed lines), specifically $g(N) = a \cdot \sqrt{N} + b$. The fit is done separately for each number of features. Visually, there is good agreement between the measured geometric differences and the dashed lines, which puts us in the regime where the geometric difference scales proportional to $\sqrt{N}$.

After finding the scaling proportional to $\sqrt{N}$, the next step in the assessment is to calculate the model complexity, defined as

$$s_K(N) = \sum_{i=1}^{N}\sum_{j=1}^{N}\left(\sqrt{K}\left(K + \lambda I\right)^{-2}\sqrt{K}\right)_{ij} y_i y_j, \tag{E3}$$

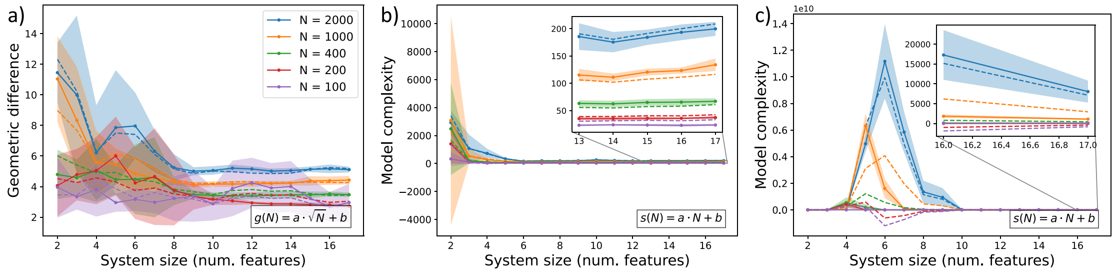

**Figure 9.** Metrics for the characterization of the 'hardness' of classifying SM and artificial anomalies. (a) Geometric difference between classical and quantum kernels for difference dataset sizes and increasing number of features. (b) Model complexity of classical kernels for different dataset sizes and increasing number of features. The inset shows the model complexities for 13 up 17 features. (c) Model complexity of quantum kernels for different dataset sizes and increasing number of features. The inset shows the model complexities only for 16 and 17 features.

where $y_{i,j}$ are the labels of the classification, $K$ is the classical or quantum kernel matrix, and $\lambda$ is again a regularization factor. Related to the model complexity we can define a training error,

$$t_K(N) = \lambda^2 \sum_{i=1}^{N} \sum_{j=1}^{N} \left( (K + \lambda I)^{-2} \right)_{ij} y_i y_j \,. \tag{E4}$$

We choose the regularization such that this training error and the model complexity are both minimized. The resulting model complexities for the classical kernel matrices and the quantum kernel matrices are shown in figures 9(b) and (c), respectively. In both cases, we fit a function proportional to $N$ to the model complexities (dashed lines), specifically $s(N) = a \cdot N + b$. Visually, we observe good agreement between the measured model complexities and the dashed lines especially for the higher number of samples, at least for the classical SVC. For the QSVC, the calculation of the model complexity is somehow not very meaningful. The model complexity is orders of magnitude bigger than in the classical case, and the values also seem to converge for larger numbers of features. However, they do not seem to be fully converged yet, and a fair assessment is not possible. Therefore, based on the geometric difference and the classical model complexity, we could end up either in the case with 'Potential quantum advantage' or the case where the problem is 'Likely hard to learn'.

From the results we obtained in the sections above, we would have expected to be in the case where both, classical and QSVCs, can learn well (either $g_{CQ} \ll \sqrt{N}$ or $s_C \ll N$). Therefore, the results obtained from the measured metrics is not conclusive and this test cannot be used to argue for (or against) quantum advantage in the studied classification problem.

## Appendix F. Hardware experiments

For the hardware experiments we used the IBM Quantum device *ibm_cairo*. The qubit layout of the device is shown in figure 10. To mitigate some of the hardware errors we apply a depolarization error mitigation strategy to the obtained kernel matrices.

