Classifying High Entropy Alloys with Quantum Machine Learning

by

Payden Lance Brown

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved October 2022 by the
Graduate Supervisory Committee:

Houlong Zhuang, Chair
Kumar Ankit
Yang Jiao

ARIZONA STATE UNIVERSITY

December 2022

ABSTRACT

With the abundance of increasingly large datasets, the ability to predict the phase of high-entropy alloys (HEAs) based solely on elemental composition could become a reliable tool for the discovery of new HEAs. However, as the amount of data expands so does the computational time and resources required to train predictive classical machine learning models. Quantum computers, which use quantum bits (qubits), could be the solution to overcoming these demands. Their ability to use quantum superposition and interference to perform calculations could be the key to handling large amounts of data. In this work, a hybrid quantum-classical machine learning algorithm is implemented on both quantum simulators and quantum processors to perform the supervised machine learning task. Their feasibility as a future tool for HEA discovery is evaluated based on the algorithm's performance. An artificial neural network (ANN), run by classical computers, is also trained on the same data for performance comparison. The accuracy of the quantum-classical model was found to be comparable to the accuracy achieved by the classical ANN with a slight decrease in accuracy when ran on quantum hardware due to qubit susceptibility to decoherence. Future developments in the applied quantum machine learning method are discussed.

ACKNOWLEDGMENTS

I would like to thank my committee chair and mentor, Houlong Zhuang, for his collaboration and support. This work utilizes resources from IBM Quantum services as well as the Agave cluster.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

The content of this work is taken from a previous publishable work completed by my thesis committee chair and myself; both works evaluate quantum machine learning as a tool for predicting the phase of high-entropy alloys (HEAs). The paper, titled *Quantum machine-learning phase prediction of high-entropy alloys*, makes up the greater part of this work and is found in Chapter 2. The subsections of Chapter 2 and their topics are introduced in the following paragraphs.

High-entropy alloys (HEAs) are defined as having four or more constituent elements all making up a nearly equal portion of the alloy. The subsection, *Introduction*, further defines HEAs and what makes them unique. It explains why these alloys have been the subject of much research and development: due to their superior mechanical and thermal properties. With the promise of improved mechanical strength even at high temperatures, there are innumerable applications for HEAs (Senkov et al., "Development"; Cheng et al.). The introduction goes on to argue that should techniques used in exploring new HEAs be limited to laboratory methods, largely trial and error based, discovery of HEAs suitable for these applications may be impossible given that as the number of constituent elements increases, the number of possible compounds increases exponentially. It also introduces computational methods such as machine learning as a tool to accelerate the design and discovery of new HEAs (Rickman et al.; Huang et al.; Chen et al.; Zhang et al.). The introduction then discusses the challenges facing classical machine learning models caused by big data and offers quantum machine learning techniques as a solution.

Machine learning is a well-known tool for predicting an outcome based on a set of input values. There are various types of machine learning, all of which require sufficient data. The term sufficient data refers to both the quantity and quality of data. The *Methods* subsection of Chapter 2 describes the dataset used and its characteristics. It describes the machine learning task of this paper as a common method known as supervised learning which uses a labeled dataset, or data that consists of input values paired with a known output value. While this black-box approach is unable to give insight into the physical phenomena that determine the outcome, it can be an aid in guiding researchers. The *Methods* section then goes on to describe the artificial neural network (ANN) and hybrid quantum-classical models that are trained and evaluated. The hybrid model is to be implemented on both quantum simulators and quantum hardware.

Results and Discussions contains all the graphical results and reports of model performance. The performance metrics focuses on accuracy of the models, but other metrics such as recall, or sensitivity, are considered. The progress of both accuracy and the loss function (upon which optimization is based) as training occurs is analyzed to show that the hybrid quantum-classical model learns and performs on the same level as the ANN. Alternative quantum machine learning techniques are discussed that have the potential to be more efficient than current ANN.

The *Conclusion* subsection offers insight into how, despite current limitations in quantum processors, quantum machine learning can perform comparably to its classical counterpart. It concludes that, while there is no advantage to the hybrid model over the ANN, quantum machine learning is a viable tool for the design and discovery of new HEAs. Chapter 3 offers a similar discussion with additional discussion of future works.

CHAPTER 2

INVESTIGATION

Quantum machine-learning phase prediction of high-entropy alloys

Payden Brown and Houlong Zhuang[†]
School for Engineering of Matter, Transport and Energy, Arizona State University,
Tempe, AZ 85287, USA
[†]zhuanghl@asu.edu

Abstract

Discovering new high-entropy alloys (HEAs) in the vast compositional space requires a growing power of classical computers for training machine learning models. The exponential increase of HEA data will pose a challenge in making the machine learning process prohibitively time consuming in the foreseeable future. Quantum computers, which use quantum superposition and interference to perform computations, hold great potential in handling big data and accelerating the optimization algorithms ubiquitous in machine learning models. Here we adopt a quantum computer simulator and quantum processors to prepare for the future challenge in new HEA discovery. We first train a classical artificial neural network (ANN), which uses HEA composition as inputs and the corresponding phases as outputs, to predict phase selection. We then apply a quantum computer simulator that implements a hybrid quantum-classical machine learning algorithm for accomplishing the same supervised machine learning task. We find that the resulting testing accuracy is comparable to that from classical ANN calculations. We finally apply quantum processors to perform the hybrid quantum-classical machine learning calculations and obtain slightly lower accuracy ascribed to the fragile nature of quantum bits in quantum processors. Our work initiates the adoption of fledgling

quantum computers in the noisy intermediate-scale quantum (NISQ) era for discovering new HEAs.

Introduction

There have been four major historical eras: Hunter-Gatherer Age, Agricultural Age, Industrial Age, and Information Age according to the way humans work. This history can also be defined via representative materials that were invented in each historical period. Correspondingly, there have been stone, iron, and bronze ages and so on. Following this definition, it may not be exaggerated to claim that we are in the High-Entropy Material (HEM) age. For example, many oxides such as the Zr–Sn–Ti–O system with high dielectric constants (Van Dover et al.) can be categorized as high-entropy oxides, which serve as critical building blocks for complementary-metal-oxide-semiconductor (CMOS) transistors that have made the internet of things possible in the Information Age. This work centers around the design of HEMs particularly high-entropy alloys (HEAs) in the Information Age and beyond.

HEAs comprise multiple elements, each having the same or nearly the same concentration. Many HEAs, for example FeMnNiCoCr (Li et al.), have been found to possess distinct properties due to added degrees of freedom from both the number and type of elements present. As a result, a wide range of applications have been proposed such as turbine engines that require high strengths at extremely high temperatures (Senkov et al., “Development”) and high corrosion resistance for surface coatings (Cheng et al.). As the number of possible constituent elements of HEAs increases, the number of potential HEAs increases exponentially. It is estimated that 25 elements in the periodic table can lead to over five billion quinary HEAs (Rickman et al.). The challenge in designing novel HEAs therefore lies in selecting a combination that forms an alloy with the desired traits. Depending on which elements are present, the phases of an HEA can vary greatly. Common phases in HEAs can be classified into three categories (Senkov et al., “Accelerated”): solid solution (SS), intermetallic (IM), and a combination of the two (SS+IM). The resulting phase plays key roles in determining the properties of an HEA. Thus, the design and selection of an HEA is tantamount to predicting its phase. Obtaining a method capable of predetermining the phase of any combination of elements and their concentrations will be extremely advantageous in the discovery of new HEAs.

Classical computers play important roles in HEA design. In particular, a number of machine learning models implemented in classical computers have been deployed for this purpose (Rickman et al.; Huang et al.; Chen et al.; Zhang et al.) . Although such existing classical computational tools for HEA discovery appear to fulfill general needs at present, we will inevitably run into dilemmas such as the curse of dimensionality as the three V’s of HEA data continues to expand. This situation is actually occurring in other scientific fields. For example, the GPT-3 autoregressive language model (Brown et al., “Language”), which is able to understand text, respond to questions, and create new writing examples, contains 175 billion parameters and results from training 45 terabytes of text data for over 30 days using hundreds of GPUs. One promising solution to avoid this predicament is to upgrade current classical computation to quantum infrastructures via embracing quantum computers and algorithms to process the quantum information.

As Moore’s law is coming to an end with the size of transistors in classical computers approaching the physical limit of an atom, quantum computers hold great potential to meet ever-growing computational need by adopting a wide range of hosts (for example, electrons and artificial atoms made of superconducting circuits) for quantum bits instead of transistors. Unlike classical computers that provide a deterministic solution to a problem (for example, a combinatorial optimization problem in drug discovery), quantum computers provide a probabilistic solution via concerted quantum phenomena of superposition, entanglement, and interference. Quantum computers are fundamentally different from classical computers. Classical computers use classical bits as information carriers, each of which is deterministically represented by either 0 or 1. By contrast, quantum computers use quantum bits (qubits), each of which is in a superposition state of 0 and 1 and collapses into only 0 or 1 with a certain probability when the qubit is measured. Due to the fragile nature of qubits, all quantum computations currently need to be done at a short timescale (e.g., about 100 microseconds for superconducting qubits (Wendin)). Steady progress has been made in the hardware of quantum computing over the last two decades. We are now in the era of Noisy Intermediate-Scale Quantum (NISQ) (Preskill) at our disposal with IBM recently reaching a milestone of over 100 gate-controllable qubits (Sparkes). Despite such a small number of available qubits, quantum computers have shown so-called ‘quantum supremacy’, where the probability distribution of bit strings resulted from 53-qubit Google’s Sycamore quantum processor cannot be simulated at the same amount of time by the most advanced classical computers in the world at the time of experiment (Arute et al.). Quantum computers have also been employed to study binding energies of small molecules (Kandala et al.) and perform machine learning tasks based on quantum versions of machine learning algorithms such as support vector machines (Havlíček et al.). Quantum computing algorithms, especially Shor’s integer factorization algorithm (Shor) and Grover’s quantum search algorithm (Grover), have been proven to show quantum speedup over classical computers and exhibit practical applications, with the former in breaking current cryptography protocols and the latter in unstructured database searches.

A typical quantum algorithm achieves quantum speedup via two key quantum mechanical phenomena: quantum superposition and quantum interference (Nielsen and Chuang). The first feature generates a superposition state and therefore enables access to the infinitely large Hilbert space with the knowledge of the amplitudes of corresponding basis in the space. Using this feature alone is insufficient for quantum speedup, since measuring the superposition state destroys the quantum state leading to a random bit string of 0s and 1s with a probability equal to a target bit string corresponding to the solution (Nielsen and Chuang). This is an unwanted outcome. Instead, we need to apply quantum gates with a uniform distribution of the basis. We therefore need to creatively use the second feature, quantum interference, which can lead to the targeted bit strings of 0s and 1s when the quantum state in a quantum circuit is measured and amplify the amplitude of targeted bit string combination. Quantum algorithms often fall into two categories of extremes in terms of their efficiency and practicality. On one hand, several quantum algorithms especially the remarkable Shor's integer factorization algorithm which poses tangible threats to the current protocols of cryptography, have an exponential quantum speedup comparing to their classical counterparts, but their practical applications heavily depend on millions of qubits, which are not likely to be available in the near future. On the other hand, for other quantum algorithms such as Grover's quantum search algorithm, the quantum speedup is not as impressive, but it can be coupled with existing classical algorithms to solve optimization problems that are ubiquitous in machine learning algorithms.

Quantum computers are particularly promising for solving combinatorial optimization problems. For example, Grover's search algorithm forms the basis of implementing adaptive global optimization algorithms (Baritompa et al.). Furthermore, the quantum approximation optimization algorithm (QAOA) (Farhi et al.) has been developed and implemented in a superconducting quantum processor (Harrigan et al.). Examples of QAOA applications include solving combinatorial optimization problems such as nondeterministic polynomial (NP)-time complete max cut and dominating set problems related to graphs (Medvidović and Carleo). Many machine learning problems can be reduced to optimization problems. As a result, quantum machine learning (QML), in addition to applications in computational chemistry, has naturally become another promising application domain of quantum computers in the NISQ era. Nearly all state-of-the-art machine learning algorithms implemented in classical computers have their quantum counterparts (Biamonte et al.). Many classical machine learning algorithms such as support vectors machines and generative adversarial networks (GANs) have been implemented with quantum computers (Hu et al.; Zoufal et al.; Yang et al.) . QML has also recently been used to discover drugs for treating COVID-19 (Batra et al.). A significant amount of time reduction in each training epoch was observed and the training accuracy was comparable to that from classical machine learning. Designing new HEAs belongs to a challenging combinatorial optimization problem in the vast compositional space. Here we explore the potential of QML being used to tackle this problem.

