



Title	A study on Quantum-Classical Inter-Domain Classification Method
Author(s)	野原, 大靖
Degree Grantor	北海道大学
Degree Name	博士(情報科学)
Dissertation Number	甲第17043号
Issue Date	2026-06-30
DOI	https://doi.org/10.14943/doctoral.k17043
Doc URL	https://hdl.handle.net/2115/100096
Type	doctoral thesis
File Information	Taisei_Nohara.pdf, 全文



A study on Quantum-Classical Inter-Domain Classification Method

Taisei Nohara

Graduate School of Information Science and Technology
Hokkaido University
April, 2026

Contents

1	Introduction	1
1.1	Background	1
1.2	Contributions	3
1.3	Organization	4
2	Preliminaries	5
2.1	Notation	5
2.2	Problem setting	6
2.3	Hilbert space and kernel method	7
2.4	Linear model	8
2.5	Support Vector Machines	8
2.6	Neural Networks	8
2.7	One-class classification methods	9
2.7.1	Support vector data description	9
2.7.2	Deep SVDD	9
2.8	Positive-unlabeled learning	9
2.8.1	Background in positive-unlabeled learning	9
2.8.2	Unbiased PU learning and non-negative PU learning	10
2.9	Quantum computation	10
2.9.1	Quantum state and density matrix representation	10
2.9.2	Properties of Quantum Computation	12
2.9.3	Quantum circuit	12
2.9.4	HHL algorithm	13
2.10	Quantum machine learning	14
2.10.1	Four regions of approaches in quantum machine learning field	14
2.10.2	Introduction to chapter 3	14
2.10.3	Introduction to chapter 4	17
3	Quantum Support Vector Machine for Pairwise Classification Problem	19
3.1	Background	19
3.1.1	Pairwise classification problem	19
3.1.2	Kernel method	19
3.1.3	Least squares Support Vector Machine	20
3.1.4	Quantum Support Vector Machine	22

3.1.5	Kronecker kernel	24
3.1.6	Computational complexity for solving pairwise classification problem with Kronecker kernel	24
3.2	Proposed model	25
3.3	Experiment	27
3.3.1	Setup	27
3.3.2	Implementation	28
3.3.3	Results	30
3.4	Discussion	34
3.5	Conclusion	34
4	Positive-Unlabeled Learning for Entanglement Detection	36
4.1	Background	37
4.1.1	Problem setting	37
4.1.2	Detailed formalization of entanglement detection	37
4.1.3	Existing approaches for entanglement detection	38
4.2	Training entanglement detector using positive-unlabeled learning framework	40
4.2.1	Proposed method	40
4.2.2	Data generation	42
4.3	Related work	43
4.4	Numerical experiments	46
4.4.1	Methodology	46
4.4.2	Architectures	47
4.4.3	Setup	47
4.4.4	Results	48
4.4.5	Ablation studies	48
4.5	Discussion	51
4.6	Conclusion	52
5	Discussion	54
6	Conclusion	57
A	Calculation of inner product	59
B	Effect of sampling error	61
	References	64

List of Figures

2.1	Classification of problem settings based on the availability of negative label information and the existence of unlabeled data samples in the training dataset	11
2.2	A sample quantum circuit	12
3.1	Outline of quantum SVM model	22
3.2	Training part in existing quantum SVM model	23
3.3	Testing part in existing quantum SVM model	23
3.4	Training part in proposed quantum SVM model	26
3.5	Testing part in proposed quantum SVM model	26
3.6	Samples in basic OCR image dataset	28
3.7	Outline of implementation of proposed model	28
3.8	Training part implementation of proposed model	29
3.9	Testing part implementation of proposed model	30
3.10	The accuracy with the classical model and the proposed model for different values of γ	32
3.11	Execution time with the classical model and the proposed model for different values of γ	33
4.1	Illustrations of types of entanglement in two-qubit and three-qubit systems	38
4.2	Overview of proposed method	40
4.3	Processes for generating separable and entangled states	42

List of Tables

2.1	Notation for set	5
2.2	Notation for complex numbers	5
2.3	Notation for matrices	5
2.4	Four regions of approaches in quantum machine learning field	14
4.1	Selected studies and a comparison on entanglement detection using machine learning methods	44
4.2	Network architecture used in numerical experiments	47
4.3	Experimental results	49
4.4	Experimental results in ablation studies	49
4.5	Metrics for estimating the difficulty of training datasets	51
4.6	Additional experimental results in ablation studies	51

Acknowledgements

While writing my doctoral dissertation, I received invaluable advice from my advisor, Professor Itsuki Noda of the Intelligent Software Laboratory. In addition to engaging in extensive discussions during regular laboratory seminars, he met with me one-on-one to discuss the research content. Furthermore, he provided significant support in coordinating the schedule for my dissertation defense.

I also received a great deal of advice from Associate Professor Hiroki Sakaji during laboratory seminar discussions. By posing questions from a neutral perspective, he helped me clarify the core contributions of my research. During our one-on-one meetings, he offered advice on how to proceed with my research.

I would like to thank Professors Masahito Yamamoto, Hidenori Kawamura, and Daisuke Sakamoto for their feedback on my presentation.

I would also like to express my deepest respect to Professor Oyama of Nagoya City University. He was my advisor during my undergraduate and master's programs, and we held weekly online meetings during my doctoral studies. Our discussions were a major driving force behind the progress of my research. Even when my research wasn't going as planned, he provided constructive and steady advice, which was a great source of mental support.

The members of the Intelligent Software Laboratory also supported me in many ways as I pursued my doctoral research. Discussions in seminars and casual daily conversations improved my experience at the laboratory. I was inspired by the other members, and we shared information with one another. It was a positive environment where we supported each other equally.

Finally, I would like to express my gratitude to my family and friends for their support. Without them, I would not have been able to devote myself to my research and complete this dissertation.

Chapter 1

Introduction

1.1 Background

Quantum machine learning is a field of study that focuses on the relationship between quantum computation and machine learning. Quantum computation is a form of computation based on quantum mechanics and it has some characteristics, such as superposition and entanglement. These properties appear uniquely in quantum computation but not in *classical* computation—a retronym for traditional computation, which is based on classical mechanics. As an early study relevant to quantum machine learning, there is research combining the concepts of quantum computation and neural networks, which was published in 1995 [1]. There are several areas of interest in the field of quantum machine learning, as follows.

The acceleration of machine learning methods by quantum computation in the future. Quantum computation is expected to be useful for the speed-up of calculation in machine learning methods from some discoveries which imply that quantum computation may solve some problems faster than existing methods. Examples include the prime factorization algorithm, also known as Shor’s algorithm [2], and the search algorithm for an unsorted array, also known as Grover’s algorithm [3]. These algorithms are based on the fast subroutines that were discovered in quantum computation, such as the quantum Fourier transform.

Analysis of data derived from quantum computation using machine learning. The idea is that learning from quantum states in a quantum computer is straightforward. However, this natural approach is hindered by the lack of error-correcting capabilities in real quantum computer devices at present. Therefore, an alternative approach is to process data obtained through quantum computation on traditional computers and analyze it using machine learning techniques. Entanglement detection is an example of such approaches [4, 5]. This can be viewed as a classification problem that takes a certain (quantum) state arising during quantum computation as input and classify that state into several patterns. Then, existing classification methods can be applied. Research also exists within the framework of learning theory. For example, efficient estimation methods are being studied for quantum state tomography, which involves estimating the input quantum state from the outputs obtained through specific operations [6].

Quantum-inspired machine learning methods. One research direction is to build machine

learning algorithms that are inspired by, or emulate, quantum machine learning algorithms. One example in that direction is based on recommendation systems using quantum machine learning. Although the main result of the work weakens the impact of speed-up achieved by utilizing quantum machine learning, it underscores the importance of grounded discussions that rigorously evaluate the advantages of quantum computers.

Approaches in the field of quantum machine learning field can be divided into four regions by their problem settings and methods (see Figure 2.4). In those four regions, approaches spanning quantum computation and machine learning are important for deepening our understanding for quantum computation and machine learning, with respect to the benefits do they bring to each other. For example, applying a quantum algorithm for classical data is a kind of approach with classical problem setting and quantum method (CQ), which is expected to make it feasible to apply methods for very high-dimensional data. Another example is the analysis of quantum state data (QC), which is an approach with quantum problem setting and classical method. This approach is expected to help verify new input quantum states and quantum algorithms. These approaches have the potential to yield new methods, which are not found in the methods from the same field of problem setting. However, there are challenges we have to address in these approaches.

- (i) In CQ: methods utilizing the common properties of the quantum computational model and problem settings. Existing research on the CQ approach have produced models that replicate the results of classical machine learning models while outsourcing some or all of their computations to quantum computers. These methods rely on general-purpose quantum algorithms, such as the Harrow-Hassidim-Lloyd (HHL) algorithm, which is used for inverse matrix calculations. Conversely, such methods have little consideration for utilizing the structure of the quantum computational model itself. Quantum computational models are usually described using specific structural constraints, such as normalized vectors, Hermitian matrices, or unitary matrices. Depending on the properties of the problem formulation, combining it with the structure of the quantum computational model could lead to improvements in computational complexity. Discovering methods based on the CQ approach that focus on the commonalities between quantum computational models and problem settings deepens our understanding of the quantum machine learning field. This can be achieved by identifying domains in which the CQ approach is more practical and suggesting new ways to improve its performance.
- (ii) In QC: the difficulty of labeling in a specific class. Most existing research on the QC approach uses methods that assume datasets consisting of input-output pairs are available, similar to conventional supervised machine learning. However, depending on the problem formulation, labeling data derived from quantum computation can be challenging to achieve experimentally. In traditional machine learning problem settings, such as when images or strings are provided as inputs, labeling is straightforward because humans can visually determine the correct labels. In problem settings such as entanglement detection, however, where the input is a quantum state represented as a Hermitian matrix and the output is a label uniquely determined by the

input, accurately labeling any given input requires solving an NP-hard problem. This difficulty persists not only when the state is represented on a quantum computer, but also in simpler cases where an exact classical representation of the quantum state is available. Although this challenge of utilizing experimental data has largely been overlooked in existing research, it is a critical issue that must be resolved to leverage the high performance of machine learning in data analysis.

1.2 Contributions

The goal of this dissertation is to develop quantum machine learning methods that are more universally applicable than existing methods by linking quantum computation and machine learning through their mathematical properties. The contributions of this dissertation are as follows.

- (i) We propose a learning algorithm with better scalability compared to existing methods for pairwise classification problems. Pairwise classification is a type of machine learning problem setting, arising in scenarios like link prediction and chemical interaction prediction. It can also be viewed as an extension of standard classification problems. Methods utilizing feature combinations have been explored for pairwise classification, and the Kronecker kernel is one useful approach, enabling kernel-based feature mapping applied to feature pairs. However, the Kronecker kernel suffers from poor scalability because its kernel matrix grows to $n^2 \times n^2$ dimensions, where n represents the number of data points per feature pair, as the number of rows and columns increases proportionally to the number of feature combinations. A key characteristic of the Kronecker kernel is that its kernel matrix can be expressed as the Kronecker product of two matrices. Since this Kronecker product operation frequently appears in quantum computation, the proposed method leverages the fact that the Kronecker kernel can be incorporated into quantum support vector machines with minimal modification to improve computational scalability. Furthermore, we implemented the proposed method on a quantum computation simulator and evaluated numerically on small-scale datasets, confirming that it achieves performance comparable to existing methods.
- (ii) We propose a method that avoids the labeling challenges inherent in existing machine learning approaches for the entanglement detection problem, which is known to be an NP-hard problem. Entanglement is one of the properties of quantum computation and is utilized, for example, in quantum communication. In quantum communication, detecting entanglement is a practical problem, serving as a method to detect when communication is not functioning properly. Approaches to entanglement detection are diverse, but recent research has focused on the versatility of machine learning methods for this task. However, much of this research scarcely addresses dataset-related issues, particularly the availability of data for entangled quantum states (accessible on classical computers) - in other words, the difficulty of labeling quantum states as *entangled*. We note that data representing valid quantum states—which are not

necessarily entangled but possess properties relevant to quantum computation-can be readily obtained. Our proposed method utilizes a learning framework known as Positive-Unlabeled (PU) Learning in machine learning, which does not require negative examples. Furthermore, numerical experiments on entanglement detection rarely use evaluation datasets that are difficult to classify, such as those where data clusters densely near the classification boundary. In contrast, we propose designing a dataset for the fully-separability problem—a genuine entanglement detection problem—where data densely clusters near the classification boundary. We conducted comparative experiments with existing one-class classification models on this dataset and confirmed that our proposed method achieves high performance.

1.3 Organization

This dissertation is organized as follows. Chapter 2 introduces basic concepts and definitions in the field of machine learning and quantum computation to aid in understanding the following contents. Chapter 3 presents a pairwise classification method using Kronecker kernel and quantum support vector machine. In Chapter 4, we propose the method for entanglement detection using PU learning. In Chapter 5, we discuss the results and clarify the limitations and future work regarding the objective of this dissertation. We conclude our dissertation in Chapter 6.

Chapter 2

Preliminaries

2.1 Notation

In this dissertation, we use the notations for sets (in Table 2.1), complex numbers (in Table 2.2), and matrices (in Table 2.3). We use $a_i \in \mathbb{C}$ and $A_{i,j} \in \mathbb{C}$ to denote the i -th element of $\mathbf{a}$ for $1 \leq i \leq d$, and the i -th row and j -th column element of A , respectively.

Table 2.1 Notation for set

$\mathbb{N}$	the set of positive integers
$\mathbb{R}$	the set of real numbers
$\mathbb{R}_{\geq 0}$	the set of nonnegative real numbers
$\mathbb{R}^d$	the set of d -dimensional real vectors
$\mathbb{R}^{a \times b}$	the set of real matrices of size a by b
$\mathbb{C}$	the set of complex numbers
$\mathbb{C}^d$	the set of d -dimensional complex vectors
$\mathbb{C}^{a \times b}$	the set of complex matrices of size a by b

Table 2.2 Notation for complex numbers

$\bar{c}$	complex conjugate of complex number c $\bar{c} = a - b i$ for $c = a + b i$
$ c $	absolute value of complex number c $ c = \sqrt{c\bar{c}} = \sqrt{a^2 + b^2}$

Table 2.3 Notation for matrices

A^T	the transpose of A $A_{i,j}^T = A_{j,i}$
A^*	the complex conjugate of A $A_{i,j}^* = \overline{A_{i,j}}$
$A^\dagger$	the conjugate transpose of A $A_{i,j}^\dagger = \overline{A_{j,i}}$

For a complex vector $\mathbf{a} \in \mathbb{R}^d$, we write the Euclidean norm of $\mathbf{a}$ as

$$\|\mathbf{a}\|_2 = \sqrt{\sum_{i=1}^d |a_i|^2}. \quad (2.1)$$

Given a complex matrix $A \in \mathbb{C}^{a \times b}$, we denote trace norm and Frobenius norm as $\|A\|_1$ and $\|A\|_F$, i.e.,

$$\|A\|_1 = \sum_{i=1}^{\min(a,b)} \sigma_i(A) \quad (2.2)$$

and

$$\|A\|_F = \sqrt{\sum_{i=1}^a \sum_{j=1}^b |A_{i,j}|^2} \quad (2.3)$$

where $\sigma_i(A)$ denotes the i -th largest singular value. For two matrices $A \in \mathbb{C}^{r_a \times c_a}$, $B \in \mathbb{C}^{r_b \times c_b}$, we write the Kronecker product of them as $A \otimes B \in \mathbb{C}^{r_a r_b \times c_a c_b}$, i.e.,

$$(A \otimes B)_{(i-1)r_b+k, (j-1)c_b+l} = A_{i,j} B_{k,l} \quad (2.4)$$

for $1 \leq i \leq r_a, 1 \leq j \leq c_a, 1 \leq k \leq r_b, 1 \leq l \leq c_b$. Given a complex square matrix $A \in \mathbb{C}^{d \times d}$, we use $\text{tr}[A] \in \mathbb{C}$ to denote the trace of A , i.e.,

$$\text{tr}[A] = \sum_{i=1}^d A_{ii}. \quad (2.5)$$

Moreover, when the matrix A is invertible, we use $A^{-1} \in \mathbb{C}^{d \times d}$ to denote the inverse matrix of A . For convenience, we denote the $d \times d$ identity matrix as I_d , where

$$(I_d)_{ij} = \begin{cases} 1 & (i = j) \\ 0 & (i \neq j). \end{cases} \quad (2.6)$$

For a positive value $x \in \mathbb{R}$ and functions $f, g : \mathbb{R} \rightarrow \mathbb{R}$, we write as

$$f(x) = O(g(x)) \quad (2.7)$$

when there exist values $x_0 \in \mathbb{R}$ and $c > 0$ which satisfy

$$f(x) \leq cg(x) \quad (2.8)$$

for $x > x_0$ (known as big O notation).

2.2 Problem setting

The studies in this dissertation are based on the supervised classification problem: let $\mathcal{X}$ be the input space, $\mathcal{Y}$ be the output space, and $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_{\geq 0}$ ⁽¹⁾ be the loss function. Then, the expected risk for a decision function $f : \mathcal{X} \rightarrow \mathcal{Y}$ is defined as

$$R(f) = \mathbb{E}_{(x,y) \sim p(\mathcal{X}, \mathcal{Y})} [\ell(f(x), y)] \quad (2.9)$$

⁽¹⁾ Although there are no explicit constraints on the range of the loss function, non-negative values are typically chosen for their intuitive ease of understanding.

where $p(\mathcal{X}, \mathcal{Y})$ is a joint probability distribution over $\mathcal{X}$ and $\mathcal{Y}$. The input and output spaces vary depending on the problem setting. For instance, $\mathcal{X}$ may be a real vector space, such as $\mathbb{R}^4$, or a complex square matrix space, such as $\mathbb{C}^{32 \times 32}$, and $\mathcal{Y}$ may be a set of diverse instances such as $\{+1, -1\}$ (binary classification), $\{1, \dots, 10\}$ (multi-class classification), or $\mathbb{R}$ (regression). As loss functions, squared loss $\ell(f(x), y) = (f(x) - y)^2$ and absolute loss $\ell(f(x), y) = |f(x) - y|$ can be used.

The main goal in the supervised classification problem is to find the best function f^* that minimizes the value of the expected risk, that is

$$R(f^*) = \min_{f \in \mathcal{F}} R(f) \quad (2.10)$$

where $\mathcal{F}$ is a set of decision functions. In practical problem settings, we usually do not have access to the expected risk. Therefore, we calculate empirical risk

$$\hat{R}(f) = \frac{1}{n} \sum_{i=1}^n [\ell(f(x_i), y_i)] \quad (2.11)$$

from a given dataset $(X, Y) = \{(x_i, y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$ of size n , which is assumed to be independently sampled from the distribution $p(\mathcal{X}, \mathcal{Y})$.

In this dissertation, we focus on supervised binary classification problems, in which $\mathcal{Y} = \{+1, -1\}$.

2.3 Hilbert space and kernel method

In this dissertation, let $\mathcal{H}$ be a Hilbert space, i.e., a complete inner product space. Hilbert space can be viewed as an extension of n -dimensional Euclidean space $\mathbb{R}^n$ for $n \in \mathbb{N}$. Furthermore, when considering a Hilbert space consisting of a set (e.g., a set of input features) and the functions defined on that set, one can define a reproducing kernel Hilbert space (RKHS) as follows.

Definition 2.1 (Reproducing kernel Hilbert space). Given a Hilbert space $\mathcal{H}$ and a set X , let $f \in \mathcal{H}$ be a function over X . Then, if for any $x \in \mathcal{X}$ and any $f_n \in \mathcal{H}$,

$$\|f_n - f\|_{\mathcal{H}} \rightarrow 0 \ (n \rightarrow \infty) \Rightarrow f_n(x) \rightarrow f(x) \ (n \rightarrow \infty) \quad (2.12)$$

holds where $\|\cdot\|_{\mathcal{H}}$ is the norm over $\mathcal{H}$, $\mathcal{H}$ is called a *reproducing kernel Hilbert space*. For an RKHS, for any $x \in \mathcal{X}$ and any $f \in \mathcal{H}$, there is a unique function (*reproducing kernel*) $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ such that

$$f(x) = \langle f, k(x, \cdot) \rangle_{\mathcal{H}} \quad (2.13)$$

where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is the inner product over $\mathcal{H}$.

The RKHS is deeply related to the kernel method through the following Moore-Aronszajn theorem:

Definition 2.2 (Moore-Aronszajn theorem). Given a (positive-definite) kernel function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, there exists a unique RKHS $\mathcal{H}$ with kernel function k .

Therefore, choosing different kernel functions is equivalent to using inner products in different RKHS.

2.4 Linear model

Consider the case in which the input space is d -dimensional Euclidean space and the output space is the set of real numbers $\mathbb{R}$. In this case, the linear model can be considered the simplest model for calculating decision functions. That is,

$$f(\mathbf{x}; \mathbf{w}, b) = \mathbf{w}^T \mathbf{x} + b = \sum_{i=1}^d w_i x_i + b \quad (2.14)$$

where $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ are parameters to be optimized. Here, the inner product of the parameter $\mathbf{w}$ and the input feature $\mathbf{x}$ is used to calculate f . In this way, the inner product naturally appears in a decision function and in the calculations of various models. It can also be interpreted as a value measuring the similarity between the two input vectors. Kernel methods have the advantage of mapping input features into another space, enabling the use of inner products in that space. In this way, we can use various similarity functions between vectors.

2.5 Support Vector Machines

A support vector machine (SVM) is a model that is deeply related to the kernel method. SVM seeks a hyperplane, which is determined by parameters to be optimized, that separates two classes of data in the feature space, maximizing the distance from the nearest data points. Mathematically, it can be written as

$$f(\mathbf{x}) = \sum_{i=1}^n \alpha k(\mathbf{x}_i, \mathbf{x}) + b \quad (2.15)$$

where α and b are the parameters to be optimized, and k is the kernel function.

2.6 Neural Networks

Neural networks (NNs) are models whose basic structure consists of parametrized linear transformations and fixed activation functions. This model is written as

$$f(\mathbf{x}) = \sigma_L(W^L \cdots \sigma_2(W^2 \cdot \sigma_1(W^1 \mathbf{x})) \cdots) \quad (2.16)$$

where $L \in \mathbb{N}$ is the number of layers. For $k = 1, \dots, L$ and $d_k \in \mathbb{N}$, $W^k \in \mathbb{R}^{d_k \times d_{k-1}}$ and $\sigma_k : \mathbb{R}^{d_k} \rightarrow \mathbb{R}^{d_k}$ represent weight parameters and nonlinear functions in the k -th layer, where $d_0 = d$. They are among the most widely used models in machine learning.

Convolutional neural networks (CNNs) are models that take images as input. They use convolutional layers for extracting information from local areas of data, which is derived from image processing. Pooling layers, which aggregate local data, are also used in CNN models.

2.7 One-class classification methods

Here we will briefly explain existing methods for one-class classification. Let $\mathcal{X} \subset \mathbb{R}^d$ denote the data space. Given unlabeled data $\{\mathbf{x}_1, \dots, \mathbf{x}_m\} \subset \mathcal{X}$, anomaly detection is the problem of determining whether newly received data are normal or anomalous.

2.7.1 Support vector data description

One commonly used method for the anomaly detection problem is the support vector data description (SVDD) method [7]. SVDD solves the following optimization problem: $\min_{R, \xi_k, \mathbf{a}} R^2 + C \sum_{k=1}^m \xi_k$ subject to $\|\mathbf{x}_k - \mathbf{a}\|_2^2 \leq R^2 + \xi_k$ for $k = 1, \dots, m$. Intuitively, SVDD searches for the minimal hypersphere that surrounds the given data $\mathbf{x}_1, \dots, \mathbf{x}_m$. In this formulation, C is a positive constant that determines the looseness of the solution, and $\{\xi_k\}_{k=1}^m$ is a variable that determines the extent to which ξ_i is allowed to deviate from the center $\mathbf{a}$. By using kernel functions in the calculation of the inner product among vectors $\mathbf{x}_1, \dots, \mathbf{x}_m, \mathbf{a}$, more diverse data boundaries can be explored in different feature spaces.