### F.1. Depolarization error mitigation

Some of the errors occurring on the hardware can be modelled by a depolarization channel

$$\mathcal{D}_\lambda[\rho] = \lambda\rho + (1 - \lambda)\frac{1}{2^n}\,, \tag{F1}$$

where $\lambda$ is the survival probability of the original quantum state $\rho$, and $n$ is the number of qubits. To mitigate the depolarization error we can exploit that in the noiseless kernel matrix $K$ all diagonal entries are 1. Therefore, if one measures the diagonal entries in a noisy settings one can gather information about the device noise [77]. The survival probability $\lambda_i$ of the noisy kernel matrix element $K^*_{ii}$ is

$$\lambda_i = \sqrt{\frac{K^*_{ii} - 2^{-n}}{1 - 2^{-n}}}\,. \tag{F2}$$

The mitigated kernel values can then be obtained with

$$K_{ij} = \frac{K^*_{ij} - 2^{-n}\left(1 - \lambda_i \lambda_j\right)}{\lambda_i \lambda_j}\,. \tag{F3}$$

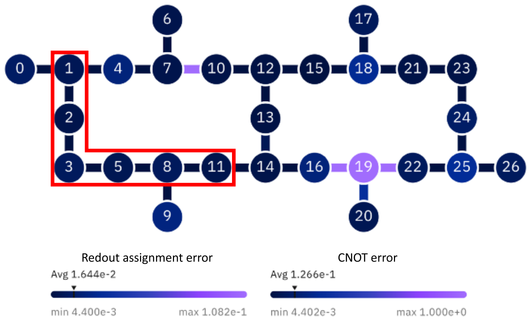

**Figure 10.** Qubit Layout of *ibm_auckland*. The highlighted qubits are the ones used in the experiments. The colour of the qubits denotes the readout assignment error, and the colour of the edges the CNOT error (colours obtained on October 04, 2022). The highlighted 6 qubits are the ones used to estimate the kernel values.

For the experiment in the main text, we assume that all survival probabilities $\lambda_i$ have the same value, which can be estimated with

$$\lambda = \frac{1}{N}\sum_{i=1}^{N}\lambda_i \,, \tag{F4}$$

where $N$ is the size of the symmetric training kernel matrix. This value can then also be used for the mitigation of the non-symmetric validation kernel matrix.

The applied error mitigation strategy has a low overhead in terms of the required quantum resources. Specifically, it requires $O(N)$ additional kernel value estimations for measuring the diagonal elements of the kernel matrix, compared to the $O(N^2)$ measurements to obtain the full kernel matrix. There also exist more elaborate error mitigation strategies, e.g. read out error mitigation [78, 79], pulse-efficient transpilation [80], zero-noise extrapolation [81–83] or probabilistic error cancellation [84]. However, a comparison between different error mitigation strategies is beyond the scope of this work.

### F.2. Resource estimates

In the following, we give rough estimates of the qubit and gate resources, as well as of the QPU execution times. In particular, we consider the circuits resulting from the application of feature maps corresponding to equations (4)–(6). After decomposing the Hadamard gates into $H = R_Z(\pi/2)\sqrt{X}R_Z(\pi/2)$ and the $R_{ZZ}$ gates into two CNOTs and one $R_Z$ gate, the feature map with $n$ qubits then consists of $2n$ $\sqrt{X}$ gates, $8n-2$ $R_Z$ gates and $4(n-1)$ CNOT gates. The final circuit, comprises the feature map, its inverse, and a concluding layer for the readout in the suited basis. Using the default transpilation in Qiskit [67], the corresponding pulse schedule has an average duration of 12.75 $\mu$s for $n=6$ qubits (which corresponds to the number of qubits used in the hardware experiment in section 4.3). The average is taken over the different available IBM Quantum backends (calculated on August 30, 2023). Note that the rotation angles of the $R_Z$ gates does not have an influence on the duration of the pulse schedule since the $R_Z$ gates are applied virtually on the IBM Quantum hardware [85]. For the experiments with 50 data samples executed on the hardware (section 4.3), the total time spend on the QPU would therefore amount to around 16.26 ms. Note that in practice, the total execution time needed to run a quantum circuit includes additional 'classical' contributions such as the time of initialization of the qubit register, the queuing time, and others.