Currently, there are two main flavors of QML. The first one is completely based on quantum algorithms such as quantum Fourier transform (Hales and Hallgren) and quantum phase estimation by Harrow, Hassidim, and Lloyd (Harrow et al.), but these algorithms require a fault-tolerant quantum computer with a number of physical qubits which are currently unavailable. We therefore adopt the second flavor of QML, which is using hybrid quantum-classical algorithms such as the aforementioned QAOA, quantum variational eigensolver and quantum circuit learning (Kandala et al.; Peruzzo et al.; Mitarai et al.) . These hybrid models formalize a machine learning problem as a variational optimization problem jointly tackled by a parameterized quantum circuit (from available NISQ resources) and classical computers (Benedetti et al.). Several advantages of using hybrid over classical models have been shown in the literature. For example, a recent report shows that, compared to classical GANs, QuGANs based on hybrid quantum-classical modes achieve similar performance to classical GANs but with a significant reduction (94.98%) on the parameter set (Stein et al.). The quantum machine learning algorithm marries quantum and classical computers and adopts their different strengths to perform quantum versions of machine learning algorithms for typical the classification task.

Methods

It has been shown in recent literature that the properties of a compound are strongly correlated to its chemical composition (Zhang et al.). For example, the ElemNet, a deep neural net used for predicting materials properties, is trained using elemental composition as the only input (Jha et al.). We apply this concept in predicting the phase of HEA compounds using classical and quantum-classical deep learning models. The data we use is the classical dataset established by Miracle and Senkov, originally containing over 600 samples of HEA data (Miracle and Senkov). A subset of samples is chosen for training and testing including only those samples with relevance to the classifier's objective. We select HEA samples that involve the elements in the first several rows of the periodic table as shown in Figure 1(a). Additional data requirements include compounds made with the "as cast" synthesis method and listed with a specified phase. This is because many properties of HEAs depend on the microstructures obtained from different experimental fabrication techniques. We therefore restrict the data by selecting only those samples resulting from a specified experimental technique. The 336-sample subset is made of ~39% SS phase compounds, ~15% IM phase compounds, and ~46% SS+IM phase compounds. When dividing the data for training and evaluation, 70% (~235 samples) of the data is randomly selected for the training dataset and the remaining 30% (~101 samples) become the testing dataset. In order to use chemical composition as input, every HEA compound is represented by an array where each member of the array corresponds to the molar ratio of an element of the periodic table that exists in the compound. Elements that are absent in the compound have a value of 0. For example, $AlCoCrCu_{0.5}Fe$ encoded using the following scheme, ['Li', 'Be', 'B', 'C', 'Mg', 'Al', 'Si', 'Ti', 'V', 'Cr', 'Mn', 'Fe', 'Co', 'Ni', 'Cu', 'Zn', 'Ge', 'Y', 'Zr', 'Nb', 'Mo', 'Ag', 'Sn', 'Hf', 'Ta', 'W', 'Au'] is represented by the array [0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0.5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]. This encoding is used for both classical ANN and quantum-classical hybrid classifiers.

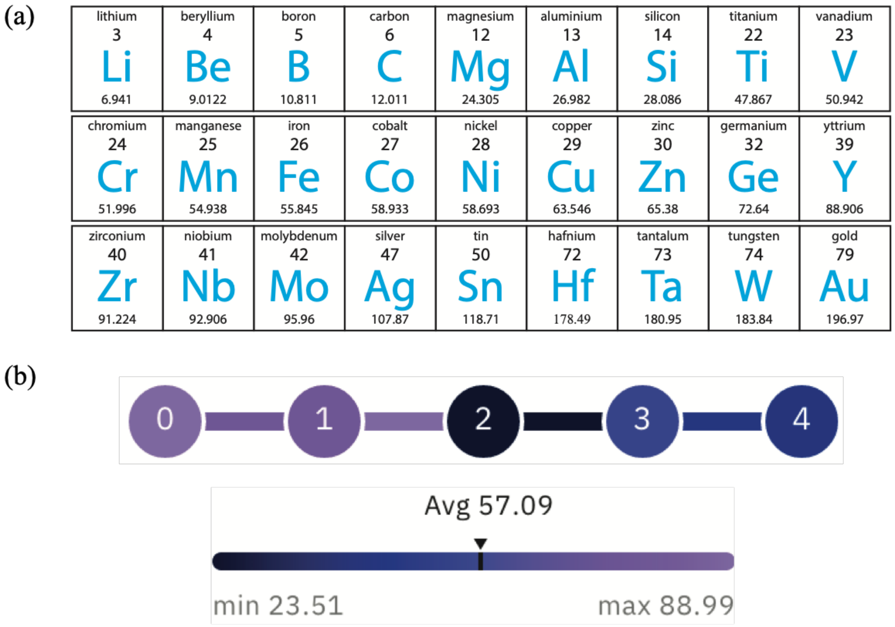


Figure 1. (a) 27 common elements that are involved in forming the high-entropy alloys listed in the dataset of Miracle and Senkov (Miracle and Senkov). These alloys exhibit three different phases. (b) Schematic of the arrangement of the five qubits in the *ibmq_manila* IBM quantum processor. The qubits are labeled by numerals from 0 to 4 and the circles are color coded by the T_2 decoherence with a time scale in the unit of microseconds (μs). As can be seen, every qubit is different having different T_2 values with the average of about 57 μs.

Figure 2(a) displays the classical ANN model used in this work. ANN architecture is made up of layers of nodes, represented by circles in Figure 2, each with an activation condition and function. Information flows from layer to layer extracting features and abstracting the data. Figure 2(a) shows a single-hidden-layer model which was chosen to closely match the architecture of the quantum-classical hybrid model for comparison. The classical ANN hidden layer uses a rectified linear unit (ReLU) activation function leading into the output layer with a SoftMax activation function. The number of nodes in the hidden layer are varied to match the complexity of each classification problem. For ternary classification problems we use a 7-node hidden layer and for binary classification problems we use 4 nodes. Categorical cross-entropy is used for the loss function, accuracy for the performance metric, and Adam optimizer for optimization. Even with this simple model, overfitting is expected to be a problem given the small size of the dataset and the disparity between the number of samples belonging to each class. To ensure that overfitting does not occur, a dropout layer is also included in the ANN model.

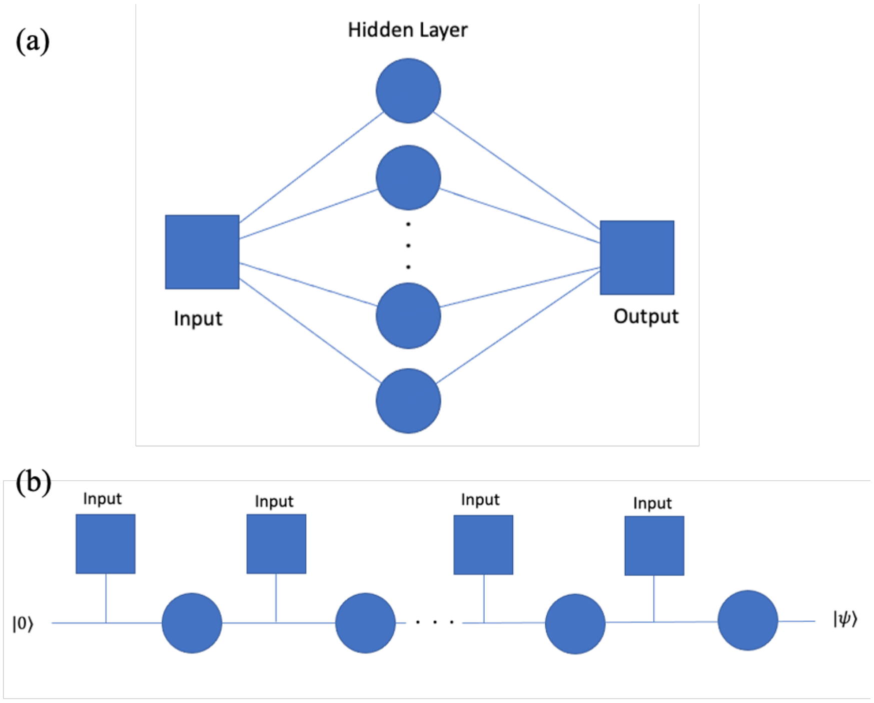


Figure 2. Graphical representations of (a) classical and (b) quantum classifiers. Connections show information flow between nodes. Classical neural network nodes in (a) are able to duplicate information and send to multiple nodes whereas quantum classifier in (b) must re-upload information for each processing node.

The categorical cross-entropy loss function can be written as

$$Loss_{cross-entropy} = -\sum_{i=1}^{n} y_i \cdot \log \hat{y}_i \tag{1}$$

where n is the number of classes, y_i is the target value of the i^{th}sample, and $\hat{y}_i$ is the corresponding model output.

With current limitations in quantum hardware in addition to information loss due to decoherence, circuit depth and complexity were considered when selecting a QML model. A quantum-classical, parameterized quantum circuit classifier was chosen because of its small circuit depth. The QML classifier selected to categorize HEA compounds is a single qubit classifier with data re-uploading (Pérez-Salinas et al.). In classical neural networks, information can flow from one node of a layer to every node in a subsequent layer (see Fig. 2a), a process that quantum computers are incapable of performing due to the no-cloning theorem (Wootters and Zurek). This quantum classifier, inspired by classical neural-networks, overcomes this limitation by simply inputting the data multiple times, once for every parameterized node as shown in Fig. 2b. Figure 3 illustrates the hybrid quantum-classical machine learning model, where classical optimization algorithms constantly communicate with the quantum circuit to update the unitary operator parameters, which evolve the input quantum states of HEA data and minimize the loss function.

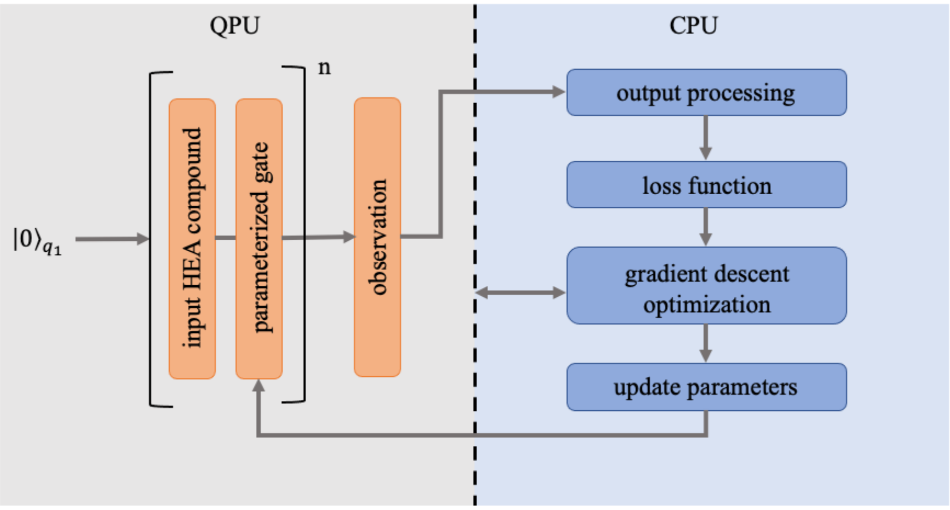


Figure 3. A hybrid quantum-classical machine learning model, where central processing unit (CPU) of classical computers provide optimized parameters resulted from standard gradient descent variants to a quantum processing circuit (QPU), in which the quantum state rapidly evolves into the state that minimize the defined loss. $|0\rangle_{q_1}$ denotes the single qubit used for classification and n denotes the number of processing nodes (“layers”) or number of times the bracketed steps are repeated in the quantum circuit.

