Quantum machine learning for quantum anomaly detection

Nana Liu[1,2,*] and Patrick Rebentrost[3,†]

[1]*Centre for Quantum Technologies, National University of Singapore, 3 Science Drive 2, Singapore 117543*
[2]*Singapore University of Technology and Design, 8 Somapah Road, Singapore 487372*
[3]*Xanadu, 372 Richmond St W, Toronto, M5V 2L7, Canada*

Anomaly detection is used for identifying data that deviate from 'normal' data patterns. Its usage on classical data finds diverse applications in many important areas like fraud detection, medical diagnoses, data cleaning and surveillance. With the advent of quantum technologies, anomaly detection of *quantum* data, in the form of quantum states, may become an important component of quantum applications. Machine learning algorithms are playing pivotal roles in anomaly detection using classical data. Two widely-used algorithms are kernel principal component analysis and one-class support vector machine. We find corresponding quantum algorithms to detect anomalies in quantum states. We show that these two quantum algorithms can be performed using resources logarithmic in the dimensionality of quantum states. For pure quantum states, these resources can also be logarithmic in the number of quantum states used for training the machine learning algorithm. This makes these algorithms potentially applicable to big *quantum* data applications.

I. Introduction

Quantum computing has achieved success in finding algorithms that offer speed-ups to classical algorithms for certain problems like factoring and searching an unstructured data-base. It relies upon techniques like quantum phase estimation, quantum matrix inversion and amplitude amplification. These tools have recently been employed in quantum algorithms for machine learning[1–4], an area which has various applications across very broad disciplines. In particular, it has proved useful in data fitting and classification problems that appear in pattern recognition [5–11], where quantum algorithms can offer speed-ups in both the dimensionality and number of data used to train the algorithm.

In the presence of a large amount of input data, some coming from unreliable or unfamiliar sources, it is important to be able to detect outliers in the data. This is especially relevant when few examples of outliers may be available to develop a prior expectation. These outliers can be indicative of some unexpected phenomena emerging in a system that has never been before identified, like a faulty system or a malicious intruder. The subject of anomaly detection is learning how to make an accurate identification of an outlier when training data of 'normal' data are given. Machine learning algorithms for anomaly detection in classical data have been widely applied in areas as diverse as fraud detection, medical diagnoses, data cleaning and surveillance [12, 13].

Quantum data, which take the form of quantum states, are prevalent in all forms of quantum computation, quantum simulation and quantum communication. Anomaly detection of quantum states is thus expected to be an important direction in the future of quantum information processing and communication, in particular, over the cloud or quantum internet. Since machine learning algorithms have proved successful for anomaly detection in classical data, a natural question arises if there exist quantum machine algorithms used for detecting anomalies in quantum systems. While classical anomaly detection techniques for quantum states can be used, they are only possible by first probing the classical descriptions of these states which require state tomography, requiring a large number of measurements [14, 15]. Thus, it would also be advantageous to reduce these resource overheads by using a quantum algorithm.

In this paper we discuss quantum machine learning methods applied to the detection of anomalies in quantum states themselves, which we call *quantum anomaly detection*. Given a training set of M unknown quantum states, each of dimension d, the task is to use a quantum computer to detect outliers on new data, occurring for example due to a faulty quantum device. Our schemes also do not require quantum random access memory [16], since the necessary data input is fully given by quantum states generated from quantum devices. In that sense, we present an instance of an algorithm for *quantum learning* [17–21].

We present two quantum algorithms for quantum anomaly detection: kernel principal component analysis (PCA) [22, 23] and one-class support vector machine (SVM) [24, 25]. We show that pure state anomaly detection algorithms can be performed using resources logarithmic in M and d. For mixed quantum states, we show how this is possible using resources logarithmic in d only. We note this can also be an exponential resource reduction compared to state tomography [26].

After introducing the classical kernel PCA and one-class SVM algorithms in Section II, we develop quantum anomaly detection algorithms based on kernel PCA and one-class SVM for pure quantum states in Section III. We generalise these algorithms for mixed quantum states in Section IV and discuss the implications of these results in Section V.

---

[*]Electronic address: nana.liu@quantumlah.org
[†]Electronic address: patrick@xanadu.ai

II. Anomaly detection in classical machine learning

Anomaly detection involves algorithms that can recognise outlier data compared to some expected or 'normal' data patterns. These unusual data signatures, or *anomalies*, can result from faulty systems, malicious intrusion into a system or from naturally occuring novel phenomena that are too rare to have been captured and classed with their own training sets. These algorithms generally provide proximity measures that quantify how 'far' the inspected data pattern is from the 'norm'. This is also closely related to the change point detection problem, which involves finding the point at which an underlying probability distribution governing an observed phenomena has changed [27, 28].

There are three broad classes of anomaly detection algorithms: supervised, unsupervised and semi-supervised [12, 13]. Supervised anomaly detection assume labelled training sets for *both* 'normal' and anomalous data and the usual supervised learning algorithms can be employed [43]. The unsupervised learning algorithms apply when neither 'normal' nor anomalous data are labelled, but assume that 'normal' cases occur far more frequently than anomalous ones. However, there are scenarios where the 'normal' data can be readily identified and gathered, but the anomalous data may be too scarce to form a training set. Here semi-supervised methods are required. We will focus on this latter scenario in this paper.

Anomaly detection algorithms can be applied to different types of anomalies: point, contextual anomalies and collective anomalies [12, 13]. Point anomalies are single data instances that can be classified as either 'normal' data or an anomaly. They are the simplest, most widely studied type of anomaly and will be the focus of this paper.

Contextual anomalies are individual data instances that are anomalous only with respect to a particular context, common in time-series data. Collective anomalies on the other hand are collections of data instances that are unusual only with respect to the whole data set. These latter types of anomalies in the quantum domain will be explored in future work.

For point anomaly detection alone, there are many different types of algorithms based on different techniques [12, 13]. Two of these algorithms, kernel PCA [22] and one-class SVM [25], resemble most closely to existing quantum machine learning algorithms that provide speed-ups over their classical counterparts [5, 29]. We will use these classical algorithms, described below, as a foundation to develop quantum kernel PCA and quantum one-class SVM algorithms to detect anomalies in quantum data.

A. Kernel PCA

Suppose we are given a training set of M vectors $\vec{x}_i \in \mathbb{R}^M$. The centroid of the training data is denoted by

$$\vec{x}_c = \frac{1}{M} \sum_{i=1}^{M} \vec{x}_i, \qquad (1)$$

and we can define $\vec{z}_i = \vec{x}_i - \vec{x}_c$ to be the centered data. The centered sample covariance matrix C is then given by

$$C = \frac{1}{M-1} \sum_{i=1}^{M} \vec{z}_i \vec{z}_i^T. \qquad (2)$$

Anomaly detection via kernel PCA is then performed using a proximity measure

$$f(\vec{x}_0) = |\vec{z}_0|^2 - \vec{z}_0^T C \vec{z}_0, \qquad (3)$$

where $\vec{z}_0 = \vec{x}_0 - \vec{x}_c$. This measure detects a difference in the distance between point $\vec{x}_0$ and the centroid of the training data and the variance of the training data along the direction $\vec{z}_0$. This can quantify how anomalous the point $\vec{x}_0$ is compared to the training data. A larger $f(\vec{x}_0)$ thus implies a more anomalous datum than a smaller $f(\vec{x}_0)$.

This method also allows us to classify anomalies in non-linear feature spaces by performing the replacement $\vec{x}_i \rightarrow \phi(\vec{x}_i)$ with a feature map $\phi(\vec{x}_i)$. The inner products are performed in an abstract linear feature space. The inner product can be represented by a kernel function $k(\vec{x}_i, \vec{x}_j)$, which can be taken to be a non-linear function, for example $k(\vec{x}_i, \vec{x}_j) = \phi(\vec{x}_i)^T \phi(\vec{x}_j)$. For some kernels like the radial basis function kernels, it has been shown that anomaly detection can be more effective using kernel PCA compared to one-class SVM [22]. While for pure quantum states we will focus on the linear kernel, this analysis might also be extended to polynomial kernels [5] and radial basis function kernels [30].