## Appendix G. Different BSM datasets

In the main text, we performed initial investigations about the generalization power of the scrambling approach by looking at its performance in detecting Higgs and Graviton events among SM events. In this section, we complement these results by considering four additional BSM datasets that were studied in [58]. These processes are:

- $A \rightarrow 4l$: Neutral scalar boson (with mass $m = 50$ GeV) decaying into two four leptons via two off-shell $Z$ bosons [86],
- $LQ \rightarrow b\tau$: Leptoquark ($m = 80$ GeV) decaying into a $b$ quark and a $\tau$ lepton [87],
- $h^{\pm} \rightarrow \tau\nu$: Charged scalar boson ($m = 60$ GeV) decaying into a $\tau$ lepton and a neutrino [88],
- $h^0 \rightarrow \tau\tau$: Scalar boson ($m = 60$ GeV) decaying into two $\tau$ leptons [89].

**Table 4.** Detection AUC for different BSM processes studied in [58].

| Algorithm | Scrambling strength | Number of features | $A \rightarrow 4l$ AUC | $LQ \rightarrow b\tau$ AUC | $h^{\pm} \rightarrow \tau\nu$ AUC | $h^0 \rightarrow \tau\tau$ AUC |
|---|---|---|---|---|---|---|
| SVC | Low | 8 | $0.604 \pm 0.015$ | $0.610 \pm 0.030$ | $0.823 \pm 0.017$ | $0.728 \pm 0.018$ |
| | Medium | 6 | $0.611 \pm 0.019$ | $0.593 \pm 0.023$ | $0.801 \pm 0.017$ | $0.701 \pm 0.015$ |
| | High | 6 | $0.611 \pm 0.018$ | $0.600 \pm 0.023$ | $0.804 \pm 0.019$ | $0.704 \pm 0.019$ |
| QSVC | Low | 8 | $0.594 \pm 0.013$ | $0.545 \pm 0.033$ | $0.776 \pm 0.020$ | $0.692 \pm 0.018$ |
| | Medium | 6 | $0.555 \pm 0.017$ | $0.514 \pm 0.032$ | $0.749 \pm 0.031$ | $0.658 \pm 0.025$ |
| | High | 6 | $0.533 \pm 0.010$ | $0.555 \pm 0.025$ | $0.754 \pm 0.028$ | $0.659 \pm 0.021$ |
| sSVC[a] | | 6 | $0.785 \pm 0.006$ | $0.843 \pm 0.002$ | $0.937 \pm 0.001$ | $0.854 \pm 0.005$ |

[a] Reference calculation: supervised training for SM/BSM process classification.

We perform the same anomaly detection procedure as for the Higgs and Graviton datasets in section 4.2. The results are reported in table 4.

We additionally report the detection AUC of an SVC that was trained on the corresponding SM/BSM classification task (sSVC in table 4). The results are similar to those reported in the main text: the SVC performs slightly better than the QSVC, and the SVC trained with a supervised approach performs better than the (Q)SVCs trained with the scrambling approach.

The classification tasks seem to be harder than the ones studied in the main text, since the AUC scores generally are considerably lower. We attribute this finding to the larger similarity between the feature distributions corresponding to the SM and BSM events, which is higher for the processes above than it was for the Higgs and Graviton events. This can also lead to situations where the scrambling approach may even fail.

However, by including some physically motivated biases into the scrambling process (instead of using a purely agnostic approach) we expect to see an improvement of the AUC scores towards the values obtained with a supervised approach.

All in all, these first results indicate that a classifier trained with the scrambling procedure can effectively be used for the detection of different BSM anomalies, and therefore is able to generalize to different scenarios.