The quantum circuit utilizes the Bloch sphere, more specifically quantum state rotations within this space, to input data and introduce optimizable parameters to the circuit, the first two QPU steps seen in Figure 3. The Bloch sphere is a common geometric representation of two-level quantum states. Most often the positive and negative *z* directions (i.e., analogous to north and south poles, respectively) correspond to the spin-up, $|0\rangle$ and spin-down, $|1\rangle$, states, respectively. The rotation operation with which the single qubit classifier operates is a three-parameter unitary operator, $R(\phi, \theta, \lambda)$, that can be represented in the following forms:

$$R(\phi, \theta, \lambda) = RZ(\lambda)RY(\theta)RZ(\phi), \tag{2}$$

written in unitary matrix form,

$$R(\phi, \theta, \lambda) = \begin{bmatrix} e^{-i\,(\phi+\lambda)/2}\cos(\theta/2) & -e^{i\,(\phi-\lambda)/2}\sin(\theta/2) \\ e^{-i\,(\phi-\lambda)/2}\sin(\theta/2) & e^{i\,(\phi+\lambda)/2}\cos(\theta/2) \end{bmatrix}. \tag{3}$$

Here, $RZ(\lambda)$ and $RY(\theta)$ are rotation functions about the z and y axis, respectively, and can be written as

$$RZ(\lambda) = \exp\left(-i\frac{\lambda}{2}Z\right) = \begin{bmatrix} e^{-i\,\lambda/2} & 0 \\ 0 & e^{i\,\lambda/2} \end{bmatrix} \tag{4}$$

and

$$RY(\theta) = \exp\left(-i\frac{\theta}{2}Y\right) = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}, \tag{5}$$

where Z and Y represent the Pauli gates that rotate a state about the z and y axis by π radians:

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{6}$$

and

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}. \tag{7}$$

Here only two axes (Z and Y) of rotation are used for simplicity, as these rotations are sufficient to create any state in the Bloch sphere. Figure 4 depicts a quantum state initialized at $|0\rangle$ and demonstrates the effects of one rotation function, $R(1\ ,\ \ 1\ \ ,\ \ 1)$.

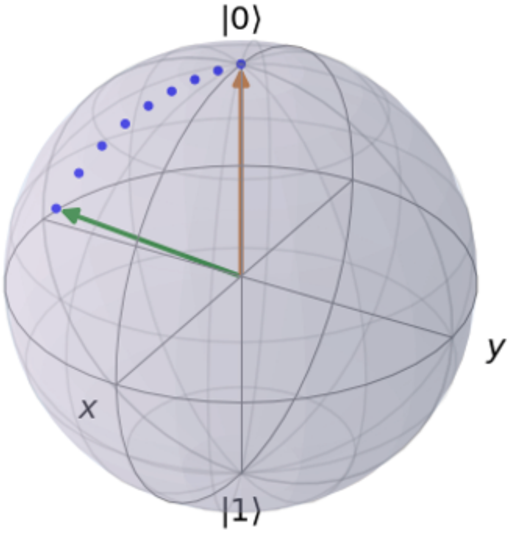


Figure 4. Visualization of the rotation function in the Bloch sphere. The rotation is applied to a quantum state initialized at $|0\rangle$, represented by the orange vector, and ends in the state represented by the green vector.

HEA compound information is transferred to a quantum state by inputting its array representation, separated into groups of three, with the rotation function. There are a total of 27 elements found in the HEA dataset, therefore nine rotations are used to enter a compound. The rotation function is also used as the parameterized gate (Figure 3), with each layer having three optimizable parameters. The resulting quantum circuit can be simplified to the form in Figure 5 (Pérez-Salinas et al.). A single data uploading and application of one parameterized gate represent one "layer" or node in the model and can be repeated an arbitrary number of times as shown in Figures 3 and 5. Changing the number of layers is analogous to changing the number of nodes in a hidden layer of a classical ANN, thus changing the total number of parameters and consequently the model's complexity.

Just as the number of layers in the quantum circuit is analogous to the number of nodes in a classical ANN, additional parameters can be introduced into the quantum model to replicate data flow weights between nodes. The dot product of the weights, w, and the compound arrays, x, are to be used as quantum circuit input. A separate set of weights exists for each layer.

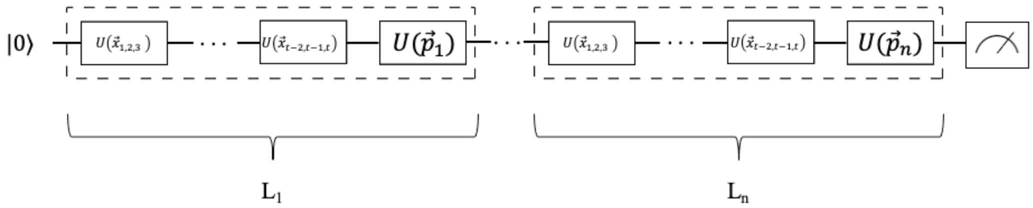


Figure 5. Quantum classifier circuit summary. Here $x = [x_1, x_2, \ldots, x_t]$ is an encoded HEA compound array and $\vec{x}_{j,k,l}$ is the vector (x_j, x_k, x_l), U is the unitary function chosen to input data and parameterize the circuit (rotation function, R), n is the number of layers, $\vec{p}_1, \vec{p}_2, \ldots, \vec{p}_n$ are the optimizable parameters in form of $\vec{p}_i = (p_i^1, p_i^2, p_i^3)$ where $\vec{p}_i$ are the parameters for the i^{th} layer.

After information passes through the node(s), the next step is the quantum state read-out (i.e., make the observation) (Figure 3). Before model training takes place, arbitrary quantum states are chosen to represent the classification target values, in our case the HEA phases. The working principle of the classifier is to align the quantum state representing the HEA compound with these arbitrary states corresponding to its phase through the use of the parameterized gates. The closeness between the state representative of the HEA compound and the arbitrary phase state is determined by measuring the Hermitian observable, which indicates overlap between the two states. For this reason, we need to choose the most orthogonal arbitrary states and the number of states depends on the number of classes. HEA phase classification was designed for both ternary and binary cases. Figure 6 depicts the states chosen for both cases.

In addition to ternary classification, we perform two types of binary classifications. In the first type, we train models to distinguish SS from IM, SS from SS+IM, and IM from SS+IM. This type of binary classification is in keeping with a previous work that used computed features as inputs for a deep learning neural network. In the second type, we merge the SS and IM data to call them pure phase to discern it from the mixed SS+IM phase. Furthermore, we combine the IM and SS+IM data to name the resulting data non-SS phase (denoted as ¬SS) to distinguish it from the SS phase. Similarly, we combine the SS and SS+IM data (denoted as ¬IM) to differentiate them from the IM data. These three different combinations of data generate another three classification tasks.

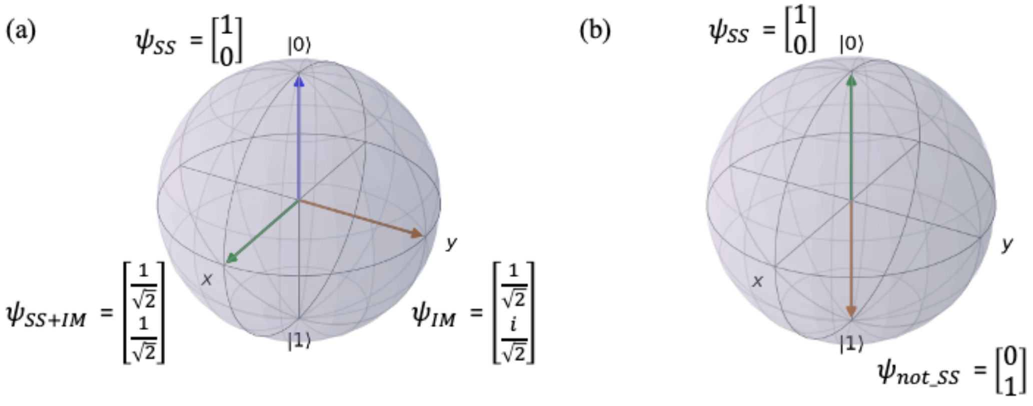

(a)
$\psi_{SS} = \begin{bmatrix}1\\0\end{bmatrix}$
|0⟩
x
y
|1⟩
$\psi_{SS+IM} = \begin{bmatrix}\frac{1}{\sqrt{2}}\\\frac{1}{\sqrt{2}}\end{bmatrix}$
$\psi_{IM} = \begin{bmatrix}\frac{1}{\sqrt{2}}\\\frac{i}{\sqrt{2}}\end{bmatrix}$
(b)
$\psi_{SS} = \begin{bmatrix}1\\0\end{bmatrix}$
|0⟩
x
y
|1⟩
$\psi_{not\_SS} = \begin{bmatrix}0\\1\end{bmatrix}$

Figure 6. Orthogonal states chosen to represent HEA phases. In the figure, SS is solid solution, IM is intermetallic, SS+IM is a mixture of the two. States are written in the form $\psi_S = \begin{bmatrix}\alpha\\\beta\end{bmatrix}$, where $\psi_S = \alpha|0\rangle + \beta|1\rangle$. (a) and (b) are used for ternary and binary classifications, respectively.

Once an observation is made, the information is used to either optimize the circuit parameters with the loss function if the model is training or make a prediction in the case that the model is being tested. These steps make up the classical portion of the hybrid algorithm (Figure 3). The same optimizer and performance metric used in the classical ANN are used to perform these tasks to make the models' performance comparable, however, as shown in Figure 3, the hybrid model optimizer must communicate with the QPU to compute the quantum gradient. Likewise, once parameters have been changed through the optimization process, they must be updated in the QPU, as shown in Figure 3. The loss functions also differ since the quantum classifier utilizes quantum observables in its loss function. The hybrid loss function is

$$Loss(\theta, w) = \sum_{i=1}^{M}(1 - |\langle\psi_s|\psi(\theta, w, x_i)\rangle|^2) \tag{8}$$

where $|\psi_s\rangle$ is the target arbitrary state corresponding to the phase of the i^{th} sample, $|\psi(\theta, w, x_i)\rangle$ is the resulting state from the parameterized circuit, $|\langle\psi_s|\psi(\theta, w, x_i)\rangle|^2$ is the Hermitian observable, θ represents the optimized parameters, and M is the number of training samples. The loss functions are defined differently for classical and quantum machine learning according to Equations (1) and (8), we therefore use normalized loss function to compare the two losses. We normalized the loss data (y-axis) with respect to its initial value before optimization was performed. The accuracy results of the models are reported as an average of the data entries from the last 20% of the epochs.

We first explore results from quantum simulators due to hardware limitations. A quantum simulator utilizes classical computers to approximate quantum states created in a quantum circuit and the manipulation of those states by quantum gates. This approximation is made by using a variety of methods including solving for Schrödinger wavefunctions of a qubit's state vector and can even account for noise present in quantum hardware through data modeling. However, even with noise modeling, quantum simulator approximations are still considered ideal as they have no decoherence time.

While simulations of quantum computers with classical hardware are not computationally efficient due to an exponential increase in required resources for each qubit, they allow for deeper investigation into the capabilities of parameterized quantum circuit deep learning models.

Results and Discussion

Figure 7(a) shows the machine learning process for the ternary classification task using the ANN model. We observe that both the training and testing losses first decrease and then converge, however the converged testing loss is larger than the training loss. This gap could be due to the high dimensionality and small size of the dataset or an imbalanced distribution of the three classes into training and testing datasets. Figure 7(b) displays the training and testing accuracy of the ANN model. We also observe similarly quick convergence of the two accuracy results. For example, the testing accuracy converges to an average value of 60.3% after 200 epochs of training. This resulting testing accuracy is smaller than that (74.3%) in previously trained ANN model for ternary classification, which used five computed features (e.g., atomic radius difference) (Huang et al.). All of these results show that composition information alone as inputs for ANN may not be sufficient to lead to similar testing accuracy. Despite this discrepancy, we focus on whether the quantum algorithms can reproduce similar predictions from classical computers based on the ANN model.

