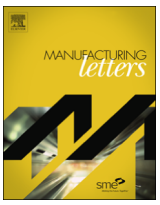
Available online at www.sciencedirect.com

Manufacturing Letters

Manufacturing Letters 00 (2023) 000–000



52nd SME North American Manufacturing Research Conference (NAMRC 52, 2024)

Quantum Machine Learning for Additive Manufacturing Process Monitoring

Eunsik Choi[a,1], Jinhwan Sul[a,1], Jungin E. Kim[a], Sungjin Hong[a], Beatriz Izquierdo Gonzalez[a], Pablo Cembellin[a], Yan Wang[a,*]

[a]*Georgia Institute of Technology, 813 Ferst Drive NW, Atlanta, GA 30047, USA*

Abstract

Machine learning is useful for analyzing and monitoring complex manufacturing processes. However, it has several limitations including the curse-of-dimensionality and lack of training data. In this paper, we propose a quantum machine learning strategy to tackle these challenges. Quantum support vector machine is applied to identify the states of machines in fused filament fabrication process based on acoustic emission data. Quantum convolutional neural network is used to detect spatters in laser powder bed fusion process based on coaxial optical images. Our results show that quantum machine learning can achieve the similar accuracy levels of predictions by classical machine learning counterparts, but with exponentially fewer parameters.

Keywords: Quantum Machine Learning; Quantum Support Vector Machine; Quantum Convolutional Neural Network; Fused Filament Fabrication; Laser Powder Bed Fusion

1. Introduction

Data-driven methods, such as machine learning, are useful to analyze and monitor complex manufacturing processes, where it is very expensive to only rely on physics-based models for predictions. Machine learning has been applied to monitor additive manufacturing processes [21]. Methods such as support vector machine (SVM) [28], hidden Markov model [29], convolutional neural network (CNN) [6, 14], and dictionary learning [17] have been applied.

However, machine learning has several limitations in manufacturing applications. First, machine learning for the prediction of complex systems requires a large amount of training data, whereas data collection in manufacturing is expensive. Second, data collected in manufacturing processes are imbalanced. Manufacturing processes are typically performed in normal conditions, and anomalies occur less frequently. The accuracy of anomaly detection will be affected by the imbalanced training dataset. Third, manufacturing equipment degrades over time. New machine states need to be discovered in real time. Therefore, machine learning models for classification should have the capability of dynamic training and model updates [30]. Fourth, machine learning suffers from the curse-of-dimensionality. As the dimension of model inputs increases, the required training data size increases exponentially. The lack of training data will significantly reduce the accuracy of model predictions.

In this paper, we propose a quantum machine learning strategy to tackle the challenges of curse-of-dimensionality and lack of training data. Quantum machine learning utilizes quantum computer and classical computer simultaneously to process data and perform predictions. Quantum computer takes advantages of quantum mechanics phenomena of superposition and entanglement to perform mass parallel computation [10, 8]. The computation is based on quantum bits, or qubits. Quantum computing can help solve complex problems such as large-scale optimization [27, 25, 13] and engineering simulations [24, 26]. Two quantum machine learning methods are used in this work, including quantum support vector machine (QSVM) and quantum convolutional neural network (QCNN). In QSVM, classi-

* Corresponding author. Tel.: +1-404-894-4714 ; fax: +1-404-894-9342.
E-mail address: yan-wang@gatech.edu (Yan Wang).

[1] Co-first authors

cal data are mapped to the quantum states in the Hilbert space of qubits based on a quantum kernel. Then a hyperplane is estimated to divide the data into two classes. In QCNN, data are similarly mapped to the quantum states. Quantum circuits are constructed as convolutional layers to reduce the dimension of Hilbert space. The quantum gates in the circuit are parameterized. The parameters are trained with classical optimization algorithms.