2.7.2 Deep SVDD

Deep SVDD [8] is a one-class classification method based on deep learning and the SVDD concept. Analogous to SVDD, Deep SVDD uses a deep neural network model to learn a feature map, enabling more flexible boundary detection. One-Class Deep SVDD was introduced in [8] as a variant of Deep SVDD under the assumption that most of the training data are normal. The objective function for One-Class Deep SVDD is defined as

$$\min_W \frac{1}{m} \sum_{k=1}^m \|\phi(\mathbf{x}_k; W) - \mathbf{a}\|_2^2 + \frac{\lambda}{2} \sum_{k=1}^L \|W^k\|_F^2, \quad (2.17)$$

where W^k represents the weights of the network $\phi(\cdot)$ in the k -th layer, and $W = \{W^k\}_{k=1}^L$ represents the set of weights across all layers. The hyperparameter λ serves as the network weight decay regularizer. This objective function was designed under the assumption that most of the training data are normal, leading to the mapping of the data near representation $\mathbf{a}$. For details on general methods related to one-class classification, refer to [9].

2.8 Positive-unlabeled learning

2.8.1 Background in positive-unlabeled learning

Positive-unlabeled learning (PU learning) is a classical machine learning approach using positive and unlabeled data that addresses the problem of learning from incomplete label information [10, 11, 12]. This problem shares core principles with regular binary classification, although it differs in terms of accessibility to label information (see figure 2.1). Two labels are used: positive and unlabeled. A positive label indicates that the datum is truly positive, and an unlabeled datum means that there is uncertainty about whether the datum belongs to the positive or negative category. It is essential to use unlabeled data

for learning in the PU learning framework. This approach can be applied to other problem settings [13, 14].

2.8.2 Unbiased PU learning and non-negative PU learning

Here, we explain uPU (unbiased positive-unlabeled) learning [15] and nnPU (non-negative positive-unlabeled) learning [16], which are closely related to our approach. Let $X \in \mathbb{R}^d$ and $Y \in \{+1, -1\}$ denote the input and output random variables, respectively. Let $p(\mathbf{x}, y)$ denote the joint probability distribution over (X, Y) , and let $p_{pos}(\mathbf{x}) = p(\mathbf{x}|Y = +1)$, $p_{neg}(\mathbf{x}) = p(\mathbf{x}|Y = -1)$, and $p_{unl}(\mathbf{x}) = p(\mathbf{x})$ denote the conditional probability density functions for the positive, negative, and unlabeled categories, respectively. We assume that the data distribution of the unlabeled data is expressed in terms of the conditional probabilities of positive and negative data, which implies that $p_{unl}(\mathbf{x}) = \pi_{pos}p_{pos}(\mathbf{x}) + \pi_{neg}p_{neg}(\mathbf{x})$ holds. In addition, let $g : \mathbb{R}^d \rightarrow \mathbb{R}$ denote a decision function, and let $\ell : \mathbb{R} \rightarrow \mathbb{R}$ denote a loss function, where $\ell(t, y)$ represents the loss value when the predicted value is t and the ground truth (correct label) is y . If the dataset consists of positive data $\{\mathbf{x}_1^{pos}, \dots, \mathbf{x}_{m_{pos}}^{pos}\}$ and negative data $\{\mathbf{x}_1^{neg}, \dots, \mathbf{x}_{m_{neg}}^{neg}\}$, the risk, which represents the expected loss value $\hat{R}_{pn}(g)$, is estimated as

$$\hat{R}_{pn}(g) = \pi_p \hat{R}_p^+(g) + \pi_n \hat{R}_n^-(g), \quad (2.18)$$

where $\hat{R}_p^+(g) = (1/m_{pos}) \sum_{k=1}^{m_{pos}} \ell(g(\mathbf{x}_k^{pos}), +1)$ and $\hat{R}_n^-(g) = (1/m_{neg}) \sum_{k=1}^{m_{neg}} \ell(g(\mathbf{x}_k^{neg}), -1)$. If positive data and unlabeled data $\{\mathbf{x}_1^{unl}, \dots, \mathbf{x}_{m_{unl}}^{unl}\}$ are given, we use the relation $\pi_{neg}p_{neg}(\mathbf{x}) = p_{unl}(\mathbf{x}) - \pi_{pos}p_{pos}(\mathbf{x})$ to estimate risk $\hat{R}_{pu}(g)$:

$$\hat{R}_{pu}(g) = \pi_p \hat{R}_p^+(g) - \pi_p \hat{R}_p^-(g) + \hat{R}_u^-(g), \quad (2.19)$$

where $\hat{R}_p^-(g) = (1/m_{pos}) \sum_{k=1}^{m_{pos}} \ell(g(\mathbf{x}_k^{pos}), -1)$ and $\hat{R}_u^-(g) = (1/m_{unl}) \sum_{k=1}^{m_{unl}} \ell(g(\mathbf{x}_k^{unl}), -1)$, which corresponds to the methods used in uPU learning [15]. In nnPU learning [16], the risk is adjusted to $\tilde{R}_{pu}(g)$, which is defined as

$$\tilde{R}_{pu}(g) = \pi_p \hat{R}_p^+(g) + \max\{0, \hat{R}_u^-(g) - \pi_p \hat{R}_p^-(g)\}, \quad (2.20)$$

to ensure a non-negative risk value because uPU learning sometimes leads to overfitting when applied to models such as deep neural networks. Since the coefficient of $\hat{R}_u^-$ in the above equation is positive, nnPU learning can be considered effective for quantum state data, for which unlabeled data is likely to be predicted as negative, and negative data is expected to be abundant.

Semi-supervised learning [17, 18] is another problem setting for utilizing unlabeled data. Unlike PU learning, it includes a small amount of negative data.

2.9 Quantum computation

2.9.1 Quantum state and density matrix representation

This subsection provides the necessary information for understanding the content of this dissertation regarding the representation of quantum states, which is a fundamental and

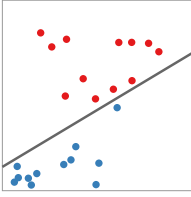
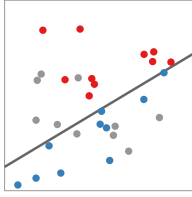
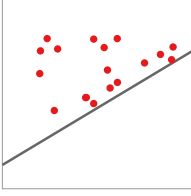
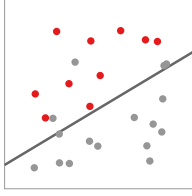
Unlabeled Negative	Non-existent	Existent
	Available	Unavailable
	Positive-Negative (ordinary binary classification) 	Positive-Negative-Unlabeled (semi-supervised learning) 
	Positive (anomaly detection) 	Positive-Unlabeled (PU learning) 

Figure 2.1 Classification of problem settings based on the availability of negative label information and the existence of unlabeled data samples in the training dataset. Positive, negative, and unlabeled data points are shown in red, blue, and gray, respectively

important concept in quantum computation. See [19] for more detailed explanations and properties.

In quantum computation, the state of a physical system is fundamentally described by a complex unit vector in a Hilbert space $\mathcal{H}$. Such a vector is called a state vector, and the space in which a state vector can take values is referred to as the state space. A state vector is usually denoted by $|\psi\rangle$, while the notation $\langle\psi|$ represents its conjugate transpose. This notation is known as bra-ket notation in the field of quantum computation.

A fundamental example of a quantum system is the one-qubit system, which has two basis states. We denote a concrete state vector as $|\psi\rangle = [\alpha \ \beta]^T = \alpha|0\rangle + \beta|1\rangle$, where $|0\rangle, |1\rangle$ are elements of the computational basis; i.e., $|0\rangle = [1 \ 0]^T$, $|1\rangle = [0 \ 1]^T$ and $\alpha, \beta \in \mathbb{C}$ satisfy $|\alpha|^2 + |\beta|^2 = 1$. A qubit is considered as an extension of the classical bit, which simply takes a value of 0 or 1. Although there are many ways to define the basis, we use the computational basis throughout this dissertation. More generally, quantum systems can have discrete bases with more than two states, and fundamental discussions regarding qubits can often be extended to such systems in essence.

These expressions can be easily extended to an n -qubit system, a system in which a quantum state is represented using 2^n basis states. An n -qubit pure state is denoted here as $|\psi\rangle = [\alpha_1, \dots, \alpha_{2^n}]^T = \sum_{k=1}^{2^n} \alpha_k |k\rangle$, where $|k\rangle$ is a basis vector with its k -th element equal to 1 and all other elements equal to 0. The coefficients $\alpha_k \in \mathbb{C} (k = 1, \dots, 2^n)$ satisfy the normalization condition $\sum_{k=1}^{2^n} |\alpha_k|^2 = 1$.

A qubit system's behavior can be described through state vector representations, but

not all of them. A quantum state that can be expressed by a single vector, such as $|\psi\rangle$, is classified as a pure state. However, there are instances in which a quantum state is represented as a probability-weighted mixture of pure states. For example, a quantum state may consist of 40% of $|0\rangle$ and 60% of $|1\rangle$. A density matrix, denoted here by ρ , can be used to describe such a quantum state. Given a set of tuples $(p_k, |\psi_k\rangle)$, where p_k denotes the probability weight of state $|\psi_k\rangle$, the density matrix for the system is defined as $\rho = \sum_k p_k |\psi_k\rangle\langle\psi_k|$. In an n -qubit system, the density matrix takes the form of a $2^n \times 2^n$ Hermitian matrix.

Similar to the constraint $\sum_{k=1}^{2^n} |\alpha_k|^2 = 1$ for the n -qubit pure state $|\psi\rangle = \sum_{k=1}^{2^n} \alpha_k |k\rangle$, density matrix ρ does not span the entire $2^n \times 2^n$ Hermitian matrix space. The constraints governing ρ are outlined as follows.

Theorem 2.3 (Characterization of n -qubit density matrices, Theorem 2.5 in [19] ⁽²⁾). A $2^n \times 2^n$ Hermitian matrix ρ is classified as a density matrix for an n -qubit system if and only if it satisfies the following two conditions: (i) $\text{tr}[\rho] = 1$ and (ii) $\langle\psi|\rho|\psi\rangle \geq 0$ for any n -qubit state vector $|\psi\rangle$.

Here we can easily confirm that each element in a density matrix, which is a complex number, is guaranteed to have an absolute value of at most 1.

2.9.2 Properties of Quantum Computation

A quantum state can be in a state such as $(|0\rangle + |1\rangle)/2$ with one qubit, which cannot be realized by one bit in traditional computation. This property is known as superposition of quantum states. Although this property seems applicable to parallel computation, we cannot do so because of measurement, which is the operation to read out the information of quantum state. By measurement, we obtain an output based on a probability distribution which is determined by measurement basis and output state before measurement. At the same time the state after measurement changes to some state, which makes it difficult to manipulate quantum states.

2.9.3 Quantum circuit

A quantum circuit is a diagram that represents quantum computation, which visualizes the target qubit(s) in operations and the relations between qubits. In this section, the fundamental perspective of quantum circuits is elucidated through the computational process, as illustrated in the circuit example depicted in Figure 2.2.

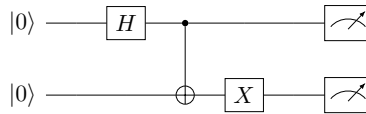


Figure 2.2 A sample quantum circuit

⁽²⁾ Although this theorem is stated in the source text as defining the conditions for the density operator, we reinterpret it in terms of the density matrix, as that is our focus.

Quantum circuits are read from left to right; that is, the leftmost part of the diagram represents the input (or initial) state, and the rightmost part represents the output state. Furthermore, each horizontal row in the array corresponds to a single quantum register, which is defined as a group consisting of one or more qubits. The objects delineated on the lines represent the operations that have been applied. Consequently, the quantum state depicted in Figure 2.2 is comprised of two distinct quantum registers. The top row of the quantum register is equivalent to the first register, while the bottom row corresponds to the second register. At this particular juncture, the initial state is

$$|\psi_0\rangle = |0\rangle|0\rangle. \quad (2.21)$$

As illustrated in Figure 2, the initial register is subjected to an H gate, the first and second registers are exposed to a $CNOT$ gate, and the second register is affected by an X gate. Here, the X and H gates are defined as

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (2.22)$$

Furthermore, the $CNOT$ gate is defined as

$$CNOT = |0\rangle\langle 0| \otimes I_2 + |1\rangle\langle 1| \otimes X \quad (2.23)$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (2.24)$$

$$(2.25)$$

The calculation results after the X gate are

$$|\psi\rangle = (I_2 \otimes X)CNOT(H \otimes I_2)|\psi_0\rangle \quad (2.26)$$

$$= \frac{1}{\sqrt{2}}(|0\rangle|1\rangle + |1\rangle|0\rangle). \quad (2.27)$$

Here the symbols located at the extreme right of each row are used to denote measurement operations. For the sake of simplicity, the focus will be on the Z-basis measurements utilized in Chapter 3. In the domain of quantum mechanics, a Z-basis measurement can be defined as a process that yields one of the basis states as the output, with a probability that is equivalent to the square of the coefficient (also known as the amplitude) of that basis state in the quantum state. Consequently, in the event that the state of all qubits is measured, the probability of obtaining the outputs 01 and 10 is equal at 1/2 each. Conversely, the probability of obtaining either 00 or 11 is zero. Accordingly, the foundational paradigm in quantum computation entails the execution of operations on a designated initial state, culminating in the acquisition of probabilistic outputs through measurement.

2.9.4 HHL algorithm

The HHL algorithm is an algorithm for solving a system of linear equations $A\mathbf{x} = \mathbf{b}$ [20]. The HHL algorithm utilizes quantum phase estimation, which encodes the eigenvalue of a

matrix A in the form of a binary fraction. The HHL algorithm decreases the computational complexity by utilizing the property that the eigenvalues of inverse matrix A^{-1} are reciprocal of those of A .

2.10 Quantum machine learning

2.10.1 Four regions of approaches in quantum machine learning field

The field of quantum machine learning can be divided into four regions based on its problem settings and methods. This concept originates from the learning classes introduced by Aïmeur et al. in [21]. These regions are shown in Table 2.4. CC (classical problem, classical

Table 2.4 Four regions of approaches in quantum machine learning field. This dissertation focuses on the CQ and QC regions, which are shaded

Problem setting \ Method	Classical	Quantum
	Classical	Quantum
Classical	CC	CQ
Quantum	QC	QQ

method) primarily refers to classical machine learning, which is the field with the most active research. Broadly speaking, CC also encompasses classical methods inspired by quantum computation. CQ (classical problem, quantum method) is a region with classical problem setting and quantum methods. In this region, improvements in the performance of machine learning methods focusing on the expressive power of quantum computation can be considered. QC (quantum problem, classical method) encompasses the field of quantum problem setting with classical machine learning methods. For example, it includes the analysis of data derived from quantum computation via machine learning. The expectation for work in this direction is efficient information processing using classical computers, on which information is easy to access information. QQ (quantum problem, quantum method) is the case in which the problem setting and methods are closed in quantum computation. Processing data derived from quantum computation using machine learning techniques on quantum computers appears to be a natural approach. However, given the current scale and error correction capabilities of actual quantum computers, verification is not straightforward. Thus, determining the significance of the QQ approach, particularly in comparison to conventional computer processing (CQ approach), remains a major future research challenge in this field.

This dissertation focuses on CQ and QC approaches, which bridge the fields of classical and quantum computation.

2.10.2 Introduction to chapter 3

Quantum computation has been studied as a way to overcome computational complexity problems of machine learning using huge training datasets. A machine learning algorithm uses training data as input, learns the rules and properties inherent in the data, and solves

problems such as regression and classification for unknown data. These algorithms are currently used in various fields, such as recommendation systems for e-commerce sites [22] and reaction prediction for chemical substances [23], and represent one of the key technologies supporting artificial intelligence.

Domingos categorized machine learning researchers into five major tribes: symbolists, connectionists, evolutionists, Bayesians, and analogists [24]. Each tribe has different ideas about learning and a core algorithm. Connectionists consider the brain to be represented by its operating principles and learn by adjusting the strength of connections between neurons. Analogists see similarities between situations as the key to learning and thus reasoning about other similarities. Their core algorithm is the support vector machine (SVM), a machine learning algorithm proposed in [25]. It is a general-purpose algorithm that can be applied to regression problems, classification problems, etc. It finds the optimal solution by maximizing the margin.

The use of SVM in combination with the kernel method, which uses a kernel function to represent the similarity between the data, enables data to be mapped into a feature space for training and prediction. The use of a kernel trick enables various feature mapping functions to be virtually used without explicitly calculating high-dimensional feature mappings. This is a major advantage of the kernel method, and the flexibility to choose a feature mapping function for the target problem makes it possible to use the appropriate feature space for the problem. This improves the classification accuracy of the learned model. Another advantage of the kernel method is that it has clear theoretical properties, such as the fact that learning with SVM is often formulated as a convex optimization problem [26], which guarantees that a global optimal solution can be found.

There are, however, disadvantages of the kernel method, such as its computational complexity. When calculating the inner product of the data using the feature mapping function, the amount of calculation is substantially higher when using large-scale datasets, which may become a bottleneck in the training process. Connectionist approaches using deep neural networks are thus often used for large-scale datasets, but they also have disadvantages, such as model complexity (which makes it impossible to guarantee the optimality of the learning results), low interpretability, and a lack of theoretical understanding of generalization performance. For these situations, recently, some research has focused on the dynamics of neural networks in learning [27].

The use of various machine learning models has been increasing along with the growth of big data. However, the increasing amount of data that is handled increases the time it takes to train machine learning models. Therefore, it is important to develop models that are scalable with the amount of training data.

Quantum computation is used in various fields, including quantum mechanics [19]. In quantum computation, the key to gaining an advantage over classical computation is to make effective use of two unique properties: superposition and entanglement. Superposition is the ability to maintain multiple states simultaneously, which is generally not easy to do with classical computation. Superposition makes it possible to compute in parallel and to use amplitudes, or degrees of overlap, in calculations. Entanglement enables a dependency to be introduced between the states of each qubit, making it possible to create more complex

superposition states.

Various research has been conducted on high-performance algorithms that use these properties, including Shor’s algorithm [2], used for prime factorization, and Grover’s algorithm [3], used for the fast search of unordered data. Many other algorithms have been proposed that can reduce computational complexity compared to the currently used algorithms. Some groups [28, 29] have recently published research results showing that quantum computation is faster than classical computation on certain tasks. There are also some quantum-inspired methods, which are classical methods based on the idea of existing quantum methods. [30] tackles the classical computational complexity tasks in [28] and shows that classical methods can tackle the same task almost as fast as a quantum model.

In quantum machine learning, exponential speedup based on the Harrow-Hassidim-Lloyd (HHL) algorithm has attracted much attention, and various models have been developed. Machine learning algorithms that aim for quadratic acceleration using Grover’s algorithm [3] have also been developed. A recent trend is not only to use fast quantum algorithms to accelerate machine learning subroutines, but also to theoretically discover machine learning tasks that are relatively easy for quantum computation but difficult for classical computation. As a result of recent algorithmic developments, expectations for quantum computation have been rising in recent years.

Hardware for quantum computers has also been the subject of intense research. A variety of materials for quantum computers are being considered, including superconductors, ion traps, and light. Each material has certain advantages and disadvantages in terms of performance, such as ease of integration, the number of clocks, and the fidelity of gate operations, and it remains to be seen which material will bring the greatest advantage.

Despite the rising expectations, the problem settings in which quantum computation can demonstrate its superiority over classical computation are very limited at present because of the noise factor and limited hardware scalability. This makes finding useful applications a major issue. The use of quantum computation in machine learning should speed up the linear algebra operations used in the computation and is thus a candidate application. The quantum SVM [31] is an algorithm for quantum machine learning and, under certain conditions, it can exponentially reduce computational complexity compared to existing SVMs. The implementation of quantum SVM has been studied in [32] and [33].

We have devised a quantum SVM with the Kronecker kernel for the pairwise classification problem, which is the problem of determining whether two sets of data satisfy a certain relationship. This problem has been studied using a number of classical computation methods. The *Kronecker kernel*, one of the kernels used in pairwise classification, is difficult to apply to large-scale data in classical computation because the Kronecker product of the kernel matrix is huge. On the other hand, in quantum computation, the Kronecker product of matrices can be expressed simply by using the tensor product of quantum states, so the computational complexity is smaller than in classical computation. This means that it is easy to scale up to large-scale data. We thus hypothesize that the pairwise problem is a candidate problem setting in which quantum computation may have an advantage.

Pairwise classification is a problem in which two data points are given as input, and the objective is to determine whether they satisfy a certain relationship [34, 35]. This

formulation is applicable to both cases where the two inputs are symmetric and where they are asymmetric in practical problem settings. An example of a problem with asymmetric inputs is the purchase prediction problem in e-commerce: determining whether a user likes an item. An example of a problem with symmetric inputs is the chemical interaction prediction [36] problem, which involves determining whether two given chemical compounds will react.

2.10.3 Introduction to chapter 4

Quantum entanglement plays a central role in quantum information processing. For example, quantum teleportation [37], quantum computation [2, 3], and quantum cryptography [38] utilize quantum entanglement to perform calculations not possible with classical computers. Furthermore, quantum entanglement is an indispensable element in frameworks that rely on quantum states. Quantum states are typically governed by quantum circuits, and verifying entanglement generation at runtime is essential to ensure that the algorithms and subroutines executed by these circuits function properly. Entanglement detection refers to the problem of determining whether a given quantum state is entangled, a problem that has been studied for an extended period [39].

In the case of a general bipartite quantum system, entanglement detection is known to be NP-hard [40]. Consequently, numerous studies on entanglement detection have been conducted to achieve higher accuracy. The Peres-Horodecki criterion [41, 42], also known as PPT (positive partial transpose) criterion, is one of the most well-known criteria for determining whether a quantum state is entangled. This criterion serves as a powerful tool for analytically determining entangled states from the density matrix within a specific system. However, applying such analytical approaches to relatively large systems is challenging, and no universally applicable method has been established.

As a result, numerous studies in recent years have explored classical machine learning techniques as alternative approaches to entanglement detection [43, 44]. In various machine learning methods, quantum states are categorized as either separable or entangled, framing the problem as a binary classification task. This framing offers the advantage of being both intuitive and readily applicable. In fact, such methods have demonstrated highly accurate entanglement detection.

Although the use of binary classification is reasonable, there is a fundamental issue regarding the assumption of training data. An ordinary binary classification problem in machine learning operates on the basis of the availability of both positive and negative example data. However, in the entanglement detection problem, explicitly preparing data representing entangled states is generally challenging. That means it is not possible to guarantee that a quantum state is entangled solely on the basis of the given density matrix. Specific types of entangled states, such as the GHZ state and the W state, can be prepared; however, identifying a quantum state as entangled requires information about the data generation process. This raises the question of how to generate a dataset for training.

In the context of entanglement detection, the density matrix data contained within the test data is typically distributed over a broad range. In such circumstances, the utilization of a train dataset comprising solely well-known entangled states, such as the GHZ and W

states, results in diminished performance metrics on test data. Furthermore, the labeling process in existing numerical entanglement detection methods may contain errors, which could lead to concerns regarding the performance of the classification model.

[45] identified challenges in the generation of entangled state data and presented a learning method that does not require explicit entangled state data in the training dataset. They proposed using an unsupervised learning setting in which only data of separable states are assumed to be available. That means no label information is provided in the training dataset. A generative adversarial network (GAN) model along with Siamese neural network is used to train the discriminator and generator using only data from separable states. We believe that avoiding the use of explicit entanglement state data is an effective approach to address the challenge of preparing it, as they demonstrated.

We present a method for addressing the entanglement detection problem in a multi-qubit system that utilizes separable and randomly generated quantum state data while accounting for data availability. The idea of utilizing random quantum states is based on learning from positive-unlabeled data, which is often referred to as PU learning, a classical machine learning framework that does not use label information from negative data [10]. In the PU learning framework, all negative data and a portion of the positive data are provided as unlabeled data, which means that methods for regular supervised learning cannot be directly applied as they assume label information is available for all data. There are several benefits of applying methods based on PU learning: a more realistic setting compared with formulation as regular binary classification and the ability to utilize a broader range of data compared with the setting that includes only separable state data.

Chapter 3

Quantum Support Vector Machine for Pairwise Classification Problem

The structure of this chapter is as follows. Section 3.1 explains the background, including the problem definition and existing approaches. Section 3.2 describes the proposed model: pairwise support vector machine model. In Section 3.3, we conduct experiments and summarize the results. In Section 3.4, we discuss future work. We conclude this chapter in Section 3.5. For an introduction to this chapter, refer to section 2.10.2.

3.1 Background

3.1.1 Pairwise classification problem

The pairwise classification problem differs from the standard supervised classification problem setting (section 2.2) in that it explicitly has two inputs. Let $\mathcal{X}, \mathcal{Z}$ be the two input spaces, $\mathcal{Y} = \{+1, -1\}$ be the output space, and R be some relation over $\mathcal{X} \times \mathcal{Z}$. Our goal here is to construct a classifier that represents the function $f : \mathcal{X} \times \mathcal{Z} \rightarrow \mathcal{Y}$ such that

$$f(\mathbf{x}, \mathbf{z}) = \begin{cases} +1 & (\mathbf{x} \text{ and } \mathbf{z} \text{ satisfy relation } R) \\ -1 & (\text{otherwise}) \end{cases} \quad (3.1)$$

from the given dataset $(X, Z, Y) = \{(x_i, z_i, y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Z} \times \mathcal{Y}$.