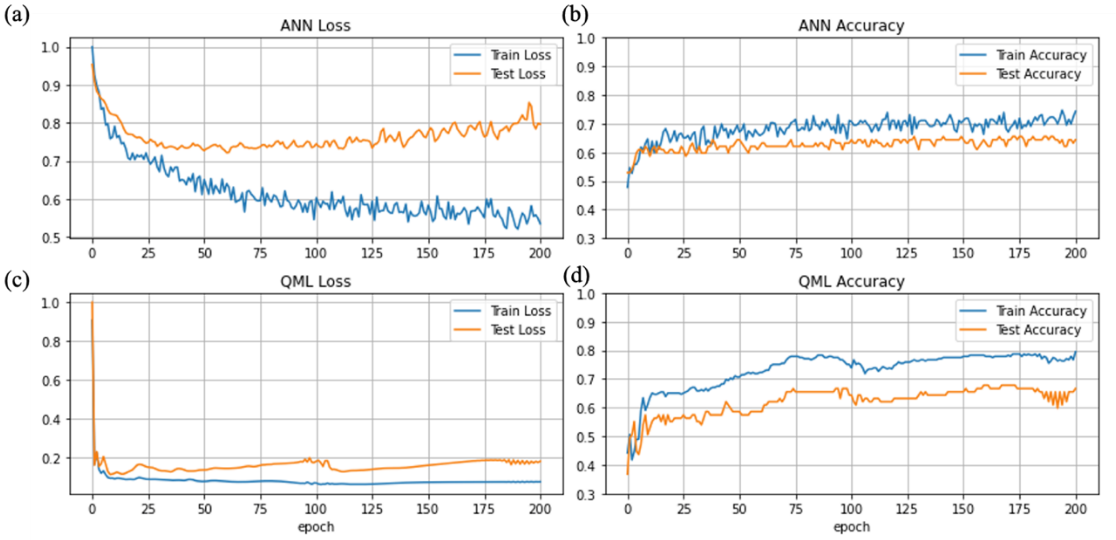


Figure 7. Training and testing (a) loss and (b) accuracy resulted from the artificial neural network (ANN) model calculations for ternary classifications of phases in high-entropy alloys. (c) and (d): corresponding loss and accuracy from quantum simulator calculations.

We now compare the results from quantum simulators to those from the above ANN calculations. Figure 7(c) displays the training and testing losses from the calculations using quantum simulators. We observe a similar trend in the machine learning process manifested by the decreased-and-then-converged training and testing losses. Notably, both losses seem to converge much more rapidly than the learning process in the classical ANN model. This comparison indicates an advantage of the hybrid quantum-classical machine learning model over the classical ANN model in terms of the efficiency of simultaneously classifying the three phases. Figure 7(d) shows the training and testing accuracy converge to 77.8% and 65.8%, respectively. Both accuracy values are comparable to the class ANN counterparts extracted from Fig. 7(b). This compatibility suggests that hybrid quantum-classical machine learning model as implemented in the quantum simulator can almost reproduce results from the classical ANN model. We can therefore conclude that the hybrid model is well debugged and ready for being deployed for test in quantum processors.

We next replace the quantum simulators with the IBM quantum processors to perform the above machine learning task for ternary classification. The number of epochs that can be completed by quantum processors is currently still limited because of the small number of available qubits. Figure 8(a) shows the training and testing losses after 15 epochs. As can be seen, despite the small number of machine learning epochs, the trends of the two curves are similar to those from ANN and quantum simulators (see e.g., Figs. 7(a)), which means that learning progress also occurs in the quantum processors. The training and testing accuracy values for ternary classification (Fig. 8(b)) are about 59.2% and 53.8%, respectively, which are both slightly smaller than those from quantum simulator computations. This slight deviation is somewhat expected due to the limited number of epochs as well as the unavoidable environmental noises that cause the quantum decoherence (see below) and damage the anticipated quantum states.

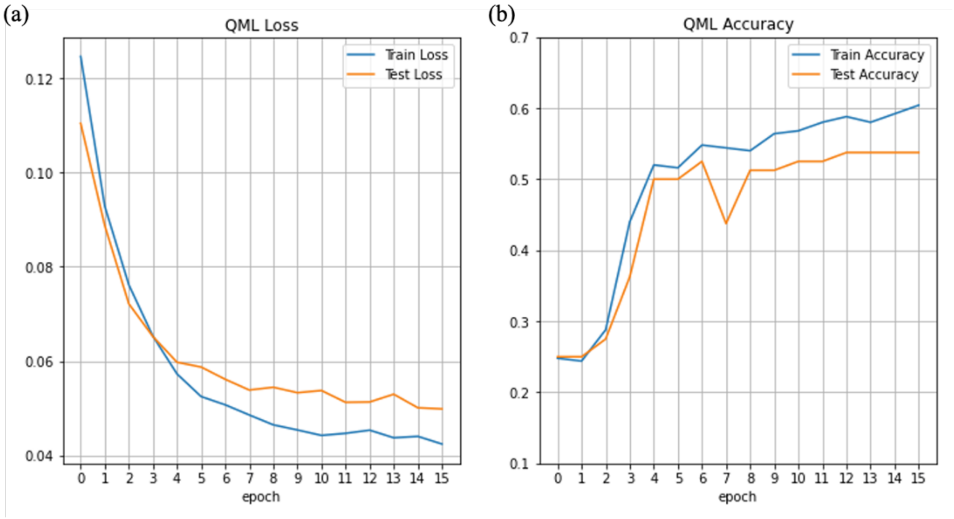


Figure 8. Quantum machine learning (a) loss and (b) accuracy for classifying the ternary phases of high-entropy alloys implemented in quantum processors.

The relatively low testing accuracy in the ternary classification task, in both quantum simulator and quantum processor calculations, may result from different sources, most likely the small dataset and imbalanced data (i.e., much less IM data than the other two categories). We therefore switch to the two types (and six tasks in total) of binary classifications. For each task, we perform the same set of machine learning calculations, which are classical ANN model calculations followed by quantum simulator calculations. We then test a single case of each binary classification task using quantum processors.

We begin with the SS vs. IM task in the first type of binary classification. We first notice from Figure 9(a) that the training and testing losses using the ANN model in this task are much smaller than their corresponding losses (see Fig. 7(a)) in the ternary classification task, portending significantly higher testing accuracy, which is 87.5% as shown in Fig. 9(b). In the quantum simulator calculations, although the converged training and testing losses are higher than their counterparts in the ternary classification task (see Fig. 7(c)), the training and testing accuracy values are once again comparable to those from ANN calculations and higher than in the ternary classification task. The larger accuracy manifests the disparity between the structures of SS and IM phases with the former exhibiting random structures whereas the latter show ordered ones. For the SS vs. SS+IM classification task (see Fig. 10), it is associated with the lowest testing accuracy of 73.3%, so it remains the most challenging among the three binary classification tasks (Huang et al.). Regarding the IM vs. SS+IM classification task (see Fig. 11), we observe nearly flat curves for the testing loss and testing accuracy because of the small number of IM data in the testing dataset. The model's sensitivity to the minority class, IM, was also evaluated to determine its generalizability to outside data. Relatively high recall values

(~70%) were found for each case showing that the imbalance did not significantly affect the model's sensitivity to either class. Note that the converged testing accuracy (81.7%) lies between that of the other two binary classification tasks, which disagree with the previous trained binary model (Huang et al.), implying the importance of including input features. Fig. 12 displays the results from quantum processors for the SS vs. SS+IM classification task, which consistently reveals the smaller testing accuracy than the ANN model or quantum simulator calculations due to quantum decoherence.

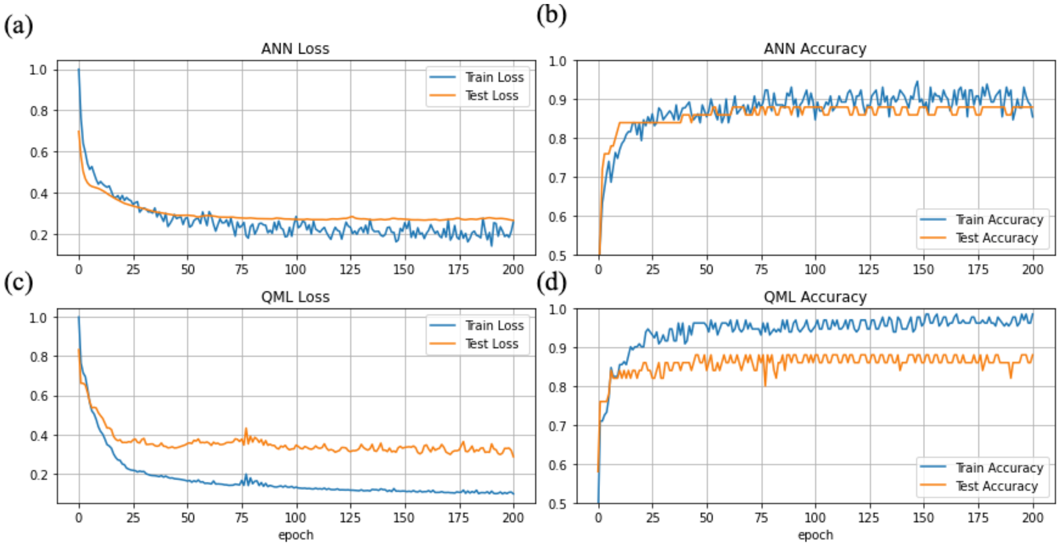


Figure 9. Training and testing (a) loss and (b) accuracy resulted from the artificial neural network (ANN) model calculations for binary classification of SS vs. IM phases in high-entropy alloys. (c) and (d): corresponding loss and accuracy from quantum simulator calculations.

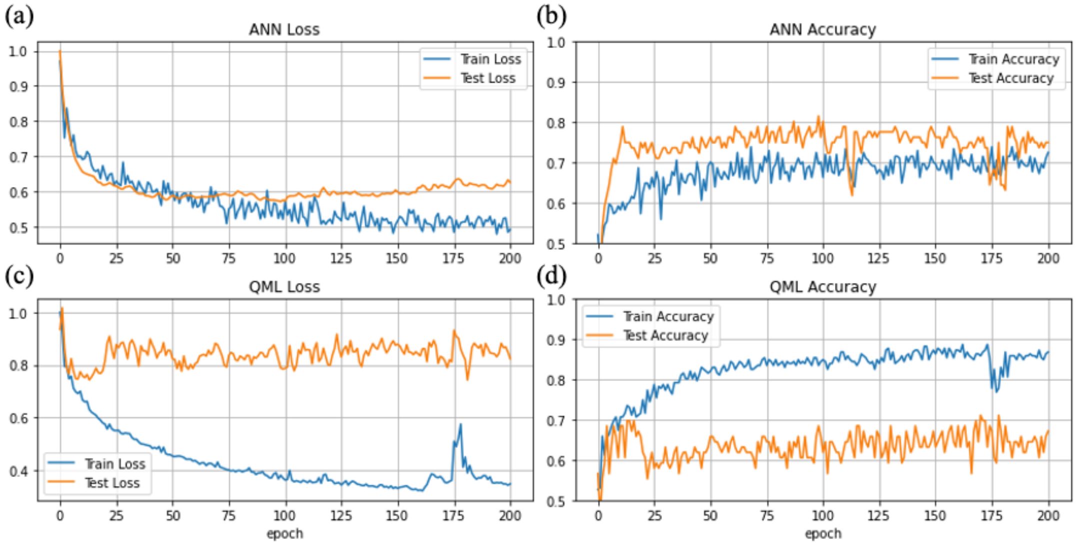


Figure 10. Training and testing (a) loss and (b) accuracy resulted from the artificial neural network (ANN) model calculations for binary classification of SS vs. SS+IM phases in high-entropy alloys. (c) and (d): corresponding loss and accuracy from quantum simulator calculations.

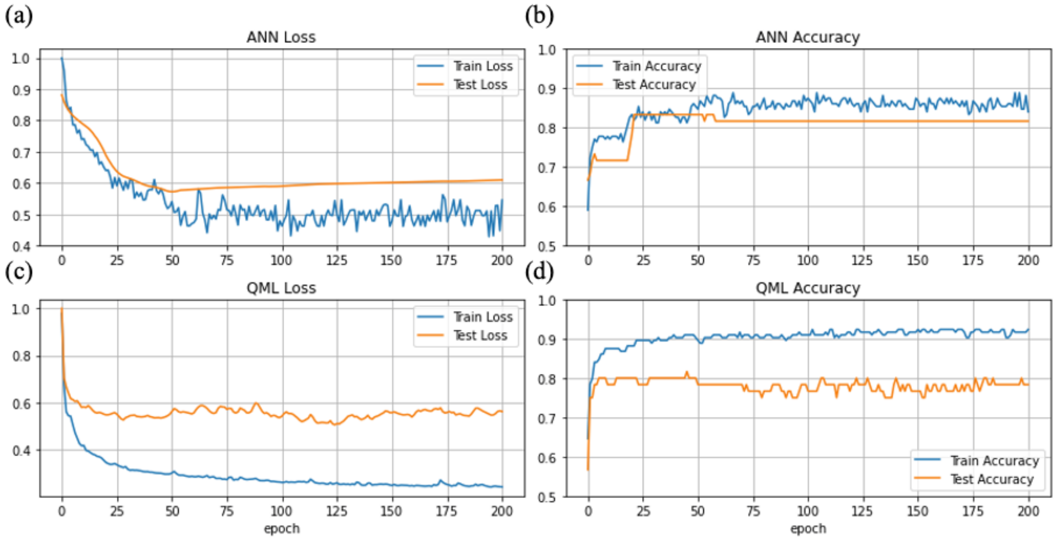


Figure 11. Training and testing (a) loss and (b) accuracy resulted from the artificial neural network (ANN) model calculations for binary classification of IM vs. SS+IM phases in high-entropy alloys. (c) and (d): corresponding loss and accuracy from quantum simulator calculations.