Here, we demonstrate the feasibility of quantum machine learning approaches for manufacturing process monitoring. Specifically, QSVM is applied to identify the states of machines in fused filament fabrication (FFF) based on acoustic emission (AE) data. QCNN is used to detect spatters in laser powder bed fusion (LPBF) based on coaxial optical images. This is the first research effort of its kind to utilize quantum machine learning in manufacturing applications. Our results show that quantum machine learning can achieve the similar accuracy levels of predictions by classical machine learning counterparts, but with exponentially fewer parameters. Therefore, quantum machine learning has the great potential to tackle the curse-of-dimensionality challenge.

In the remainder of the paper, quantum computing and quantum machine learning are introduced in Section 2. The QSVM method for FFF machine diagnosis is described in Section 3. The QCNN method for spatter detection in LPBF is presented in Section 4. The results and the comparisons with classical machine learning methods are discussed in Section 5. Section 6 provides conclusions.

2. Background

In this section, quantum computing and quantum machine learning are introduced. The basic concepts of quantum computing, superposition and entanglement are introduced in Section 2.1. The fundamentals of QSVM and QCNN are introduced in Section 2.2.

2.1. Quantum Computing

Superposition is the phenomenon that the state of a qubit can be represented as a linear combination of computational basis states. In contrast to a classical bit, where the state is either 0 or 1, a qubit has a state defined as

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{1}$$

where $|0\rangle$ and $|1\rangle$ are computational basis states, and α and β are complex-valued amplitudes.

Entanglement is the phenomenon where the state of one qubit is not independent of another qubit. In other words, a state of the entangled qubit is correlated with another one. Entanglement allows quantum systems to process an exponentially large amount of information at once. For instance, the most general form of a two qubit state can be written as

$$|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle, \tag{2}$$

where α, β, γ, and δ are the amplitudes of basis states $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$, respectively. When a two-qubit system can be factored into two uncoupled one-qubit systems, then the two qubits are not entangled. In this case, the amplitudes satisfy the requirement of $\alpha\delta = \beta\gamma$. In contrast, the amplitudes of the entangled system are four independent complex numbers.

An n-qubit quantum computer encodes information in the 2^n-dimensional Hilbert space. The dimension of the Hilbert space increases exponentially as the number of qubits increases. Therefore, quantum computer has the potential to speed up computation exponentially. In a quantum computer, a quantum circuit is constructed to manipulate the state of the quantum system. Similar to electrical circuits, quantum circuits consist of wires and unitary quantum gates. Each wire represents a qubit which contains information about the system. The amplitudes of the system's state change as the wires pass through each quantum gate. Mathematically, a quantum circuit is defined as

$$|\psi\rangle = U_M \cdots U_2 U_1 |\psi_0\rangle, \tag{3}$$

where each U_i $(i = 1, 2, \ldots, M)$ is a unitary operator. $|\psi_0\rangle$ is the initial state, and $|\psi\rangle$ is the final state. After U_i's are performed sequentially, a measurement operator is applied at the end, which collapses the system's superposition state into a computational basis state.

There are several types of quantum gates which perform different operations in quantum circuits. One gate is the Hadamard gate, which is defined as

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{4}$$

If the initial state of a qubit is $|0\rangle$, the Hadamard gate transforms the state into the equal superposition state $(|0\rangle + |1\rangle)/\sqrt{2}$. In the Bloch sphere, which geometrically represents the state space of a qubit, the Hadamard gate performs a π-rotation about the axis in the $(x+z)/\sqrt{2}$ direction. Another gate is the Pauli gate which performs a π-rotation about an axis in the Bloch sphere. The Pauli-X, Pauli-Y, and Pauli-Z gates rotate the qubit by an angle π about the x-, y-, and z-axes, respectively. The three Pauli gates are defined as

$$P_X = \begin{bmatrix} 0 & 1 \\ 1 & 0, \end{bmatrix}, \; P_Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \; P_Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{5}$$