## ORCID iDs

Julian Schuhmacher https://orcid.org/0000-0002-7011-6477
Laura Boggia https://orcid.org/0000-0002-9924-7489
Vasilis Belis https://orcid.org/0000-0001-5920-8998
Ema Puljak https://orcid.org/0000-0002-6011-9965
Michele Grossi https://orcid.org/0000-0003-1718-1314
Maurizio Pierini https://orcid.org/0000-0003-1939-4268
Sofia Vallecorsa https://orcid.org/0000-0002-7003-5765
Francesco Tacchino https://orcid.org/0000-0003-2008-5956
Panagiotis Barkoutsos https://orcid.org/0000-0001-9428-913X
Ivano Tavernelli https://orcid.org/0000-0001-5690-1981

## References

[1] Glashow S L 1961 Partial-symmetries of weak interactions *Nucl. Phys.* **22** 579–88
[2] Weinberg S 1967 A model of leptons *Phys. Rev. Lett.* **19** 1264
[3] Salam A 1968 Elementary particle physics: relativistic groups and analyticity *Eighth Nobel Symp., Almquvist and Wiksell*, Ed N Svartholm
[4] Hooft G 't and Veltman. M 1972 Regularization and renormalization of gauge fields *Nucl. Phys.* B **44** 189–213
[5] Langacker P 2017 *The Standard Model and Beyond* (Taylor & Francis)
[6] Beacham J *et al* 2019 Physics beyond colliders at CERN: beyond the standard model working group report *J. Phys. G: Nucl. Part. Phys.* **47** 010501
[7] Béjar Alonso I *et al* (eds) 2020 High-luminosity large hadron collider (HL-LHC): Technical Design Report, 10 *CERN Yellow Reports: Monographs*
[8] Radovic A, Williams M, Rousseau D, Kagan M, Bonacorsi D, Himmel A, Aurisano A, Terao K and Wongjirad T 2018 Machine learning at the energy and intensity frontiers of particle physics *Nature* **560** 41–48
[9] Guest D, Cranmer K and Whiteson D 2018 Deep learning and its application to LHC physics *Annu. Rev. Nucl. Part. Sci.* **68** 161–81
[10] Albertsson K *et al* 2019 Machine learning in high energy physics community white paper *J. Phys.: Conf. Ser.* **1085** 022008
[11] Bourilkov D 2019 Machine and deep learning applications in particle physics *Int. J. Mod. Phys.* A **34** 1930019

[12] Carleo G, Cirac I, Cranmer K, Daudet L, Schuld M, Tishby N, Vogt-Maranto L and Zdeborová L 2019 Machine learning and the physical sciences *Rev. Mod. Phys.* **91** 045002

[13] Nachman B 2022 *Artificial Intelligence for High Energy Physics* (World Scientific) pp 85–112

[14] Kasieczka G *et al* 2021 The LHC olympics 2020 a community challenge for anomaly detection in high energy physics *Rep. Prog. Phys.* **84** 124201

[15] Aarrestad T *et al* 2022 The dark machines anomaly score challenge: benchmark data and model independent event classification for the large hadron collider *SciPost Phys.* **12** 043

[16] D'Agnolo R T and Wulzer A 2019 Learning new physics from a machine *Phys. Rev.* D **99** 015014

[17] Collins J, Howe K and Nachman B 2018 Anomaly detection for resonant new physics with machine learning *Phys. Rev. Lett.* **121** 241803

[18] Collins J H, Howe K and Nachman B 2019 Extending the search for new resonances with machine learning *Phys. Rev.* D **99** 014038

[19] Farina M, Nakai Y and Shih D 2020 Searching for new physics with deep autoencoders *Phys. Rev.* D **101** 075021

[20] Blance A, Spannowsky M and Waite P 2019 Adversarially-trained autoencoders for robust unsupervised new physics searches *J. High Energy Phys.* JHEP10(2019)047

[21] Hajer J, Li Y-Y, Liu T and Wang H 2020 Novelty detection meets collider physics *Phys. Rev.* D **101** 076015

[22] De Simone A and Jacques T 2019 Guiding new physics searches with unsupervised learning *Eur. Phys. J.* C **79** 1–15