B. One-class SVM

One-class SVM [31, 32] has also been applied to anomaly detection in classical data. For instance, it has been used as part of change point detection algorithms and detecting novel signals in a time series [33]. We focus on the least-squares formulation of the one-class SVM [25] (using the linear kernel for now). This involves finding a hyperplane in feature space such that it both maximises the distance from the hyperplane to the origin of the feature space, as well as minimising the least-squares distance ξ_i between the data point $\vec{x}_i$ and the hyperplane.

This is equivalent to extremising a Lagrangian $\mathcal{L}$

$$\mathcal{L} = \mathcal{L}(\vec{w}, r, \xi_i, \alpha_i) = \frac{1}{2}||\vec{w}||^2 + \frac{1}{2P_T M}\sum_{i=1}^{M}\xi_i^2$$
$$- r - \sum_{i=1}^{M}\alpha_i(\vec{w}^\dagger \cdot \vec{x}_i + \xi_i - r), \qquad (4)$$

where $\vec{w} \cdot \vec{x}_i = r$ defines a hyperplane in feature space, which 'best characterises' the set of training data and r is a bias term. In the presence of non-zero ξ_i, the hyperplane we want to find is subject to the constraint $\xi_i = r - \vec{w}^\dagger \cdot \vec{x}_i$. The Lagrange multipliers α_i can be represented in vectorised form as $\vec{\alpha}$ and P_T denotes a 'threshold acceptance probability', related to the fraction of data expected to be outliers [31]. From the standard method, one arrives at the corresponding matrix equation

$$\tilde{F}\begin{pmatrix} -r \\ \vec{\alpha} \end{pmatrix} = \begin{pmatrix} 0 & \vec{e}^T \\ \vec{e} & K + P_T M\mathbf{1} \end{pmatrix}\begin{pmatrix} -r \\ \vec{\alpha} \end{pmatrix} = \begin{pmatrix} 1 \\ \vec{0} \end{pmatrix}, \quad (5)$$

where K is a matrix whose elements are the kernel functions $K_{ij} = k(\vec{x}_i, \vec{x}_j)$ and $\vec{e} = (1, ..., 1)^T$. The matrix inversion of $\tilde{F}$ applied to $(1, \vec{0})^T$ then reveals $(-r, \alpha_i)$. Once α_i and r are determined, one can compute a proximity measure $f(\vec{x}_0)$ from the new data $\vec{x}_0$ to the hyperplane

$$f(\vec{x}_0) = |\vec{w}^\dagger \cdot \vec{x}_0 - r|, \qquad (6)$$

which can also be generalised to non-linear kernels.

It is possible to look at restricted versions of this algorithm, where the bias r is set to a constant number c. This reduces Eq. (5) to $(K + P_T M\mathbf{1})\vec{\alpha} = c\vec{e}$. This should produce the same ordering of proximity measures when only the relative distance to the hyperplane is relevant. For simplicity, in the rest of the paper we focus on the case $r = c = 1$ [44].

III. Pure state anomaly detection

We begin with the following set-up. Suppose we are given access to both the unitaries $\{U_i\}$ and the control unitary $U_C = \sum_{i=1}^{M}|i\rangle\langle i| \otimes U_i$, where $U_i|0\rangle = |\psi_i\rangle$ are the training quantum states. We also suppose that given any two unitaries u_1 and u_2, it is possible to generate a control unitary of the form $|0\rangle\langle 0| \otimes u_1 + |1\rangle\langle 1| \otimes u_2$ [45]. The states $|\psi_i\rangle$ are labelled 'normal'. If we are also given U_0 where $U_0|0\rangle = |\psi_0\rangle$, our task in quantum anomaly detection is to quantify how anomalous the state $|\psi_0\rangle$ is compared to the 'normal' states. One scenario where this may be relevant is in the testing of manufactured quantum circuits.

The control unitary U_C [46] allows one to create superpositions of training states of the form $\sum_{i=1}^{M}|\psi_i\rangle|i\rangle$. This is required for the generation of the necessary kernel density matrices, the computation of some normalisation constants and for the anomaly classification stage.

With these assumptions, one can create multiple copies of the training states $\{|\psi_i\rangle\}$, the test state $|\psi_0\rangle$ and all the other states necessary to perform quantum anomaly detection. Requiring multiple identical copies of states is a purely quantum characteristic stemming from the no-cloning theorem and the imperfect distinguishability of non-orthogonal quantum states. For instance, multiple copies of two different states are required to improve the success probability in distinguishing between these two states, yet these multiple copies cannot be generated from a single copy due to the no-cloning constraint.

A. Quantum kernel PCA (pure state)

1. Algorithm using resources logarithmic in d

For a quantum version of the kernel PCA algorithm for anomaly detection, first we define the analogous centroid state of the training quantum data by

$$|\psi_c\rangle = \mathcal{N}_c\sum_{i=1}^{M}|\psi_i\rangle = \mathcal{N}_c\sum_{i=1}^{M}\sum_{j=1}^{d}(\psi_i)_j|j\rangle. \qquad (7)$$

Here $(\psi_i)_j$ denotes the jth component of a vector $\vec{\psi}_i$. We demonstrate in Appendix A that the normalisation $|\mathcal{N}_c|^2 = 1/\sum_{i,j=1}^{M}\langle\psi_i|\psi_j\rangle$ can be found using $\mathcal{O}(\log M)$ resources. We also show a method for preparing the centroid state in Appendix A.

We can write the centered quantum data as

$$|z_i\rangle = \mathcal{N}_i(|\psi_i\rangle - (1/(M\mathcal{N}_c))|\psi_c\rangle) = \mathcal{N}_i\sum_{j=1}^{d}(z_i)_j|j\rangle, \quad (8)$$

where $|\mathcal{N}_i|^2 = 1/(1 + 1/(M^2|\mathcal{N}_c|^2) - (2/M)\text{Re}[\langle\psi_i|\psi_c\rangle]/\mathcal{N}_c)$ and $(z_i)_j$ is the jth component of the vector $\vec{z}_i = \vec{\psi}_i - (1/M)\sum_{k=1}^{M}\vec{\psi}_k$. If the new quantum state to be classified is $|\psi_0\rangle = \sum_{i=1}^{d}(\psi_0)_i|i\rangle$, we can denote its centered equivalent by $|z_0\rangle = \mathcal{N}_0(|\psi_0\rangle - (1/(M\mathcal{N}_c))|\psi_c\rangle) = \mathcal{N}_0\sum_{j=1}^{d}(z_0)_j|j\rangle$, where $|\mathcal{N}_0|^2 = 1/(1 + 1/(M^2|\mathcal{N}_c|^2) - (2/M)\text{Re}[\langle\psi_0|\psi_c\rangle]/\mathcal{N}_c)$.

The centered sampled covariance matrix can be rewritten in terms of quantum states as $C = (1/(M - 1))\sum_{i=1}^{M}(1/|\mathcal{N}_i|^2)|z_i\rangle\langle z_i|$. It is also proportional to a density matrix $\mathcal{C} = C/\text{tr}(C)$, where $\text{tr}(C) = (1/(M - 1))\sum_{i=1}^{d}\sum_{j=1}^{M}(z_j)_i(z_j)_i^*$.

By rewriting Eq. (3) in terms of quantum states $|z_0\rangle$, $\mathcal{C}$ and dividing the total expression by $|\mathcal{N}_0|^2$, we can define a proximity measure $f(\psi_0)$, which obeys $0 \leq f(\psi_0) \leq 1$. This quantifies how anomalous the quantum state $|\psi_0\rangle$ is from the set $\{|\psi_i\rangle\}$ and can be written as

$$f(\psi_0) = \langle z_0|(\mathbf{1} - C)|z_0\rangle = 1 - \text{tr}(C)\text{tr}(\mathcal{C}|z_0\rangle\langle z_0|)$$
$$= 1 - \frac{\text{tr}(\mathcal{C}|z_0\rangle\langle z_0|)}{|\mathcal{N}_{\chi_c}|^2(M - 1)} = 1 - \frac{1}{M - 1}\sum_{i=1}^{M}\frac{|\langle z_i|z_0\rangle|^2}{|\mathcal{N}_i|^2}. \quad (9)$$

In the first line, we use the fact that $|z_0\rangle$ is a normalised quantum state, which is true in all cases except $|z_0\rangle = 0$. The special limit $|z_0\rangle = 0$ corresponds to when all the training states and $|\psi_0\rangle$ are identical. In this case $f(\psi_0) = 0$, indicating no anomaly, as expected.