Here we remark on the assumptions for our problem setting: the given dataset is a Cartesian product. That is, let $|X|$ and $|Z|$ be the numbers of distinct elements in X and Z , respectively; then $n = |X| \times |Z|$ holds. There are also problem settings where this assumption does not hold [46].

3.1.2 Kernel method

One approach to solving pairwise classification problems is to use the kernel method [47], which enables classification in a high-dimensional space by replacing the inner product of data points, which appear in the model equation, with a kernel function. Different kernel functions yield different classification boundaries.

Let k be a function of two variables over $\mathbb{R}^d$ (this explanation also applies when k is a two-variable function over an arbitrary set X ; however, for simplicity, we describe here the case $X = \mathbb{R}^d$). Function k is a kernel function over $\mathbb{R}^d$ if it satisfies two conditions:

(i) symmetry, i.e.,

$$k(\mathbf{x}, \mathbf{y}) = k(\mathbf{y}, \mathbf{x}) \quad (3.2)$$

holds for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$.

(ii) positive semi-definiteness, i.e.,

$$\sum_{i,j=1}^n c_i c_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0 \quad (3.3)$$

holds for any $n \in \mathbb{N}$ and all $\{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \mathbb{R}^d, \{c_1, \dots, c_n\} \subset \mathbb{R}$.

Here we define $\langle \cdot | \cdot \rangle$ as a regular inner product in $\mathbb{R}^d$; i.e., for $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ such that $\mathbf{x} = [x_1, \dots, x_n]^T, \mathbf{y} = [y_1, \dots, y_n]^T$,

$$\langle \mathbf{x} | \mathbf{y} \rangle = \sum_{i=1}^n x_i y_i. \quad (3.4)$$

Then, for any mapping $\Phi : \mathbb{R}^d \rightarrow \mathcal{H}$ (Hilbert space),

$$k(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}) | \Phi(\mathbf{y}) \rangle \quad (3.5)$$

is a kernel function. From this, we can consider the inner products of the elements of $\mathbb{R}^d$ in $\mathcal{H}$ by using k . An example kernel is a linear kernel $k(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x} | \mathbf{y} \rangle$, the simplest type of kernel. There are many kinds of kernel functions, such as polynomial kernel $k(\mathbf{x}, \mathbf{y}) = (\langle \mathbf{x} | \mathbf{y} \rangle + b)^c$ where $b \in \mathbb{R}$ and $c \in \mathbb{N}$, which is an extension of the linear kernel, and Gaussian kernel $k(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \|\mathbf{x} - \mathbf{y}\|_2^2)$ where $\gamma \in \mathbb{R}$ is the hyperparameter.

This means that we have many mapping functions to choose from. We can select one that suits the problem we focus on. The mapping of data in another space is called the *kernel trick*. This mapping Φ is called *feature mapping*, and the SVM model is based on this trick.

We also define the $n \times n$ matrix $K_{\mathcal{X}}$ for the dataset $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ so that (i, j) element of $K_{\mathcal{X}}$ is $\langle \mathbf{x}_i | \mathbf{x}_j \rangle$. We call $K_{\mathcal{X}}$ the *kernel matrix over $\mathcal{X}$* .

3.1.3 Least squares Support Vector Machine

The least squares support vector machine (LS-SVM) is a classical classification model derived from the SVM, proposed in [48]. In this section, we consider an LS-SVM model for the binary classification problem. For the training data point $\mathbf{x} \in \mathbb{R}^d$, we assume that the correct answer label is determined by $t \in \{+1, -1\}$. In this case, the linear model

$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b \quad (3.6)$$

can be used to estimate the label for the data point $\mathbf{x}$. Here, $b \in \mathbb{R}$ is the bias parameter and $\mathbf{w} = [w_1, \dots, w_{\dim \mathcal{H}}] \in \mathcal{H}$ ($\dim \mathcal{H}$ is the dimension of Hilbert space $\mathcal{H}$) is the weight vector.

The function $\phi : \mathbb{R}^d \rightarrow \mathcal{H}$, the *feature mapping function*, is used to move the raw data into another (usually high-dimensional) space, where it may become linearly separable. The sign of $y(\mathbf{x})$ calculated using these values is then used as the prediction of the classification class. That is, the prediction label $\tilde{y}(\mathbf{x})$ for data point $\mathbf{x}$ is given by

$$\tilde{y}(\mathbf{x}) = \begin{cases} +1 & (y(\mathbf{x}) \geq 0) \\ -1 & (y(\mathbf{x}) < 0). \end{cases} \quad (3.7)$$

In the LS-SVM problem setting, the mathematical optimization formulation is given as

$$\min_{\mathbf{w}, b, \mathbf{e}} L(\mathbf{w}, b, \mathbf{e}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + \frac{\gamma}{2} \sum_{i=1}^N e_i \quad (3.8)$$

subject to the constraints

$$y_i \{ \mathbf{w}^T \phi(\mathbf{x}_i) + b \} = 1 - e_i. \quad (3.9)$$

Given $y = \{+1, -1\}$, then

$$e_i^2 = [1 - (y_i \{ \mathbf{w}^T \phi(\mathbf{x}_i) + b \})]^2 \quad (3.10)$$

$$= \{y_i^2 - (y_i \{ \mathbf{w}^T \phi(\mathbf{x}_i) + b \})\}^2 \quad (3.11)$$

$$= (y_i - \{ \mathbf{w}^T \phi(\mathbf{x}_i) + b \})^2 \quad (3.12)$$

holds, and thus

$$e_i = y_i - \{ \mathbf{w}^T \phi(\mathbf{x}_i) + b \}. \quad (3.13)$$

Solving this problem using the Lagrange multiplier method,

$$\mathbf{w} = \sum_{i=1}^N a_i \phi(\mathbf{x}_i) \quad (3.14)$$

holds by the condition that $\frac{\partial L'}{\partial \mathbf{w}} = 0$ for

$$L'(\mathbf{w}, b, \mathbf{e}, \mathbf{a}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + \frac{\gamma}{2} \sum_{i=1}^N e_i - \sum_{i=1}^N a_i (e_i - y_i + \{ \mathbf{w}^T \phi(\mathbf{x}_i) + b \}). \quad (3.15)$$

Then, the prediction model in Eq. (3.6) can be rewritten as

$$y(\mathbf{x}) = \sum_{i=1}^n a_i \phi(\mathbf{x}_i)^T \phi(\mathbf{x}) + b = \sum_{i=1}^n a_i \langle \phi(\mathbf{x}_i) | \phi(\mathbf{x}) \rangle + b, \quad (3.16)$$

where $a_1, \dots, a_n \in \mathbb{R}$ and $b \in \mathbb{R}$ are parameters that need to be obtained in order to perform the classification. In the LS-SVM model, the parameters are obtained by dividing the problem into multiple linear equations and solving the equations. Specifically, we solve

$$\begin{bmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & K_{\mathcal{X}} + \gamma^{-1} I_n \end{bmatrix} \begin{bmatrix} b \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{t} \end{bmatrix}, \quad (3.17)$$

where $\mathbf{1} = [1, \dots, 1]^T$, $\mathbf{a} = [a_1, \dots, a_n]^T$, $\mathbf{t} = [t_1, \dots, t_n]^T$, and $K_{\mathcal{X}}$ is the kernel matrix. γ is a positive number (hyperparameter), and I_n is the $n \times n$ identity matrix. We define the matrix $F_{\mathcal{X}}$ as

$$F_{\mathcal{X}} = \begin{bmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & K_{\mathcal{X}} + \gamma^{-1} I_n \end{bmatrix}. \quad (3.18)$$

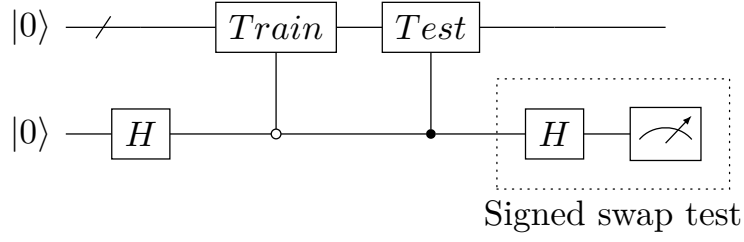


Figure 3.1 Outline of quantum SVM model comprising three main parts: train, test, and signed swap test (prediction). In train part, quantum state is calculated using training data, and training data is encoded as a quantum state for prediction. In test part, test data is encoded as a quantum state. In signed swap test part, measurement is repeated many times and inner product of the states is estimated from the result of measurement

Then equation 3.17 can be expressed as

$$F_{\mathcal{X}} \begin{bmatrix} b \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{t} \end{bmatrix}. \quad (3.19)$$

We can now solve the equation (3.17) by inverse matrix calculation; then

$$\begin{bmatrix} b \\ \mathbf{a} \end{bmatrix} = (F_{\mathcal{X}})^{-1} \begin{bmatrix} 0 \\ \mathbf{t} \end{bmatrix} \quad (3.20)$$

follows (we can choose the hyperparameter γ so that $F_{\mathcal{X}}$ has an inverse matrix). Doing this gives the values of the parameters $\mathbf{a}$ and b , and the prediction can be made using the equation 3.16.

3.1.4 Quantum Support Vector Machine

Quantum SVM uses quantum computation and was proposed in [31]. The basic idea is the same as that of the LS-SVM classification model. Its use of quantum computation speeds up the calculation of the inverse matrix, a computationally expensive part of the classical LS-SVM algorithm.

The whole quantum SVM model is shown in Figure 3.1. This model is composed of four registers, and the qubits in the first and fourth registers are used as ancilla qubits. Here, we note that the figures of quantum circuits are drawn using Quantikz [49] in this dissertation. The behavior of the quantum SVM can be summarized as follows:

1) Prepare a quantum state from the training data. The model is shown in 3.2. At that time, training data are pre-normalized and encoded in the third register. When the training data is $\mathbf{x} = [x_1, \dots, x_n]^T$ and $C = \sum_{i=1}^n |x_i|^2$, this training data is encoded as $|\mathbf{x}\rangle = \frac{1}{\sqrt{C}} \sum_{i=1}^n x_i |i\rangle$. After encoding, obtain parameters for prediction by using the training data encoded in the quantum state and applying the HHL algorithm [20]. This algorithm is a powerful quantum algorithm that enables the inverse matrix to be calculated more quickly than classical methods. It solves linear equations in the form $A|\mathbf{y}\rangle = |\mathbf{b}\rangle$ in a quantum state, and the result is the quantum state equivalent to $|\mathbf{y}\rangle = A^{-1}|\mathbf{b}\rangle$. Using the HHL algorithm

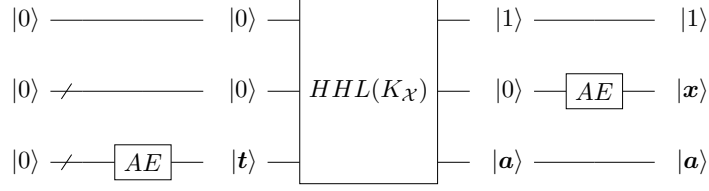


Figure 3.2 Outline of training part in existing quantum SVM model. First, applying amplitude encoding(AE), encoding training data label as amplitude in third register, the state becomes $|0\rangle|0\rangle|t\rangle$. And then applying HHL algorithm with K_X , the state becomes $|1\rangle|0\rangle|a\rangle$. 1 in first register mean success of HHL algorithm. Lastly, by amplitude encoding in second register, the state becomes $|1\rangle|x\rangle|a\rangle$

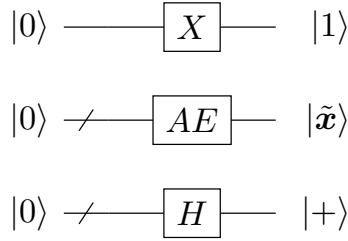


Figure 3.3 Outline of testing part in existing quantum SVM model. X gate is applied for first register to match $|1\rangle$ in first register of training part. Amplitude encoding for second register encodes test data. H gate for third register makes superposition state to evaluate inner product

and the quantum state corresponding to $[0 \ t]^T$, the state corresponds to $[b \ a]^T$ is obtained as a quantum state $|a\rangle$ (see equation 3.20). Lastly, apply amplitude encoding of training data in the second register.

2) Prepare a quantum state from the test data. The model is shown in 3.3. Applying amplitude encoding of test data in the second register and H gates in the third register to make a superposition state. X gate in the first register is applied to take the inner product with the state for which the HHL algorithm succeeded.

3) Determine the sign of the inner product between the quantum state $|\tilde{x}\rangle$ corresponding to data point $\tilde{x}$ of the test data and $A^{-1}|b\rangle = |a\rangle$ obtained in 1). *Swap test* operation is performed to measure the inner product of the parameters and the new data point. The sign of the inner product is used in the classification problem with the SVM, so a variant of the swap test is used to obtain the sign of the inner product. We call this a *signed swap test*, a method proposed in [50].

Signed swap test is performed as follows. We assume that $|\psi_1\rangle$ and $|\psi_2\rangle$ in the following are real-valued vectors and aim to obtain the (signed) value of $\langle\psi_1|\psi_2\rangle$.

- 1) Prepare the state $\frac{1}{\sqrt{2}}(|0\rangle|\psi_1\rangle + |1\rangle|\psi_2\rangle)$.
- 2) Apply a H gate to the first qubit so that the state changes to $|\psi'\rangle = \frac{1}{2}((|0\rangle + |1\rangle)|\psi_1\rangle + (|0\rangle - |1\rangle)|\psi_2\rangle) = \frac{1}{2}(|0\rangle(|\psi_1\rangle + |\psi_2\rangle) + |1\rangle(|\psi_1\rangle - |\psi_2\rangle))$.
- 3) Measure the first qubit. The probability of obtaining zero from the first qubit is

$p_0 = |\langle 0|\psi'\rangle|^2 = \frac{1}{4}(\langle\psi_1| + \langle\psi_2|)(|\psi_1\rangle + |\psi_2\rangle) = \frac{1}{2} + \frac{1}{2}\langle\psi_1|\psi_2\rangle$. This means that we know that $\langle\psi_1|\psi_2\rangle = 2p_0 - 1$. Therefore, we can predict the value of the inner product of $|\psi_1\rangle$ and $|\psi_2\rangle$ by again measuring the first qubit and calculating the probability of obtaining 0.

The final prediction class is determined on the basis of the sign of the inner product obtained in this way. This means, for example, that we can predict that the label is +1 when the value of $y(\mathbf{x}) \geq 0$ and -1 otherwise.

3.1.5 Kronecker kernel

The Kronecker kernel is used to solve problems such as pairwise classification. When two pairs of data points $((\mathbf{x}_1, \mathbf{z}_1), (\mathbf{x}_2, \mathbf{z}_2))$ are extracted from the product set $\mathcal{X} \times \mathcal{Z}$ ($\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$, $\mathbf{z}_1, \mathbf{z}_2 \in \mathcal{Z}$, $(\mathbf{x}_1, \mathbf{z}_1), (\mathbf{x}_2, \mathbf{z}_2) \in \mathcal{X} \times \mathcal{Z}$), the Kronecker kernel function $k_{\otimes}$ is defined as

$$k_{\otimes}((\mathbf{x}_1, \mathbf{z}_1), (\mathbf{x}_2, \mathbf{z}_2)) = k_{\mathcal{X}}(\mathbf{x}_1, \mathbf{x}_2)k_{\mathcal{Z}}(\mathbf{z}_1, \mathbf{z}_2). \quad (3.21)$$

This function satisfies the two conditions discussed in 3.1.2. On the basis of this representation, the Kronecker kernel matrix can be expressed as

$$K_{\mathcal{X} \otimes \mathcal{Z}} = K_{\mathcal{X}} \otimes K_{\mathcal{Z}}, \quad (3.22)$$

where $\otimes$ denotes the Kronecker product. A standard kernel matrix stores the inner products between the data, and the Kronecker kernel stores the inner products between data pairs. From the above definition, the inner product of $Pair1(\mathbf{x}_1, \mathbf{z}_1)$ and $Pair2(\mathbf{x}_2, \mathbf{z}_2)$ can be expressed as

$$\langle Pair1 | Pair2 \rangle = \langle \mathbf{x}_1 | \mathbf{x}_2 \rangle \langle \mathbf{z}_1 | \mathbf{z}_2 \rangle. \quad (3.23)$$

This expression of the inner product of data pairs intuitively tells us that, when $\mathbf{x}_1$ and $\mathbf{x}_2$ are similar and $\mathbf{z}_1$ and $\mathbf{z}_2$ are similar, then $Pair1$ and $Pair2$ can be considered similar. In later discussion, to simplify the problem, we assume that $\mathcal{X} = \mathcal{Z}$. In the general case, $\mathcal{X} \neq \mathcal{Z}$, our model can be applied with little change. In the following part, we define the dimensions of $\mathbf{x} \in \mathcal{X}$ and $\mathbf{z} \in \mathcal{Z}$ to be d , and the numbers of elements in $\mathcal{X}$ and $\mathcal{Z}$ to be n , respectively.

3.1.6 Computational complexity for solving pairwise classification problem with Kronecker kernel

Here, we analyze the computational complexity of the inverse calculation for a kernel matrix, an $n \times n$ semidefinite matrix. To simplify the problem, we discuss the case in which the model is used without an offset term. In classical computation, Gaussian elimination is one of the general methods used to calculate the inverse of a general regular matrix, and its complexity is $O(n^3)$. As a classical approximation method, *conjugate gradient method* [51] is used, whose complexity is $O(n\kappa \log(1/\epsilon))$ [52]. The conjugate gradient method can be applied under the condition that the matrix is real symmetric and s -sparse, which means that the number of non-zero elements in the matrix is at most s . The value κ is called the *condition number* and is defined as the absolute value of the ratio of the largest singular value to the smallest singular value; ϵ implies the error between the obtained values and the exact values.

In quantum computation, we can apply the HHL algorithm, the calculation of which is the bottleneck in the quantum SVM, in $O(s\kappa \text{poly}(\log(s\kappa/\epsilon)))$ time complexity [53]. We thereby achieve a computational complexity of $O(s\kappa \text{poly}(\log(s\kappa/\epsilon)))$ by assuming the use of quantum random access memory (QRAM) [54], by which we can encode the training and test data in $O(\log n)$ time. Therefore, when the kernel matrix is sparse, especially when $s = O(\log n)$, quantum computation can perform the inverse matrix calculation exponentially faster than classical computation for n .

For the pairwise classification problem with the Kronecker kernel, the matrix of the pairwise kernels has $O(n^2)$ rows and $O(n^2)$ columns, which causes the large computational complexity. In this case, the computational complexity of the SVM is $O(n^6)$, which can be improved to $O(n^3)$ by applying vec-trick, a technique for reducing computational complexity of Kronecker product of matrices [46]. With the conjugate gradient method, the complexity is $O(n^2 s^2 \kappa^2 \log(1/\epsilon))$. For the complexity of the space, if the elements of the Kronecker kernel matrix are stored explicitly, at least $O(n^4)$ memory is needed. It is thus difficult to prepare memory large enough to store the Kronecker kernel matrix for a large dataset. Applying these algorithms to large datasets is thus difficult due to time and space complexities.

3.2 Proposed model

We have devised a quantum SVM model with a Kronecker kernel for solving pairwise classification problems. The outline of our model is the same as the existing model (Figure 3.1), and our model also comprises three parts; train, test, and signed swap test.

First, initial state $|0\rangle|0 \dots 0\rangle|0 \dots 0\rangle|0\rangle$ is prepared using $O(\log n)$ qubits to store information. Next, a H gate is applied to the fourth register, and the state becomes $\frac{1}{\sqrt{2}}|0\rangle|0 \dots 0\rangle|0 \dots 0\rangle(|0\rangle + |1\rangle)$. The fourth register is used to estimate the sign of the inner product.

The circuit in the training part (Figure 3.4) uses three registers $|0\rangle|0 \dots 0\rangle|0 \dots 0\rangle$ and here we only focus on these three registers. First, amplitude encoding of the training data label is applied and the state becomes $|0\rangle|0 \dots 0\rangle|\mathbf{t}\rangle$. Next, HHL algorithm is applied. In the phase estimation part of the HHL algorithm, controlled- $e^{i\theta F_{\mathcal{X} \otimes \mathcal{Z}}}$ gates are applied; this calculation is performed as in the original quantum SVM model. Here, $F_{\mathcal{X} \otimes \mathcal{Z}}$ is defined as

$$F_{\mathcal{X} \otimes \mathcal{Z}} = \begin{bmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & K_{\mathcal{X} \otimes \mathcal{Z}} + \gamma^{-1} I_{n^2} \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & K_{\mathcal{X}} \otimes K_{\mathcal{Z}} + \gamma^{-1} I_{n^2} \end{bmatrix}. \quad (3.24)$$

By applying the HHL algorithm to the state $(|0\rangle|0 \dots 0\rangle|\mathbf{t}\rangle)$, the state becomes $|1\rangle|0 \dots 0\rangle|\mathbf{a}\rangle$, where $|\mathbf{a}\rangle = b|1\rangle + \sum_{i=1}^{n^2} a_i|i+1\rangle$. The training data is then again encoded as a quantum state so that the state becomes $|1\rangle|\mathbf{x}, \mathbf{z}\rangle|\mathbf{a}\rangle$, here $|\mathbf{x}, \mathbf{z}\rangle = |\mathbf{x}\rangle|\mathbf{z}\rangle$. We define this state $|\psi_{train}\rangle$, and at this point the overall quantum state is $\frac{1}{\sqrt{2}}(|\psi_{train}\rangle|0\rangle + |0\rangle|0 \dots 0\rangle|0 \dots 0\rangle|1\rangle)$.

The circuit in the test part (Figure 3.5) also uses three registers $|0\rangle|0 \dots 0\rangle|0 \dots 0\rangle$. The test data are encoded as a quantum state and superposed equally. The state is then

$$\frac{1}{\sqrt{n^2+1}} \sum_{i=1}^{n^2+1} |1\rangle|\tilde{\mathbf{x}}, \tilde{\mathbf{z}}\rangle|i\rangle. \quad (3.25)$$

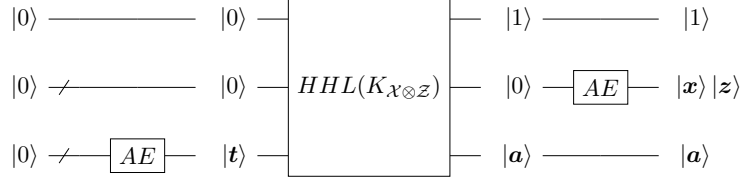


Figure 3.4 Outline of training part in proposed quantum SVM model. First, encoding training data label as amplitude in third register, the state becomes $|0\rangle|0\rangle|t\rangle$. And then applying HHL algorithm with $K_{X \otimes Z}$, the state becomes $|1\rangle|0\rangle|a\rangle$. 1 in first register mean success of HHL algorithm. Lastly, encoding training data as amplitude in second register, the state becomes $|1\rangle|x, z\rangle|a\rangle$

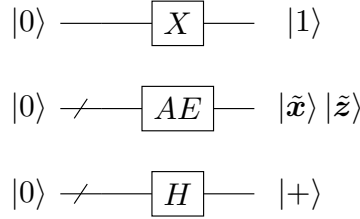


Figure 3.5 Outline of testing part in proposed existing quantum SVM model. X gate is applied for first register to match $|1\rangle$ in first register of training part. By amplitude encoding for second register, encoding test data in second register. H gate for third register makes superposition state to evaluate inner product

We define this state $|\psi_{test}\rangle$, and at this point the overall quantum state is $\frac{1}{\sqrt{2}}(|\psi_{train}\rangle|0\rangle + |\psi_{test}\rangle|1\rangle)$. After this, the inner product $\langle\psi_{train}|\psi_{test}\rangle$ is calculated. The evaluation is done using signed swap test, described in section 3.1.4, and finally the prediction is obtained.