[23] Andreassen A, Nachman B and Shih D 2020 Simulation assisted likelihood-free anomaly detection *Phys. Rev.* D **101** 095004

[24] Nachman B and Shih D 2020 Anomaly detection with density estimation *Phys. Rev.* D **101** 075042

[25] Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N and Lloyd S 2017 Quantum machine learning *Nature* **549** 195–202

[26] Guan W, Perdue G, Pesah A, Schuld M, Terashi K, Vallecorsa S and Vlimant J-R 2021 Quantum machine learning in high energy physics *Mach. Learn.: Sci. Technol.* **2** 011003

[27] Mangini S, Tacchino F, Gerace D, Bajoni D and Macchiavello C 2021 Quantum computing models for artificial neural networks *Europhys. Lett.* **134** 10002

[28] Delgado A *et al* 2022 Quantum computing for data analysis in high-energy physics (arXiv:2203.08805)

[29] Rebentrost P, Mohseni M and Lloyd S 2014 Quantum support vector machine for big data classification *Phys. Rev. Lett.* **113** 130503

[30] Havlíček V, Córcoles A D, Temme K, Harrow A W, Kandala A, Chow J M and Gambetta J M 2019 Supervised learning with quantum-enhanced feature spaces *Nature* **567** 209–12

[31] Schuld M and Killoran N 2018 Quantum machine learning in feature hilbert spaces *Phys. Rev. Lett.* **122** 040504

[32] Cong I, Choi S and Lukin M D 2019 Quantum convolutional neural networks *Nat. Phys.* **15** 1273–8

[33] Romero J, Olson J P and Aspuru-Guzik A 2017 Quantum autoencoders for efficient compression of quantum data *Quantum Sci. Technol.* **2** 045001

[34] Wu S L *et al* 2021 Application of quantum machine learning using the quantum kernel algorithm on high energy physics analysis at the LHC *Phys. Rev. Res.* **3** 033221

[35] Heredge J, Hill C, Hollenberg L and Sevior M 2021 Quantum support vector machines for continuum suppression in B meson decays *Comput. Softw. Big Sci.* **5** 27

[36] Terashi K , Kaneda M, Kishimoto T, Saito M, Sawada R and Tanaka J 2021 Event classification with quantum machine learning in high-energy physics *Comput. Softw. Big Sci.* **5** 2

[37] Gianelle A *et al* 2022 Quantum machine learning for *b*-jet charge identification *J. High Energ. Phys.* **2022** 14

[38] Blance A and Spannowsky M 2021 Quantum machine learning for particle physics using a variational quantum classifier *J. High Energy Phys.* JHEP02(2021)212

[39] Wu S L *et al* 2021 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: Nucl. Part. Phys.* **48** 125003

[40] Belis V, González-Castillo S, Reissel C, Vallecorsa S, Combarro E F, Dissertori G and Reiter F 2021 Higgs analysis with quantum classifiers *EPJ Web Conf.* **251** 03070

[41] Chen S Y C , Wei T-C, Zhang C, Yu H and Yoo S 2022 Quantum convolutional neural networks for high energy physics data analysis *Phys. Rev. Res.* **4** 013231

[42] Chen S Y C *et al* 2021 Hybrid quantum-classical graph convolutional network (arXiv:2101.06189)

[43] Ngairangbam V S, Spannowsky M and Takeuchi M 2022 Anomaly detection in high-energy physics using a quantum autoencoder *Phys. Rev.* D **105** 095004

[44] Mott A, Job J, Vlimant J-R, Lidar D and Spiropulu M 2017 Solving a higgs optimization problem with quantum annealing for machine learning *Nature* **550** 375–9

[45] Zlokapa A *et al* 2021 Quantum adiabatic machine learning by zooming into a region of the energy surface *Phys. Rev. A* **102** 062405

[46] Kim M *et al* 2021 Leveraging quantum annealer to identify an event-topology at high energy colliders (arXiv:2111.07806)