Any operator consists of a combination of universal quantum gates. One type of universal gates are rotation gates, $R_X(\theta)$, $R_Y(\theta)$, and $R_Z(\theta)$, which correspond to rotations about x-, y-, and z-axes on the Bloch sphere, defined as

$$R_X(\theta) = \begin{bmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{bmatrix}, \tag{6}$$

$$R_Y(\theta) = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}, \tag{7}$$

and

$$R_Z(\theta) = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}, \tag{8}$$

respectively. Another type of universal gates is the CNOT gate, defined as

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{9}$$

The CNOT gate operates on two qubits simultaneously. If the control qubit is 1, the target qubit will flip from 0 to 1 or 1 to 0.

Despite the potential speed up of quantum computing, its scalability is currently limited by the hardware implementation. The performance of noisy intermediate-scale quantum (NISQ) devices is limited on two aspects [20]. First, NISQ devices are prone to decoherence, where quantum states are lost due to interactions with their surroundings. As a result, quantum algorithms cannot be executed for a long period of time. Second, the number of available qubits is limited. With these limitations, quantum algorithms are currently infeasible to solve large-scale problems. In addition, similar to classical computer, error correction mechanisms must be included in quantum computer to improve the reliability. However, due to the limited number of qubits, quantum error correction techniques are difficult to implement in NISQ devices.

2.2. Quantum Machine Learning

Quantum machine learning refers to the machine learning methods that are based on quantum computer, such as quantum kernels and variational quantum algorithms [5]. The general framework of variational quantum algorithms is illustrated in Fig. 1. Classical data $\boldsymbol{x}$ are first preprocessed and mapped to quantum states through quantum kernel $V(\boldsymbol{x})$ and stored in a quantum register. Quantum operations, which are parameterized with rotational angles $\boldsymbol{\theta}$ and denoted as $U(\boldsymbol{\theta})$, are applied to the register. The measurements then give us the distribution of cost function values. Classical optimizers are used to train the parameters to minimize the cost function

$$f(\boldsymbol{x};\boldsymbol{\theta}) = \langle \hat{B} \rangle = \langle \mathbf{0} | V^\dagger(\boldsymbol{x}) U^\dagger(\boldsymbol{\theta}) \hat{B} U(\boldsymbol{\theta}) V(\boldsymbol{x}) | \mathbf{0} \rangle, \tag{10}$$

where the n-qubit register is usually initialized as $|\mathbf{0}\rangle = |0\rangle^{\otimes n}$. $U^\dagger$ is the complex conjugate of U. The cost function is the expectation value of a pre-defined observable $\hat{B}$ for a specific problem.

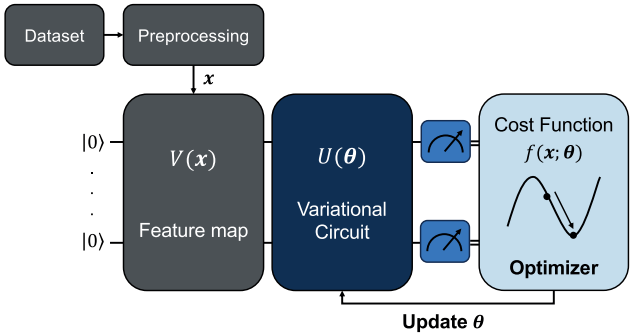


Figure 1. Quantum machine learning with variational quantum circuit, where parameters $\boldsymbol{\theta}$ are trained to minimize cost function $f(\mathbf{x};\boldsymbol{\theta})$.

The parameterized quantum circuit is also referred to as quantum neural network. The training of quantum neural network is to find the optimal rotation angles $\boldsymbol{\theta}$ so that the chance of obtaining the best prediction after measurement is maximized. To reduce the number of qubits in the circuit, QCNN was also proposed [7]. QCNN is illustrated with a simple example in Fig. 2. After the quantum feature mapping, the information is encoded with 8 qubits. In the convolutional layer, the qubits form a ring and the neighboring qubits on the ring are entangled. In the pooling layer, the number of qubits is reduced by half so that the dimension of the feature space is reduced. The convolution and pooling processes can be applied recursively until only one qubit remains as a binary output.