The complexity of the proposed model is reflected in the bottleneck, i.e., calculation of the HHL algorithm, so changing the implementation of the HHL algorithm affects the total complexity of our model. The time complexity of the fastest HHL algorithm is $O(s\kappa \text{poly}(\log(s\kappa/\epsilon)))$. Therefore, if the kernel matrix has $O(n^2)$ rows and $O(n^2)$ columns, the complexity of the fastest HHL algorithm is $O(s^2\kappa^2 \text{poly}(\log(s\kappa/\epsilon)))$. This is better than $O(n^2s^2\kappa^2 \log(1/\epsilon))$ of the conjugate gradient method, the fastest classical matrix inversion algorithm. Especially in the case $s, \kappa, 1/\epsilon = O(\text{poly}(\log n))$, when $s, \kappa, 1/\epsilon$ grow exponentially smaller than n , the proposed algorithm achieves exponential acceleration compared to classical algorithms [52]. This implies the possibility of quantum advantage in time complexity for the dataset such that the kernel matrix is sparse. In this case, the time complexity of the proposed quantum model for pairwise classification is $O(\text{poly}(\log n))$, compared with $O(n^2 \text{poly}(\log n))$ of the existing classical model. It is a larger reduction than in the case of binary classification, where $O(n \text{poly}(\log n))$ in the classical model is reduced to $O(\text{poly}(\log n))$ in the quantum model.

We use $O(\log n)$ qubits to store the training and test data, some (constant) qubits for binary expression of the eigenvalue in the HHL algorithm, and some qubits as ancilla qubits. Therefore, if only $O(\log n)$ qubits are needed for the binary representation of the eigenvalue

$K_{\mathcal{X}} \otimes K_{\mathcal{Z}}$, the space complexity of our model is $O(\log n)$, smaller than the $O(n^4)$ of the classical methods. In terms of space complexity, our model can be applied to large datasets.

3.3 Experiment

3.3.1 Setup

In an experiment, we performed pairwise classification using a quantum SVM with a Kronecker kernel and a classical LS-SVM with a Kronecker kernel, both on the same dataset. For a Kronecker kernel, we use two different base kernels, the linear kernel $K_{\mathcal{X} \otimes \mathcal{Z}}(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x} | \mathbf{z} \rangle$ and the quadratic kernel (polynomial kernel with degree 2 and no bias term) $K_{\mathcal{X} \otimes \mathcal{Z}}(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x} | \mathbf{z} \rangle^2$. The results were used to determine whether the quantum SVM with a Kronecker kernel has the same classification rate as the classical LS-SVM. We use Gaussian elimination for the calculation of the inverse matrix in the classical LS-SVM model.

In terms of hardware, classical and quantum methods use the same computer, WA9A-G200/WT. For the execution of quantum circuits, we use the simulator (`qasm_simulator`) in Qiskit, a quantum computation framework provided by IBM. Note that we did not use an actual quantum computer but rather a simulator due to the size of the circuit needed in the experiment. An actual quantum computer without error-correction system would not be able to provide accurate calculation results because of its high error rate, which causes unintentional changes of quantum state and measurement results. The results come probabilistically, so the prediction result is determined by the results of 8192 runs.

In the experiment, we used the basic OCR image dataset (see Figure 3.6), where each example contains an image of a digit, used in a previous study [32]. Each image in the dataset has 128×128 size, and each pixel has two values 0 and 1, which corresponds to white and black. Since this dataset is too large to process the data directly on a simulator, we used the features calculated from the data as input data instead of the original image data. By feature extraction, we used two regularized values instead of raw image input. An extracted feature vector, not the raw image input, was also used for the classical model.

First, we calculated the horizontal (HR) and vertical (VR) ratios of pixels. HR is the ratio of the number of black pixels in the upper half of the image to the number of black pixels in the lower half of the image, and VR is the ratio of the number of black pixels in the left half of the image to the number of black pixels in the right half of the image. Then, we calculate elements of the feature vector v'_1 and v'_2 by $v'_1 = HR \times 1.3 - 0.62$ and $v'_2 = VR \times 0.95 - 0.42$ (linear mapping). Next, v'_1 and v'_2 are scaled to v_1 and v_2 so that $(v_1)^2 + (v_2)^2 = 1$ holds. This regularized feature vector (v_1, v_2) is used as input for training and test data. These feature-vector definitions are based on the definitions used in [32] and [33].

We use only the images of “6” and “9”. Two images (an image of “6” and an image of “9”) in the training dataset were used, so four pairs comprised the training dataset. Twenty images (ten images of “6” and ten images of “9”) were prepared for the test dataset, so we tested 400 pairs in total. In this experiment, we define the pairwise relationship between

Training data	6	9
Test data	6	9

Figure 3.6 Samples in basic OCR image dataset. Training data consists of Times New Roman font characters and Test data consists of handwritten characters

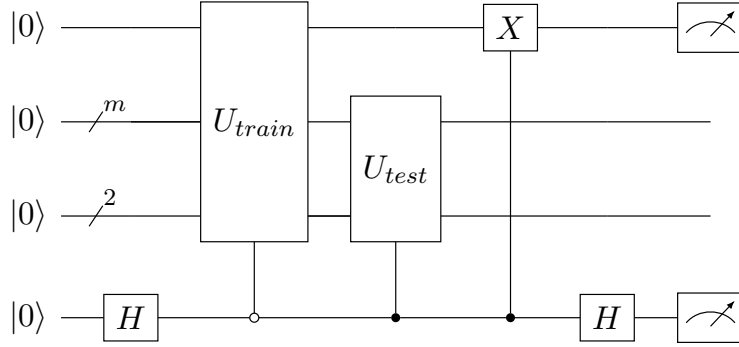


Figure 3.7 Outline of implementation of proposed model. In our experiment, the second register has m qubits, and third register has 2 qubits. First, an H gate is applied to fourth register to create a superposition. Then, unitary circuits U_{train} and U_{test} are applied controlled by the state of fourth register. After, X gate is applied to the qubit in first register for the evaluation of inner product. Lastly, an H gate is applied to fourth register again and the states of first and fourth register are measured

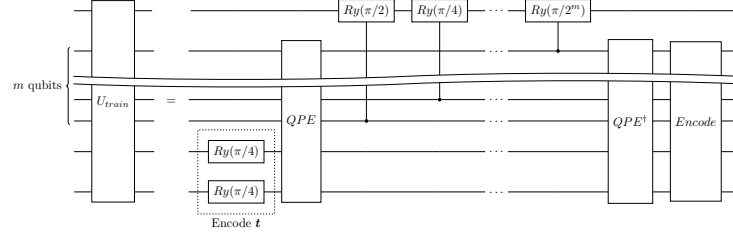
two digits as whether the two numbers are the same or not. For example, we assign the value $+1$ for the relationship between two images of 6 and the value -1 for the relationship between an image of “6” and an image of “9”.

We remark that the classical model also uses the extracted feature vector, not raw image input. In the experiment, we use one classical model and three quantum models, which vary in the size of circuit, and set the value of γ with $\gamma = 2^{-5}, 2^{-4}, \dots, 2^9$, to investigate the relationship with the hyperparameter.

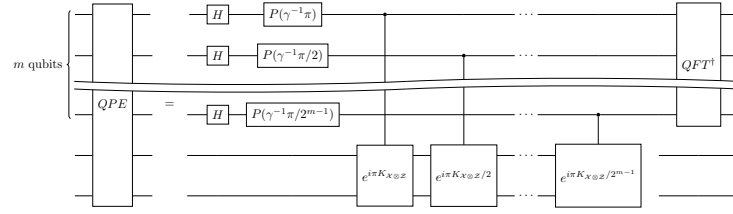
3.3.2 Implementation

Our implementation model refers to [32] in many parts, but has some changes, so we mainly note the differences of the models. The general circuit of our model is shown in Figure 3.7, which contains four registers.

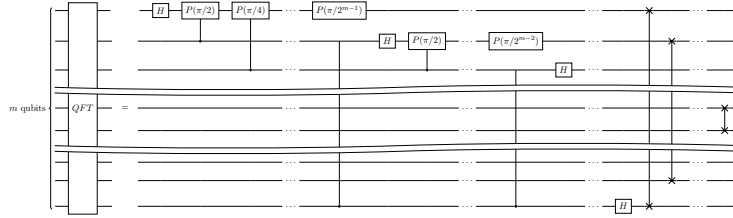
In the model, in total, $m + 4$ qubits are used (m is the number of qubits used for binary encoding of eigenvalues in quantum phase estimation (QPE)). In the implementation, we omit the bias term b for simplicity of implementation. The details of U_{train} is shown in Figure 3.8.



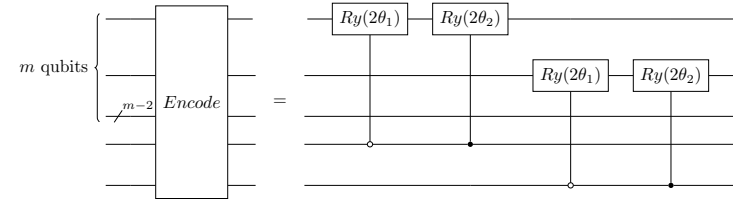
(a) General circuit of train part



(b) Circuit of Quantum Phase Estimation(QPE) part



(c) Circuit of Quantum Fourier Transform(QFT) part



(d) Circuit of encoding training data

Figure 3.8 Outline of training part implementation of proposed model. Encoding training data in the beginning, and applying HHL algorithm, then encoding training data in second register. In the part of encoding training data, encode training data based on the index of the third register. We remark that measurement operation is not applied in train part to avoid collapse of the superposition for signed swap test. θ_1, θ_2 are defined as $\theta_1 = \arctan(\frac{x_1}{x_2})$ and $\theta_2 = \arctan(\frac{z_1}{z_2})$, x_i is i -th element of train data input x and similarly for z

In the implementation model, $P(\theta)$ and $Ry(\theta)$ gate is defined as

$$P(\theta) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix}, Ry(\theta) = \begin{bmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix}, \quad (3.26)$$

and the $A^\dagger$ gate is defined as a complex conjugate of A gate, that is, $AA^\dagger = A^\dagger A = I$. First, the training data labels are encoded to the amplitudes of the third register. After that, the state of the third register is $\frac{1}{2} \sum_{i=0}^3 t_i |i\rangle$. Then quantum phase estimation (QPE) is applied to write the eigenvalue of $F_{\mathcal{X} \otimes \mathcal{Z}}$ in the second register in the form of a binary expression. Next, using the state of the second register, write the inverse of the eigenvalues to the amplitude using the Ry (Rotation around the y-axis) gate. Then, uncomputation of QPE is performed, and lastly encode training data to qubits in second register. At this stage, we encode data, not data labels. In the implementation, we separate controlled- $e^{i\theta F_{\mathcal{X} \otimes \mathcal{Z}}}$ gates into two parts, controlled- $e^{i\theta \gamma^{-1} I}$ gates and controlled- $e^{i\theta K_{\mathcal{X} \otimes \mathcal{Z}}}$ gates. In implementation, we use phase gate to express controlled- $e^{i\theta \gamma^{-1} I}$ gates. And the detail of U_{test} is shown in Figure 3.9.

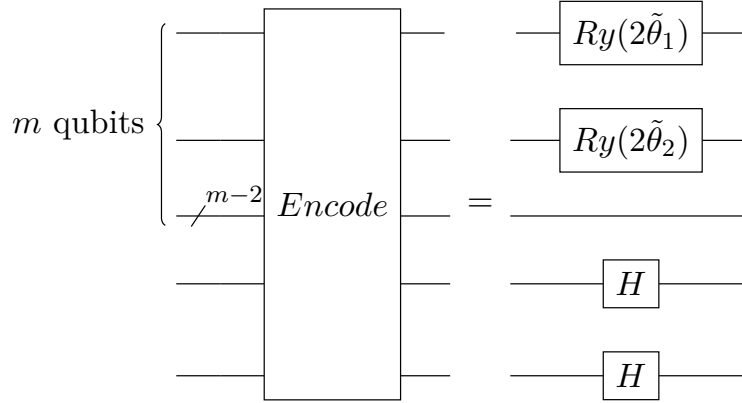


Figure 3.9 Outline of testing part implementation of proposed model. Encoding test data in second (first in this figure) register, and make superposition in third (second in this figure) register. $\tilde{\theta}_1, \tilde{\theta}_2$ are defined as $\tilde{\theta}_1 = \arctan(\frac{\tilde{x}_1}{\tilde{x}_2})$ and $\tilde{\theta}_2 = \arctan(\frac{\tilde{z}_1}{\tilde{z}_2})$, $\tilde{x}_i$ is i -th element of test data input $\tilde{x}$ and similarly for $\tilde{z}$

After applying U_{train} and U_{test} , signed swap test is needed to estimate inner product but preparing quantum state $\frac{1}{\sqrt{2}}(|0\rangle|\psi_{train}\rangle + |1\rangle|\psi_{test}\rangle)$ is difficult because measurement, done in HHL algorithm, breaks the superposition. However, in reality, in this problem setting, the evaluation of the inner product can be done from the measurement results similarly to the case that superposition $\frac{1}{\sqrt{2}}(|0\rangle|\psi_{train}\rangle + |1\rangle|\psi_{test}\rangle)$ can be obtained (see Appendix A). The quadratic kernel is implemented by replacing $|\mathbf{x}\rangle$ with $|\mathbf{x}\rangle|\mathbf{x}\rangle$, and $|\mathbf{z}\rangle$ with $|\mathbf{z}\rangle|\mathbf{z}\rangle$ in the linear kernel. We have conducted experiments with the quadratic kernel only for $m = 4$ and 6 because at least 4 qubits are needed to encode data for the quadratic kernel.

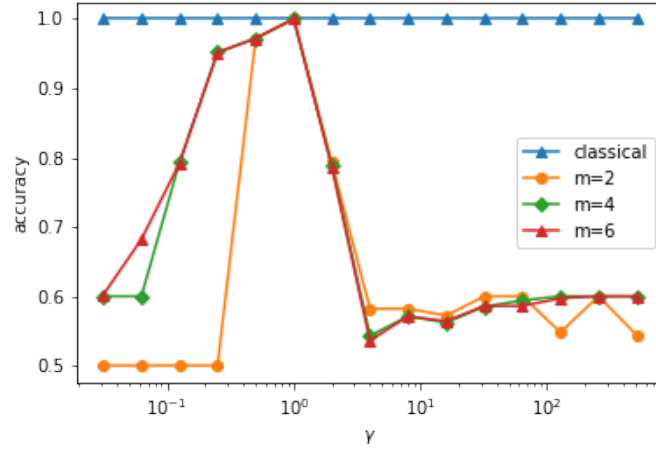
3.3.3 Results

Results are shown in Figure 3.10 and Figure 3.11. Figure 3.10 shows the relationship between the strength of regularization and accuracy for each model. The horizontal axis

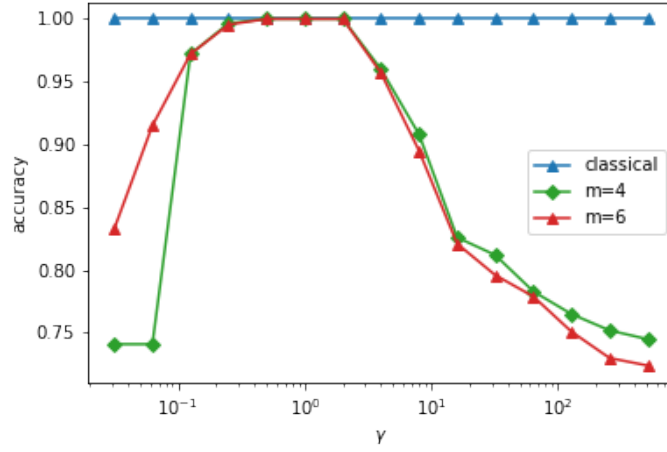
(γ label) represents the logarithm value of regularization parameter γ and the vertical axis (accuracy label) represents the accuracy. Figure 3.10(a) shows the result in linear kernel setting. The blue line shows the result of classical model and the other lines show the results of proposed model for different hyper-parameter m . The accuracy of the proposed model seems sensitive to the value of γ . In the case of $\gamma = 1$ the proposed model achieves an accuracy of 1.0 for all m values. On the other hand, if the value of γ is small or large, the accuracy of the proposed model decreases, while the accuracy of the classical model remains high. For classical model, accuracy is 1.0 in all cases, that means the classical model achieves perfect classification in this dataset. In addition, accuracy in the classical model does not change by the value of γ , which differs from the tendency of the proposed quantum model. Figure 3.10(b) shows the result in quadratic kernel setting. When using the quadratic kernel, the accuracy of the proposed model becomes more stable than in the linear-kernel setting. Particularly, the accuracy for small and large values of γ is significantly improved. For the classical model, the accuracy is still not sensitive to γ .

We suppose this phenomenon is caused by the limitations on the representation of the eigenvalues in quantum SVM. Eigenvalues are precisely calculated in the classical model, but not necessarily in quantum SVM. In quantum phase estimation subroutine in quantum SVM, all eigenvalues of the kernel matrix, which is a Hermitian matrix, are assumed to be included in the range $[0, 1)$ to be calculated precisely. In fact, when the value of γ is too small, the eigenvalues of $K_{\mathcal{X} \otimes \mathcal{Z}} + \gamma^{-1}I$ become too large (close to γ^{-1}) and exceed the range $[0, 1)$, which causes inaccurate results of calculation. When the value of γ is too large, the eigenvalues of $K_{\mathcal{X} \otimes \mathcal{Z}} + \gamma^{-1}I$ are close to those of $K_{\mathcal{X} \otimes \mathcal{Z}}$, which are approximately (0.561, 0.188, 0.188, 0.063). In quantum SVM, the eigenvalues of $K_{\mathcal{X} \otimes \mathcal{Z}}/2^m$ are binary encoded using m qubits, for example, 0.625 is expressed as 0.1010 when $m = 4$. This expression causes cancellation of significant digits because the eigenvalues are expressed in $1/2^m$ increments, whereas the maximum eigenvalue of $K_{\mathcal{X} \otimes \mathcal{Z}}/2^m$ is about $0.561/2^m$. These explanations give the reason why the accuracy is insensitive to γ in classical SVM, but sensitive in quantum SVM.

Figure 3.11 shows the relation between the strength of regularization and execution time for each model. The horizontal axis (γ label) represents the logarithm value of regularization parameter γ and the vertical axis (execution time label) represents the execution time. Same as Figure 3.10, blue line shows the result of classical model and the others show the results of proposed model for different hyper-parameter m . In both of the kernel settings, execution time in proposed model per datum can be seen proportional to hyper-parameter m , which seems natural because the depth of quantum circuit in proposed model increases proportional to hyper-parameter m . The execution time does not vary by the value of γ . In addition, execution time of the classical model is more than a thousand times faster than the proposed model executed using the quantum simulator. We suppose that the advantage of the quantum method will be kept theoretically. The main purpose of the experiment with small data and the quantum simulator is to confirm implementability. The advantage of the quantum method will become apparent when we apply the method to large data with a real quantum device in the future.

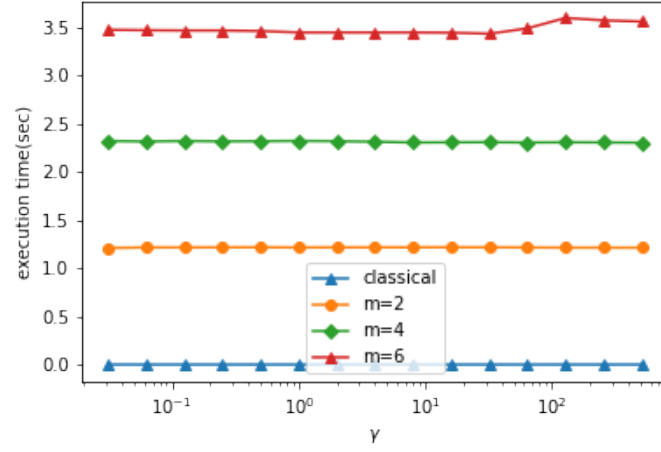


(a) Linear kernel

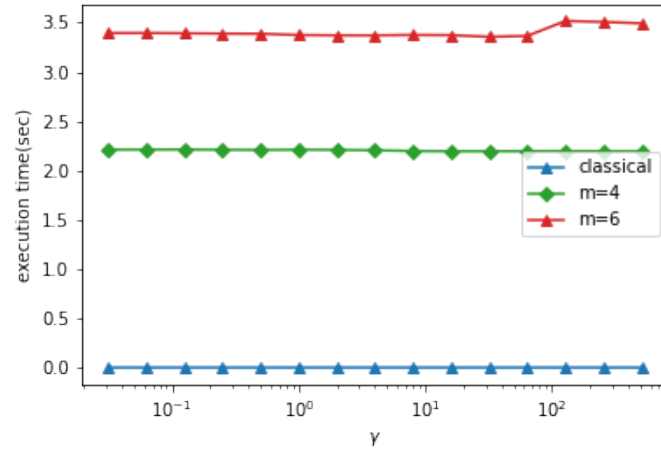


(b) Quadratic kernel

Figure 3.10 The accuracy with the classical model and the proposed model for different values of γ . The accuracy of the proposed model is sensitive to the value of γ while that of the classical model is insensitive



(a) Linear kernel



(b) Quadratic kernel

Figure 3.11 Execution time with the classical model and the proposed model for different values of γ . The execution time is insensitive to the value of γ . The execution time of the proposed model is proportional to the value of m , which controls the size of the quantum circuit in the model

3.4 Discussion

We investigated the theoretical speedup of pairwise learning obtained by using the Kronecker kernel and showed that the accuracy of a model with a Kronecker kernel can approach that of the classical one. However, it is necessary to investigate whether an actual quantum device has performance comparable to that of conventional devices. From the results and findings obtained through this study, we present several points related to the performance of quantum machine learning.

One such discussion is about kernel expression. Although constructing controlled- $e^{i\theta F_{\mathcal{X} \otimes \mathcal{Z}}}$ gates is difficult in general, there is an algorithm that approximates such gates, known as density matrix exponentiation [55]. However, applying this algorithm to our model seems difficult because density matrix exponentiation needs to calculate traces many times and then the superposition collapses by measurement. One way to solve this problem is using an algorithm which makes quantum state $\frac{1}{\sqrt{2}}(|\psi_1\rangle + |\psi_2\rangle)$ from the quantum state $|\psi_1\rangle|\psi_2\rangle$, and using this algorithm, we can make the state $\frac{1}{\sqrt{2}}(|\psi_{train}\rangle + |\psi_{test}\rangle)$ from the state $|\psi_{train}\rangle|\psi_{test}\rangle$, which can be prepared easily. Currently, such an algorithm has not been found as far as we investigated. Therefore, finding such an algorithm (or proving that such algorithm cannot be made) helps and improves our understanding of quantum machine learning.

We should also focus on the complexity of adding the bias term b to the learning model. The matrix $F_{\mathcal{X} \otimes \mathcal{Z}}$ is an $(n^2 + 1) \times (n^2 + 1)$ matrix, and preparing such a matrix as a quantum state (in the form of density matrix) is complicated compared to the case without a bias term. Finding an effective implementation to prepare such a matrix will improve usability of the quantum SVM model.

We also should think about performance of other types of pairwise kernels, such as the Cartesian kernel [56]. Identifying which type of models benefit from quantum computation is very important as a step to developing quantum machine learning. We assumed that the Kronecker product can be computed faster in quantum computation, but more detailed research is needed on which machine learning models and problems this approach can be applied to. In addition, the size of the problems that can be handled by quantum computers is currently quite limited, so evaluating learning performance experimentally requires defining appropriate problems.

3.5 Conclusion

In this chapter, we explored an application of quantum computation in the field of pairwise classification and proposed a Harrow-Hassidim-Lloyd (HHL)-based quantum support vector machine model. Pairwise classification models play an important role in practical problems, but its huge computational complexity weakens the practicality for large datasets. We focused on the structural commonality between Kronecker kernel and quantum computation and proposed a model for applying a quantum support vector machine with a Kronecker kernel to pairwise classification. We theoretically demonstrated that the proposed model is faster and more scalable than the conventional models. Especially in the case that the kernel

matrix is guaranteed to be very sparse, the proposed model can be exponentially faster than the existing models. Under specific conditions, our model works in $O(\text{poly}(\log n))$ for the size of training data n , while the existing classical model works in $O(n^2 \text{poly}(\log n))$. This reduction has a larger impact than the reduction from $O(n \text{poly}(\log n))$ to $O(\text{poly}(\log n))$ by the standard quantum SVM. In addition, we experimentally demonstrated that the accuracy of a model with a Kronecker kernel can approach that of the classical one. As for space complexity, the proposed model requires a very small amount of space compared to the existing models. This connection of pairwise learning with quantum machine learning is the first step towards developing a quantum pairwise learning domain.