[47] Blance A and Spannowsky M 2021 Unsupervised event classification with graphs on classical and photonic quantum computers *J. High Energy Phys.* JHEP08(2021)170

[48] Araz J Y and Spannowsky M 2022 Quantum-probabilistic hamiltonian learning for generative modelling & anomaly detection (arXiv:2211.03803)

[49] Peixoto M C *et al* 2022 Fitting a collider in a quantum computer: tackling the challenges of quantum machine learning for big datasets (arXiv:2211.03233)

[50] Cugini D, Gerace D, Govoni P, Perego A and Valsecchi D 2023 Comparing quantum and classical machine learning for vector boson scattering background reduction at the large hadron collider *Quantum Mach. Intell.* **5** 35

[51] Shashua A 2009 Introduction to machine learning: class notes 67577

[52] Liu Y, Arunachalam S and Temme K 2021 A rigorous and robust quantum speed-up in supervised machine learning *Nat. Phys.* **17** 1013–7

[53] Kübler J, Buchholz S and Schölkopf B 2021 The inductive bias of quantum kernels *Advances in Neural Information Processing Systems* vol 34 pp 12661–73

[54] Huang H-Y, Broughton M, Mohseni M, Babbush R, Boixo S, Neven H and McClean J R 2021 Power of data in quantum machine learning *Nat. Commun.* **12** 2631

[55] Glick J R *et al* 2021 Covariant quantum kernels for data with group structure (arXiv:2105.03406)

[56] Lloyd S *et al* 2020 Quantum embeddings for machine learning (arXiv:2001.03622)

[57] Cerri O, Nguyen T Q, Pierini M, Spiropulu M and Vlimant J-R 2019 Variational autoencoders for new physics mining at the large hadron collider *J. High Energy Phys.* JHEP05(2019)036

[58] Knapp O, Cerri O, Dissertori G, Nguyen T Q, Pierini M and Vlimant J R 2021 Adversarially learned anomaly detection on CMS open data: re-discovering the top quark *Eur. Phys. J. Plus* **136** 236

[59] CMS collaboration 2016 Simulated dataset VBFHiggs0MToBB_M-125p6_7TeV-JHUGenV4-pythia6-tauola in AODSIM format for 2011 collision data (BSM Higgs) *CERN Open Data Portal* (https://doi.org/10.7483/OPENDATA.CMS.3R3P.5JYR)

[60] Randall L and Sundrum R 1999 A large mass hierarchy from a small extra dimension *Phys. Rev. Lett.* **83** 3370–3

[61] CMS collaboration 2016 Simulated dataset Graviton2BPqqbarToZZTo4L_M-125p6_7TeV-JHUGenV3-pythia6 in AODSIM format for 2011 collision data (BSM Higgs) *CERN Open Data Portal* (https://doi.org/10.7483/OPENDATA.CMS.SZWT.H9MC)

[62] Sjöstrand T, Ask S, Christiansen J R, Corke R, Desai N, Ilten P, Mrenna S, Prestel S, Rasmussen C O and Skands P Z 2015 An introduction to PYTHIA 8.2 *Comput. Phys. Commun.* **191** 159–77

[63] De Favereau J, Delaere C, Demin P, Giammanco A, Lemaitre V, Mertens A and Selvaggi M 2014 DELPHES 3: a modular framework for fast simulation of a generic collider experiment *J. High Energy Phys.* JHEP02(2014)057

[64] Sjostrand T, Mrenna S and Skands P Z 2006 PYTHIA 6.4 physics and manual *J. High Energy Phys.* JHE05(2006)026

[65] Shaydulin R and Wild S M 2021 Importance of Kernel bandwidth in quantum machine learning *Phys. Rev. A* **106** 042407

[66] Peters E , Caldeira J, Ho A, Leichenauer S, Mohseni M, Neven H, Spentzouris P, Strain D and Perdue G N 2021 Machine learning of high dimensional data on a noisy quantum processor *npj Quantum Inf.* **7** 161

[67] Abraham H *et al* 2021 Qiskit: an open-source framework for quantum computing