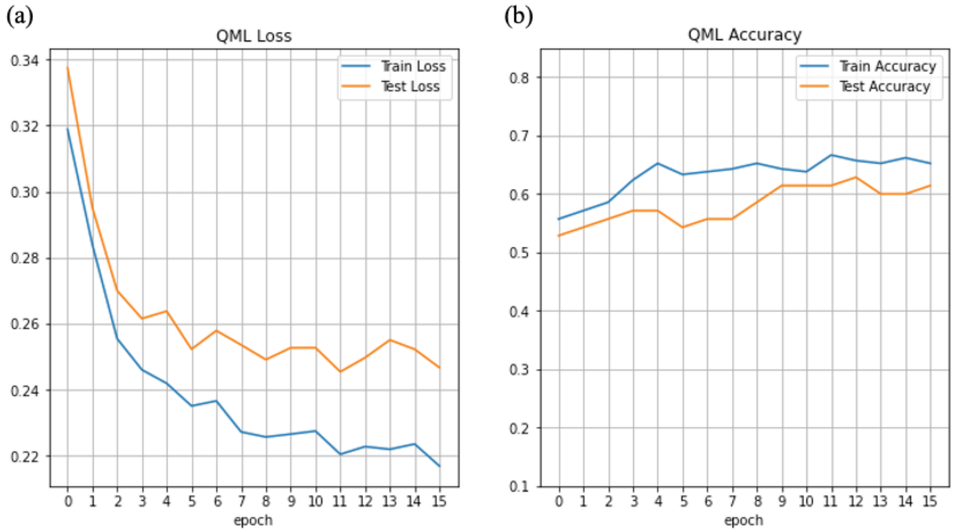


Figure 12. Quantum machine learning (a) loss and (b) accuracy for binary classification of SS vs. SS+IM phases in high-entropy alloys implemented in quantum processors.

The above three binary classification tasks result in generally higher accuracy than that from the ternary classification task. We now perform the second type of binary classification task to examine the resulting accuracy. Figure 13 shows that the pure vs. mixed task is a challenging one, which is reflected by the large training losses (either from ANN or quantum simulator calculations; see Fig. 13 (a) and (c)) and also by the low testing accuracy of 65.8% (Fig. 13(b)) and 72.5% (Fig. 13(d)) for ANN and quantum simulators, respectively. The challenge of this classification task indicates the absence of a clear boundary between the pure and mixed phases. For example, an SS phase may exhibit a short-range order to some extent, as a result of which the structure can be legitimately labeled as SS+IM. By contrast, Figure 14 reveals that the IM vs. ¬IM task is associated with the highest training and testing accuracy (for example, respectively, 95.4% and 88.6% from the quantum simulator calculations) and low training and testing losses. This suggests the distinctive feature (including a fixed stoichiometry along with an ordered structure) of IM phase facilitating the identification of the phase in experimental characterization. Figure 15 shows that the SS vs. ¬SS task leads to intermediate training and testing losses and accuracy comparing to the other two tasks in this type of binary classifications. For example, the testing accuracy from ANN or quantum simulator calculations are 76.6% and 79.3%, respectively. Similar to the ternary classification, all six binary classifications show that quantum simulator results are consistent with the ANN model results. Fig. 16 shows the results of quantum processors training the model to classify the SS vs. ¬SS task. Though the processors were able to successfully train the model, the converged accuracies are lower than that of the ANN and quantum simulation models due to decoherence.

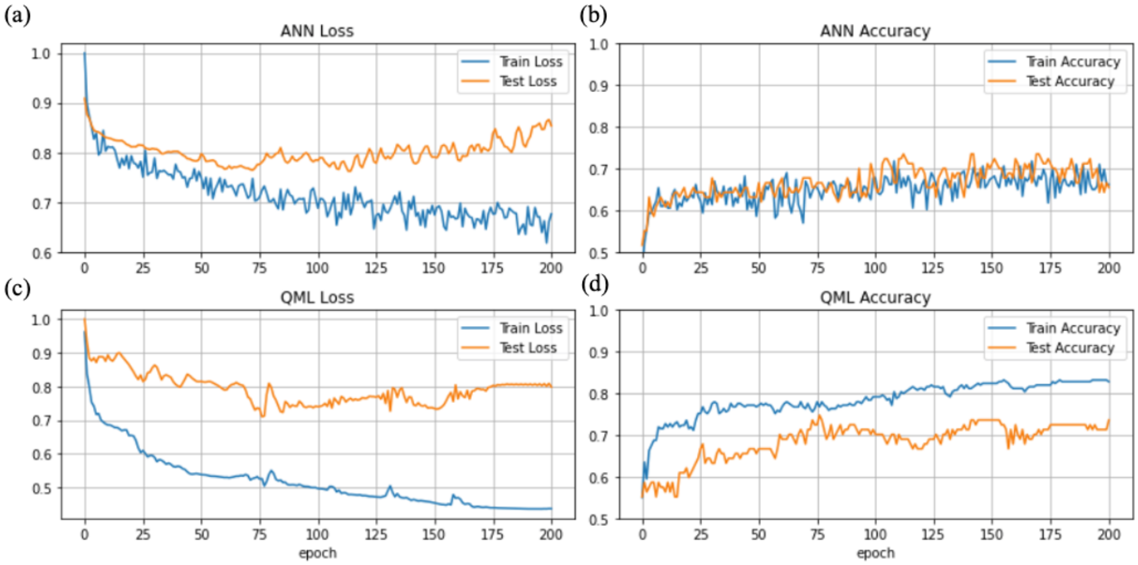


Figure 13. Training and testing loss and accuracy in binary classification of pure vs. mixed phases of high-entropy alloys using (a) and (b) artificial neural network and (c) and (d) quantum machine learning model as implemented in quantum simulators.

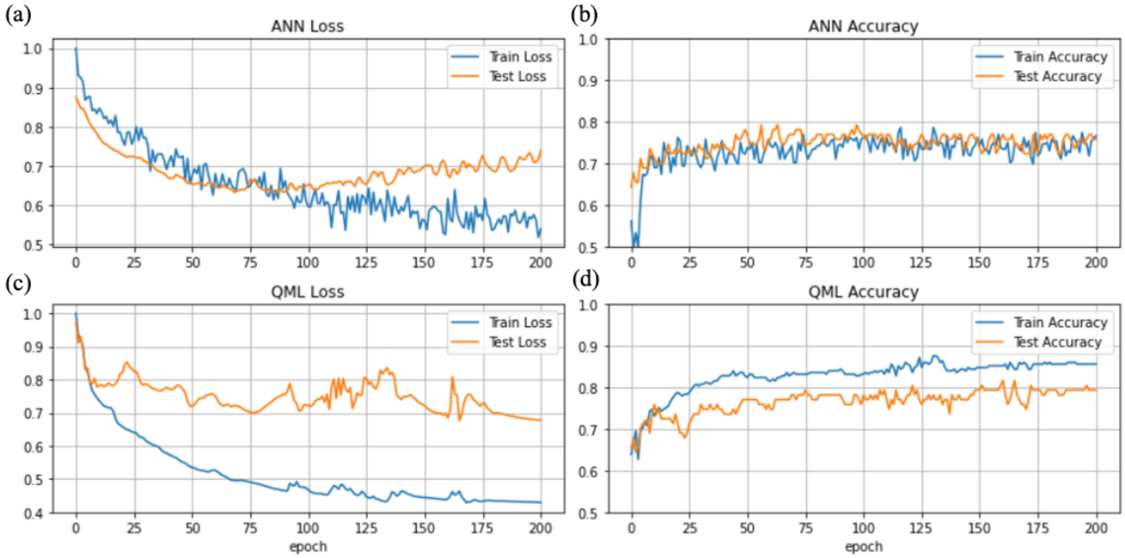


Figure 14. Training and testing loss and accuracy in binary classification of SS vs. ¬SS phases of high-entropy alloys using (a) and (b) artificial neural network and (c) and (d) quantum machine learning model as implemented in quantum simulators.

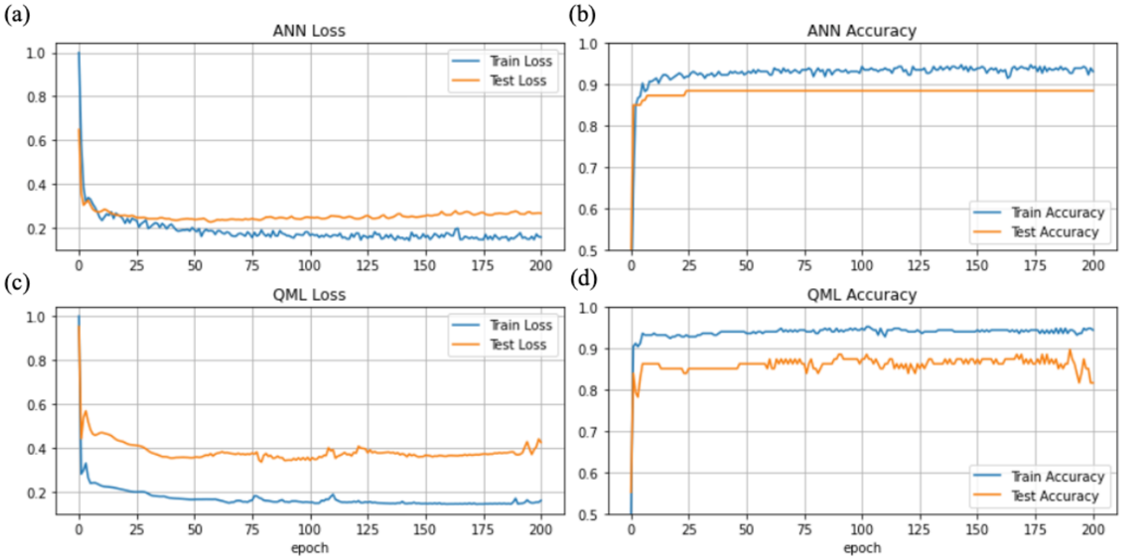


Figure 15. Training and testing loss and accuracy in binary classification of IM vs ¬IM phases of high-entropy alloys using (a) and (b) artificial neural network and (c) and (d) quantum machine learning model as implemented in quantum simulators.

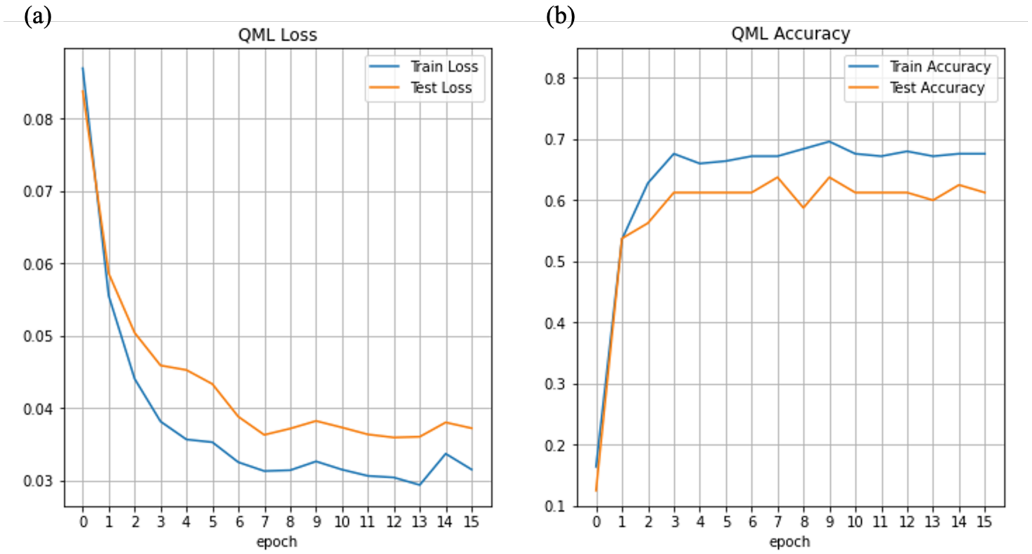


Figure 16. Quantum machine learning (a) loss and (b) accuracy for binary classification of SS vs. ¬SS phases in high-entropy alloys implemented in quantum processors.