This expression in Eq. (9) is composed of a sum of $\mathcal{O}(M)$ inner product terms $\langle\psi_i|\psi_\kappa\rangle$ where $i = 1,...,M$ and $\kappa = 0,...,M$. Since the phases of these inner products are also required, the standard swap test [34] is insufficient. Each inner product can be found instead using a modified swap test. For this we prepare the state $|\Psi_{i\kappa}\rangle = (1/2)[|0\rangle(|\psi_i\rangle + \zeta|\psi_\kappa\rangle) + |1\rangle(|\psi_i\rangle - \zeta|\psi_\kappa\rangle)]$ by applying the control unitary $|0\rangle\langle0|\otimes U_i + |1\rangle\langle1|\otimes U_\kappa$ and then a Hadamard gate onto the state $(1/\sqrt{2})(|0\rangle + \zeta|1\rangle)|0\rangle$, where ζ is a complex number. The success probability of measuring $|1\rangle$ in the ancilla of state $|\Psi_{i\kappa}\rangle$ is given by $P_{i\kappa} = (1/2)(1 - \text{Re}(\zeta\langle\psi_i|\psi_\kappa\rangle))$. Then $\zeta = 1$ and $\zeta = i$ recover respectively the real and imaginary part of $\langle\psi_i|\psi_\kappa\rangle$. The final measurement outcomes of the ancilla state satisfy a Bernoulli distribution so the number of measurements $\mathcal{N}$ required to estimate $P_{i\kappa}$ to precision ϵ is $P_{i\kappa}(1 - P_{i\kappa})/\epsilon^2$ and is upper bounded by $\mathcal{O}(1/\epsilon^2)$. Therefore $\mathcal{N} \sim \mathcal{O}(\text{poly}(\log d))$ is sufficient if an error of order $\mathcal{O}(1/\log d)$ is accepted.

Each modified swap test requires $\mathcal{O}(\text{poly}(\log d))$ number of measurements and copies of $|\psi_\kappa\rangle$. The required normalisation constants $|\mathcal{N}_i|^2$ can be computed using inner products between the training states and $\mathcal{N}_c$, which require resources costing $\mathcal{O}(\text{poly}(M\log d))$. Thus the proximity measure $f(\psi_0)$ can be computed using resources scaling as $\mathcal{O}(\text{poly}(M\log d))$.

2. Algorithm using resources logarithmic in d and M

Next we show an alternative protocol for computing $f(\psi_0)$, now using $\mathcal{O}(\log M)$ resources. This can be achieved by preparing a superposition of the centered data. A method to generate $\mathcal{C}$ is to create the state $|\chi_c\rangle = \mathcal{N}_{\chi_c}\sum_{i=1}^{M}(1/\mathcal{N}_i)|i\rangle|z_i\rangle$ where $|\mathcal{N}_{\chi_c}|^2 = 1/\sum_{i=1}^{M}(1/|\mathcal{N}_i|^2) = M + 1/(2M|\mathcal{N}_c|^2)$. In the reduced space of the second register we find

$$\text{tr}_1(|\chi_c\rangle\langle\chi_c|) = |\mathcal{N}_{\chi_c}|^2\sum_{i=1}^{M}(1/|\mathcal{N}_i|^2)|z_i\rangle\langle z_i|$$

$$= |\mathcal{N}_{\chi_c}|^2(M-1)C = \mathcal{C}, \qquad (10)$$

where $\text{tr}(C) = 1/(|\mathcal{N}_{\chi_c}|^2(M - 1))$ since $\text{tr}(\text{tr}_1(|\chi_c\rangle\langle\chi_c|)) = 1$. The proximity measure $f(\psi_0) = 1 - \text{tr}(C)\text{tr}(\mathcal{C}|z_0\rangle\langle z_0|)$ can be measured using a standard swap test with $\mathcal{O}(\log d)$ copies of $|z_0\rangle$ and $\mathcal{C}$. For methods of generating the states $|z_0\rangle$, $|\chi_c\rangle$ using $\mathcal{O}(\log M)$ resources see Appendices B and C respectively.

B. Quantum one-class SVM (pure state)

Most classical data sets consist of real-valued numbers. In such a setting it is often sufficient to use the kernel function $k(\vec{x}_i, \vec{x}_j) = \vec{x}_i^T\vec{x}_j$ or its non-linear variants. For quantum states, it is beneficial to use well known fidelity measures such as $|\langle\psi_i|\psi_j\rangle|^2$ [26]. Thus, the kernel matrix corresponding to pure quantum states is defined by

$$K = \sum_{i,j=1}^{M}|\langle\psi_i|\psi_j\rangle|^2|i\rangle\langle j|. \qquad (11)$$

This kernel matrix is positive semi-definite, because the kernel function $|\langle\psi_i|\psi_j\rangle|^2$ can be represented as an inner product of an appropriately defined feature map on the states $|\psi_i\rangle$. Thus $\mathcal{K} = K/\text{tr}(K)$ can be considered as a density matrix, where $\text{tr}(K) = M$. See Appendix D for a derivation which also applies to mixed states.

Similarly to the classical algorithm for one-class SVM, one requires a matrix inversion algorithm before computing the proximity measure, which we demonstrate below. We first present an algorithm that computes the relevant proximity measure via a swap test using resources logarithmic in the dimensionality. Then we show an algorithm potentially using resources logarithmic in the dimensionality and the size of training data based on a quantum matrix inversion algorithm.

1. Algorithm using resources logarithmic in d

Suppose we look for an algorithm using $\mathcal{O}(\text{poly}(M\log(d)))$ resources when computing the proximity measure. Then it is possible to use the standard swap test to measure the fidelity $|\langle\psi_i|\psi_j\rangle|^2$, which costs $\sim \mathcal{O}(\log(d))$ in resources for each pair $i,j = 1,...,M$, or $\sim \mathcal{O}(M^2\log(d))$ resources for every i,j. Then to find the constants α_i necessary to compute the proximity measure, one can perform a classical matrix inversion, at cost $\mathcal{O}(M^3)$. Although use of only $\mathcal{O}(\log M)$ resources is not achieved in this protocol, it does not require the assumption of having access to U_C. It is sufficient to need only the unitaries $\{U_i\}$ and U_0. If the unitary U_C is available, we potentially obtain a logarithmic dependence in the number of quantum data as shown in the next section.

2. Algorithm using resources logarithmic in d and M

a. Matrix inversion algorithm

The problem is given by the linear system:

$$(K + P_T M\mathbf{1})\vec{\alpha} = r\vec{e}, \qquad (12)$$

where we set $r = 1$. We can convert this into its quantum counterpart beginning with

$$(K + P_T M \mathbf{1})|\vec{\alpha}\rangle = |\vec{e}\rangle, \tag{13}$$

where $|\vec{e}\rangle = (1/\sqrt{M}) \sum_{i=1}^{M} |i\rangle$, $|\vec{\alpha}\rangle = \mathcal{N}_\alpha \sum_{i=1}^{M} \alpha_i |i\rangle$ and $\mathcal{N}_\alpha \equiv 1/\sum_{j=1}^{M} |\alpha_j|^2$. We can then use the quantum matrix inversion algorithm (HHL) [35] to obtain $|\vec{\alpha}\rangle$ with a runtime of $\mathcal{O}(\log M)$. This algorithm requires the efficient exponentiation of K, which we show in Section III B 2 b.