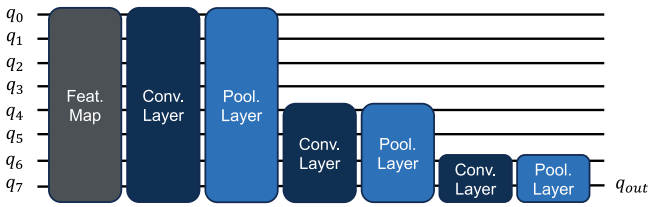


Figure 2. QCNN structure of 8 qubits with convolutional layers.

Another important category of quantum machine learning methods is QSVM. In QSVM, quantum feature map function $V(\boldsymbol{x})$ maps the classical data into the 2^n-dimensional Hilbert space with n qubits. Quantum kernel is defined to quantify the difference between data points $\boldsymbol{x}$ and $\boldsymbol{x'}$, as

$$K(\boldsymbol{x}, \boldsymbol{x'}) = |\langle V(\boldsymbol{x}) | V(\boldsymbol{x'}) \rangle|^2. \tag{11}$$

For hybrid QSVM, quantum kernel is used in the classical SVM to find the optimal hyperplane for classification. The exponentially high-dimensional feature space after quantum feature mapping is necessary to achieve quantum advantage of data encoding, which can improve the classification accuracy of SVM [11].

Other quantum machine learning algorithms have also been developed, such as quantum decision tree [9], quantum clustering [12], quantum Born machines [2], and quantum reinforcement learning [15]. Quantum machine learning methods have been utilized in various applications, including facial expression classification [22], cybersecurity [23], quantum phase de-

tection [16], disease diagnosis [1], drug development [3], and seismic damage detection [4]. This study is the first attempt to use quantum machine learning for manufacturing applications.

3. Quantum Support Vector Machine for FFF Anomaly Detection

Here, QSVM is applied to anomaly detection, where the FFF machine states are identified based on AE sensor data.

3.1. FFF Machine States

The AE sensor data for FFF machine states are obtained from Wu et al. [28]. The four conditions or states of 3D printer include 'Normal', 'Run-out-of-Material', 'Semi-Blocked' (the extruder is semi-blocked), and 'Blocked' (the extruder is completely blocked). The data were collected by attaching an AE sensor to a 3D printer. Each anomaly condition was intentionally introduced and the corresponding datasets were collected. Three important attributes are identified from the raw AE sensor data, including ABS-Energy, RMS, and AE counts. ABS-Energy of an AE hit is the absolute energy calculated as the integration of the output voltage within an AE hit, which indicates its strength. RMS is the root mean square of the AE signal. AE counts are the numbers of pulses over a certain threshold. All the AE sensor data are given as time-series data points. A time window of 500 ms is used to calculate the mean and the standard deviation of each attribute. A total of six features, which include the means and standard deviations of the three attributes, are used for the demonstration. Fig. 3 shows the mean and standard deviation of each time window for all four states and three attributes. For instance, Fig. 3a to Fig. 3d show the RMS of the four states, respectively. Both mean and standard deviation are plotted in the figure, where the center lines are mean values and shaded regions are standard deviations. The signals for the 'Blocked' state have larger mean values than those for the 'Normal' state. The ones for the 'Run-out-of-Material' state have larger standard deviations than those for the 'Normal' state. The total numbers of data points for the 'Normal', 'Semi-Blocked', 'Blocked', and 'Run-out-of-Material' are 302, 371, 221, and 130, respectively. 75% of the data points are used for training, and the remaining 25% are the test data. All the features are min-max normalized.