In the above hybrid quantum-machine learning calculations, data encoding allows us to use a single qubit to represent each HEA compound. Alternatively, we explore another scheme to encode an HEA compound via deploying multiple qubits that are available on the IBM quantum processors (e.g., the 5-qubit *ibmq_manila* quantum hardware). This alternative scheme depends on the fact that any material (not limited to HEA compounds in the present context) with a specific chemical formula can be represented as a quantum superposition state. For example, one of the most well studied HEA, CoCrFeMnNi (Otto et al.) can be represented by the ket vector in the Dirac notation: $|\mathrm{CoCrFeMnNi}\rangle = 1/\sqrt{5}\, |\mathrm{Co}\rangle + 1/\sqrt{5}\, |\mathrm{Cr}\rangle + 1/\sqrt{5}\, |\mathrm{Fe}\rangle + 1/\sqrt{5}\, |\mathrm{Mn}\rangle + 1/\sqrt{5}\, |\mathrm{Ni}\rangle$. For a non-equimolar compound such as another important HEA, $Al_{0.5}CoCrCuFeNi$ (Hemphill et al.), can be written in a similar manner: $|\mathrm{Al_{0.5}CoCrCuFeNi}\rangle = 1/\sqrt{21}\, |\mathrm{Al}\rangle + 2/\sqrt{21}\, |\mathrm{Co}\rangle + 2/\sqrt{21}\, |\mathrm{Cr}\rangle + 2/\sqrt{21}\, |\mathrm{Fe}\rangle + 2/\sqrt{21}\, |\mathrm{Mn}\rangle + 2/\sqrt{21}\, |\mathrm{Ni}\rangle$. As can be seen, the molar ratio associated with each element in the two chemical formulae have been transformed into the information of amplitude for a superposition state and normalized so that the sum of the squared amplitudes (i.e., probability) is equal to one. Furthermore, equal molar HEAs have a uniform amplitude whereas non-equal molar HEAs have different amplitudes. Following this encoding approach, each element in the periodic table constitutes a basis vector (i.e., a string of 0s and 1s) in the 2^n Hilbert space, where n is the number of qubits used in the encoding. For example, if an HEA compound contains eight elements, three qubits are needed for the representation. More generally, because there are 118 elements in the periodic table, seven qubits ($2^7 = 128$) are sufficient to represent the quantum states of any existing material and those that await discovery. Returning to the HEA dataset in this work, we assign integer decimal numbers from 0 to 26 to the 27 elements from Li to Au in the ascending order of the atomic number according to Fig.1(a). We then transform the 27 decimal numbers to 5-digit binary representations, i.e., from $|00000\rangle$ for $|\mathrm{Li}\rangle$ to $|11010\rangle$ for $|\mathrm{Au}\rangle$. As a result, the five qubits with a maximum total dimensionality of 32 is sufficiently large and only 27 dimensions (basis vectors) are needed to encode all the 336 compounds in the data set. Each HEA differs from each other in the amplitudes of the 27 basis vectors.

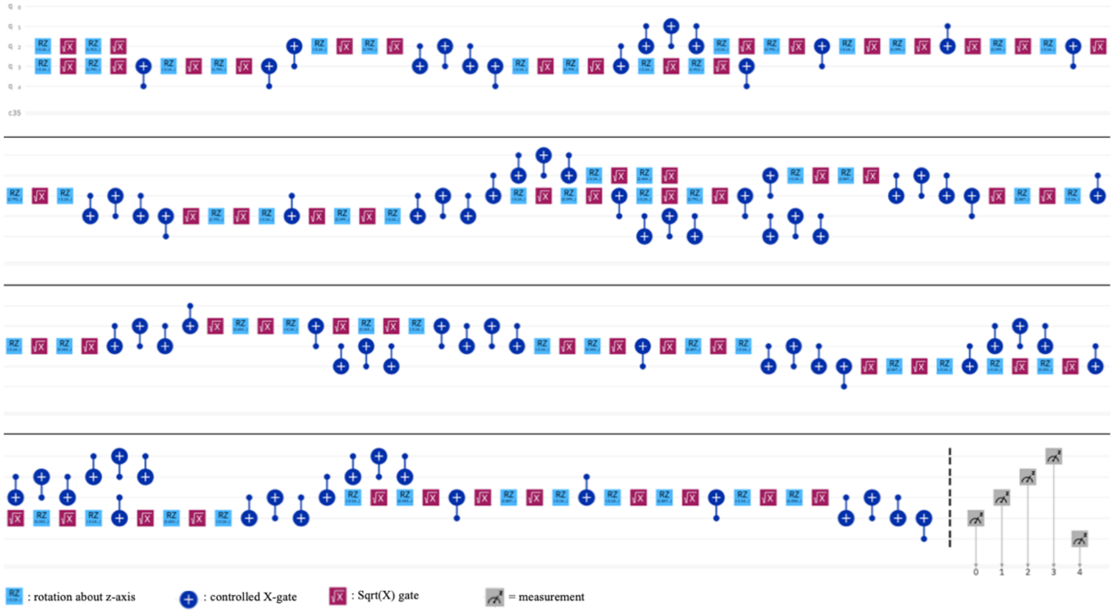


Figure 17. Quantum circuit diagram of five qubits used to encode high-entropy alloys. The diagram consists of 204 gates belonging to four types: rotation about the z-axis, controlled X-gate, Sqrt(X) gate, and measurement.

With this new HEA data encoding approach, we use CoCrFeMnNi and $Al_{0.5}$CoCrCuFeNi HEAs again as examples to create quantum superposition states in quantum circuits and measure these states after these states are prepared. We perform such preparation-measurement calculations for 2×10^4 times (each calculation is called a "shot" in the quantum processor), which are the maximally available shots for the *ibmq_manila* IBM quantum processor. Each shot corresponds to a full process for the quantum processor to implement all the 204 quantum gates illustrated in Fig. 17. We use both the quantum computer simulator and IBM quantum processors for the calculations and the measured probability distributions are shown in Fig. 18. In the quantum simulator calculations, because there is no noise, the heights of the bars, which refer to the probability of elements, are nearly uniform for CoCrFeMnNi and non-uniform for $Al_{0.5}$CoCrCuFeNi HEAs. Furthermore, In the $Al_{0.5}$CoCrCuFeNi HEA, the height for Al is about a quarter of the heights of the other five bars because of its smaller molar ratio comparing to other elements (0.5 vs. 1). In the quantum processor calculations, ideally, we expect to observe only five bars with non-zero probabilities as in quantum simulators. However, since the qubits are fragile and associated with a number of decoherence sources caused by the interactions of the qubits with the environment, we observe from Fig.18(b) and 18(d) that all the 32 possible strings of 0s and 1s show up with non-zero probability magnitudes for the two HEAs. Nevertheless, we can still see the five/six bars representing the five/six elements in the two HEAs have dominant heights over the other bars. This dominance implies that NISQ quantum computers remain useful to represent HEAs.

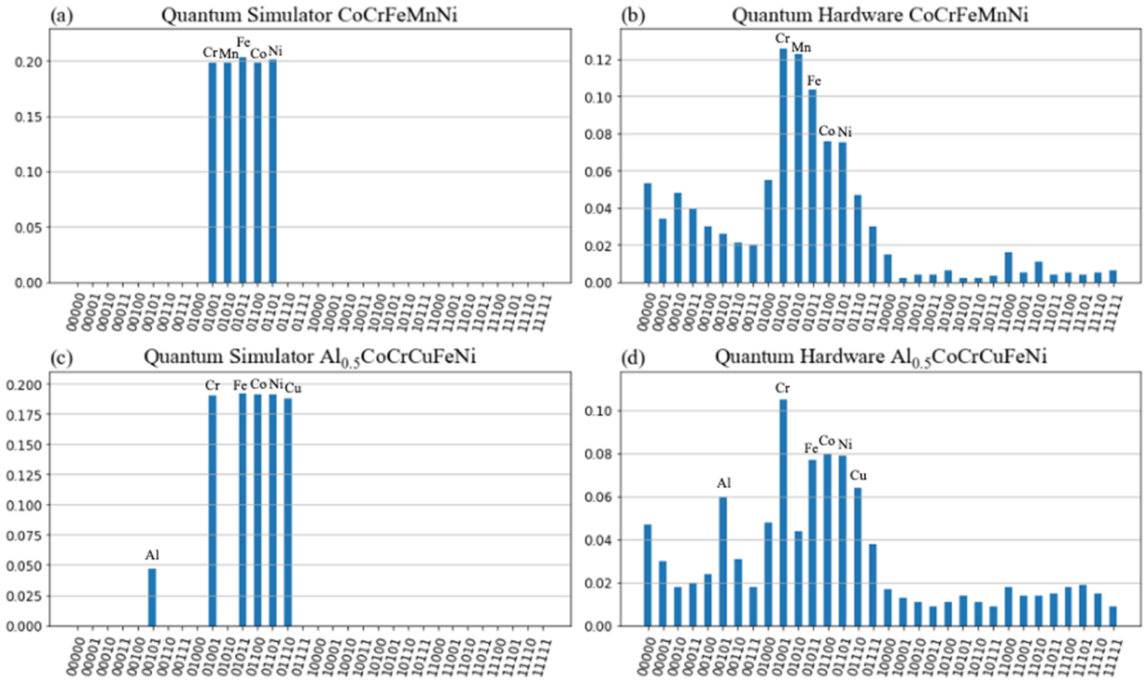


Figure 18. Probability distributions of constituent elements in two high-entropy alloys: (a, c) CoCrFeMnNi and (b, d) Al0.5CoCrCuFeNi, which are encoded as quantum superpositions states. The probability is measured from (a, b) quantum simulator and (c, d) quantum hardware calculations.

In our future work, quantum algorithms will be developed to introduce tricks of quantum interference. For example, 5 out of the 32 dimensions in the 5-qubit quantum processor have been idle without any encoded information. We can choose three of them, for example $|11010\rangle$, and use them to label the three possible phases. The goal of the quantum algorithm is efficiently (i.e., using a minimum number of shots) converting the quantum superposition states to one of the three target states. Furthermore, the encoding scheme can be generalized to any property of HEAs to solve regression problems in machine learning. For example, we can represent an HEA in terms of a linear combination of its all properties or dependent factors. Therefore, the quantum state of HEA's properties is $|\text{HEA}\rangle = \alpha\,|\text{yield strength}\rangle + \beta\,|\text{hardness}\rangle + \ldots$, where α, β refer to the values of the corresponding entries in the databases.

Conclusions

We have performed quantum machine learning calculations to predict phase selection in HEAs. Our trained ANN model and quantum simulator shows that the hybrid quantum-classical machine learning calculations performed by quantum simulators nearly reproduce the results from the classical ANN model. The algorithm implemented in quantum hardware can lead to slightly lower testing accuracy. Finally, we propose an alternative data encoding scheme that can be adopted for future studies. We emphasize that the test cases in this work undoubtedly can also be handled by classical computers and no drastic speed up using quantum hardware is obtained. This is because of the small size of the example database we used and also of the limited number of qubits in the noisy intermediate-scale quantum (NISQ) era. The trained hybrid quantum-classical machine learning model therefore should be treated as a ‘well-debugged’ model, awaiting to be implemented on future generations of quantum hardware that contains millions of fault-tolerant qubits. By then, we will most likely be able to take full advantage of these quantum algorithms over their classical counterparts.

Acknowledgements

This work was supported by the National Science Foundation (NSF) (Grant No. DMR-2020277). We acknowledge the use of IBM Quantum services for this work. The views expressed are those of the authors, and do not reflect the official policy or position of IBM or the IBM Quantum team. This research also used computational resources of the Agave cluster.

References

Arute, Frank, et al. “Quantum Supremacy Using a Programmable Superconducting Processor.” *Nature*, vol. 574, no. 7779, 2019, pp. 505–510., https://doi.org/10.1038/s41586-019-1666-5.