The performance of the quantum matrix inversion algorithms relies on four basic conditions discussed in [35] and also summarized in [36]. First, the right hand side must be prepared efficiently. In the present case of the uniform superposition this step is easily performed via a Hadamard operation on all relevant qubits. Second, the quantum matrix inversion algorithm uses phase estimation as a subroutine and thus requires controlled operations of the unitary generated by the matrix under consideration. Here the controlled operations are made possible in the following way. We first rescale the overall matrix by the trace $\text{tr}(K) = M$ so we are now solving the problem

$$(\mathcal{K} + P_T \mathbf{1})|\vec{\alpha}\rangle = |\vec{e}\rangle. \tag{14}$$

Note that $\|\mathcal{K} + P_T \mathbf{1}\| = O(1)$ since P_T is a constant. Since $\mathcal{K}$ is a density matrix, we can simulate $\exp(-i\mathcal{K}t)$, with time t, via a variant of the quantum PCA algorithm [29], which we describe in Section III B 2 b. For simulating this gate to precision ϵ, this method requires $O(t^2/\epsilon)$ copies of $\mathcal{K}$, as long as K is sufficiently low-rank. The evolution $\exp(i P_T \mathbf{1} t)$ is trivial to generate.

As a third basic criterion, the condition number of the matrix should be at most $O(\log M)$. In our case the largest eigenvalue is $O(1 + P_T)$ while the smallest eigenvalue is P_T, since the kernel matrix is assumed to be low-rank with most eigenvalues being 0. Thus we require $1/P_T \lesssim \mathcal{O}(\log M)$. The fourth criterion is that measuring the desired anomaly quantifier requires at most $O(\log M)$ repetitions, which we show in Section III B 2 c.

b. Exponentiating the kernel matrix

The kernel matrix K can be related to another kernel matrix $K_0 = \sum_{i,j=1}^{M} |i\rangle\langle j|\langle \psi_i|\psi_j\rangle$ which has been used in previous discussions of the quantum support vector machine [5]. Note that

$$K = K_0^T * K_0, \tag{15}$$

where $*$ is the element wise (or Hadamard) product of two matrices, $(A * B)_{ij} = A_{ij} B_{ij}$. This fact allows us to simulate the matrix exponential $\exp(-i\mathcal{K}t)$ by using multiple copies of the density matrix $\mathcal{K}_0 = K_0/M$ in a modified quantum PCA method. With the efficiently simulable operator

$$S' = \sum_{j,k=1}^{N} |k\rangle\langle j| \otimes |j\rangle\langle k| \otimes |k\rangle\langle j|, \tag{16}$$

we can perform

$$\text{tr}_{1,2}\{e^{-iS'\Delta t}(\mathcal{K}_0 \otimes \mathcal{K}_0 \otimes \sigma)e^{iS'\Delta t}\}$$
$$= \sigma - i[(\mathcal{K}_0^T * \mathcal{K}_0), \sigma]\Delta t + O(\Delta t^2). \tag{17}$$

See Appendix E for a derivation. Concatenating these steps allows us to perform $\exp(-i\mathcal{K}t)$ to error $O(t^2/n)$ using $2n$ copies of $\mathcal{K}_0$.

One preparation method for $\mathcal{K}_0$ [5] is by starting with the state $|\chi\rangle = (1/\sqrt{M}) \sum_{i=1}^{M} |i\rangle|\psi_i\rangle$. Then the partial trace of $|\chi\rangle\langle\chi|$ with respect to second register gives

$$\text{tr}_2(|\chi\rangle\langle\chi|) = \frac{1}{M} \sum_{i,j=1}^{M} |i\rangle\langle j|\langle\psi_i|\psi_j\rangle = K_0/M = \mathcal{K}_0. \tag{18}$$

The state $|\chi\rangle$ can be generated by applying U_C to the uniform superposition

$$\frac{1}{\sqrt{M}} \sum_{i=1}^{M} |i\rangle|0\rangle \rightarrow \frac{1}{\sqrt{M}} \sum_{i=1}^{M} |i\rangle|\psi_i\rangle = |\chi\rangle. \tag{19}$$

c. Proximity measure computation

The proximity measure for the pure state $|\psi_0\rangle$ when $r = 1$ can be rewritten

$$f(\psi_0) = |\sum_{i=1}^{M} \alpha_i|\langle\psi_i|\psi_0\rangle|^2 - 1|$$
$$= |\langle\phi_1|\phi_2\rangle - 1|, \tag{20}$$
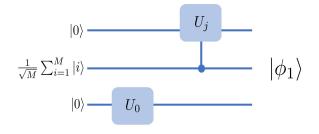
where

$$|\phi_1\rangle = \frac{1}{\sqrt{M}} \sum_{i=1}^{M} |\psi_i\rangle|i\rangle|\psi_0\rangle$$

$$|\phi_2\rangle = \frac{1}{\sqrt{M \sum_{j=1}^{M} |\alpha_j|^2}} |\psi_0\rangle \sum_{i=1}^{M} \alpha_i|i\rangle|\psi_i\rangle. \tag{21}$$

The state $|\phi_1\rangle$ can be generated by acting the unitary $\tilde{U}_1 = \sum_{j=1}^{M} U_j \otimes |j\rangle\langle j| \otimes U_0$ onto state $(1/\sqrt{M}) \sum_{i=1}^{M} |0\rangle|i\rangle|0\rangle$. The state $|\phi_2\rangle$ can likewise be generated by acting the unitary $\tilde{U}_2 = U_0 \otimes U_C$ onto state $|0\rangle|\vec{\alpha}\rangle|0\rangle$, where $U_0|0\rangle = |\psi_0\rangle$ and $U_i|0\rangle = |\psi_i\rangle$. See Fig. 1.

We can compute $f(\psi_0)$ if we can then find the overlap $\langle\phi_1|\phi_2\rangle$ in Eq. (20) using the same kind of modified swap test in Section III A 1. With access to the state $|\tilde{\phi}_R\rangle = (1/\sqrt{2})(|0\rangle|\phi_1\rangle + |1\rangle|\phi_2\rangle)$ and applying a
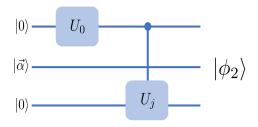
FIG. 1: Quantum circuits for generating states $|\phi_1\rangle$ and $|\phi_2\rangle$. See text for more details.

Hadamard on the ancilla qubit, the probability of measuring the first ancilla qubit in state $|1\rangle$ is given by $P_R = (1/2)(1 - \text{Re}\langle\phi_1|\phi_2\rangle)$. Similarly, with access with the state $|\tilde{\phi}_I\rangle = (1/\sqrt{2})(|0\rangle|\phi_1\rangle + i|1\rangle|\phi_2\rangle)$ and then applying a Hadamard on the ancilla qubit, the probability of measuring the first ancilla qubit in state $|1\rangle$ is given by $P_I = (1/2)(1 + \text{Im}\langle\phi_1|\phi_2\rangle)$. The states $|\tilde{\phi}_R\rangle$ and $|\tilde{\phi}_I\rangle$ can be created by applying the control unitary $\tilde{U}_c = |0\rangle\langle0| \otimes \tilde{U}_1(\mathbf{1} \otimes \tilde{H} \otimes \mathbf{1}) + |1\rangle\langle1| \otimes \tilde{U}_2$ onto the states $|A_R\rangle = (1/\sqrt{2})(|0\rangle|0\rangle|0\rangle|0\rangle + |1\rangle|0\rangle|\vec{\alpha}\rangle|0\rangle)$ and $|A_I\rangle = (1/\sqrt{2})(|0\rangle|0\rangle|0\rangle|0\rangle + i|1\rangle|0\rangle|\vec{\alpha}\rangle|0\rangle)$ respectively, where $\tilde{H}|0\rangle = (1/\sqrt{M})\sum_{i=1}^{M}|i\rangle$. See Appendix F for a method on generating states $|A_R\rangle$ and $|A_I\rangle$. Note that one way of implementing $\tilde{U}_c$ using U_C is via a two-path interferometer with $\tilde{U}_1(\mathbf{1} \otimes \tilde{H} \otimes \mathbf{1})$ acting along one path and $\tilde{U}_2$ acting along the other path.