[68] Schuhmacher J, Mazzola G, Tacchino F, Dmitriyeva O, Bui T, Huang S and Tavernelli I 2022 Extending the reach of quantum computing for materials science with machine learning potentials *AIP Adv.* **12** 115321

[69] Woźniak K *et al* 2023 Quantum anomaly detection in the latent space of proton collision events at the LHC (arXiv:2301.10780)

[70] Aaronson S 2022 How much structure is needed for huge quantum speedups? (arXiv:2209.06930)

[71] Slattery L *et al* 2022 Numerical evidence against advantage with quantum fidelity kernels on classical data *Phys. Rev. A* **107** 062417

[72] Tacchino F, Mangini S, Barkoutsos P K, Macchiavello C, Gerace D, Tavernelli I and Bajoni D 2021 Variational learning for quantum artificial neural networks *IEEE Trans. Quantum Eng.* **2** 1–10

[73] Wu Y *et al* 2023 Quantum Phase Recognition via Quantum Kernel Methods *Quantum* **7** 981

[74] Huang H-Y *et al* 2022 Quantum advantage in learning from experiments *Science* **376** 1182–6

[75] ECFA Detector R&D Roadmap Process Group 2020 The 2021 ECFA detector research and development roadmap *Technical Report*

[76] Di Marcantonio F *et al* 2023 Quantum Advantage Seeker with Kernels (QuASK): a software framework to speed up the research in quantum machine learning *Quantum Mach. Intell.* **5** 20

[77] Hubregtsen T *et al* 2022 Training quantum embedding kernels on near-term quantum computers *Phys. Rev. A* **106** 042431

[78] Maciejewski F B, Zimborás Z and Oszmaniec M 2020 Mitigation of readout noise in near-term quantum devices by classical post-processing based on detector tomography *Quantum* **4** 257

[79] Bravyi S, Sheldon S, Kandala A, Mckay D C and Gambetta J M 2021 Mitigating measurement errors in multiqubit experiments *Phys. Rev.* A **103** 042605

[80] Earnest N, Tornow C and Egger D J 2021 Pulse-efficient circuit transpilation for quantum applications on cross-resonance-based hardware *Phys. Rev. Res.* **3** 043088

[81] Dumitrescu E F , McCaskey A, Hagen G, Jansen G, Morris T, Papenbrock T, Pooser R, Dean D and Lougovski P 2018 Cloud quantum computing of an atomic nucleus *Phys. Rev. Lett.* **120** 210501

[82] He A, Nachman B, de Jong W A and Bauer C W 2020 Zero-noise extrapolation for quantum-gate error mitigation with identity insertions *Phys. Rev.* A **102** 012426

[83] Giurgica-Tiron T *et al* 2020 Digital zero noise extrapolation for quantum error mitigation *2020 IEEE Int. Conf. on Quantum Computing and Engineering (QCE)* (IEEE) pp 306–16

[84] Berg E *et al* 2023 Probabilistic error cancellation with sparse pauli-lindblad models on noisy quantum processors *Nature Physics* **19** 1116–21

[85] McKay D C, Wood C J, Sheldon S, Chow J M and Gambetta J M 2017 Efficient $z$ gates for quantum computing *Phys. Rev.* A **96** 022330

[86] Cerri O *et al* 2020 New physics mining at the large Hadron collider: $A \to 4l$ (Zenodo) (https://doi.org/10.5281/zenodo.3675159)

[87] Cerri O *et al* 2020 New physics mining at the large Hadron collider: $LQ \to b\tau$ (Zenodo) (https://doi.org/10.5281/zenodo.3675196)

[88] Cerri O *et al* 2020 New physics mining at the large Hadron collider: $h\pm \to \tau\nu$ (Zenodo) (https://doi.org/10.5281/zenodo.3675178)

[89] Cerri O *et al* 2020 New physics mining at the large Hadron collider: $h^0 \to \tau\tau$ (https://doi.org/10.5281/zenodo.3675190)