Fig. 4 shows the distributions of four states for the three attributes. For RMS and ABS-Energy in Fig. 4a and Fig. 4b, the data for the 'Normal' and 'Semi-Blocked' states overlap. For AE counts in Fig. 4c, there are overlaps among the 'Normal', 'Semi-Blocked', and 'Blocked' states.

3.2. Quantum Support Vector Machine

We use the quantum feature map proposed by Havlíček et al. [11], which is implemented with ZZfeaturemap in Qiskit. Fig. 5 shows the two-qubit ZZfeaturemap, which consists of Hadamard gate H, phase gate P, and CNOT gate. ϕ is a classical mapping function such that $\phi(x) = x$ and $\phi(x, y) = (\pi - x)(\pi - y)$. In this implementation, the number of qubits is equal to the number of features. Two cases are demonstrated. For the first case, four features are used, which include the means and standard deviations of RMS and ABS-Energy. Four qubits are used to encode the four features. For the second case, all six features in the data are used. Two additional qubits are utilized to encode the mean and standard deviation of AE counts. The quantum kernel in Eq. (11) is calculated with the quantum feature map. QSVM is then trained with the quantum kernel. The classification accuracy of QSVM is compared with those of two classical SVMs. One classical SVM is trained with the linear kernel, and the other is trained with the radial basis function (RBF) kernel. The same SVM model was used for both QSVM and SVM, where each method involves a different feature mapping kernel.

4. Quantum Convolutional Neural Network for LPBF Monitoring

As the second quantum machine learning method, QCNN is used to identify the spatters in LPBF process based on the optical images captured by a co-axial high-speed camera.

4.1. LPBF Melt Pool Monitoring Dataset

The dataset includes the images of LPBF melt pool captured by the high-speed camera at 1,000 frames per second [18]. Each frame was manually labeled as either 'Spatter' or 'No Spatter' by an expert. As shown in Fig. 6, images of 20×20 pixels focused on the melt pool are cropped from the original 128×48 pixels, and then resized to 4×2 pixels in order to encode them with 8 qubits. The training and test data were randomly sampled from the original data to prevent class imbalance. The training dataset consists of 100 images for each class, whereas the test dataset contains 50 images for each class.

4.2. Quantum Convolutional Neural Network

The quantum circuit of QCNN is shown in Fig. 7. QCNN starts with Hadamard gates, as shown in Fig. 7a, to map the basis states into equal superposition. Subsequently, Pauli-Z gate, as shown in Eq. (5), is applied. The feature mapping circuit in Fig. 7a is performed on two qubits. The gates are applied twice.

After data are encoded with feature mapping, convolution and pooling layers are iteratively applied to the qubits. The convolution and pooling layers are constructed with two-qubit unitary gates to reduce the number of parameters and training time. To obtain the best predictions, the parameters are optimized so that the loss function is minimized.

Quantum convolution circuit consists of rotation gates and CNOT gates, as shown in Fig. 7b. Rotation gates include $R_Y(\theta)$ and $R_Z(\theta)$, which are shown in Eqs. (7) and (8), respectively. The CNOT gate, as shown in Eq. (9), entangles qubits so that fewer qubits are used to maintain information about features.

Quantum pooling circuit is used, as shown in Fig. 7c, to reduce the number of qubits in QCNN while maintaining information, similar to conventional CNN [7]. In quantum machine