An anomaly has been detected with error ϵ if we can find the value $f(\psi_0)$ to precision ϵ. $f(\psi_0)$ depends only on the inner product $\langle\phi_1|\phi_2\rangle$ and this comes from binary measurement outcomes with probabilities P_R and P_I. Since the measurement outcomes satisfy a Bernoulli distribution, the number of measurements $\mathcal{N}_f$ to find $\text{Re}\langle\phi_1|\phi_2\rangle$ and $\text{Im}\langle\phi_1|\phi_2\rangle$ to precision ϵ are

$$\mathcal{N}_f \sim \frac{P_R(1 - P_R)}{\epsilon^2} = \frac{1}{4\epsilon^2}(1 - (\text{Re}\langle\phi_1|\phi_2\rangle)^2) \lesssim \frac{1}{4\epsilon^2}$$

$$\mathcal{N}_f \sim \frac{P_I(1 - P_I)}{\epsilon^2} = \frac{1}{4\epsilon^2}(1 - (\text{Im}\langle\phi_1|\phi_2\rangle)^2) \lesssim \frac{1}{4\epsilon^2}. \quad (22)$$

Thus $\mathcal{N}_f \lesssim 1/(4\epsilon^2) \sim \mathcal{O}(\log(Md))$ if the error we tolerate is of order $\mathcal{O}(1/(\log(Md)))$.

IV. Mixed state anomaly detection

Up to this point, we have investigated anomaly detection for pure states. One can also consider the generalised problem of anomaly detection for mixed quantum states. This can find applications in the presence of unknown noise sources or when the output of a given quantum process is designed to be a mixed state, like in NMR [37].

The performance of pure state anomaly detection is potentially logarithmic in both the dimension of the Hilbert space and the number of pure states used for training. These exponential speedups are possible as long as conditional subroutines are available to prepare the pure states and additional requirements for quantum phase estimation and matrix inversion are satisfied. However, outputs of quantum devices are often likely to be statistical mixtures of pure states. It is then natural to extend anomaly detection to mixed states as described by density matrices.

For mixed states, we have to use state similarity measures different from the pures state ones discussed above. For the kernel PCA we propose an analogous measure for mixed states and an algorithm for finding this. We show how this algorithm can be executed using $\mathcal{O}(\log d)$ resources.

For the one-class SVM classification, one requires K to be a proper kernel measure and a positive semidefinite matrix. For this we show that a quantum state fidelity measure can be used as a kernel function. We show that so long as is interested only in an algorithm that runs using $\mathcal{O}(\log d)$ resources and not necessarily efficient in $\mathcal{O}(\log M)$, the kernel matrix can be prepared. The proximity measures corresponding to mixed states can then be computed using $\mathcal{O}(\log d)$ resources.

However, for a big quantum data 'speedup' in the number of training states, there are additional requirements to satisfy. For example, one needs an efficient way of *generating* multiple copies of a density matrix proportional to the kernel matrix for the mixed states, which take a more complex form. It is unclear how these can be generated with minimal assumptions. One might also require a simple experimental procedure for computing the corresponding proximity measure. These remain interesting challenges to be explored in future work.

A. Quantum kernel PCA (mixed state)

Suppose we want to find the proximity measure between the training states represented by density matrices ρ_i, $i = 1, ..., M$ and the test density matrix ρ_0. For $\kappa = 0, 1, ..., M$, we can define these density matrices as $\rho_\kappa = \sum_{l=1}^{N} E_l^{(\kappa)}|0\rangle\langle0|E_l^{(\kappa)\dagger}$, where $l = 1, ..., N$ and $E_l^{(\kappa)}$ are Kraus operators obeying $\sum_{l=1}^{N} E_l^{(\kappa)\dagger}E_l^{(\kappa)} = \mathbf{1}$.

We note that the pure state proximity measure in Eq. (9) is composed of $\mathcal{O}(\text{poly}(M))$ inner product terms

of the form $\langle\psi_i|\psi_\kappa\rangle$. We can denote a mixed state analogue of this inner product to be $\langle\rho_i,\rho_\kappa\rangle$. Then a proximity measure that reproduces Eq.(9) in the pure state limit can be written

$$f(\rho_0) = 1 - \frac{1}{M-1}\sum_{i=1}^{M}|\tilde{\mathcal{N}}_0|^2 \times$$

$$|\langle\rho_i,\rho_0\rangle - \sum_{j=1}^{M}\frac{(\langle\rho_i,\rho_j\rangle + \langle\rho_j,\rho_0\rangle)}{M} + \frac{1}{M^2}\sum_{j,l=1}^{M}\langle\rho_j,\rho_l\rangle|^2,$$
(23)

where $|\tilde{\mathcal{N}}_0|^2 = 1/(1 + 1/(M^2|\tilde{\mathcal{N}}_c|^2) - (2/M)\text{Re}[\sum_{i=1}^{M}\langle\rho_0,\rho_i\rangle/(M\tilde{\mathcal{N}}_c)]$ and $|\tilde{\mathcal{N}}_c|^2 = 1/\sum_{i,j=1}^{M}\langle\rho_i,\rho_j\rangle$, which reduce to $|\mathcal{N}_0|^2$ and $|\mathcal{N}_c|^2$ in the pure state limit. In the following, we define $\langle\rho_i,\rho_\kappa\rangle$ and propose a mixed state version of the modified swap test in Section III A 1 to find $\langle\rho_i,\rho_\kappa\rangle$.

It is important to observe that $\langle\rho_i,\rho_\kappa\rangle \neq \text{tr}(\rho_i\rho_\kappa)$, since $\text{tr}(\rho_i\rho_\kappa)$ reduces not to $\langle\psi_i|\psi_\kappa\rangle$, but to the fidelity $|\langle\psi_i|\psi_\kappa\rangle|^2$. Thus a simple swap test between ρ_i and ρ_κ is insufficient. Instead, we define

$$\langle\rho_i,\rho_\kappa\rangle \equiv \sum_{l=1}^{N}\langle 0|E_l^{(i)\dagger}E_l^{(\kappa)}|0\rangle,$$
(24)

which obeys all the properties of an inner product. In the pure state limit, $N = 1$ and the Kraus operators become unitary $E_l^{(\kappa)} \to U_\kappa, E_l^{(i)} \to U_i$, thus we can recover $\langle\psi_i|\psi_\kappa\rangle$.

To measure $\langle\rho_i,\rho_\kappa\rangle$, we start with an initial state $|0\rangle\langle 0| \otimes |0\rangle\langle 0|$ and apply a Hadamard to the ancilla state to get $\rho_{\text{in}} = |+\rangle\langle+| \otimes |0\rangle\langle 0|$. Suppose we are given access to quantum operations $\mathcal{E}^{(\kappa,i)}$ such that $\mathcal{E}^{(\kappa,i)}(\rho_{\text{in}}) = \sum_{l=1}^{N}P_l^{(\kappa,i)}\rho_{\text{in}}(P_l^{(\kappa,i)})^\dagger$ where $P_l^{(\kappa,i)} = |0\rangle\langle 0| \otimes E_l^{(\kappa)} + |1\rangle\langle 1| \otimes E_l^{(i)}$. Here $\sum_{l=1}^{N}(P_l^{(\kappa,i)})^\dagger P_l^{(\kappa,i)} = \mathbf{1}\otimes\mathbf{1}$ follows from completeness of the Kraus operators $E_l^{(\kappa)}$. After applying a Hadamard to the ancilla in $\mathcal{E}^{(\kappa,i)}(\rho_{\text{in}})$, the probability of finding the ancilla in state $|1\rangle$ is $(1/2)(1 - \text{Re}(\sum_{l=1}^{N}\langle 0|E_l^{(i)\dagger}E_l^{(\kappa)}|0\rangle))$. Similarly, the imaginary components $\text{Im}(\sum_{l=1}^{N}\langle 0|E_l^{(i)\dagger}E_l^{(\kappa)}|0\rangle)$ can be found by applying $(1/\sqrt{2})\left(\begin{smallmatrix}1 & 1\\ i & -i\end{smallmatrix}\right)$ to the ancilla instead of using the Hadamard gate.

Since the final measurement outcomes of the ancilla state satisfy a Bernoulli distribution, the number of measurements $\mathcal{N}$ required to estimate $\langle\rho_i,\rho_\kappa\rangle$ to precision ϵ is upper bounded by $\mathcal{O}(1/\epsilon^2)$, thus $\mathcal{N} \lesssim \mathcal{O}(\text{poly}(\log d))$ if the error tolerated is of order $\mathcal{O}(1/\log d)$. Since there are $\mathcal{O}(\text{poly}(M))$ of these terms to estimate for the new proximity measure $f(\rho_0)$, the total resource cost for the algorithm is $\mathcal{O}(\text{poly}(M\log d))$.