Chapter 4

Positive-Unlabeled Learning for Entanglement Detection

Our contributions in this chapter are as follows:

- We devised a learning method for entanglement detection that does not rely on explicit entangled state data and uses the concept of PU learning. Since reliably generating arbitrary entangled state data in the quantum state generation process is challenging, constructing a training dataset presents difficulties. However, the challenge of generating quantum state data that is definitively either separable or entangled—that is, valid quantum state data—is relatively minor. Our approach prioritizes the availability of quantum state data and offers a method that is more practical for experimental implementation than conventional approaches requiring explicitly entangled state data.
- We developed a method for generating dense datasets that define classification boundaries in fully-separable state detection, enabling the evaluation of a trained model for entanglement detection. Through experimental verification, we demonstrated that the proposed method achieves satisfactory performance on such datasets.
- The effectiveness of the proposed method was verified using a synthetic dataset representing mixed states ranging from 2 to 6 qubits. The proposed PU learning-based method exhibited significantly higher AUC scores compared with two baseline one-class methods, reinforcing the effectiveness of the proposed method.

The structure of this chapter is as follows. Section 4.1 explains the background, including the problem definition and existing approaches for entanglement detection. Section 4.2 explains the proposed model and the data generation methods for the evaluation dataset in numerical experiments. Section 4.3 explains related work in entanglement detection, which takes different approaches to the problem setting or methods due to data availability. Section 4.4 describes the numerical experiments conducted on the prepared dataset using one-class classification methods as baselines. Section 4.5 discusses the limitations and future work of the proposed method. We conclude the chapter in section 4.6. For an introduction to this chapter, refer to section 2.10.3.

4.1 Background

4.1.1 Problem setting

In this chapter, we consider a variant of PU learning and aim to obtain accurate inference for new input data, provided as density matrices representing quantum states. For a given integer $n \geq 2$, which determines the dimensions of the input matrix, let $\mathcal{X} \subset \mathbb{C}^{2^n \times 2^n}$ denote the data space and let $\mathcal{Y} = \{+1, -1\}$ denote the output space. We assume that the input data consist of $2^n \times 2^n$ complex matrices in $\mathcal{X}$, with some of the data provided as positive examples of size m_{pos} , denoted by $\mathcal{D}_{pos} = \{\rho_1^{pos}, \dots, \rho_{m_{pos}}^{pos}\}$, and as unlabeled examples of size m_{unl} , denoted by $\mathcal{D}_{unl} = \{\rho_1^{unl}, \dots, \rho_{m_{unl}}^{unl}\}$. Defining the decision function and true labeling function as $f, f^* : \mathcal{X} \rightarrow \mathcal{Y}$, which means $f^*(\rho_k^{pos}) = +1$ for $k = 1, \dots, m_{pos}$, our goal is to identify f that minimizes the expectation $\mathbb{E}_{\rho \sim P(\rho)}[f(\rho) - f^*(\rho)]$ under the data probability distribution $P(\rho)$.

In addition, we assume access to class prior information $\pi_P = \Pr[f^*(\rho) = +1]$ and that the data distribution is consistent between the training and test datasets. Also, we note that this chapter does not address the specifics of obtaining the density matrix. Instead, we assume access to its exact values. In experimental settings, estimates—such as expected values—derived from repeated measurements that depend on the elements of the exact density matrix are often used [57, 58].

4.1.2 Detailed formalization of entanglement detection

Entanglement is a fundamental characteristic of quantum systems, and identifying its presence in an input quantum state is a critical challenge in the field of quantum information. The simplest formulation of the entanglement detection problem occurs in a two-qubit system, which is defined as follows.

Definition 4.1 (Entanglement detection in a two-qubit system, Definition 3 in [39]). Given a quantum state represented by a two-qubit density matrix ρ , entanglement detection is the problem of determining whether there exists a set of tuples $(p_k, \rho_k^A, \rho_k^B)_{k=1}^M$ that satisfies $\rho = \sum_{k=1}^M p_k \rho_k^A \otimes \rho_k^B$, subject to the conditions that $p_k \in (0, 1]$ and $\sum_{k=1}^M p_k = 1$ and that ρ_k^A and ρ_k^B are one-qubit density matrices for $k = 1, \dots, M$. The state ρ is classified as separable if such a set exists and entangled otherwise.

For systems containing more than two qubits, the classification of quantum states becomes increasingly complex due to the presence of multiple types of separability and entanglement. In this chapter we primarily focus on distinguishing fully-separable states from other quantum states and define the problem in this case as follows.

Definition 4.2 (Entanglement detection problem for fully-separability). Given a quantum state represented by an n -qubit density matrix ρ , the entanglement detection problem for fully-separability is to determine whether there exists a set of tuples $(p_k, \rho_k^1, \rho_k^2, \dots, \rho_k^n)_{k=1}^M$ that satisfies $\rho = \sum_{k=1}^M p_k \rho_k^1 \otimes \rho_k^2 \otimes \dots \otimes \rho_k^n$, subject to the conditions that $p_k \in (0, 1]$, that $\sum_{k=1}^M p_k = 1$, and that $\rho_k^1, \rho_k^2, \dots, \rho_k^n$ are one-qubit density matrices for $k = 1, \dots, M$. The state ρ is classified as fully-separable if such a set exists; otherwise, it is not fully-separable.

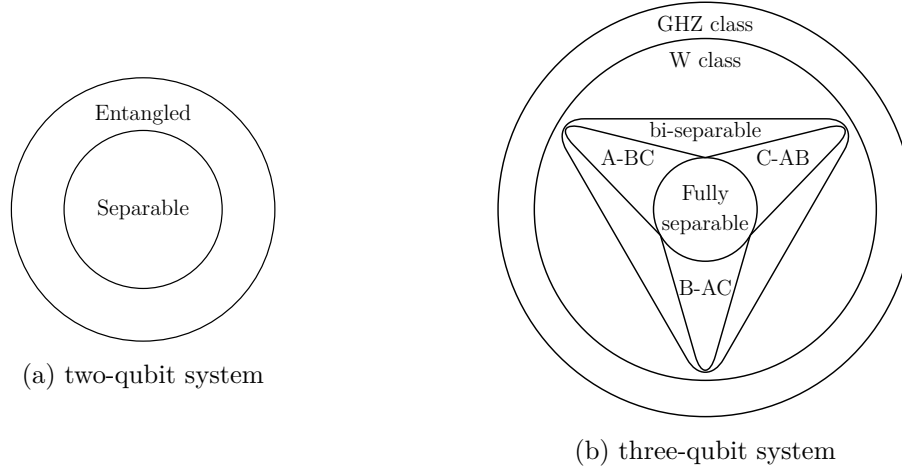


Figure 4.1 Illustrations of types of entanglement in two-qubit and three-qubit systems. The set of separable states definitively forms a convex set

Illustrations of the types of entanglement in two-qubit and three-qubit systems are shown in Figure 4.1. In a two-qubit system, as defined in Definition 4.1, each quantum state is classified as either separable or entangled. In a three-qubit system, as defined in [59], the classification of quantum states is more complicated due to the presence of six types of entanglement: fully-separable, bi-separable (A-BC, B-AC, C-AB), and genuine multipartite entanglement (GME) (GHZ class, W class). States exhibiting GME possess the highest degree of entanglement among three-qubit states. Bi-separable states, on the other hand, are those that are neither fully-separable in Definition 4.2, nor classified as GME states. Strictly speaking, some states in the mixed state set are neither bi-separable nor GME (in three-qubit systems, neither GHZ class nor W class). However, these states are typically regarded as part of the bi-separable set, which ensures convexity and simplifies their mathematical treatment. Similarly, for a general n -qubit system, we can define a set of k ($1 \leq k \leq n$)-separable states, with all of them being regarded as convex [60].

4.1.3 Existing approaches for entanglement detection

Entanglement detection has been investigated over the years within the field of quantum information science, and various approaches have been taken. Here, we briefly describe the methods that are most relevant to this study.

PPT criterion. The PPT criterion [41, 42] is among the earliest established methods for detecting entanglement in small systems and is defined as follows.

Theorem 4.3 (PPT criterion [41, 42]). Given an n -qubit density matrix ρ , ρ exhibits positive partial transposition if ρ is separable. In particular, when $n = 2$, ρ is separable if ρ exhibits PPT; that is, these conditions are both necessary and sufficient.

By using the PPT criterion, we can examine examples that illustrate the boundary between separable and entangled states in two-qubit systems, as explained below.

Example 4.4 (Bell state under depolarizing noise). Let $\rho(p)$ be a two-qubit density matrix expressed as $\rho(p) = (1 - p)\rho_{\text{bell}} + p\frac{I}{4}$, where ρ_{bell} represents the Bell state density matrix

and I denotes the identity matrix.

$$\rho_{bell} = \frac{1}{2} \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}, \quad I = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (4.1)$$

By calculating the partial transpose of $\rho(p)$, we determine that its eigenvalues are $\frac{2-p}{4}$ and $\frac{2-3p}{4}$. Consequently, $\rho(p)$ is separable when $p \in [\frac{2}{3}, 1]$ and entangled when $p \in [0, \frac{2}{3})$.

The PPT criterion is a powerful tool in that it can perfectly identify the separability of a given state; however, it cannot be applied to systems with more than two qubits. In quantum systems comprising three or more qubits, an alternative approach is required to detect entanglement.

Entanglement witness. An entanglement witness is an operator that distinguishes a particular entangled state from fully-separable states and is defined as follows.

Definition 4.5 (Entanglement witness). Given an n -qubit density matrix ρ_{ent} known to be entangled (that is, not fully-separable) and letting S denote the set of density matrices corresponding to fully-separable states on n qubits, an operator W is called an entanglement witness if $\text{tr}[W\rho_{sep}] \geq 0$ for any $\rho_{sep} \in S$.

Conceptually, entanglement witnesses have no upper limit on the size of the quantum system to which they can be applied, so they can be applied to systems with three or more qubits. In addition, we can determine that an unknown input state ρ is entangled if the output satisfies $\text{tr}[W\rho] < 0$. On the other hand, an entanglement witness cannot distinguish entangled states when the range of possible entangled states is extensive.⁽¹⁾ Furthermore, even when an entanglement witness is applicable, determining the appropriate operator W to serve as the witness is generally nontrivial, and research is underway to identify a more suitable witness operator [61].

Entanglement measures. There are various indicators for quantifying the strength of entanglement, such as entanglement entropy (Von Neumann entropy) and relative entropy. Among the various entanglement metrics, negativity is particularly relevant to our problem setting. Negativity is defined as follows.

Definition 4.6 (Negativity, Eq. (1) in [62]). Given a density matrix ρ , negativity for subsystem A is defined as $\mathcal{N}(\rho) = (\|\rho^{TA}\|_1 - 1)/2$, where ρ^{TA} denotes the partial transpose of ρ for subsystem A .

Negativity can be interpreted as a type of penalty within the PPT criterion. Since an input density matrix ρ being PPT and the negativity of ρ being zero are equivalent, in a two-qubit system, a negativity of zero corresponds to ρ being separable. A notable advantage of negativity is that it is a computable measure for mixed state inputs, whereas other measures, such as entanglement entropy, primarily apply to pure states.

⁽¹⁾Specifically, in cases where the convex hull of the possible entangled states overlaps with the set of fully-separable states, no witness exists that can perfectly distinguish these states by the value of $\text{tr}[W\rho]$, which is a linear function of ρ .

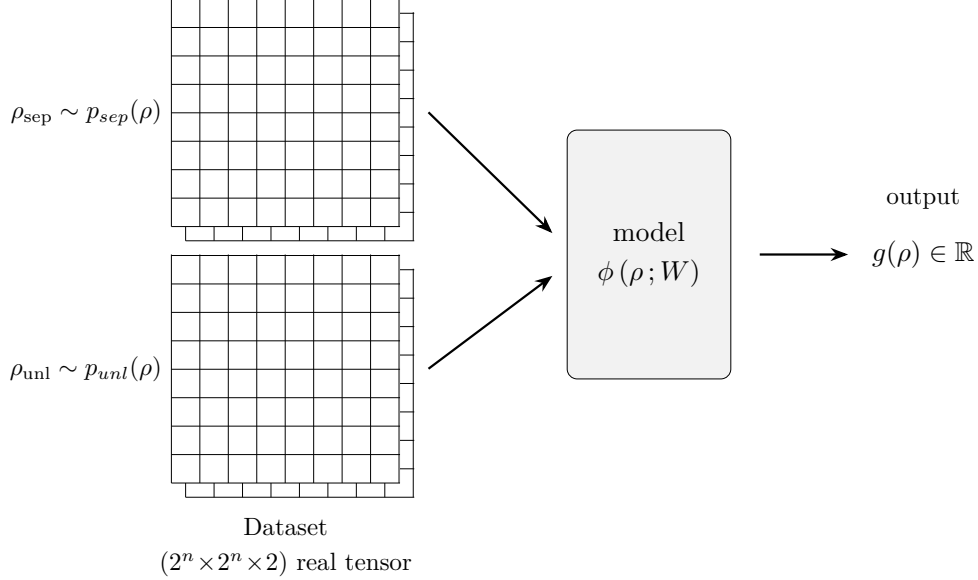


Figure 4.2 Overview of proposed method

4.2 Training entanglement detector using positive-unlabeled learning framework

For the problem setting described in Section 4.1.1, we devised a method that integrates entanglement detection with nnPU learning [16] to train an entanglement detector using only a realistically available dataset. In this section, we describe the proposed method and explain the process of data generation.

4.2.1 Proposed method

First, we briefly describe the proposed method. This method is designed for learning and inference in a classical system and is based on the assumption that exact values can be obtained for the density matrix used as input. This assumption facilitates model implementation based on the nnPU learning method [16]. An overview of the proposed method is shown in Figure 4.2.

The proposed method is primarily divided into two phases: data generation and model application. In the data generation phase, data corresponding to the density matrix is generated as positive data in separable states and as unlabeled data in states that are valid quantum states (refer to Theorem 2.3), although it is unknown whether these states are separable or entangled. Here, we examine the process of data generation using quantum circuits with random operations under depolarizing noise, starting from an initial state, such as the state $|\psi_{\text{init}}\rangle = |0\rangle^{\otimes n}$. When constructing a dataset for entanglement detection, it is desirable that the dataset be capable of spanning all possible states that may exist as either separable or entangled from the perspective of data quality. In this problem setting, labeling entangled states is generally more difficult than labeling separable states due to the following factors.

Difficulty of labeling based solely on input information. In classical machine learning applications, annotation work is often performed to assign labels to unlabeled data. For example, when determining whether characters or specific features are present in an input image, human experts can often assess the accuracy of an image simply by looking at it. However, in entanglement detection, it is common to use density matrices or estimates of expected values of certain physical quantities as input. In such cases, it is often impossible for humans to determine entanglement solely on the basis of these numerical data.

Non-guaranteed separable/entangled states after global random operations. When generating data to span the possible separable states, these states can be spanned by performing local unitary operations within each qubit. However, when generating data to span the possible entangled states, global random operations across qubits are required. These operations do not necessarily guarantee that the resulting states are entangled (see Figure 4.3).

In considering ways to avoid the difficulty of assigning labels to negative data, it is logical to develop learning methods, such as PU learning, that do not depend on negative labels so that they can be applied to more practical problem settings.

In the model application phase, the generated density matrix data are used as input, and the final output is a real scalar value indicating the degree of separability or entanglement. In an n -qubit system, the density matrix is represented as a complex matrix of size $2^n \times 2^n$. In the proposed method, this matrix is decomposed into its real and imaginary components and used as input as a real tensor of size $2^n \times 2^n \times 2$. The reasons for adopting this input format are as follows.

Input format of density matrices. The proposed method is designed to take a density matrix input. However, vectors consisting of the expected values of physical quantities, such as Pauli observables, can also be used to represent quantum states. We chose the density matrix format for input due to its compatibility with classical shadow methods [58]. These methods store snapshots of the density matrix of a quantum state, enabling estimation of the original density matrix values from their expectation values. The number of snapshots can be adjusted in accordance with the required accuracy, enabling the generation of density matrices with arbitrary accuracy. Given this flexibility, we adopted the density matrix format in this study.

Input format: Real tensor representation of density matrices. While density matrices are inherently complex, our method represents them as real-valued tensors by decomposing them into their real and imaginary parts. This approach offers several advantages. 1) Ease of combination with machine learning frameworks: In machine learning research, real-valued input settings are more common than complex number input ones, and model implementation is mainly based on the assumption that the inputs are given as real values. Additionally, models that take color images as input often use input formats, such as $32 \times 32 \times 3$ and $224 \times 224 \times 3$, that are similar to the $2^n \times 2^n \times 2$ format of real-valued density matrices. This suggests that image input models may be readily adapted for density matrix input models. 2) Adaptability and performance considerations: Although some studies suggest that methods tailored for complex inputs might offer performance improvements [63], such benefits are highly dependent on the specific problem setting and dataset. We used



(a) process to generate separable states (b) process to generate entangled states

Figure 4.3 Processes for generating separable and entangled states. Although preparing entangled data is challenging due to verification difficulties, the output state is reliably classified as unlabeled

the real number input format in the proposed method, considering both the convenience of using real number inputs and the performance variations when using complex number inputs.

This method produces real scalar values that are used to calculate the AUC—a metric commonly used in PU learning and anomaly detection. When binary classification is desired (i.e., assigning labels $\{+1, -1\}$), applying an appropriate threshold to these real values enables evaluation using the AUC or F1 score.

4.2.2 Data generation

When framing the entanglement detection problem as a machine learning problem, one key challenge is that the distribution ranges of positive and negative example data are adjacent. This closeness can lead to misclassification even with slight shifts in the classification boundary, creating a substantial obstacle to reliable classification. For robust evaluation of models trained on such problems, it is crucial that the dataset contains a substantial number of samples near the true classification boundary. Standard entangled states—such as GHZ states, W states, and their locally transformed variants—typically do not span the region near the classification boundary. Relying solely on such states may therefore adversely affect model evaluation. To address this limitation, we generate a synthetic dataset based on a Hilbert-Schmidt measure-based data distribution combined with the PPT criterion (Theorem 4.3). This approach ensures that a large amount of the generated data is near the classification boundary between separable and entangled states, thereby enabling a more comprehensive assessment of model performance.

Data distribution of density matrix in Hilbert-Schmidt measure. Here we describe the data distribution used in our experiments. Density matrices for an n -qubit state were sampled using the following procedures [64] with $N = 2^n$.

1. Let A, B be $N \times N$ matrices in which each element A_{jk} and B_{jk} ($1 \leq j, k \leq N$) is independently drawn from the standard normal distribution, i.e., $\mathcal{N}(0, 1)$.
2. Construct an $N \times N$ complex matrix C with entries defined as $C_{jk} = A_{jk} + iB_{jk}$.
3. Define density matrix ρ as $\rho = CC^\dagger / \text{tr}[CC^\dagger]$, which ensures that ρ meets the conditions for a valid density matrix (see Theorem 2.3).

For two qubits, it has been suggested that the probability of a density matrix generated by this method being separable is $8/33$, or approximately 24% [65].

Creation of dense synthetic datasets using the PPT criterion. To create the dense datasets described above, we used the PPT criterion. This method enables perfect classification when the exact values of the density matrix elements are clearly identified, as in specific systems such as two-qubit systems. In the context of determining fully-separability, we use the PPT criterion to construct a dataset including entangled state data near the classification boundary. [45] reported an implementation for two-qubit systems based on a similar approach, and we believe that it can be extended to systems with three or more qubits. Specifically, we would generate density matrices ρ for an n -qubit system as follows: For predetermined indices a, b ($1 \leq a < b \leq n$),

$$\rho_{j_1 \dots j_n, k_1 \dots k_n} = \rho_{j_1, k_1}^1 \cdots \rho_{j_{a-1}, k_{a-1}}^{a-1} \rho_{j_a, k_a}^{a+1} \cdots \rho_{j_{b-1}, k_{b-1}}^{b-1} \rho_{j_b, k_b}^{b+1} \cdots \rho_{j_n, k_n}^n \rho_{j_a j_b, k_a k_b}^{ab}, \quad (4.2)$$

where $j_1, \dots, j_n, k_1, \dots, k_n \in \{1, 2\}$ and $\rho_{j_1 \dots j_n, k_1 \dots k_n}$ represents the element located at the $(\sum_{l=1}^n (j_l - 1) * 2^{n-l} + 1)$ -th row and $(\sum_{l=1}^n (k_l - 1) * 2^{n-l} + 1)$ -th column in ρ . This indexing convention is similarly applied to the following matrices: ρ^{ab} , which is a 4×4 density matrix according to Hilbert-Schmidt measure described above, and 2×2 density matrices, which are sampled similarly. When $b = a + 1$, the overall density matrix can be expressed more simply as

$$\rho = \rho^1 \otimes \cdots \otimes \rho^{a-1} \otimes \rho^{ab} \otimes \rho^{b+1} \otimes \cdots \otimes \rho^n \quad (4.3)$$

A dataset generated using this formulation is adjacent to the fully-separable data on the entangled side, ensuring coverage near the target classification boundary.

4.3 Related work

Various methods have been explored for using machine learning to detect entanglement. This section summarizes the most relevant research. Table 4.1 provides an overview of related work. The abbreviations used in the table are as follows: FCNN (fully-connected neural network), AE (autoencoder), LSTM (long short-term memory), CNN (convolutional neural network), SNN (Siamese neural network), SVM (support vector machine), AutoML (automated machine learning), QNN (quantum neural network), QSVM (quantum support vector machine), and RBM (restricted Boltzmann machine).

Neural network-based approaches are among the most prevalent methods for entanglement detection using machine learning. [66] investigated the effect of the number of measurements on classification performance, while [67] examined how deep models respond to variations in data distribution. [88] presented methods for classifying GME using a neural network, and [68] introduced an approach based on deep neural networks with feature selection. Additionally, [69] presented a method using features based on relative entropy, while [4] introduced a method using features derived from Bell's inequality within neural networks. Further, [5] explored methods based on convex hull approximation in connection with ensemble learning, and this technique has also been employed in [89], which utilizes SLOCC (stochastic local operations and classical communication) for data generation. [90] also proposed a method based on convex hull approximation for systems other than qubits. [91] demonstrated the use of neural networks to assess the entanglement structure of the noisy quantum generalized GHZ state. [92] concentrated on the standard form of input

Table 4.1 Selected studies and a comparison on entanglement detection using machine learning methods

References	Methods	Data availability		
		P	N	U
[66, 67, 68, 69, 4, 70, 71]	FCNN	✓	✓	
[72]	FCNN,AE	✓	✓	
[73]	FCNN,LSTM	✓	✓	
[74]	CNN	✓	✓	
[75]	CNN,Transformer	✓	✓	
[76]	CNN,SNN	✓	✓	
[43, 44, 77, 78]	SVM	✓	✓	
[5]	Bagging	✓	✓	
[79]	AutoML	✓	✓	
[80, 81]	QNN	✓	✓	
[82]	QSVM	✓	✓	
[83, 84]	RBM			✓
[85]	AE			✓
[45]	CNN,SNN	✓		
[86]	FCNN	✓	✓	✓
[87]	SVM	✓	✓	✓
Ours	CNN	✓		✓

features and developed a technique for reducing dimensionality. [93] developed an optimization method for models that accounts for the effects of experimental noise. [94] proposed a detection method that utilizes supervised learning through estimation of negativity. [95] proposed a classification approach predicated on the characteristics of observables in relation to entanglement states. [96] focused on detecting fundamental quantum resources, which are more general than entanglement detection, and proposed a detection method using fully connected neural networks. [73] presented a method of detecting entanglement using only information from local measurements.

Several studies have explored entanglement detection methods designed specifically for matrix-based input representations. [74] introduced a convolutional neural network model for GME verification and demonstrated its effectiveness in environments with specified noise. [97] proposed a classification method based on statistical images derived from the transverse-field ferromagnetic Ising model. [75] presented a hybrid approach combining a convolutional neural network and a Transformer model to distinguish between W states and GHZ states. They later proposed a multi-view neural network model with the objective of reducing computational cost [71]. XpookyNet, as introduced in [98], focuses on reducing the number of measurements required during training. [72] developed a complex-valued autoencoder model applicable to qubit systems of up to four qubits. Additionally, [76] presented a model that accounts for invariances in entanglement properties under local

unitary operations and qubit permutations. [99] advanced a computational resource-aware classification method for entanglement.