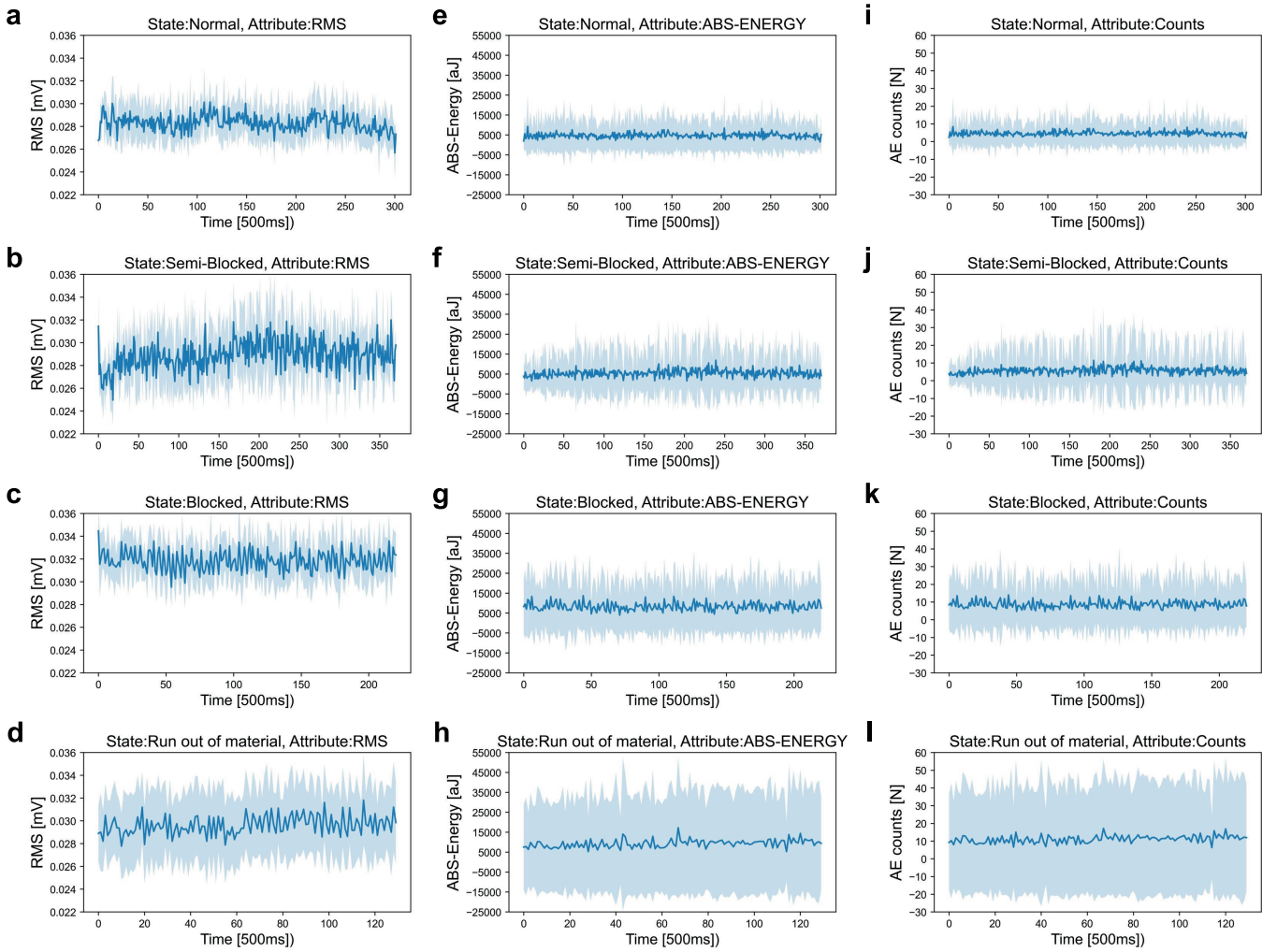


Figure 3. Time series data for different FFF machine states. Center lines are the means of 500 ms time window data and shaded regions are standard deviations. Each column corresponds to a different attribute: (a-d) RMS, (e-h) ABS-Energy, (i-l) AE Counts. Each row corresponds to a different state: (a),(e),(i) Normal, (b),(f),(j) Semi-Blocked, (c),(g),(k) Blocked, (d),(h),(l) Run-out-of-Material.