B. Quantum one-class SVM (mixed state)

We begin by identifying the kernel matrix to the superfidelity [38], which reduces to the fidelity in the pure

state limit. It is possible to show that the superfidelity $F(\rho_i,\rho_j) = \text{tr}(\rho_i\rho_j) + \sqrt{1 - \text{tr}(\rho_i^2)}\sqrt{1 - \text{tr}(\rho_j^2)}$ between states ρ_i and ρ_j obey all the properties of a kernel matrix. See Appendix D. We can then extend the proximity measure in Eq. (20) for pure states to mixed states

$$f(\rho_0) = \left|\sum_{i=1}^{M}\alpha_i\mathcal{F}_{i0}(\rho_i,\rho_0) - 1\right|^2.$$
(25)

If we can only access $\mathcal{O}(\text{poly}(M)\log(d))$ resources when computing the proximity measure, we can apply the same method as for pure states in Section III B 1. It is possible to use the same swap test, but now with mixed state inputs measuring $\text{tr}(\rho_i\rho_j)$. This again costs $\sim \mathcal{O}(\log d)$ in resources for each pair $i,j = 1,...,M$, or $\sim \mathcal{O}(M^2\log d)$ resources for every i,j. A classical algorithm for matrix inversion can be used, costing $\mathcal{O}(M^3)$ in resources, to find the constants α_i. Like the pure state case, we no longer require the assumption of having access to U_C.

V. Discussion

We have shown that there exist quantum machine learning algorithms for anomaly detection to detect outliers in both pure and mixed quantum states. This can be achieved using resources only logarithmic in the dimension of the quantum states. For pure states, the resources can also be logarithmic in the number of training quantum states used.

There is a wide variety of anomaly detection algorithms based on machine learning which can be extended to the quantum realm and remain yet unexplored. These new quantum algorithms might find applications not only in identifying novel quantum phenomena, but also in secure quantum data transfer, secure quantum computation and verification over the cloud. These questions may become even more important as cloud quantum computing systems evolve into a quantum internet.

A. Generating centroid quantum state $|\psi_c\rangle$

Assume we are given the state $|\chi\rangle = (1/\sqrt{M})\sum_{i=1}^{M}|\psi_i\rangle|i\rangle$. We can apply a Hadamard operation H on each of the qubits of the second register so $|\chi\rangle \rightarrow (1/M)\sum_{i=1}^{M}|\psi_i\rangle\sum_{j=1}^{M}(-1)^{ij}|j\rangle$. Then by making a projective measurement $|0\ldots0\rangle\langle0\ldots0|$ on the second register we can recover the centroid quantum state $|\psi_c\rangle|0\ldots0\rangle$.

The success probability $\mathcal{P}_\chi$ of this measurement is

$$\mathcal{P}_\chi = \text{tr}(|0\ldots0\rangle\langle0\ldots0|H^{\otimes\log M}|\chi\rangle\langle\chi|H^{\otimes\log M})$$

$$= \frac{1}{M^2}\sum_{i,j=1}^{M}\langle\psi_i|\psi_j\rangle. \tag{A1}$$

The use of $\mathcal{O}(\log M)$ Hadamard gates means that our gate resource count is $\mathcal{O}(\log M)$.

In the scenario where the covariance matrix C is low rank, we expect most training samples to be rather similar, i.e., $\langle\psi_i|\psi_j\rangle \sim$ constant. This is also the case in anomaly detection where we expect the training samples to belong to a single 'type' and hence expect high state fidelities between these states. In this case, $\mathcal{P}_\chi \sim \mathcal{O}(1)$.

We observe that Eq. (A1) also allows us to recover $|\mathcal{N}_c|^2 = 1/(M^2\mathcal{P}_\chi)$ using $\mathcal{O}(\log M)$ resources.

B. Generating centered quantum states $|z_i\rangle$ and $|z_0\rangle$

Assume the state preparation operations as discussed in the main part. First we prepare

$$\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)|0\ldots0\rangle \rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle|i\rangle - \frac{1}{\sqrt{M}}|1\rangle\sum_{j=1}^{M}|j\rangle\right) \tag{B1}$$

which we obtain by performing $|0\rangle \rightarrow |i\rangle$ conditioned on the first ancilla being in $|0\rangle$ and performing a Hadamard conditioned on the first ancilla being in $|1\rangle$. Note that the later operation $|0...0\rangle \rightarrow (1/\sqrt{M})\sum_{j=1}^{M}|j\rangle$ requires $\mathcal{O}(\log M)$ Hadamard gates, thus the gate resource count is $\mathcal{O}(\log M)$.

Now we use U_C conditioned on the non-ancilla states to prepare

$$\frac{1}{\sqrt{2}}\left(|0\rangle|i\rangle - \frac{1}{\sqrt{M}}|1\rangle\sum_{j=1}^{M}|j\rangle\right)|0\ldots0\rangle$$

$$\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle|i\rangle|\psi_i\rangle - \frac{1}{\sqrt{M}}|1\rangle\sum_{j=1}^{M}|j\rangle|\psi_j\rangle\right). \tag{B2}$$

Now perform another Hadamard conditioned on the an-

cilla being in $|1\rangle$ to arrive at

$$\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle|i\rangle|\psi_i\rangle - \frac{1}{M}|1\rangle\sum_{j,k=1}^{M}(-1)^{j\cdot k}|k\rangle|\psi_j\rangle\right). \tag{B3}$$

Conditioned on the ancilla being in $|0\rangle$ uncompute the $|i\rangle$

$$\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle|0\rangle|\psi_i\rangle - \frac{1}{M}|1\rangle\sum_{j,k=1}^{M}(-1)^{j\cdot k}|k\rangle|\psi_j\rangle\right). \tag{B4}$$

Measure $|0\cdots0\rangle$ in the label register to obtain

$$\rightarrow \frac{1}{\sqrt{1+\frac{1}{M^2}\sum_{j,j'=1}^{M}\langle\psi_{j'}|\psi_j\rangle}}\left(|0\rangle|\psi_i\rangle - \frac{1}{M}|1\rangle\sum_{j=1}^{M}|\psi_j\rangle\right). \tag{B5}$$

The success probability of this measurement is then given by

$$P_{0\cdots0} = \frac{1}{2}\left(1 + \frac{1}{M^2}\sum_{j,j'=1}^{M}\langle\psi_{j'}|\psi_j\rangle\right). \tag{B6}$$

As the final step measure the ancilla in $(|0\rangle + |1\rangle)/\sqrt{2}$ to obtain

$$\rightarrow \mathcal{N}_i\left(|\psi_i\rangle - \frac{1}{M}\sum_{j=1}^{M}|\psi_j\rangle\right) \equiv |z_i\rangle, \tag{B7}$$

where $|\mathcal{N}_i|^2 = 1/(1 - 2\text{Re}[\sum_{j=1}^{M}\langle\psi_i|\psi_j\rangle]/M + \sum_{j,k=1}^{M}\langle\psi_j|\psi_k\rangle)$. The success probability of the final measurement is given by

$$P_0^{(i)} = \frac{1}{2(1+\frac{1}{M^2}\sum_{j,j'=1}^{M}\langle\psi_{j'}|\psi_j\rangle)} \times$$
$$\left(1 - \frac{2}{M}\sum_{j=1}^{M}\text{Re}(\langle\psi_i|\psi_j\rangle) + \frac{1}{M^2}\sum_{j,j'=1}^{M}\langle\psi_{j'}|\psi_j\rangle\right) \tag{B8}$$

Thus the total probability of success is $P_{0\cdots0}P_0^{(i)}$, which scales as $\sim \mathcal{O}(1)$ following a similar argument to Appendix A. Note that in the limit where the training states are exactly the same, the state $|z_i\rangle$ does not exist, i.e., $|z_i\rangle = 0$ and thus the success probability $P_0^{(i)} = 0$ as expected.