In addition to neural network-based methods, several studies have explored alternative machine learning techniques for entanglement detection. [43] proposed constructing entanglement witness operators using a support vector machine to classify unknown input quantum states. Other SVM-based approaches have been reported [77, 100]. [44] introduced a method that calculates the expectations of selected Pauli observables—quantities that are computationally efficient to evaluate—and uses them as features for entanglement detection tasks. In relation to this method, [78] enhanced efficiency in terms of computational complexity. [101] proposed a method employing a support vector machine for entanglement detection in high-dimensional systems. These approaches use a two-class SVM to solve classical binary classification problems. Additionally, [83] presented a method using restricted Boltzmann machines for entanglement detection in pure states. Subsequently, they expanded the methodology to encompass the application to mixed states [84]. [102] proposed a hybrid approach combining quantum neural networks with classical fully connected neural networks. [103] investigated the efficacy of Transformer models when presented with millions of training data points for the purposes of pre-training and fine-tuning. [85] proposed a methodology employing autoencoders to ascertain the separability of quantum states. [104] presented an empirical discussion of the challenges associated with training entanglement classifiers based solely on information from local measurements. [105] proposed a method leveraging neural networks to ascertain optimal measurements in the entanglement detection problem. [106] demonstrated the generation and detection of entanglement on quantum hardware using manifold learning. [79] reported on the effectiveness of AutoML in the detection of entanglement. [107] proposed utilizing quantum extreme learning machines for entanglement witnessing. [108] conducted exhaustive experiments and discussions on a range of topics, including the classification of quantum states in optical systems. [109] verified the performance of detection in scenarios where measurements are incomplete. [110] proposed the concept of a universal quantum witness machine, which would be utilized for the study of fundamental properties of quantum systems.

For formulation as one-class classification, [45] introduced a deep learning approach based on ideas from Siamese networks and GANs. Their method frames entanglement detection as an anomaly detection problem, a type of unsupervised learning. To achieve this they developed methods that rely exclusively on separable data.

The concept of utilizing quantum states as unlabeled data has been explored in various research studies. [86] devised a semi-supervised learning method and used it to demonstrate how using unlabeled data can enhance entanglement detection. [87] enhanced the performance of semi-supervised SVMs for GME state classification using techniques such as grouping unlabeled data and using iterative labeling.

Related research has expanded in various directions. [111] characterized the separability of states using methods based on manifold learning embedding techniques. [80] formulated the entanglement detection problem as a two-player zero-sum game, and solved it using a quantum neural network. [70] applied a neural network to a continuous-variable problem setting. [81] investigated parametrized quantum circuits as a model for constructing en-

tanglement witness operators. [112] formulated entanglement detection as a multi-armed bandit problem and proposed a detection algorithm with probabilistic theoretical guarantees. [82] proposed a cross-domain classification method that utilizes quantum support vector machines. They later proposed a classification method employing quantum neural networks in conjunction with a problem-inspired data generation circuit [113]. [114] investigated the feasibility of entanglement detection on a real quantum computer. [115] investigated the use of discrete-variable quantum neural networks and continuous-variable quantum neural networks for the entanglement detection problem.

4.4 Numerical experiments

We conducted numerical experiments to evaluate the effectiveness of the proposed PU learning-based framework for entanglement detection. To assess whether incorporating unlabeled data as training data enhances classifier performance, we compared the performance of two baseline methods that use only positive data with that of the proposed method, which uses both positive data and unlabeled data. In ablation studies, we verified the behavior of the proposed method when changing the network and input features.

4.4.1 Methodology

Baseline methods. We evaluated the proposed method using two one-class classification methods. Their hyperparameter settings were primarily based on their original implementations unless otherwise specified. One baseline method was the network introduced in [45], which combines a Siamese network with a complex neural network. We refer to it as PS-CxGAN on the basis of its elements—a pseudo-Siamese network, a complex neural network, and a GAN. We used RMSProp optimizer with a momentum of 0.9, and set the learning rate to 10^{-4} for all 100 training epochs. Weight decay was not applied to PS-CxGAN. The other baseline method was the DeepSVDD model introduced in [8], in which the DeepSVDD model is pretrained using autoencoder followed by training with a convolutional neural network. We used the Adam optimizer with a weight decay of 5×10^{-7} for pretraining and training, while keeping the default parameter settings for the other hyperparameters. The initial learning rate was set to 10^{-4} and reduced to 10^{-5} after 250 pretraining epochs and 100 training epochs. The DeepSVDD model was pretrained for 350 epochs and trained for 150 epochs. *Proposed method.* We used nnPU learning [16] as the PU learning method for entanglement detection. We used the Adam optimizer with a weight decay of 5×10^{-4} while the other hyperparameters were kept at their default values. The learning rate was set to 10^{-4} for all 100 training epochs, a value lower than that used in the original implementation. For all three methods, the batch size was fixed at 128. To further investigate the effectiveness of unlabeled data, comparative experiments were also conducted using a method that made label information available for all data. The same model as the proposed method was used, except that the PN risk (equation 2.18) was applied for risk estimation on the nnPU.

Table 4.2 Network architecture used in numerical experiments. The same network architecture was used for each method except for the number of units in the final layer (*). Only the nnPU method changed the final output dimension to 1

	PS-CxGAN-based network
2-qubit	$C[20 \times (2 \times 2)] - C[60 \times (2 \times 2)] - 240 - 192 - 20^*$
3-qubit	$C[20 \times (3 \times 3)] - C[60 \times (3 \times 3)] - C[60 \times (3 \times 3)] - 240 - 192 - 20^*$
	DeepSVDD-based network
4-qubit	$C[32 \times (5 \times 5)] - C[64 \times (5 \times 5)] - 1024 - 64^*$
5-qubit	$C[32 \times (5 \times 5)] - C[64 \times (5 \times 5)] - C[128 \times (5 \times 5)] - 2048 - 128^*$
6-qubit	$C[32 \times (5 \times 5)] - C[64 \times (5 \times 5)] - C[128 \times (5 \times 5)] - C[256 \times (5 \times 5)] - 4096 - 256^*$

4.4.2 Architectures

To minimize performance differences arising from variations in model parameters and structure, we aligned the architecture of the three methods with those of PS-CxGAN and DeepSVDD for each comparison. The resulting architectures are shown in Table 4.2.

Here we denote a convolutional layer with kernel size k and d output channels as $C[d \times (k \times k)]$, while a linear layer is specified solely by its number of output units. For the PS-CxGAN-based method, we applied batch normalization between convolutional layers and used the ReLU activation function between layers, including linear layers. In this method, half of the channel and unit numbers are used to represent the real and imaginary parts, respectively. Therefore, the total number of channels and units remains constant. For the DeepSVDD-based method, we applied batch normalization, followed by the leaky ReLU activation function, and a 2×2 max pooling operation, in that order between convolutional layers. We omitted both the batch normalization parameters and the bias term in the linear layer of the DeepSVDD method to prevent hypersphere collapse [8].

4.4.3 Setup

Our synthesized dataset included PU data representing density matrices of fully-separable states and entangled states—meaning the dataset contained two labels. We used only positive-labeled data for the one-class classification methods and both positive and unlabeled data for the proposed method. The dataset consisted of 60,000 training samples and 12,000 test samples, both of which included the same number of positive and negative samples, i.e., $\pi_p = 0.5$. The synthesized dataset contained 20,400 explicit positive samples and 39,600 unlabeled samples, with all 30,000 negative samples included in the unlabeled data. This means that the ratio of positive samples within the unlabeled data was 8/33, based on sampling from the Hilbert-Schmidt measure. To ensure equality in the amount of data between the compared methods, we conducted additional experiments using a sub-dataset of 6,000 positive-unlabeled samples (1/10 samples) for the proposed method. These samples were selected to maintain the same ratio of labeled data as in the original set of 60,000 samples (full samples).

4.4.4 Results

The experimental results are shown in Table 4.3, where each value represents the average and standard deviation from five trials using different seeds. As shown in Table 4.3, the proposed PU learning-based method achieved better performance than the baseline methods. Although the results of nnPU using 1/10 samples were lower than those using full samples, this decrease is reasonable; nevertheless, they showed high performance compared to the baseline methods. In this problem setting, it was assumed that the positive data spanned most of the true positive data set, leading to the expectation that the baseline methods would perform similarly to the proposed method. However, the results contradict this expectation. One possible explanation is as follows. According to [12], in one-class classification, the negative class (e.g., entangled state) is not included in the given sample (positive class, e.g., separable state) and is instead considered as all other possible classes. Conversely, in the context of PU learning, unlabeled samples offer insights into the global distribution of data, encompassing the potential presence of negative samples. From this standpoint, access to valid density matrix data may have been a contributing factor to the high performance of the PU learning method in this study. Additionally, we remark that data points from the positive and negative classes are close to each other in the feature space. Consequently, in one-class classification methods that exclusively utilize positive samples, even if the given positive data can be classified as belonging to the positive class (data closer to the positive boundary), it is expected to be challenging to correctly classify negative data as negative. In PS-CxGAN, for instance, the classification of data outside the region of positive data as negative requires the generation of data within the negative data region and the training of the discriminator to accurately differentiate between the two. However, in this problem setting, there is no explicit incentive, such as a loss function design, to generate negative data uniformly across all regions where negative data could potentially exist. In scenarios where positive and negative data are distributed near the classification boundary, ambiguity in learning the boundary may contribute to suboptimal performance. Conversely, the PU learning approach involves the provision of information regarding unlabeled samples. Moreover, given that the information regarding negative samples is provided through a class prior, in conjunction with the positive data information, it is conceivable to predict points that are likely to be negative. When we compared methods that used label information for all data to estimate PN risk with the proposed method for estimating PU risk, the PN risk method naturally had a higher AUC score. Meanwhile, the proposed method achieved an AUC score similar to that of labeled data despite lacking negative data labels. These results suggest the effectiveness of the proposed method compared to situations using labeled data, as it adheres to the formulation of PU learning.

4.4.5 Ablation studies

To analyze the impact of the components of the proposed method, we conducted ablation studies in a setting with 1/10 samples of 5-qubit quantum states. We examined the effects of introducing complex neural networks [63] and input features. 1) Application of complex neural networks: complex neural networks handle input and network parameters within

Table 4.3 Experimental results. The values below show the averages and standard deviations of the execution results for five different seeds. The nnPU results show the case of using 6,000 samples (1/10 samples) and 60,000 samples (full samples) as training data

	2-qubit	3-qubit	4-qubit
PS-CxGAN	0.9889 ± 0.0217	0.5153 ± 0.0617	0.4738 ± 0.0216
DeepSVDD	0.9713 ± 0.0067	0.6521 ± 0.0324	0.5371 ± 0.0072
Proposed(1/10 samples)	0.9967 ± 0.0005	0.9579 ± 0.0103	0.9150 ± 0.0070
Proposed(full samples)	0.9998 ± 0.0001	0.9972 ± 0.0003	0.9928 ± 0.0012
+Negative (entangled) label	1.0000 ± 0.0000	0.9991 ± 0.0001	0.9967 ± 0.0006
	5-qubit	6-qubit	
PS-CxGAN	0.4619 ± 0.0277	0.4884 ± 0.0127	
DeepSVDD	0.4179 ± 0.0147	0.3925 ± 0.0096	
Proposed(1/10 samples)	0.8783 ± 0.0125	0.8359 ± 0.0116	
Proposed(full samples)	0.9862 ± 0.0050	0.9846 ± 0.0031	
+Negative (entangled) label	0.9950 ± 0.0012	0.9912 ± 0.0013	

Table 4.4 Experimental results in ablation studies. The values below show the averages and standard deviations of the execution results for five different seeds

1) Complex NNs		2) Input features	
Complex NNs	Real NNs	Pauli observables	Density matrices
0.8246 ± 0.0190	0.8783 ± 0.0125	0.6309 ± 0.0040	0.6521 ± 0.0096

the range of complex numbers. These networks perform operations based on complex numbers and are also used in the PS-CxGAN baseline method [45]. In this ablation study, we replaced the convolutional layer, activation function, and batch normalization with a complex neural network. We conducted experiments using the CReLU activation function [63]. 2) Input features: In addition to density matrices, quantum states can be represented as sets of expected values of observed quantities. One such representation is the expected values of Pauli observables, which can concisely represent quantum states. Here, we evaluate the performance of using Pauli observables as features, where they are represented as 4^5 -dimensional vectors. We used a fully connected neural network with the output sizes of $200 - 200 - 200 - 1$ as the network architecture. For comparison, we conducted additional experiments using feature vectors consisting of 4^5 values. These values were obtained by extracting the diagonal components and the real and imaginary parts of the non-diagonal components from the density matrix. The experimental results are shown in Table 4.4.

The experimental results for complex neural networks were lower than those obtained using conventional neural networks. This result suggests that it is appropriate to treat the inputs as real numbers in this problem setting.

One factor contributing to the poorer performance of complex neural networks compared to real neural networks is the issue of inductive bias in complex neural networks. Specifically, the limited parameter freedom in the convolution operations within complex neural networks did not align well with the problem setting at hand. In complex neural

networks, given a complex filter matrix $W = A + iB$ ($A, B \in \mathbb{R}^{k \times k}$) of kernel size k and the $m \times m$ complex matrix $H = X + iY$ ($X, Y \in \mathbb{R}^{m \times m}$), the convolution operation between each channel pair within complex neural networks is expressed as

$$W * H = (A + iB) * (X + iY) = (A * X - B * Y) + i(B * X + A * Y) \quad (4.4)$$

where $*$ denotes the convolution operation between two real matrices. Note that in the above convolution operation, the contribution to the real part of X and the contribution to the imaginary part of Y are performed through the same single real matrix A (the same applies to B). That is, the ratio of contribution to the real and imaginary parts of the input values is fixed. Conversely, the density matrix, which is the input feature in this problem setting, has real diagonal entries by definition so the distribution of real and imaginary values differs for each matrix element. Considering this, convolving the real part X and imaginary part Y with values of the same scale may excessively restrict the model's representation. Meanwhile, in real neural networks, the filter matrix parameters are independent between each channel pair. This suggests that the model's expressive power is higher, leading to superior performance compared to complex neural networks. Here we remark that [63] states that the difference in classification performance between real and complex models regarding experimental results on complex neural networks varies depending on the dataset, task, and architecture. We believe it is important to note that our results do not necessarily imply outcomes for different problem settings.

Regarding the input-feature results, a significant difference was observed in the one-sided t -test at the significance level 0.05. This result was a bit surprising because the two types of features can be mapped one-to-one by an affine transformation. That is, considering that the features $\mathbf{x}_d$ obtained from the density matrix and the features $\mathbf{x}_p$ obtained from the Pauli observables, there exist weights W_d, W_p and biases $\mathbf{b}_d, \mathbf{b}_p$ that satisfy the relation $\mathbf{z} = W_d \mathbf{x}_d + \mathbf{b}_d = W_p \mathbf{x}_p + \mathbf{b}_p$. These results seem to suggest that the performance of entanglement detection may be sensitively affected by feature selection. To strengthen our confidence in this point, we conducted an additional ablation study.

Further investigation and experimentation on this point led us to conclude that there is no significant difference in the information held by at least the Pauli observables and density matrices regarding label inference, and that the key point lies in the scale of the features. To analyze the impact of feature selection on model performance, we first calculated N2 and N3 for our 5-qubit test dataset; these are complexity metrics for binary classification problem proposed by [116]. These metrics estimate the difficulty of learning based solely on the input features, with N2 and N3 specifically calculated from the information of nearest-neighbor samples in the feature space. Specifically, N2 is the average ratio of intra-class to inter-class distances for each data point. N3, on the other hand, is the proportion of data points for which the nearest neighbor belongs to a different class. In other words, large values of N2 and N3 indicate that training the dataset is difficult. The specific calculation results are shown in Table 4.5.

As a result, highly similar values were obtained. These results suggest that feature selection does not significantly affect classification based on nearest neighbors. Further investigation into the difference in the distribution of input feature values confirmed that

Table 4.5 Metrics for estimating the difficulty of training datasets

	Pauli observables	Density matrices
N2	0.4944	0.4945
N3	0.4368	0.4329

Table 4.6 Additional experimental results in ablation studies. The values below show the averages and standard deviations of the execution results for five different seeds

Input	Pauli observables			Density matrices
Weight decay	5×10^{-4}	5×10^{-3}	5×10^{-2}	5×10^{-4}
AUC	0.6309 ± 0.0040	0.6475 ± 0.0089	0.6592 ± 0.0063	0.6521 ± 0.0096

the standard deviation of features based on Pauli operators is tens of times larger than that of features based on the density matrix. In this study, which utilizes quantum states distributed in or near fully separable regions, the values of each element in the density matrix decrease exponentially with the number of qubits. Conversely, the absolute values of the Pauli operator often become larger. This is thought to cause the difference in standard deviation. In machine learning models incorporating weight decay, increasing the absolute magnitude of features can excessively boost the model’s expressive power, as features with larger absolute values contribute more significantly to the output when weights are fixed. To verify this, we conducted experiments examining performance changes when features were fixed and weight decay was increased. The results are shown in Table 4.6.

Increasing weight decay from the initial setting raised the AUC score, and in the one-sided t-test at the significance level 0.05, no significant difference was observed compared to using the density-matrix features. We consider these additional observations to support the notion that there is no significant difference between Pauli observables and density matrices regarding feature design. At the same time, we also believe they suggest that attention should be paid to the scale and distribution of values when designing features, even in situations where the range of output is fixed to the range of $[-1, +1]$ due to the use of density matrices and Pauli observables.

4.5 Discussion

We proposed using a PU learning-based method to address the problem of generating data for entanglement detection. Here, we clearly state the limitations of our method. One of the major limitations of the proposed method is that it is not scalable for larger systems. In short, applying the proposed method directly to larger systems is likely to face scalability issues. Consider a simple example: the behavior of the proposed method when increasing the number of qubits while keeping the number of steps and the sample size in the dataset constant. First, the size of a single density matrix increases exponentially, meaning the dataset size would also increase exponentially. Furthermore, the size of the neural network model used—specifically, the number of parameters within the model—also increases exponentially relative to the qubit count when scaled using a design like the

DeepSVDD-based network described in this chapter. Overall, computational complexity, training time, and memory usage are expected to scale exponentially with the number of qubits. We discuss approaches to the scaling problem in the conclusion.

Furthermore, obtaining an exact density matrix is generally impossible in more realistic problem settings, so we must account for sampling error. Although increasing the number of measurements seems likely to improve the accuracy of the density matrix estimate, doing so consumes computational resources. This naturally leads to the following question: When computational resources are limited—for instance, when the product of the number of measurements and the number of obtainable density matrices must remain below a given constant—how should the number of measurements (or obtainable density matrices) be chosen? The solution to this basic case is described in the Appendix B.

4.6 Conclusion

Through this chapter, we have demonstrated that learning methods based on positive-unlabeled learning are well suited to the entanglement detection problem, particularly given the constraints of realistically obtainable datasets. Numerous studies have highlighted the potential of machine learning approaches for entanglement detection, establishing them as promising candidates for creating general-purpose entanglement detectors. However, as noted in [45], it is essential to consider the practical challenges of data acquisition when implementing these approaches. From this perspective, we propose treating states with no guarantee of separability or entanglement as unlabeled data and incorporating them into a PU learning framework. PU learning is a well-established field in machine learning that provides insights applicable to other fields. In particular, the nnPU learning method [16] used in our experiments is designed to avoid overfitting when applying deep models, making it an effective approach for the application of these models in classical systems. Moreover, our comparative experiments demonstrated the superiority of the proposed method. Methods based on PU learning effectively utilize information in unlabeled data and tend to maintain high performance in entanglement detection even when data are concentrated near the classification boundary.

As future work, we may explore the following problems:

Scaling problem. Scaling is a fundamental challenge in quantum computation, and the entanglement detection approach presented in this chapter is no exception. One straightforward solution to scaling is to perform most of the learning and inference directly in the quantum domain. In this case, constructing a high-quality dataset becomes a crucial issue to be addressed. Although dataset generation was touched upon in our description of the proposed method, ensuring a comprehensive representation of entangled states through quantum operations remains a difficult task, as spanning the entire space of entangled states is inherently complex. Although research on dataset construction has been conducted [117, 118], preparing datasets for large systems remains a key challenge for future studies. As an alternative approach, exploring memory-efficient classical methods could be beneficial. Feature selection [68], a well-established technique in classical machine learning, may also be applicable to entanglement detection. If an efficient feature selection method

can be identified, it could enable the application of classical frameworks to relatively large systems.

Variations in problem setting. Since there are various types of entanglement, it is important to investigate its detailed properties using variational formulations in entanglement detection. Examples include formulations as regression problems and with generalized classification boundaries. In formulating the task as a regression problem, we can consider predicting values such as a certain type of entropy. Various studies have quantified entropy values using deep learning [119, 120], so applying the PU learning framework to this field is conceivable. Generalizing the classification boundary involves, for example, determining whether a given state is k -separable [60], which includes the fully-separable classification problem. The ability to classify using more diverse boundaries may make the methods described in this chapter applicable to a wider range of situations.

Chapter 5

Discussion

Here we evaluate the research and its outcomes presented in Chapters 3 and 4. Specifically, we discuss how we addressed the dissertation’s objectives, what the results and limitations are, and the applications and challenges for future work.

Chapter 3 focused on developing a quantum machine learning method for the pairwise classification problem. Pairwise classification involves classifying the relationship pattern between two given data points. Examples include link prediction and chemical reaction prediction. Pairwise classification encompasses general classification problems as subproblems. Furthermore, problems such as link prediction are being researched in the field of quantum computation. This chapter contributes to this research by proposing a feature-based learning approach for quantum computation. The proposed approach is an extension of the idea that quantum support vector machines and quantum kernels can easily be applied to pairwise inputs. The proposed method focuses on the computationally intensive part of solving the linear equation system, from training to inference. Theoretically, it demonstrates an improvement in the computational complexity order of that part. Additionally, the proposed method was implemented and tested on a small-scale quantum computer simulator. These experiments confirmed that, with hyperparameter tuning, the proposed method achieves performance comparable to existing methods. The limitations and challenges of this research include:

- (i) Operation in environments without error correction capabilities. Like the existing QSVM approach, the proposed method is designed for execution on a fault-tolerant quantum computer (FTQC). However, such a device has not yet been realized. Consequently, execution on current hardware is expected to be significantly affected by noise. One potential future direction is to develop noise-tolerant methods that incorporate error mitigation techniques.
- (ii) The necessity of state preparation for each inference. Furthermore, both the proposed method and existing QSVMs share the common challenge of requiring state preparation, or training, for each inference. This differs significantly from classical machine learning methods, such as LS-SVM, where pre-trained parameters can be reused for multiple inferences. One solution to this problem is to execute similar circuits in parallel, *consuming* one circuit per inference. While this approach is easy to implement, requiring only multiple copies of the same circuit, it presents the challenge of high

memory usage, meaning it consumes many qubits. Therefore, developing efficient parallelization methods that address memory usage is a promising area for future research.

- (iii) The impact of sampling error depends on the number of measurements taken. Quantum computational models typically require multiple measurements to estimate an output value. Since the variance of this estimate depends on the number of measurements, the number of measurements indirectly affects prediction accuracy. A future challenge will be to investigate the impact of measurement count on prediction accuracy from both theoretical and experimental perspectives.
- (iv) Numerical experiments using large-scale data. Because of the limitations of current quantum computers and quantum computer simulators on classical computers, the experiments in Chapter 3 used small-scale datasets. Finding an immediate solution to this challenge is difficult. Scaling up quantum computer hardware is a critical area of research for advancing this field.

Chapter 4 introduces a learning method that uses PU learning to address the entanglement detection problem, which is one of the challenges in quantum computation. Although the entanglement detection problem is of theoretical interest, it also arises in applications such as quantum communication, making it a significant issue. This research builds on the extensive background of entanglement detection using machine learning due to its applicability to diverse problem settings. In this context, the research focuses on quantum state data and the difficulty of generating diverse entangled states. The proposed method is based on the idea that, although it is difficult to determine whether a state is entangled based solely on the information in the density matrix, the state is a valid quantum state. In other words, it focuses on the fact that the given density matrix takes on valid values. It treats certain quantum states as unlabeled data in machine learning. Combined with the ease of obtaining separable state data, this naturally leads to the formulation of the problem as PU learning, in which a binary classification problem is learned from labeled data for one class and unlabeled data for the other. By utilizing unlabeled data, the proposed method avoids the challenge of preparing large amounts of entangled state data, which is a limitation of conventional approaches. Numerical experiments comparing the proposed method with a one-class classification method suggest that the PU learning-based approach has high detection performance. This research has limitations and challenges as follows:

- (i) Covariate shift. In this study, we assume that the input features of separable and entangled states remain largely unchanged between the training and test data. This assumption implies that there is no distribution shift. However, in the entanglement detection problem, the probability distribution of the unknown input quantum state is not necessarily sampled from the same distribution as the quantum states included in the training data. In this problem setting, where the random variables of the input and output are denoted as X and Y , we assume that $p_{train}(Y|X) = p_{test}(Y|X)$. This is a well-known problem setting called covariate shift [121]. Generally, in covariate shift, $p_{train}(X) \neq p_{test}(X)$, meaning the distribution of input features can change. A key

challenge under covariate shift is that minimizing empirical risk does not necessarily lead to minimizing expected risk. One future research challenge is adapting this method to handle covariate shift while relating it to research on PU learning under covariate shift [122].