Baritompa, W. P., et al. “Grover's Quantum Algorithm Applied to Global Optimization.” *SIAM Journal on Optimization*, vol. 15, no. 4, 2005, pp. 1170–1184., https://doi.org/10.1137/040605072.

Batra, Kushal, et al. “Quantum Machine Learning Algorithms for Drug Discovery Applications.” *Journal of Chemical Information and Modeling*, vol. 61, no. 6, 2021, pp. 2641–2647., https://doi.org/10.1021/acs.jcim.1c00166.

Benedetti, Marcello, et al. “Parameterized Quantum Circuits as Machine Learning Models.” *Quantum Science and Technology*, vol. 4, no. 4, 2019, p. 043001., https://doi.org/10.1088/2058-9565/ab4eb5.

Biamonte, Jacob, et al. “Quantum Machine Learning.” *Nature*, vol. 549, no. 7671, 2017, pp. 195–202., https://doi.org/10.1038/nature23474.

Brown, T.B., et al. “Language Models Are Few-Shot Learners.” *ArXiv*, 2020, https://arxiv.org/abs/2005.14165. Accessed 2022.

Chen, Shuai, et al. “Machine Learning for High-Entropy Alloys.” *Artificial Intelligence for Materials Science*, 2021, pp. 21–58., https://doi.org/10.1007/978-3-030-68310-8_2.

Cheng, Hongxu, et al. “Review—Corrosion-Resistant High-Entropy Alloy Coatings: A Review.” *Journal of The Electrochemical Society*, vol. 168, no. 11, 2021, p. 111502., https://doi.org/10.1149/1945-7111/ac34d0.

Farhi, E., et al. “A Quantum Approximate Optimization Algorithm.” *ArXiv*, 2014, Accessed 2022.

Grover, Lov K. “Quantum Mechanics Helps in Searching for a Needle in a Haystack.” *Physical Review Letters*, vol. 79, no. 2, 1997, pp. 325–328., https://doi.org/10.1103/physrevlett.79.325.

Hales, L., and S. Hallgren. “An Improved Quantum Fourier Transform Algorithm and Applications.” *Proceedings 41st Annual Symposium on Foundations of Computer Science*, 2000, https://doi.org/10.1109/sfcs.2000.892139.

Harrigan, Matthew P., et al. “Quantum Approximate Optimization of Non-Planar Graph Problems on a Planar Superconducting Processor.” *Nature Physics*, vol. 17, no. 3, 2021, pp. 332–336., https://doi.org/10.1038/s41567-020-01105-y.

Harrow, Aram W., et al. “Quantum Algorithm for Linear Systems of Equations.” *Physical Review Letters*, vol. 103, no. 15, 2009, https://doi.org/10.1103/physrevlett.103.150502.

Havlíček, Vojtěch, et al. “Supervised Learning with Quantum-Enhanced Feature Spaces.” *Nature*, vol. 567, no. 7747, 2019, pp. 209–212., https://doi.org/10.1038/s41586-019-0980-2.

Hemphill, M.A., et al. “Fatigue Behavior of al0.5cocrcufeni High Entropy Alloys.” *Acta Materialia*, vol. 60, no. 16, 2012, pp. 5723–5734., https://doi.org/10.1016/j.actamat.2012.06.046.

Hu, Ling, et al. “Quantum Generative Adversarial Learning in a Superconducting Quantum Circuit.” *Science Advances*, vol. 5, no. 1, 2019, https://doi.org/10.1126/sciadv.aav2761.

Huang, Wenjiang, et al. “Machine-Learning Phase Prediction of High-Entropy Alloys.” *Acta Materialia*, vol. 169, 2019, pp. 225–236., https://doi.org/10.1016/j.actamat.2019.03.012.

Jha, Dipendra, et al. “ElemNet: Deep Learning the Chemistry of Materials from Only Elemental Composition.” *Scientific Reports*, vol. 8, no. 1, 2018, https://doi.org/10.1038/s41598-018-35934-y.

Kandala, Abhinav, et al. “Hardware-Efficient Variational Quantum Eigensolver for Small Molecules and Quantum Magnets.” *Nature*, vol. 549, no. 7671, 2017, pp. 242–246., https://doi.org/10.1038/nature23879.

Li, Zhiming, et al. “Metastable High-Entropy Dual-Phase Alloys Overcome the Strength–Ductility Trade-Off.” *Nature*, vol. 534, no. 7606, 2016, pp. 227–230., https://doi.org/10.1038/nature17981.

Medvidović, Matija, and Giuseppe Carleo. “Classical Variational Simulation of the Quantum Approximate Optimization Algorithm.” *Npj Quantum Information*, vol. 7, no. 1, 2021, https://doi.org/10.1038/s41534-021-00440-z.

Miracle, D.B., and O.N. Senkov. “A Critical Review of High Entropy Alloys and Related Concepts.” *Acta Materialia*, vol. 122, 2017, pp. 448–511., https://doi.org/10.1016/j.actamat.2016.08.081.

Mitarai, K., et al. “Quantum Circuit Learning.” *Physical Review A*, vol. 98, no. 3, 2018, https://doi.org/10.1103/physreva.98.032309.

Nielsen, Michael A., and Isaac L. Chuang. “Quantum Computation and Quantum Information.” 2012, https://doi.org/10.1017/cbo9780511976667.

Otto, F., et al. “The Influences of Temperature and Microstructure on the Tensile Properties of a Cocrfemnni High-Entropy Alloy.” *Acta Materialia*, vol. 61, no. 15, 2013, pp. 5743–5755., https://doi.org/10.1016/j.actamat.2013.06.018.

Peruzzo, Alberto, et al. “A Variational Eigenvalue Solver on a Photonic Quantum Processor.” *Nature Communications*, vol. 5, no. 1, 2014, https://doi.org/10.1038/ncomms5213.

Preskill, John. “Quantum Computing in the NISQ Era and Beyond.” *Quantum*, vol. 2, 2018, p. 79., https://doi.org/10.22331/q-2018-08-06-79.

Pérez-Salinas, Adrián, et al. “Data Re-Uploading for a Universal Quantum Classifier.” *Quantum*, vol. 4, 2020, p. 226., https://doi.org/10.22331/q-2020-02-06-226.

Rickman, J. M., et al. “Machine Learning Strategies for High-Entropy Alloys.” *Journal of Applied Physics*, vol. 128, no. 22, 2020, p. 221101., https://doi.org/10.1063/5.0030367.

Senkov, O.N., et al. “Accelerated Exploration of Multi-Principal Element Alloys with Solid Solution Phases.” *Nature Communications*, vol. 6, no. 1, 2015, https://doi.org/10.1038/ncomms7529.

Senkov, Oleg, et al. “Development of a Refractory High Entropy Superalloy.” *Entropy*, vol. 18, no. 3, 2016, p. 102., https://doi.org/10.3390/e18030102.

Shor, Peter W. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer.” *SIAM Journal on Computing*, vol. 26, no. 5, 1997, pp. 1484–1509., https://doi.org/10.1137/s0097539795293172.

Sparkes, Matthew. “A New Quantum Leader?” *New Scientist*, vol. 252, no. 3361, 2021, p. 7., https://doi.org/10.1016/s0262-4079(21)02045-5.

Stein, Samuel A., et al. “Qugan: A Quantum State Fidelity Based Generative Adversarial Network.” *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 2021, https://doi.org/10.1109/qce52317.2021.00023.

Van Dover, R. B., et al. “Discovery of a Useful Thin-Film Dielectric Using a Composition-Spread Approach.” *Nature*, vol. 392, no. 6672, 1998, pp. 162–164., https://doi.org/10.1038/32381.

Wendin, G. “Quantum Information Processing with Superconducting Circuits: A Review.” *Reports on Progress in Physics*, vol. 80, no. 10, 2017, p. 106001., https://doi.org/10.1088/1361-6633/aa7e1a.

Wootters, W. K., and W. H. Zurek. “A Single Quantum Cannot Be Cloned.” *Nature*, vol. 299, no. 5886, 1982, pp. 802–803., https://doi.org/10.1038/299802a0.

Yang, J., et al. “Support Vector Machines on Noisy Intermediate Scale Quantum Computers.” *ArXiv*, 2019, Accessed 2022.

Zhang, Jie, et al. “Composition Design of High-Entropy Alloys with Deep Sets Learning.” *Npj Computational Materials*, vol. 8, no. 1, 2022, https://doi.org/10.1038/s41524-022-00779-7.

Zoufal, Christa, et al. “Quantum Generative Adversarial Networks for Learning and Loading Random Distributions.” *Npj Quantum Information*, vol. 5, no. 1, 2019, https://doi.org/10.1038/s41534-019-0223-2.

CHAPTER 3

SUMMARY

Findings

The hybrid quantum-classical machine learning method implemented on quantum simulators and quantum processors was able to successfully perform phase prediction of HEAs. While this method does not claim to be more efficient than the classical ANN model, it is shown that current quantum computers can perform machine learning tasks with comparable results. The average difference between ANN testing accuracies and the testing accuracies of the hybrid quantum-classical model performed on quantum simulators was 3.9%. The average difference between ANN testing accuracies and the testing accuracies of the hybrid model performed on quantum processors was 11.6%. Unlike the quantum simulator results, where some accuracies outperformed the ANN model, quantum processor accuracies were always lower than those of the ANN model. As would be expected, the quantum simulator resulted in higher accuracies than the quantum processor which can be attributed to the large amount of noise present in current qubit technology. Out of the seven classification tasks, the quantum processor performed the best on the SS vs. ¬SS task with a training and testing accuracy of 67.5% and 61.3%, respectively. With the accuracy of both ANN and hybrid models reaching similar limits, we can speculate that these limits may be caused by attributes of the dataset. The complexity of the problem, the relatively small size of the dataset, and insufficient input features may all be factors limiting the resulting accuracy.

Future works

As quantum processors increase their number of qubits and become less susceptible to decoherence, current quantum machine learning techniques will shed their dependence on classical computers and fully utilize the advantages that quantum algorithms offer. These advantages, such as increased efficiency, are accomplished through the utilization of quantum interference and superposition. This may lead to improved machine learning and optimization techniques which could utilize the novel encoding method described in the *Results and Conclusions* subsection of Chapter 2.

Another key factor that will affect HEA machine learning tasks in the future is increasing amounts of data. Continued collection of quality data will greatly improve the performance of machine learning models, making them more accurate and robust. In order to investigate the full potential of a machine learning model to classify HEA phases, a larger dataset should be established.

In addition to the information found in a compound's chemical formula, there are a number of factors that may directly determine the phase of HEA compounds that could increase a model's accuracy when used as input. This additional information includes atomic information, such as atomic radius, and formation enthalpy and entropy. A future work could investigate the effects of including this information in the input features used for model training.

REFERENCES

Arute, Frank, et al. “Quantum Supremacy Using a Programmable Superconducting Processor.” *Nature*, vol. 574, no. 7779, 2019, pp. 505–510., https://doi.org/10.1038/s41586-019-1666-5.

Baritompa, W. P., et al. “Grover's Quantum Algorithm Applied to Global Optimization.” *SIAM Journal on Optimization*, vol. 15, no. 4, 2005, pp. 1170–1184., https://doi.org/10.1137/040605072.

Batra, Kushal, et al. “Quantum Machine Learning Algorithms for Drug Discovery Applications.” *Journal of Chemical Information and Modeling*, vol. 61, no. 6, 2021, pp. 2641–2647., https://doi.org/10.1021/acs.jcim.1c00166.

Benedetti, Marcello, et al. “Parameterized Quantum Circuits as Machine Learning Models.” *Quantum Science and Technology*, vol. 4, no. 4, 2019, p. 043001., https://doi.org/10.1088/2058-9565/ab4eb5.

Biamonte, Jacob, et al. “Quantum Machine Learning.” *Nature*, vol. 549, no. 7671, 2017, pp. 195–202., https://doi.org/10.1038/nature23474.

Brown, T.B., et al. “Language Models Are Few-Shot Learners.” *ArXiv*, 2020, https://arxiv.org/abs/2005.14165. Accessed 2022.

Chen, Shuai, et al. “Machine Learning for High-Entropy Alloys.” *Artificial Intelligence for Materials Science*, 2021, pp. 21–58., https://doi.org/10.1007/978-3-030-68310-8_2.