C. Generating state $|\chi_c\rangle$ for kernel PCA

The state preparation of $|\chi_c\rangle$ works essentially the same way as in Appendix B, except that we need a superposition of the label i. We prepare by calling

the controlled state preparation operation and controlled Hadamards in the following way

$$\frac{1}{\sqrt{2M}} \sum_{i=1}^{M} |i\rangle \left(|0\rangle - |1\rangle\right) |0\ldots 0\rangle |0\ldots 0\rangle$$

$$\rightarrow \frac{1}{\sqrt{2M}} \times$$

$$\sum_{i=1}^{M} |i\rangle \left(|0\rangle |0\ldots 0\rangle |\psi_i\rangle - \frac{1}{\sqrt{M}} |1\rangle \sum_{j=1}^{M} |j\rangle |0\ldots 0\rangle \right)$$
(C1)

$$\rightarrow \frac{1}{\sqrt{2M}} \sum_{i=1}^{M} |i\rangle \left(|0\rangle |0\ldots 0\rangle |\psi_i\rangle - \frac{1}{\sqrt{M}} |1\rangle \sum_{j=1}^{M} |j\rangle |\psi_j\rangle \right)$$

$$\rightarrow \frac{1}{\sqrt{2M}} \times$$

$$\sum_{i=1}^{M} |i\rangle \left(|0\rangle |0\ldots 0\rangle |\psi_i\rangle - \frac{1}{M} |1\rangle \sum_{j,k=1}^{M} (-1)^{j \cdot k} |k\rangle |\psi_j\rangle \right).$$
(C2)

Here, like in Appendices A and B we employed $\mathcal{O}(\log M)$ Hadamard gates for the transformation $|0...0\rangle \rightarrow (1/\sqrt{M}) \sum_{j=1}^{M} |j\rangle$, thus our gate resource count scale as $\mathcal{O}(\log M)$.

We then measure $|0 \cdots 0\rangle$ in the label register to obtain the state

$$\frac{\sum_{i=1}^{M} |i\rangle \left(|0\rangle |\psi_i\rangle - (1/M) |1\rangle \sum_{j=1}^{M} |\psi_j\rangle \right)}{\sqrt{M + 1/M \sum_{j,j'=1}^{M} \langle \psi_j | \psi_{j'} \rangle}}$$
(C3)

The success probability of this measurement is given by

$$P_{0\cdots 0} = \frac{1}{2} + \frac{1}{2M^2} \sum_{j,j'=1}^{M} \langle \psi_j | \psi_{j'} \rangle$$
(C4)

As the final step measure the ancilla in $(|0\rangle + |1\rangle)/\sqrt{2}$ to obtain:

$$\rightarrow \mathcal{N}_c \sum_{i=1}^{M} |i\rangle \left(|\psi_i\rangle - \frac{1}{M} \sum_{j=1}^{M} |\psi_j\rangle \right) \equiv |\chi_c\rangle,$$
(C5)

where $|\mathcal{N}_c|^2 = 1/(M - 1/M \sum_{j,j'=1}^{M} \langle \psi_j | \psi'_j \rangle])$. The success probability of the final measurement is then given by

$$P_0 = \frac{1 - 1/M^2 \sum_{j,j'=1}^{M} \langle \psi_j | \psi'_j \rangle}{1 + 1/M^2 \sum_{j,j'=1}^{M} \langle \psi_j | \psi_{j'} \rangle}$$
(C6)

where $P_0 \sim \mathcal{O}(1)$ following a similar argument to Appendix A.

D. Positive semi-definite kernel matrix K and fidelity measure

Let $k(A, B)$ be a kernel function, where A and B are matrices. This kernel function is called positive semidefinite if the kernel matrix $K_{ij} = k(\rho_i, \rho_j)$ is positive semidefinite for any training set $\{\rho_i\}$, where ρ_i are density matrices. For showing that a kernel function corresponds to a positive kernel matrix, we have by definition to show that $\sum_{ij} c_i c_j K_{ij} \geq 0$ for all vectors $\vec{c}$ and all training sets. Alternatively, positivity holds if there exist some feature map function $\phi(\rho_i)$ such that the kernel matrix can be defined as a matrix of inner products $K_{ij} = \phi(\rho_i)^\dagger \phi(\rho_j)$.

A large number of density matrix fidelity and distance measures have been discussed [38–41]. Here, we take the kernel function given by the *superfidelity* [38], defined by the symmetric function

$$F(\rho_i, \rho_j) = \text{tr}(\rho_i \rho_j) + \sqrt{1 - \text{tr}(\rho_i^2)} \sqrt{1 - \text{tr}(\rho_j^2)}. \quad (D1)$$

The matrix entries of superfidelity are all *real*, since the expression $\text{tr}(\rho_i \rho_j)$ is real and positive. We can see this by observing there exist matrices A, B such that $\rho_i = A^T A$ and $\rho_j = B^T B$, since ρ_i, ρ_j are all positive semidefinite. Then $\text{tr}(\rho_i \rho_j) = \text{tr}((BA^T)(AB^T)) = \text{tr}(C^T C)$ where $C = AB^T$ and $C^T C$ is by definition positive semidefinite. It also turns out that the super fidelity can indeed be written via a feature map, by noting that

$$\text{tr}(\rho_i \rho_j) = \vec{\rho}_i^\dagger \vec{\rho}_j, \quad (D2)$$

where $\vec{\rho}$ is a vectorized form of a matrix ρ by stacking the columns of ρ. So the feature map is

$$\phi(\rho_i) = \begin{pmatrix} \vec{\rho}_i \\ \sqrt{1 - \text{tr}(\rho_i^2)} \end{pmatrix} \quad (D3)$$

and the inner product

$$\phi(\rho_i)^\dagger \phi(\rho_j)$$
$$= \vec{\rho}_i^\dagger \vec{\rho}_j + \sqrt{1 - \text{tr}(\vec{\rho}_i^2)} \sqrt{1 - \text{tr}(\vec{\rho}_j^2)} = K_{ij} \quad (D4)$$

leads to a valid positive kernel matrix.

E. Hadamard product and transpose

Given two matrices ρ_1 and ρ_2. The task is to apply the element-wise product of these matrices as a Hamiltonian to another quantum state. This is for the purpose of simulating the kernel matrix related to the one-class SVM.

The standard swap matrix for quantum PCA is $S = \sum_{j,k=1}^{M} |j\rangle\langle k| \otimes |k\rangle\langle j|$. Now, take the modified swap operator

$$S' = \sum_{j,k=1}^{M} |k\rangle\langle j| \otimes |j\rangle\langle k| \otimes |k\rangle\langle j|. \quad (E1)$$

It is at most 1-sparse because each $|jkj\rangle$ gets mapped to a unique $|kjk\rangle$ while states $|jkl\rangle$ get mapped to zero for $j \neq k \neq l$. It is also efficiently computable, thus efficiently simulatable as a Hamiltonian. With an arbitrary state σ, we can perform the following operation

$$\mathrm{tr}_{1,2}\{e^{-iS'\Delta t}(\rho_1 \otimes \rho_2 \otimes \sigma)e^{iS'\Delta t}\}. \qquad (E2)$$

We perform the infinitesimal operation with the matrix S'. The trace is over the ρ_1 and ρ_2 subspaces. Expanding to $O(\Delta t^2)$ leads to

$$\mathrm{tr}_{1,2}\{e^{-iS'\Delta t}(\rho_1 \otimes \rho_2 \otimes \sigma)e^{iS'\Delta t}\} =$$
$$1 - i\,\mathrm{tr}_{1,2}\{S'(\rho_1 \otimes \rho_2 \otimes \sigma)\}\Delta t$$
$$+ i\,\mathrm{tr}_{1,2}\{(\rho_1 \otimes \rho_2 \otimes \sigma)S'\}\Delta t + O(\Delta t^2). \qquad (E3)$$

The first $O(\Delta t)$ term is

$$\mathrm{tr}_{1,2}\{S'(\rho_1 \otimes \rho_2 \otimes \sigma)\}$$
$$= \mathrm{tr}_{1,2}\{\sum_{j,k=1}^{M} |k\rangle\langle j| \otimes |j\rangle\langle k| \otimes |k\rangle\langle j|(\rho_1 \otimes \rho_2 \otimes \sigma)\}$$
$$= \sum_{n,m,j,k=1}^{M} \langle n|k\rangle\langle j|\rho_1|n\rangle\langle m|j\rangle\langle k|\rho_2|m\rangle|k\rangle\langle j|\sigma$$
$$= \sum_{j,k=1}^{M} \langle j|\rho_1|k\rangle\langle k|\rho_2|j\rangle|k\rangle\langle j|\sigma$$
$$= (\rho_1^T * \rho_2)\sigma. \qquad (E4)$$