- (ii) Further utilization of the problem setting including well-known entangled states and symmetry for qubit permutation. The No Free Lunch Theorem and other findings suggest that leveraging domain knowledge and problem settings is essential to improving performance when solving specific problems within a given field. This perspective applies to our research as well. For instance, we could use methods that employ known entangled states, such as GHZ or W states, as training data. Initially, these states were considered unsuitable because they occupy only a very small portion of the entire entangled state space. Interestingly, a related topic in classical machine learning is the PUBN method [18], which uses biased negative data. Treating certain entangled states as biased entangled data could allow us to incorporate them into the PUBN problem setting. Another advantageous property we consider exploitable is label invariance under qubit permutations. In this study, we determine whether an n -qubit set is fully separable, a property that remains unchanged when the qubits are permuted. The output requires the prediction model to produce the same result when the qubits of the input quantum state are permuted. In other words, calculating the quantum state for all $n!$ permutations of n qubits increases the amount of training data. This technique of augmenting data by applying specific operations to the input that do not affect the output is well known in classical machine learning as data augmentation. It is also used in an entanglement-structure classification problem [123]. Exploiting problem properties like the one mentioned here should lead to useful proposals for methods for detecting entanglement in settings where measurements are costly.

In essence, the findings in this study address the challenges associated with methods that leverage the common properties of the quantum computational model and problem settings, as well as the complexity of labeling within a particular class. By identifying commonalities between quantum computation properties and machine learning methods, we obtained results that contribute to solving these challenges. In addition, we have identified specific challenges that must be addressed in order to expand the effective scope and improve convenience. Given the absence of a universally applicable algorithm for all problem settings, the utilization of commonalities between machine learning and quantum computation as domain knowledge represents a fruitful direction for enhancing methods in interdisciplinary research. This perspective is imperative for identifying the distinctive strengths of quantum machine learning.

Chapter 6

Conclusion

In this dissertation, we address two primary issues in CQ and QC approaches. First, we explore methods that utilize the common properties of the quantum computational model and problem settings. Second, we analyze the intricacies of labeling in a specific class.

For pairwise classification problems, in Chapter 3 we propose a quantum machine learning method combining the Kronecker kernel and quantum support vector machines. Our method improves the computational complexity of the bottleneck part of LS-SVM learning under certain assumptions, and numerical experiments suggest its potential usefulness for future approaches. This approach attempts to bridge quantum computation and machine learning by focusing on the Kronecker product structure, which frequently appears in quantum computation. Furthermore, this method extends the quantum support vector machine to pairwise classification problems and can also be interpreted as a feature-based link prediction algorithm in the quantum computation domain. Furthermore, through experiments and implementation, we identified challenges arising when implementing quantum support vector machine models at the gate level on quantum computers. We also discussed the impact of the number of qubits used in the eigenvalue representation on classification performance in quantum phase estimation, an aspect not addressed in existing research.

In Chapter 4, we propose a method that repurposes the machine learning framework of PU learning for entanglement detection. Our approach addresses the difficulty of labeling entangled states, which has not been explicitly tackled in many existing geometry-based entanglement detection methods. We formulate the entanglement detection problem as a PU learning problem and introduce the concept of unlabeled quantum states as data. This formulation enabled us to utilize information from quantum states whose entanglement status is unknown for learning the classification boundary of entanglement. Indeed, experimental results comparing our PU learning-based approach with a baseline one-class classification method suggest that unlabeled quantum states are useful. Furthermore, we proposed a dataset where data points are densely distributed along the classification boundary for the fully-separability classification problem of quantum states.

One potential avenue for further research in this area involves the exploration of methodologies that capitalize on the inherent convexity problem structure in entanglement detection. In the field of mathematical optimization, problems involving convexity in the domain or objective function are of particular interest and are collectively referred to as convex

optimization. A series of related studies on entanglement detection have proposed methodologies associated with convexity, which may facilitate the development of more efficient classification methods.

The results of this dissertation present an idea for problem-solving through problem transfer between the fields of machine learning and quantum computation, and should contribute to a deeper understanding of how these two fields relate.

Appendix A

Calculation of inner product

Here we prove that the inner product of quantum states can be predicted using our implemented model.

Proof. After applying the X gate to the first register, the state can be written as

$$a_{00}|0\rangle|\phi\rangle|0\rangle + a_{10}|1\rangle|\psi_{train}\rangle|0\rangle + \frac{1}{\sqrt{2}}|1\rangle|\psi_{test}\rangle|1\rangle. \quad (\text{A.1})$$

The coefficient of $\frac{1}{\sqrt{2}}$ for the state $|1\rangle|\psi_{test}\rangle|0\rangle$ is obvious because U_{test} does not affect the state of the first register. Here, from the constraint of amplitudes,

$$a_{00}^2 + a_{10}^2 = \frac{1}{2} \quad (\text{A.2})$$

holds. Applying Hadamard gate to the fourth register, the state is as follows:

$$|\psi\rangle = a_{00}|0\rangle|\phi\rangle H|0\rangle + a_{10}|1\rangle|\psi_{train}\rangle H|0\rangle + \frac{1}{\sqrt{2}}|1\rangle|\psi_{test}\rangle H|1\rangle \quad (\text{A.3})$$

$$= \frac{1}{\sqrt{2}} \left\{ a_{00}|0\rangle|\phi\rangle(|0\rangle + |1\rangle) + a_{10}|1\rangle|\psi_{train}\rangle(|0\rangle + |1\rangle) + \frac{1}{\sqrt{2}}|1\rangle|\psi_{test}\rangle(|0\rangle - |1\rangle) \right\} \quad (\text{A.4})$$

$$= \frac{1}{\sqrt{2}} \left(a_{00}|0\rangle|\phi\rangle + a_{10}|1\rangle|\psi_{train}\rangle + \frac{1}{\sqrt{2}}|1\rangle|\psi_{test}\rangle \right) |0\rangle \quad (\text{A.5})$$

$$+ \frac{1}{\sqrt{2}} \left(a_{00}|0\rangle|\phi\rangle + a_{10}|1\rangle|\psi_{train}\rangle - \frac{1}{\sqrt{2}}|1\rangle|\psi_{test}\rangle \right) |1\rangle. \quad (\text{A.6})$$

For this state, the probability of measuring $i, j (i, j \in \{0, 1\})$ in the first and last registers (p_{ij}) is calculated as follows:

$$p_{00} = p_{01} = \frac{1}{2} a_{00}^2 \quad (\text{A.7})$$

$$p_{10} = \frac{1}{2} \left(a_{10}^\dagger \langle \psi_{train} | + \frac{1}{\sqrt{2}} \langle \psi_{test} | \right) \left(a_{10} |\psi_{train}\rangle + \frac{1}{\sqrt{2}} |\psi_{test}\rangle \right) \quad (\text{A.8})$$

$$= \frac{1}{2} \left(a_{10}^2 + \frac{1}{2} + \sqrt{2} a_{10} \langle \psi_{train} | \psi_{test} \rangle \right) \quad (\text{A.9})$$

$$p_{11} = \frac{1}{2} \left(a_{10}^\dagger \langle \psi_{train} | - \frac{1}{\sqrt{2}} \langle \psi_{test} | \right) \left(a_{10} |\psi_{train}\rangle - \frac{1}{\sqrt{2}} |\psi_{test}\rangle \right) \quad (\text{A.10})$$

$$= \frac{1}{2} \left(a_{10}^2 + \frac{1}{2} - \sqrt{2} a_{10} \langle \psi_{train} | \psi_{test} \rangle \right). \quad (\text{A.11})$$

Under the condition that a_{00} and a_{10} are real numbers and $|\psi_{train}\rangle, |\psi_{test}\rangle$ are real-valued vectors (in the experimental setting, these constraints are guaranteed). Therefore,

$$\langle \psi_{train} | \psi_{test} \rangle = \frac{2p_{10} - (a_{10}^2 + \frac{1}{2})}{\sqrt{2}a_{10}} \quad (\text{A.12})$$

$$= \frac{2p_{10} - (p_{10} + p_{11})}{\sqrt{2(p_{10} + p_{11} - \frac{1}{2})}} \quad (\text{A.13})$$

$$= \frac{p_{10} - p_{11}}{\sqrt{2(p_{10} + p_{11}) - 1}}. \quad (\text{A.14})$$

p_{10} and p_{11} are easily obtained from the calculation, so the inner product can be calculated.

□

Appendix B

Effect of sampling error

Here, we consider the case in which a finite number of measurements is permitted. We assume an n_{qubit} -qubit system and let $d = 2^{n_{\text{qubit}}}$. We also assume that we can perform B measurements and prepare n_{data} data samples, with each data sample estimated by n_{shot} , i.e., $n_{\text{data}}n_{\text{shot}} = B$ and $1 \leq n_{\text{data}}, n_{\text{shot}} \leq B$. (Here, we only consider the case in which each sample is measured the same number of times). We denote the dataset as $\{(X_i + E_i, y_i)\}_{i=1}^{n_{\text{data}}}$, where $E_i \in \mathbb{C}^{d \times d}$ are the errors from the true values of the density matrices. We also adopt Pauli-basis measurements, which are often used to estimate the expectation values of operators and density matrices. Here, we assume that the density matrix is $d \times d$. Let U_ℓ be a positive constant, and let the loss function $\ell : \mathbb{R} \rightarrow [0, U_\ell]$ be expressed as $\ell = \ell(f(X) \cdot y)$, and $\mathcal{F} \subset \mathbb{C}^{d \times d} \times \mathbb{R}$ be a decision function space. We assume that any composite function $\ell \circ f$, where $f \in \mathcal{F}$, is L -Lipschitz⁽¹⁾, i.e., there exists positive constant $L < +\infty$ that satisfies $|(\ell \circ f)(X)| \leq L\|X\|$ for any input (density matrices) X .

We define the risks $\hat{R}(f)$ and $\hat{R}_{\text{noisy}}(f)$ as follows:

$$\begin{aligned}\hat{R}(f) &:= \frac{1}{n_{\text{data}}} \sum_{i=1}^{n_{\text{data}}} \ell(f(X_i), y_i) \\ \hat{R}_{\text{noisy}}(f) &:= \frac{1}{n_{\text{data}}} \sum_{i=1}^{n_{\text{data}}} \ell(f(X_i + E_i), y_i).\end{aligned}$$

For risks defined above, we prove the following proposition:

Proposition B.1 (Upper bound of approximation loss). For the predefined risks $\hat{R}(f), \hat{R}_{\text{noisy}}(f)$, with the probability of at least $1 - \delta$,

$$\sup_{f \in \mathcal{F}} |\hat{R}(f) - \hat{R}_{\text{noisy}}(f)| \leq dL \sqrt{\frac{2d^c \pi}{3n_{\text{shot}}}} + \sqrt{\frac{U_\ell^2}{2n_{\text{data}}} \ln \frac{1}{\delta}}$$

holds, where $c = \log_2 3$. Additionally, if $B \geq \max\left(\frac{4d^{2+c}L^2\pi}{3U_\ell^2 \ln(1/\delta)}, \frac{3U_\ell^2 \ln(1/\delta)}{4d^{2+c}L^2\pi}\right)$, then with probability at least $1 - \delta$,

$$\sup_{f \in \mathcal{F}} |\hat{R}(f) - \hat{R}_{\text{noisy}}(f)| \leq \left(\frac{16d^{2+c}L^2U_\ell^2\pi \ln(1/\delta)}{3} \cdot \frac{1}{B} \right)^{\frac{1}{4}}$$

⁽¹⁾Here, we define the Lipschitz constant in the terms of the spectral norm. Using the equivalence of norms, we can analyze other norms, such as the Frobenius norm, similarly.

holds by choosing $n_{data} = \sqrt{\frac{3U_\ell^2 \ln(1/\delta)}{4d^{2+c}L^2\pi}}B$ and $n_{shot} = \sqrt{\frac{4d^{2+c}L^2\pi}{3U_\ell^2 \ln(1/\delta)}}B$.

Proof. From the definitions,

$$\begin{aligned}
& \sup_{f \in \mathcal{F}} |\hat{R}(f) - \hat{R}_{noisy}(f)| \\
&= \sup_{f \in \mathcal{F}} \left| \frac{1}{n_{data}} \sum_{i=1}^{n_{data}} \ell(f(X_i), y_i) - \frac{1}{n_{data}} \sum_{i=1}^{n_{data}} \ell(f(X_i + E_i), y_i) \right| \\
&= \frac{1}{n_{data}} \sup_{f \in \mathcal{F}} \left| \sum_{i=1}^{n_{data}} [\ell(f(X_i), y_i) - \ell(f(X_i + E_i), y_i)] \right| \\
&\leq \frac{1}{n_{data}} \sum_{i=1}^{n_{data}} \sup_{f \in \mathcal{F}} |\ell(f(X_i), y_i) - \ell(f(X_i + E_i), y_i)| \\
&= \frac{1}{n_{data}} \sum_{i=1}^{n_{data}} \Delta_i. \tag{B.1}
\end{aligned}$$

Here we define Δ_i as $\Delta_i = \sup_{f \in \mathcal{F}} |\ell(f(X_i), y_i) - \ell(f(X_i + E_i), y_i)|$.

From the result (the case with $3^{n_{qubit}}$ Pauli basis) of [124, 125], for $t \geq 0^{(2)}$,

$$\Pr(\|E_i\| \geq t) \leq d \exp\left(-\frac{3n_{shot}t^2}{8d^c}\right).$$

where $c = \log_2 3^{(3)(4)}$. From this inequality and the assumption that $\ell \circ f$ is L -Lipschitz,

$$\begin{aligned}
\Pr(\Delta_i \geq t) &\leq \Pr(\|E_i\| \geq t/L) \\
&\leq d \exp\left(-\frac{3n_{shot}t^2}{8d^cL^2}\right).
\end{aligned}$$

Considering the upper bound of expectation $\mathbb{E}[\Delta_i]$,

$$\begin{aligned}
\mathbb{E}[\Delta_i] &= \int_0^{+\infty} \Pr(\Delta_i \geq t) dt \\
&\leq d \int_0^{+\infty} \exp\left(-\frac{3n_{shot}t^2}{8d^cL^2}\right) dt \\
&= dL \sqrt{\frac{2d^c\pi}{3n_{shot}}}.
\end{aligned}$$

First equality follows by simple calculation and $\Delta_i \geq 0$. In last equality, we used Gaussian integral $\int_0^{+\infty} e^{-at^2} dt = (1/2)\sqrt{(\pi/a)}$. Then,

$$\mathbb{E}\left[\frac{1}{n_{data}} \sum_{i=1}^{n_{data}} \Delta_i\right] = \frac{1}{n_{data}} \sum_{i=1}^{n_{data}} \mathbb{E}[\Delta_i] \leq dL \sqrt{\frac{2d^c\pi}{3n_{shot}}}.$$

⁽²⁾ Although original paper explains that this upper bound holds for $t \in [0, 1]$, using the fact that $\|E_i\|$ satisfies the conditions of density matrix, we can easily prove this bound holds for $t \geq 0$.

⁽³⁾ Here we use $3^{n_{qubit}} = 3^{\log_2 d} = d^{\log_2 3}$.

⁽⁴⁾ It is assumed that the number of samples for $3^{n_{qubit}}$ Pauli basis is the same, which constrains the value of B as multiples of $3^{n_{qubit}}$ in original paper. Actually, we can easily confirm that similar inequality holds for any positive integer n_{shot} .

By $\Pr(0 \leq \Delta_i \leq U_\ell) = 1$, from Hoeffding's inequality, for $t > 0$,

$$\Pr\left(\frac{1}{n_{data}} \sum_{i=1}^{n_{data}} \Delta_i - \mathbb{E}\left[\frac{1}{n_{data}} \sum_{i=1}^{n_{data}} \Delta_i\right] \geq t\right) \leq \exp\left(-\frac{2n_{data}t^2}{U_\ell^2}\right).$$

Here,

$$\begin{aligned} \Pr\left(\frac{1}{n_{data}} \sum_{i=1}^{n_{data}} \Delta_i - \mathbb{E}\left[\frac{1}{n_{data}} \sum_{i=1}^{n_{data}} \Delta_i\right] \geq t\right) &= \Pr\left(\frac{1}{n_{data}} \sum_{i=1}^{n_{data}} \Delta_i \geq \mathbb{E}\left[\frac{1}{n_{data}} \sum_{i=1}^{n_{data}} \Delta_i\right] + t\right) \\ &\geq \Pr\left(\frac{1}{n_{data}} \sum_{i=1}^{n_{data}} \Delta_i \geq dL\sqrt{\frac{2d^c\pi}{3n_{shot}}} + t\right). \end{aligned}$$

By defining ϵ as $\epsilon := dL\sqrt{\frac{2d^c\pi}{3n_{shot}}} + t$, with $\epsilon > dL\sqrt{\frac{2d^c\pi}{3n_{shot}}}$,

$$\begin{aligned} \Pr\left(\frac{1}{n_{data}} \sum_{i=1}^{n_{data}} \Delta_i \geq \epsilon\right) &\leq \exp\left(-\frac{2n_{data}t^2}{U_\ell^2}\right) \\ &= \exp\left(-\frac{2n_{data}}{U_\ell^2} \left(\epsilon - dL\sqrt{\frac{2d^c\pi}{3n_{shot}}}\right)^2\right). \end{aligned}$$

Combining this inequality and B.1, we get

$$\Pr\left(\sup_{f \in \mathcal{F}} |\hat{R}(f) - \hat{R}_{noisy}(f)| \geq \epsilon\right) \leq \exp\left(-\frac{2n_{data}}{U_\ell^2} \left(\epsilon - dL\sqrt{\frac{2d^c\pi}{3n_{shot}}}\right)^2\right).$$

By defining this right hand side as $\delta \in (0, 1)$ and applying some calculations, we get first inequality of main claim.

Here, by AM-GM inequality, we can evaluate the lower bound of right hand side of the first claim as

$$\begin{aligned} &dL\sqrt{\frac{2d^c\pi}{3n_{shot}}} + \sqrt{\frac{U_\ell^2}{2n_{data}} \ln \frac{1}{\delta}} \\ &\geq 2\sqrt{dL\sqrt{\frac{2d^c\pi}{3n_{shot}} \frac{U_\ell^2}{2n_{data}} \ln \frac{1}{\delta}}} \\ &= \left(\frac{16d^{2+c}L^2U_\ell^2\pi \ln(1/\delta)}{3} \cdot \frac{1}{B}\right)^{\frac{1}{4}}. \end{aligned}$$

By substituting the assumption $n_{data}n_{shot} = B$ to this, we get the bound of second claim.

Here we check the condition that the equality in AM-GM inequality holds. That is,

$$dL\sqrt{\frac{2d^c\pi}{3n_{shot}}} = \sqrt{\frac{U_\ell^2}{2n_{data}} \ln \frac{1}{\delta}}.$$

After rearranging for n_{data} and n_{shot} , we obtain

$$\frac{n_{data}}{n_{shot}} = \frac{3U_\ell^2 \ln(1/\delta)}{4d^{2+c}L^2\pi}.$$

From the constraints $1 \leq n_{data}, n_{shot} \leq B$, it immediately follows $1/B \leq (n_{data}/n_{shot}) \leq B$. Here we note that the values (n_{data}, n_{shot}) and the value n_{data}/n_{shot} correspond one-to-one under the assumption of $n_{data}n_{shot} = B$. From these, we get the condition of B by rearranging the conditions $1/B \leq (n_{data}/n_{shot}) \leq B$. $\square$

References

- [1] Subhash Kak. On quantum neural computing. *Information Sciences*, 83(3):143–160, 1995. ISSN 0020-0255. doi: [https://doi.org/10.1016/0020-0255\(94\)00095-S](https://doi.org/10.1016/0020-0255(94)00095-S). URL <https://www.sciencedirect.com/science/article/pii/002002559400095S>.
- [2] P.W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994.
- [3] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, pages 212–219, Philadelphia, Pennsylvania, USA, 1996. doi: 10.1145/237814.237866.
- [4] Yue-Chi Ma and Man-Hong Yung. Transforming bell’ s inequalities into state classifiers with machine learning. *npj Quantum Information*, 4(1):34, 2018. ISSN 2056-6387. doi: 10.1038/s41534-018-0081-3.
- [5] Sirui Lu, Shilin Huang, Keren Li, Jun Li, Jianxin Chen, Dawei Lu, Zhengfeng Ji, Yi Shen, Duanlu Zhou, and Bei Zeng. Separability-entanglement classifier via machine learning. *Phys. Rev. A*, 98:012315, Jul 2018. doi: 10.1103/PhysRevA.98.012315.
- [6] Scott Aaronson. The learnability of quantum states. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 463(2088):3089–3114, 09 2007. ISSN 1364-5021. doi: 10.1098/rspa.2007.0113. URL <https://doi.org/10.1098/rspa.2007.0113>.
- [7] David M.J. Tax and Robert P.W. Duin. Support vector data description. *Machine Learning*, 54:45–66, 2004. doi: 10.1023/B:MACH.0000008084.60811.49.
- [8] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4393–4402. PMLR, 10–15 Jul 2018.
- [9] Naeem Seliya, Azadeh Abdollah Zadeh, and Taghi M. Khoshgoftaar. A literature review on one-class classification and its potential applications in big data. *Journal of Big Data*, 8, 2021. doi: 10.1186/s40537-021-00514-x.

- [10] Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, page 213 - 220, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605581934. doi: 10.1145/1401890.1401920.
- [11] Yunrui Zhao, Qianqian Xu, Yangbangyan Jiang, Peisong Wen, and Qingming Huang. Dist-pu: Positive-unlabeled learning from a label distribution perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14461–14470, June 2022.
- [12] Jessa Bekker and Jesse Davis. Learning from positive and unlabeled data: a survey. *Machine Learning*, 109:719–760, 2020. doi: 10.1007/s10994-020-05877-5.
- [13] Hyunjun Ju, Dongha Lee, Junyoung Hwang, Junghyun Namkung, and Hwanjo Yu. Pumad: Pu metric learning for anomaly detection. *Information Sciences*, 523:167–183, 2020. ISSN 0020-0255. doi: <https://doi.org/10.1016/j.ins.2020.03.021>.
- [14] Hiroshi Takahashi, Tomoharu Iwata, Atsutoshi Kumagai, and Yuuki Yamanaka. Deep positive-unlabeled anomaly detection for contaminated unlabeled data, 2024. Preprint at <https://arxiv.org/abs/2405.18929>.
- [15] Marthinus C. du Plessis, Gang Niu, and Masashi Sugiyama. Analysis of learning from positive and unlabeled data. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.
- [16] Ryuichi Kiryo, Gang Niu, Marthinus C du Plessis, and Masashi Sugiyama. Positive-unlabeled learning with non-negative risk estimator. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
- [17] Lukas Ruff, Robert A. Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. Deep semi-supervised anomaly detection. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.
- [18] Yu-Guan Hsieh, Gang Niu, and Masashi Sugiyama. Classification from positive, unlabeled and biased negative data. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2820–2829. PMLR, 09–15 Jun 2019.
- [19] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.
- [20] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103:150502, 2009. doi: 10.1103/PhysRevLett.103.150502.

- [21] Esma Aïmeur, Gilles Brassard, and Sébastien Gambs. Machine learning in a quantum world. In *Advances in Artificial Intelligence*, pages 431–442, Berlin, Heidelberg, 2006.
- [22] Abdul-Saboor Sheikh, Romain Guigourès, Evgenii Koriagin, et al. A deep learning system for predicting size and fit in fashion e-commerce. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 110–118, Copenhagen, Denmark, 2019. Association for Computing Machinery. doi: 10.1145/3298689.3347006.
- [23] Sina Stocker, Gábor Csányi, Karsten Reuter, et al. Machine learning in chemical reaction space. *Nature Communications*, 11:5505, 2020. doi: 10.1038/s41467-020-19267-x.
- [24] P. Domingos. *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*. Basic Books, Inc., New York, 2015.
- [25] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995. doi: 10.1007/BF00994018.
- [26] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, Singapore, 2006.
- [27] Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.
- [28] Frank Arute, Kunal Arya, Ryan Babbush, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574:505–510, 2019. doi: 10.1038/s41586-019-1666-5.
- [29] Han-Sen Zhong, Hui Wang, Yu-Hao Deng, et al. Quantum computational advantage using photons. *Science*, 370:1460–1463, 2020. doi: 10.1126/science.abe8770.
- [30] Yong (Alexander) Liu, Xin (Lucy) Liu, Fang (Nancy) Li, et al. Closing the “quantum supremacy” gap: Achieving real-time simulation of a random quantum circuit using a new sunway supercomputer. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, New York, NY, USA, 2021. Association for Computing Machinery.
- [31] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Physical Review Letters*, 113:130503, 2014. doi: 10.1103/physrevlett.113.130503.
- [32] Zhaokai Li, Xiaomei Liu, Nanyang Xu, and Jiangfeng Du. Experimental realization of a quantum support vector machine. *Physical Review Letters*, 114:140504, 2015. doi: 10.1103/PhysRevLett.114.140504.
- [33] Jiaying Yang, Ahsan Javed Awan, and Gemma Vall-Llosera. Support vector machines on noisy intermediate scale quantum computers, 2019. Preprint at <http://arxiv.org/abs/1909.11988>.