Cheng, Hongxu, et al. “Review—Corrosion-Resistant High-Entropy Alloy Coatings: A Review.” *Journal of The Electrochemical Society*, vol. 168, no. 11, 2021, p. 111502., https://doi.org/10.1149/1945-7111/ac34d0.

Grover, Lov K. “Quantum Mechanics Helps in Searching for a Needle in a Haystack.” *Physical Review Letters*, vol. 79, no. 2, 1997, pp. 325–328., https://doi.org/10.1103/physrevlett.79.325.

Hales, L., and S. Hallgren. “An Improved Quantum Fourier Transform Algorithm and Applications.” *Proceedings 41st Annual Symposium on Foundations of Computer Science*, 2000, https://doi.org/10.1109/sfcs.2000.892139.

Harrigan, Matthew P., et al. “Quantum Approximate Optimization of Non-Planar Graph Problems on a Planar Superconducting Processor.” *Nature Physics*, vol. 17, no. 3, 2021, pp. 332–336., https://doi.org/10.1038/s41567-020-01105-y.

Harrow, Aram W., et al. “Quantum Algorithm for Linear Systems of Equations.” *Physical Review Letters*, vol. 103, no. 15, 2009, https://doi.org/10.1103/physrevlett.103.150502.

Havlíček, Vojtěch, et al. “Supervised Learning with Quantum-Enhanced Feature Spaces.” *Nature*, vol. 567, no. 7747, 2019, pp. 209–212., https://doi.org/10.1038/s41586-019-0980-2.

Hemphill, M.A., et al. “Fatigue Behavior of al0.5cocrcufeni High Entropy Alloys.” *Acta Materialia*, vol. 60, no. 16, 2012, pp. 5723–5734., https://doi.org/10.1016/j.actamat.2012.06.046.

Hu, Ling, et al. “Quantum Generative Adversarial Learning in a Superconducting Quantum Circuit.” *Science Advances*, vol. 5, no. 1, 2019, https://doi.org/10.1126/sciadv.aav2761.

Huang, Wenjiang, et al. “Machine-Learning Phase Prediction of High-Entropy Alloys.” *Acta Materialia*, vol. 169, 2019, pp. 225–236., https://doi.org/10.1016/j.actamat.2019.03.012.

Jha, Dipendra, et al. “ElemNet: Deep Learning the Chemistry of Materials from Only Elemental Composition.” *Scientific Reports*, vol. 8, no. 1, 2018, https://doi.org/10.1038/s41598-018-35934-y.

Kandala, Abhinav, et al. “Hardware-Efficient Variational Quantum Eigensolver for Small Molecules and Quantum Magnets.” *Nature*, vol. 549, no. 7671, 2017, pp. 242–246., https://doi.org/10.1038/nature23879.

Li, Zhiming, et al. “Metastable High-Entropy Dual-Phase Alloys Overcome the Strength–Ductility Trade-Off.” *Nature*, vol. 534, no. 7606, 2016, pp. 227–230., https://doi.org/10.1038/nature17981.

Medvidović, Matija, and Giuseppe Carleo. “Classical Variational Simulation of the Quantum Approximate Optimization Algorithm.” *Npj Quantum Information*, vol. 7, no. 1, 2021, https://doi.org/10.1038/s41534-021-00440-z.

Miracle, D.B., and O.N. Senkov. “A Critical Review of High Entropy Alloys and Related Concepts.” *Acta Materialia*, vol. 122, 2017, pp. 448–511., https://doi.org/10.1016/j.actamat.2016.08.081.

Mitarai, K., et al. “Quantum Circuit Learning.” *Physical Review A*, vol. 98, no. 3, 2018, https://doi.org/10.1103/physreva.98.032309.

Nielsen, Michael A., and Isaac L. Chuang. “Quantum Computation and Quantum Information.” 2012, https://doi.org/10.1017/cbo9780511976667.

Otto, F., et al. “The Influences of Temperature and Microstructure on the Tensile Properties of a Cocrfemnni High-Entropy Alloy.” *Acta Materialia*, vol. 61, no. 15, 2013, pp. 5743–5755., https://doi.org/10.1016/j.actamat.2013.06.018.

Peruzzo, Alberto, et al. “A Variational Eigenvalue Solver on a Photonic Quantum Processor.” *Nature Communications*, vol. 5, no. 1, 2014, https://doi.org/10.1038/ncomms5213.

Preskill, John. “Quantum Computing in the NISQ Era and Beyond.” *Quantum*, vol. 2, 2018, p. 79., https://doi.org/10.22331/q-2018-08-06-79.

Pérez-Salinas, Adrián, et al. “Data Re-Uploading for a Universal Quantum Classifier.” *Quantum*, vol. 4, 2020, p. 226., https://doi.org/10.22331/q-2020-02-06-226.

Rickman, J. M., et al. “Machine Learning Strategies for High-Entropy Alloys.” *Journal of Applied Physics*, vol. 128, no. 22, 2020, p. 221101., https://doi.org/10.1063/5.0030367.

Senkov, O.N., et al. “Accelerated Exploration of Multi-Principal Element Alloys with Solid Solution Phases.” *Nature Communications*, vol. 6, no. 1, 2015, https://doi.org/10.1038/ncomms7529.

Senkov, Oleg, et al. “Development of a Refractory High Entropy Superalloy.” *Entropy*, vol. 18, no. 3, 2016, p. 102., https://doi.org/10.3390/e18030102.

Shor, Peter W. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer.” *SIAM Journal on Computing*, vol. 26, no. 5, 1997, pp. 1484–1509., https://doi.org/10.1137/s0097539795293172.

Sparkes, Matthew. “A New Quantum Leader?” *New Scientist*, vol. 252, no. 3361, 2021, p. 7., https://doi.org/10.1016/s0262-4079(21)02045-5.

Stein, Samuel A., et al. “Qugan: A Quantum State Fidelity Based Generative Adversarial Network.” *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 2021, https://doi.org/10.1109/qce52317.2021.00023.

Van Dover, R. B., et al. “Discovery of a Useful Thin-Film Dielectric Using a Composition-Spread Approach.” *Nature*, vol. 392, no. 6672, 1998, pp. 162–164., https://doi.org/10.1038/32381.

Wendin, G. “Quantum Information Processing with Superconducting Circuits: A Review.” *Reports on Progress in Physics*, vol. 80, no. 10, 2017, p. 106001., https://doi.org/10.1088/1361-6633/aa7e1a.

Wootters, W. K., and W. H. Zurek. “A Single Quantum Cannot Be Cloned.” *Nature*, vol. 299, no. 5886, 1982, pp. 802–803., https://doi.org/10.1038/299802a0.

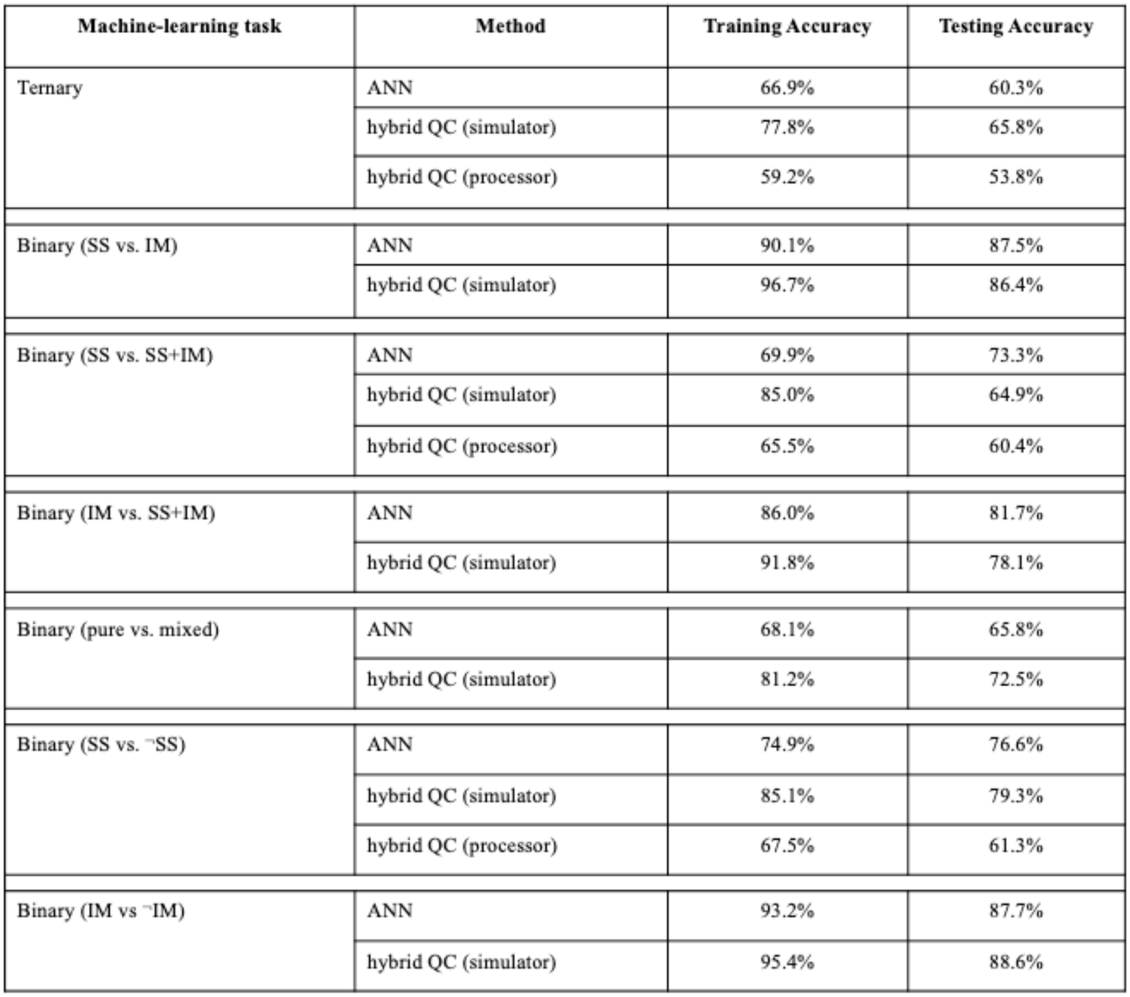
Yang, J., et al. “Support Vector Machines on Noisy Intermediate Scale Quantum Computers.” *ArXiv*, 2019, Accessed 2022.

Zhang, Jie, et al. “Composition Design of High-Entropy Alloys with Deep Sets Learning.” *Npj Computational Materials*, vol. 8, no. 1, 2022, https://doi.org/10.1038/s41524-022-00779-7.

Zoufal, Christa, et al. “Quantum Generative Adversarial Networks for Learning and Loading Random Distributions.” *Npj Quantum Information*, vol. 5, no. 1, 2019, https://doi.org/10.1038/s41534-019-0223-2.

APPENDIX A

TABLES



Machine-learning task	Method	Training Accuracy	Testing Accuracy
Ternary	ANN	66.9%	60.3%
	hybrid QC (simulator)	77.8%	65.8%
	hybrid QC (processor)	59.2%	53.8%
Binary (SS vs. IM)	ANN	90.1%	87.5%
	hybrid QC (simulator)	96.7%	86.4%
Binary (SS vs. SS+IM)	ANN	69.9%	73.3%
	hybrid QC (simulator)	85.0%	64.9%
	hybrid QC (processor)	65.5%	60.4%
Binary (IM vs. SS+IM)	ANN	86.0%	81.7%
	hybrid QC (simulator)	91.8%	78.1%
Binary (pure vs. mixed)	ANN	68.1%	65.8%
	hybrid QC (simulator)	81.2%	72.5%
Binary (SS vs. ¬SS)	ANN	74.9%	76.6%
	hybrid QC (simulator)	85.1%	79.3%
	hybrid QC (processor)	67.5%	61.3%
Binary (IM vs ¬IM)	ANN	93.2%	87.7%
	hybrid QC (simulator)	95.4%	88.6%

Table of accuracies. This table lists all the average accuracies of each model for each of the seven machine-learning tasks. ‘ANN’ corresponds to the artificial neural-network model, ‘hybrid QC (simulator)’ corresponds to the hybrid quantum-classical model run on a quantum simulator, and ‘hybrid QC (processor)’ is the same model run on quantum processors. The average is calculated with the last 20% of the epochs’ accuracies.

APPENDIX B

PERMISSIONS

The work *Quantum machine-learning phase prediction of high-entropy alloys* was reprinted in Chapter 2 of this document with permission of co-author Houlong Zhuang.