In the same manner we can show

$$\mathrm{tr}_{1,2}\{(\rho_1 \otimes \rho_2 \otimes \sigma)S'\} = \sigma(\rho_1^T * \rho_2). \qquad (E5)$$

Thus in summary, we have shown that

$$\mathrm{tr}_{1,2}\{e^{-iS'\Delta t}(\rho_1 \otimes \rho_2 \otimes \sigma)e^{iS'\Delta t}\}$$
$$= \sigma - i[(\rho_1^T * \rho_2), \sigma]\Delta t + O(\Delta t^2). \qquad (E6)$$

F. Generating states $|A_R\rangle$ and $|A_I\rangle$

In Sec. III B 2 c, we required the generation of states $|A_R\rangle = (1/\sqrt{2})(|0\rangle|0\rangle|0\rangle|0\rangle + |1\rangle|0\rangle|\vec{\alpha}\rangle|0\rangle)$ and $|A_I\rangle = (1/\sqrt{2})(|0\rangle|0\rangle|0\rangle|0\rangle + i|1\rangle|0\rangle|\vec{\alpha}\rangle|0\rangle)$. We begin with a control-unitary operation creating the state

$$(|0\rangle + |1\rangle)|000\rangle|0\rangle$$
$$\longrightarrow |0\rangle|000\rangle|1\rangle + |1\rangle|0\vec{e}0\rangle\mathcal{R}|0\rangle, \qquad (F1)$$

where $|\vec{e}\rangle = (1/\sqrt{M})\sum_{i=1}^{M}|i\rangle$ and $\mathcal{R}$ is the control-rotation on the ancilla (right-most qubit) used in the quantum matrix inversion algorithm [35]. Upon measuring this ancilla in the state $|1\rangle$

$$|0\rangle|000\rangle|1\rangle + |1\rangle|0\vec{e}0\rangle\mathcal{R}|0\rangle$$
$$\longrightarrow |0\rangle|000\rangle|1\rangle + |1\rangle|0\vec{\alpha}0\rangle|1\rangle. \qquad (F2)$$

Thus by eliminating the ancilla in state $|1\rangle$, we can generate $|A_R\rangle$. Similarly, if we instead begin with state $(1/\sqrt{2})(|0\rangle + i|1\rangle)|000\rangle|0\rangle$ in Eq. (F1), then we can generate the state $|A_I\rangle$.

[1] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Nature **549**, 195 (2017).
[2] V. Dunjko, J. M. Taylor, and H. J. Briegel, Phys. Rev. Lett. **117**, 130501 (2016).
[3] V. Dunjko and H. J. Briegel, arXiv:1709.02779 (2017).
[4] C. Ciliberto, M. Herbster, A. D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossnig, arXiv:1707.08561 (2017).
[5] P. Rebentrost, M. Mohseni, and S. Lloyd, Phys. Rev. Lett. **113**, 130503 (2014).
[6] S. Lloyd, M. Mohseni, and P. Rebentrost, arXiv:1307.0411 (2013).
[7] N. Wiebe, D. Braun, and S. Lloyd, Phys. Rev. Lett **109**, 050505 (2012).
[8] N. Wiebe, A. Kapoor, and K. Svore, arXiv:1401.2142 (2014).
[9] M. Schuld, I. Sinayskiy, and F. Petruccione, Phys. Rev. A **94**, 022342 (2016).
[10] E. Aïmeur, G. Brassard, and S. Gambs, in *Proceedings of the 24th international conference on machine learning* (ACM, 2007), pp. 1–8.
[11] E. Aïmeur, G. Brassard, and S. Gambs, Machine Learn-
ing **90**, 261 (2013).
[12] V. Chandola, A. Banerjee, and V. Kumar, ACM computing surveys (CSUR) **41**, 15 (2009).
[13] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, Signal Processing **99**, 215 (2014).
[14] S. Hara, T. Ono, R. Okamoto, T. Washio, and S. Takeuchi, Phys. Rev. A **89**, 022104 (2014).
[15] S. Hara, T. Ono, R. Okamoto, T. Washio, and S. Takeuchi, Phys. Rev. A **94**, 042341 (2016).
[16] V. Giovannetti, S. Lloyd, and L. Maccone, Phys. Rev. Lett. **100**, 160501 (2008).
[17] A. Monràs, G. Sentís, and P. Wittek, Phys. Rev. Lett. **118**, 190503 (2017).
[18] G. Sentís, J. Calsamiglia, R. Munoz-Tapia, and E. Bagan, Sci. Rep. **2**, 708 (2012).
[19] A. Bisio, G. Chiribella, G. M. DAriano, S. Facchini, and P. Perinotti, Phys. Rev. A **81**, 032324 (2010).
[20] M. Sasaki, A. Carlini, and R. Jozsa, Phys. Rev. A **64**, 022317 (2001).
[21] F. Shahi and A. T. Rezakhani, arXiv:1704.01965 (2017).
[22] H. Hoffmann, Pattern Recognit. **40**, 863 (2007).
[23] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and

L. Chang, Tech. Rep., DTIC Document (2003).

[24] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt, in *Advances in neural information processing systems* (2000), pp. 582–588.

[25] Y.-S. Choi, Pattern Recogn. Lett. **30**, 1236 (2009).

[26] M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, UK (2010).

[27] M. Basseville, I. V. Nikiforov, et al., *Detection of abrupt changes: theory and application*, vol. 104 (Prentice Hall Englewood Cliffs, 1993).

[28] G. Sentís, E. Bagan, J. Calsamiglia, G. Chiribella, and R. Munoz-Tapia, Phys. Rev. Lett. **117**, 150502 (2016).

[29] S. Lloyd, M. Mohseni, and P. Rebentrost, Nature Phys. **10**, 631 (2014).

[30] R. Chatterjee and T. Yu, arXiv:1612.03713 (2016).

[31] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, Neural Comput. **13**, 1443 (2001).

[32] K. Muandet and B. Schölkopf, arXiv:1303.0309 (2013).

[33] J. Ma and S. Perkins, in *Neural Networks, 2003. Proceedings of the International Joint Conference on* (IEEE, 2003), vol. 3, pp. 1741–1745.

[34] H. Buhrman, R. Cleve, J. Watrous, and R. De Wolf, Phys. Rev. Lett. **87**, 167902 (2001).

[35] A. W. Harrow, A. Hassidim, and S. Lloyd, Phys. Rev. Lett. **103**, 150502 (2009).

[36] S. Aaronson, Nature Phys. **11**, 291 (2015).

[37] W. S. Warren, Science **277**, 1688 (1997).

[38] J. L. A. Miszczak, Z. P. LA, P. Horodecki, A. Uhlmann, and K. Zyczkowski, Quantum Inf. Comput. **9**, 0103 (2009).

[39] R. Jozsa, J. Mod. Opt. **41**, 2315 (1994).

[40] X. Wang, C.-S. Yu, and X. Yi, Phys. Lett. A **373**, 58 (2008).

[41] Z. Puchała, J. A. Miszczak, P. Gawron, and B. Gardas, Quantum Inf. Process **10**, 1 (2011).

[42] M. Araújo, A. Feix, F. Costa, and Č. Brukner, New J. Phys. **16**, 093026 (2014).

[43] For a simplified version of a supervised learning algorithm for quantum states, see [21]. Also see [21] for an unsupervised learning algorithm for quantum states.

[44] We note that the special case $c = 0$ can be interpreted as when the training data are all clustered around the origin of the feature space. However, it is not an appropriate choice, since solving for $(K + P_T M\mathbf{1})\vec{\alpha} = \vec{0}$ would imply the positive semidefinite matrix K has negative eigenvalues $-P_T M$, which is a contradiction.

[45] For example, in a photonic setting, one can act u_1 along one path of an interferometer and u_2 acting along the other path.

[46] We note that U_C can be implemented for example using passive linear optical elements [42].