- [34] Satoshi Oyama and Manning C. D. Using feature conjunctions across examples for learning pairwise classifiers, 2004. Paper presented at the 15th European Conference on Machine Learning(ECML), 322–333 September 2004.
- [35] Sho Yokoi, Hiroshi Kajino, and Hisashi Kashima. Link prediction in sparse networks by incidence matrix factorization. *Journal of Information Processing*, 25:477–485, 2017. doi: 10.2197/ipsjjip.25.477.
- [36] Ben-Hur A and Noble WS. Kernel methods for predicting protein-protein interactions. *JBioinformatics*, 21:38–46, 2005. doi: 10.1093/bioinformatics/bti1016.
- [37] Michał Horodecki, Paweł Horodecki, and Ryszard Horodecki. General teleportation channel, singlet fraction, and quasidistillation. *Phys. Rev. A*, 60:1888–1898, Sep 1999. doi: 10.1103/PhysRevA.60.1888.
- [38] Charles H. Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. *Theoretical Computer Science*, 560:7–11, 2014. ISSN 0304-3975. doi: 10.1016/j.tcs.2014.05.025.
- [39] Otfried Gühne and Géza Tóth. Entanglement detection. *Physics Reports*, 474(1): 1–75, 2009. ISSN 0370-1573. doi: 10.1016/j.physrep.2009.02.004.
- [40] Leonid Gurvits. Classical deterministic complexity of edmonds’ problem and quantum entanglement. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, STOC ’03, page 10 – 19, New York, NY, USA, 2003. Association for Computing Machinery. ISBN 1581136749. doi: 10.1145/780542.780545. The long version is available at <https://arxiv.org/abs/quant-ph/0303055>.
- [41] Asher Peres. Separability criterion for density matrices. *Phys. Rev. Lett.*, 77:1413–1415, Aug 1996. doi: 10.1103/PhysRevLett.77.1413.
- [42] Michał Horodecki, Paweł Horodecki, and Ryszard Horodecki. Separability of mixed states: necessary and sufficient conditions. *Physics Letters A*, 223(1):1–8, 1996. ISSN 0375-9601. doi: 10.1016/S0375-9601(96)00706-2.
- [43] Claudio Sanavio, Edoardo Tignone, and Elisa Ercolessi. Entanglement classification via witness operators generated by support vector machine. *The European Physical Journal Plus*, 138:936, Oct 2023. doi: 10.1140/epjp/s13360-023-04546-5.
- [44] Alexander C.B. Greenwood, Larry T.H. Wu, Eric Y. Zhu, Brian T. Kirby, and Li Qian. Machine-learning-derived entanglement witnesses. *Phys. Rev. Appl.*, 19:034058, Mar 2023. doi: 10.1103/PhysRevApplied.19.034058.
- [45] Yiwei Chen, Yu Pan, Guofeng Zhang, and Shuming Cheng. Detecting quantum entanglement with unsupervised learning. *Quantum Science and Technology*, 7(1): 015005, nov 2021. doi: 10.1088/2058-9565/ac310f.

- [46] Antti Airola and Tapio Pahikkala. Fast kronecker product kernel methods via generalized vec trick. *IEEE Transactions on Neural Networks and Learning Systems*, 29(8):3374–3387, 2018. doi: 10.1109/TNNLS.2017.2727545.
- [47] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press, Cambridge, MA, 2002.
- [48] J. A. K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9:293–300, 1999. doi: 10.1023/A:1018628609742.
- [49] Alastair Kay. Quantikz, 2019. Tutorial is in <https://arxiv.org/abs/1809.03842>.
- [50] Zhikuan Zhao, Jack K. Fitzsimons, and Joseph F. Fitzsimons. Quantum-assisted gaussian process regression. *Phys. Rev. A*, 99:052331, 2019. doi: 10.1103/PhysRevA.99.052331.
- [51] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of research of the National Bureau of Standards*, 49:409–436, 1952.
- [52] Danial Dervovic, Mark Herbster, Peter Mountney, et al. Quantum linear systems algorithms: a primer, 2018. Preprint at <https://arxiv.org/abs/1802.08227>.
- [53] Andrew M. Childs, Robin Kothari, and Rolando D. Somma. Quantum algorithm for systems of linear equations with exponentially improved dependence on precision. *SIAM Journal on Computing*, 46:1920–1950, 2017. doi: 10.1137/16M1087072.
- [54] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Architectures for a quantum random access memory. *Phys. Rev. A*, 78:052310, 2008. doi: 10.1103/PhysRevA.78.052310.
- [55] S. Lloyd, M. Mohseni, and P. Rebentrost. Quantum principal component analysis. *Nature physics*, 10:631–633, 2014. doi: 10.1038/nphys3029.
- [56] Hisashi Kashima, Satoshi Oyama, Yoshihiro Yamanishi, et al. Cartesian kernel: An efficient alternative to the pairwise kernel. *IEICE Transactions on Information and Systems*, E93.D:2672–2679, 2010. doi: 10.1587/transinf.E93.D.2672.
- [57] R. T. Thew, K. Nemoto, A. G. White, and W. J. Munro. Qudit quantum-state tomography. *Phys. Rev. A*, 66:012303, Jul 2002. doi: 10.1103/PhysRevA.66.012303.
- [58] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16:1050–1057, 2020. doi: 10.1038/s41567-020-0932-7.
- [59] W. Dür, G. Vidal, and J. I. Cirac. Three qubits can be entangled in two inequivalent ways. *Phys. Rev. A*, 62:062314, Nov 2000. doi: 10.1103/PhysRevA.62.062314.
- [60] Andreas Gabriel, Beatrix C. Hiesmayr, and Marcus Huber. Criterion for k-separability in mixed multipartite systems, 2010. Preprint at <https://arxiv.org/abs/1002.2953>.

- [61] You Zhou, Qi Zhao, Xiao Yuan, and Xiongfeng Ma. Detecting multipartite entanglement structure with minimal resources. *npj Quantum Information*, 5(1):83, Oct 2019. ISSN 2056-6387. doi: 10.1038/s41534-019-0200-9.
- [62] G. Vidal and R. F. Werner. Computable measure of entanglement. *Phys. Rev. A*, 65: 032314, Feb 2002. doi: 10.1103/PhysRevA.65.032314.
- [63] Chiheb Trabelsi, Olexa Bilaniuk, Ying Zhang, Dmitriy Serdyuk, Sandeep Subramanian, João Felipe Santos, Soroush Mehri, Negar Rostamzadeh, Yoshua Bengio, and Christopher J Pal. Deep complex networks. In *6th International Conference on Learning Representations, ICLR 2018*, 2018.
- [64] Karol Zyczkowski and Hans-Jürgen Sommers. Induced measures in the space of mixed quantum states. *Journal of Physics A: Mathematical and General*, 34(35):7111 - 7125, August 2001. ISSN 1361-6447. doi: 10.1088/0305-4470/34/35/335.
- [65] Paul B Slater. Dyson indices and hilbert - schmidt separability functions and probabilities. *Journal of Physics A: Mathematical and Theoretical*, 40(47):14279 - 14308, November 2007. ISSN 1751-8121. doi: 10.1088/1751-8113/40/47/017.
- [66] Jan Roik, Karol Bartkiewicz, Antonín Černoč, and Karel Lemr. Accuracy of entanglement detection via artificial neural networks and human-designed entanglement witnesses. *Phys. Rev. Appl.*, 15:054006, May 2021. doi: 10.1103/PhysRevApplied.15.054006.
- [67] Julio Ureña, Antonio Sojo, Juani Bermejo-Vega, and Daniel Manzano. Entanglement detection with classical deep neural networks. *Scientific Reports*, 14:18109, 2024. doi: <https://doi.org/10.1038/s41598-024-68213-0>.
- [68] Jorawar Singh, Vaishali Gulati, Kavita Dorai, and Arvind. Entanglement classification of arbitrary three-qubit states via artificial neural networks. *Phys. Rev. Appl.*, 24:034069, Sep 2025. doi: 10.1103/xfks-snj1.
- [69] Naema Asif, Uman Khalid, Awais Khan, Trung Q. Duong, and Hyundong Shin. Entanglement detection with artificial neural networks. *Scientific Reports*, 13:1562, 2023. doi: 10.1038/s41598-023-28745-3.
- [70] Xiaoting Gao, Mathieu Isoard, Fengxiao Sun, Carlos E. Lopetegui, Yu Xiang, Valentina Parigi, Qiongyi He, and Mattia Walschaers. Correlation-pattern-based continuous variable entanglement detection through neural networks. *Phys. Rev. Lett.*, 132:220202, May 2024. doi: 10.1103/PhysRevLett.132.220202.
- [71] Rui Li, Shikun Zhang, Zheng Qin, Chunxiao Du, Yang Zhou, and Zhisong Xiao. Low-cost detection of high-dimensional multipartite entanglement structures. *Phys. Rev. Appl.*, 23:044033, Apr 2025. doi: 10.1103/PhysRevApplied.23.044033.
- [72] Yue-Di Qu, Rui-Qi Zhang, Shu-Qian Shen, Juan Yu, and Ming Li. Entanglement detection with complex-valued neural networks. *International Journal of Theoretical Physics*, 62(9):206, Sep 2023. ISSN 1572-9575. doi: 10.1007/s10773-023-05460-3.

- [73] Yulei Huang, Liangyu Che, Chao Wei, Feng Xu, Xinfang Nie, Jun Li, Dawei Lu, and Tao Xin. Direct entanglement detection of quantum systems using machine learning. *npj Quantum Information*, 11(1):29, Feb 2025. ISSN 2056-6387. doi: 10.1038/s41534-025-00970-w.
- [74] Yi-Jun Luo, Xuan Leng, and Chengjie Zhang. Genuine multipartite entanglement verification with convolutional neural networks. *Phys. Rev. A*, 110:042412, Oct 2024. doi: 10.1103/PhysRevA.110.042412.
- [75] Rui Li, Junling Du, Zheng Qin, Shikun Zhang, Chunxiao Du, Yang Zhou, and Zhisong Xiao. Entanglement structure detection via computer vision. *Phys. Rev. A*, 110:012448, Jul 2024. doi: 10.1103/PhysRevA.110.012448.
- [76] Jarosław Pawłowski and Mateusz Krawczyk. Identification of quantum entanglement with siamese convolutional neural networks and semisupervised learning. *Phys. Rev. Appl.*, 22:014068, Jul 2024. doi: 10.1103/PhysRevApplied.22.014068.
- [77] S. V. Vintskevich, N. Bao, A. Nomerotski, P. Stankus, and D. A. Grigoriev. Classification of four-qubit entangled states via machine learning. *Phys. Rev. A*, 107:032421, Mar 2023. doi: 10.1103/PhysRevA.107.032421.
- [78] Aiden R Rosebush, Alexander C B Greenwood, Brian T Kirby, and Li Qian. An exponential reduction in training data sizes for machine learning derived entanglement witnesses. *Machine Learning: Science and Technology*, 5(3):035068, sep 2024. doi: 10.1088/2632-2153/ad7457.
- [79] Caio B. D. Goes, Askery Canabarro, Eduardo I. Duzzioni, and Thiago O. Maciel. Automated machine learning can classify bound entangled states with tomograms. *Quantum Information Processing*, 20(3), March 2021. ISSN 1573-1332. doi: 10.1007/s11128-021-03037-9.
- [80] Xu-Fei Yin, Yuxuan Du, Yue-Yang Fei, Rui Zhang, Li-Zheng Liu, Yingqiu Mao, Tongliang Liu, Min-Hsiu Hsieh, Li Li, Nai-Le Liu, Dacheng Tao, Yu-Ao Chen, and Jian-Wei Pan. Efficient bipartite entanglement detection scheme with a quantum adversarial solver. *Phys. Rev. Lett.*, 128:110501, Mar 2022. doi: 10.1103/PhysRevLett.128.110501.
- [81] Francesco Scala, Stefano Mangini, Chiara Macchiavello, Daniele Bajoni, and Dario Gerace. Quantum variational learning for entanglement witnessing. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2022. doi: 10.1109/IJCNN55064.2022.9892080.
- [82] Diksha Sharma, Vivek Balasaheb Sabale, Parvinder Singh, and Atul Kumar. Harnessing quantum support vector machines for cross-domain classification of quantum states. *Quantum Machine Intelligence*, 7(1), April 2025. ISSN 2524-4914. doi: 10.1007/s42484-025-00274-4.

- [83] Cillian Harney, Stefano Pirandola, Alessandro Ferraro, and Mauro Paternostro. Entanglement classification via neural network quantum states. *New Journal of Physics*, 22(4):045001, apr 2020. doi: 10.1088/1367-2630/ab783d.
- [84] Cillian Harney, Mauro Paternostro, and Stefano Pirandola. Mixed state entanglement classification using artificial neural networks. *New Journal of Physics*, 23(6):063033, jun 2021. doi: 10.1088/1367-2630/ac0388.
- [85] Mateusz Krawczyk, Jarosław Pawłowski, Maciej M. Maśka, and Katarzyna Roszak. Data-driven criteria for quantum correlations. *Phys. Rev. A*, 109:022405, Feb 2024. doi: 10.1103/PhysRevA.109.022405.
- [86] Lifeng Zhang, Zhihua Chen, and Shao-Ming Fei. Entanglement verification with deep semisupervised machine learning. *Phys. Rev. A*, 108:022427, Aug 2023. doi: 10.1103/PhysRevA.108.022427.
- [87] Yi-Jun Luo, Jin-Ming Liu, and Chengjie Zhang. Detecting genuine multipartite entanglement via machine learning. *Phys. Rev. A*, 108:052424, Nov 2023. doi: 10.1103/PhysRevA.108.052424.
- [88] Zhenyu Chen, Xiaodie Lin, and Zhaohui Wei. Certifying unknown genuine multipartite entanglement by neural networks, 2022. Preprint at <https://arxiv.org/abs/2210.13837>.
- [89] Pingxun Wang. A convex hull-based machine learning algorithm for multipartite entanglement classification. *Applied Sciences*, 12(24), 2022. ISSN 2076-3417. doi: 10.3390/app122412778.
- [90] Balthazar Casalé, Giuseppe Di Molfetta, Sandrine Anthoine, and Hachem Kadri. Large-scale quantum separability through a reproducible machine learning lens, 2023. Preprint at <https://arxiv.org/abs/2306.09444>.
- [91] Changbo Chen, Changliang Ren, Hongqing Lin, and He Lu. Entanglement structure detection via machine learning, 2020. Preprint at <https://arxiv.org/abs/2012.00526>.
- [92] Vaishali Gulati, Shivanshu Siyanwal, Arvind, and Kavita Dorai. Ann-enhanced detection of multipartite entanglement in a three-qubit nmr quantum processor, 2024. Preprint at <https://arxiv.org/abs/2409.19739>.
- [93] Yu Tian, Liangyu Che, Xinyue Long, Changliang Ren, and Dawei Lu. Machine learning experimental multipartite entanglement structure. *Advanced Quantum Technologies*, 5(10):2200025, 2022. doi: <https://doi.org/10.1002/qute.202200025>.
- [94] Vojtěch Trávníček, Jan Roik, Karol Bartkiewicz, Antonín Černoš, Paweł Horodecki, and Karel Lemr. Sensitivity versus selectivity in entanglement detection via collective witnesses. *Phys. Rev. Res.*, 6:033056, Jul 2024. doi: 10.1103/PhysRevResearch.6.033056.

- [95] Syed Muhammad Abuzar Rizvi, Naema Asif, Muhammad Shohibul Ulum, Trung Q. Duong, and Hyundong Shin. Multiclass classification of metrologically resourceful tripartite quantum states with deep neural networks. *Sensors*, 22(18), 2022. ISSN 1424-8220. doi: 10.3390/s22186767.
- [96] Uman Khalid, Trung Q. Duong, and Hyundong Shin. Artificial neural networks for quantum sensing: Metrologically resourceful state detection. In *2024 International Conference on Quantum Communications, Networking, and Computing (QCNC)*, pages 259–264, 2024. doi: 10.1109/QCNC62729.2024.00048.
- [97] Jun Yong Khoo and Markus Heyl. Quantum entanglement recognition. *Phys. Rev. Res.*, 3:033135, Aug 2021. doi: 10.1103/PhysRevResearch.3.033135.
- [98] Ali Kookani, Yousef Mafi, Payman Kazemikhah, Hossein Aghababa, Kazim Fouladi, and Masoud Barati. Xpookynet: Advancement in quantum system analysis through convolutional neural networks for detection of entanglement, 2024. Preprint at <https://arxiv.org/abs/2309.03890>.
- [99] Jan Soubusta, Antonín Černoč, and Karel Lemr. Deep-learned classification of quantum correlations from collective measurements. *Physics Letters A*, 559:130911, 2025. ISSN 0375-9601. doi: 10.1016/j.physleta.2025.130911.
- [100] Jue Xu and Qi Zhao. Towards efficient and generic entanglement detection by machine learning, 2022. Preprint at <https://arxiv.org/abs/2211.05592>.
- [101] Ana Martínez-Sabiote, Michalis Skotiniotis, Jara J. Bermejo-Vega, Daniel Manzano, and Carlos Cano. Entanglement detection with quantum-inspired kernels and svms, 2025. Preprint at <https://arxiv.org/abs/2508.17909>.
- [102] Mohammad Rezaei Shokou and Hossein Davoodi Yeganeh. Hybrid quantum-classical learning of nonlinear entanglement witnesses via continuous-variable quantum neural networks, 2025. Preprint at <https://arxiv.org/abs/2509.05924>.
- [103] Przemysław Sekuła, Michał Romaszewski, Przemysław Głomb, Michał Cholewa, and Łukasz Paweł. Quantum-aware transformer model for state classification, 2025. Preprint at <https://arxiv.org/abs/2502.21055>.
- [104] Nahum Sá and Itzhak Roditi. β -variational autoencoder as an entanglement classifier. *Physics Letters A*, 417:127697, 2021. ISSN 0375-9601. doi: <https://doi.org/10.1016/j.physleta.2021.127697>.
- [105] Mohammad Yosefpor, Mohammad Reza Mostaan, and Sadegh Raeisi. Finding semi-optimal measurements for entanglement detection using autoencoder neural networks. *Quantum Science and Technology*, 5(4):045006, July 2020. ISSN 2058-9565. doi: 10.1088/2058-9565/aba34c.

- [106] Andreas J C Fuchs, Eric Brunner, Jiheon Seong, Hyeokjea Kwon, Seungchan Seo, Joonwoo Bae, Andreas Buchleitner, and Edoardo G Carnio. Machine-learning certification of multipartite entanglement for noisy quantum hardware. *New Journal of Physics*, 27(7):074501, jul 2025. doi: 10.1088/1367-2630/adde80.
- [107] Danilo Zia, Luca Innocenti, Giorgio Minati, Salvatore Lorenzo, Alessia Suprano, Rosario Di Bartolo, Nicolò Spagnolo, Taira Giordani, Valeria Cimini, G. Massimo Palma, Alessandro Ferraro, Fabio Sciarrino, and Mauro Paternostro. Quantum extreme learning machines for photonic entanglement witnessing, 2025. Preprint at <https://arxiv.org/abs/2502.18361>.
- [108] Shah Nawaz Ahmed, Carlos Sánchez Muñoz, Franco Nori, and Anton Frisk Kockum. Classification and reconstruction of optical quantum states with deep neural networks. *Phys. Rev. Res.*, 3:033278, Sep 2021. doi: 10.1103/PhysRevResearch.3.033278.
- [109] Qian Sun, Yanyan Song, Zhichuan Liao, and Nan Jiang. High-accuracy entanglement detection via a convolutional neural network with noise resistance. *Applied Sciences*, 14(20), 2024. ISSN 2076-3417. doi: 10.3390/app14209418.
- [110] Uman Khalid, Junaid Ur Rehman, Haejoon Jung, Trung Q. Duong, Octavia A. Dobre, and Hyundong Shin. Quantum property learning for nisc networks: Universal quantum witness machines. *IEEE Transactions on Communications*, 73(4):2207–2221, 2025. doi: 10.1109/TCOMM.2024.3469555.
- [111] Eric Brunner, Aaron Xie, Gabriel Dufour, and Andreas Buchleitner. Data-driven approach to mixed-state multipartite entanglement characterisation, 2024. Preprint at <https://arxiv.org/abs/2407.18014>.
- [112] Bharati. K, Vikesh Siddhu, and Krishna Jagannathan. Classical bandit algorithms for entanglement detection in parameterized qubit states, 2024. Preprint at <https://arxiv.org/abs/2406.19738>.
- [113] Diksha Sharma, Vivek Balasaheb Sabale, Thirumalai M., and Atul Kumar. Quantum neural networks facilitating quantum state classification, 2025. Preprint at <https://arxiv.org/abs/2504.06622>.
- [114] Mahmoud Mahdian and Zahra Mousavi. Entanglement detection with quantum support vector machine (qsvm) on near-term quantum devices. *Scientific Reports*, 15(1): 11931, Apr 2025. ISSN 2045-2322. doi: 10.1038/s41598-025-95897-9.
- [115] Peng-Hui Qiu, Xiao-Guang Chen, and Yi-Wei Shi. Detecting entanglement with deep quantum neural networks. *IEEE Access*, 7:94310–94320, 2019. doi: 10.1109/ACCESS.2019.2929084.
- [116] Tin Kam Ho and M. Basu. Complexity measures of supervised classification problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):289–300, 2002. doi: 10.1109/34.990132.

- [117] Louis Schatzki, Andrew Arrasmith, Patrick J. Coles, and M. Cerezo. Entangled datasets for quantum machine learning, 2021. Preprint at <https://arxiv.org/abs/2109.03400>.
- [118] Elija Perrier, Akram Youssry, and Chris Ferrie. Qdataset, quantum datasets for machine learning. *Scientific Data*, 9(1):582, Sep 2022. ISSN 2052-4463. doi: 10.1038/s41597-022-01639-1.
- [119] Dominik Koutný, Laia Ginés, Magdalena Moczala-Dusanowska, Sven Höfling, Christian Schneider, Ana Predojević, and Miroslav Ježek. Deep learning of quantum entanglement from incomplete measurements. *Science Advances*, 9(29):eadd7131, 2023. doi: 10.1126/sciadv.add7131.
- [120] Johnnie Gray, Leonardo Banchi, Abolfazl Bayat, and Sougato Bose. Machine-learning-assisted many-body entanglement measurement. *Phys. Rev. Lett.*, 121:150503, Oct 2018. doi: 10.1103/PhysRevLett.121.150503.
- [121] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000. ISSN 0378-3758. doi: [https://doi.org/10.1016/S0378-3758\(00\)00115-4](https://doi.org/10.1016/S0378-3758(00)00115-4). URL <https://www.sciencedirect.com/science/article/pii/S0378375800001154>.
- [122] Tomoya Sakai and Nobuyuki Shimizu. Covariate shift adaptation on learning from positive and unlabeled data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):4838–4845, Jul. 2019. doi: 10.1609/aaai.v33i01.33014838. URL <https://ojs.aaai.org/index.php/AAAI/article/view/4411>.
- [123] Xiaoting Gao, Mingsheng Tian, Feng-Xiao Sun, Ya-Dong Wu, Yu Xiang, and Qiongyi He. Classifying multipartite continuous variable entanglement structures through data-augmented neural networks, 2024. URL <https://arxiv.org/abs/2409.07909>.
- [124] Joel A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, Aug 2012. ISSN 1615-3383. doi: 10.1007/s10208-011-9099-z. URL <https://doi.org/10.1007/s10208-011-9099-z>.
- [125] M Guță, J Kahn, R Kueng, and J A Tropp. Fast state tomography with optimal error bounds. *Journal of Physics A: Mathematical and Theoretical*, 53(20):204001, apr 2020. doi: 10.1088/1751-8121/ab8111. URL <https://dx.doi.org/10.1088/1751-8121/ab8111>.