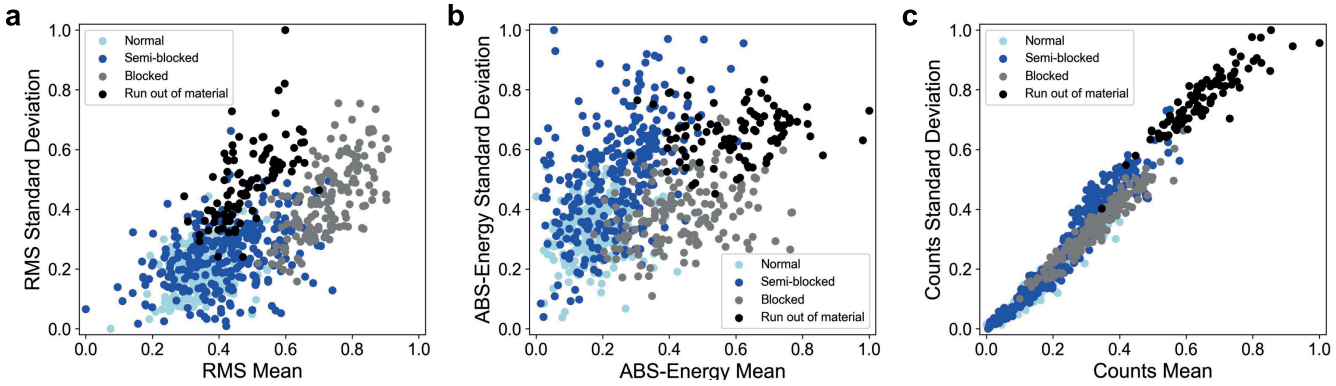


Figure 4. Distributions of features for different FFF machine states. (a) RMS, (b) ABS-Energy, (c) AE counts.

learning, it is essential to reduce the computational cost. The information in two qubits can be combined and encoded with one qubit. In the end, QCNN leaves only one qubit for binary classification problems. More qubits can be used for multi-class classification. The convolution layers of QCNN structure are illustrated in Fig. 2.

The QCNN was trained for 100 epochs to optimize the parameters of the variational quantum circuit. Constrained opti-

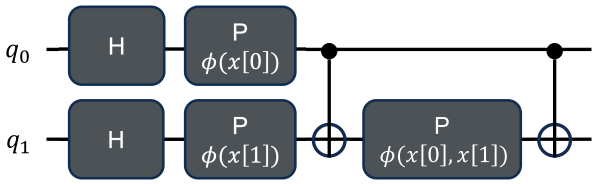


Figure 5. Two qubit ZZfeature map circuit.

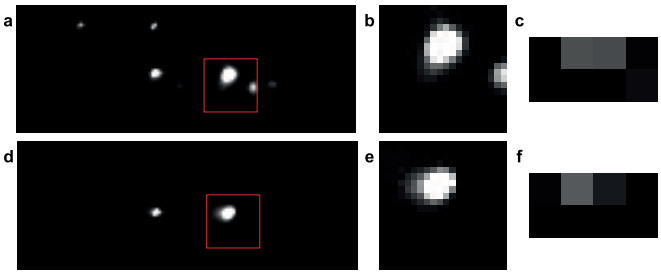


Figure 6. Images for spatter detection in laser powder bed fusion process. Images of 20 × 20 pixels focused on the melt pool are cropped from the original 128 × 48 pixels, and then resized to 4 × 2 pixels for (a-c) 'Spatter' and (d-f) 'No Spatter'. Red boxes were added on top of the images to visualize the cropped regions.

mization by linear approximation (COBYLA) optimizer [19] was used. To compare the performance, conventional neural network (NN) was trained with the same dataset. The image data are converted into 1D vectors for both QCNN and NN. QCNN has three convolution and pooling layers, whereas NN has three hidden linear layers. The number of nodes in each hidden layer of NN is set to 8, 6, and 1 so that NN has the same number of parameters as QCNN. Different learning rates for NN were investigated and the best one was chosen. In addition, the numbers of nodes and layers of NN were varied to further optimize its performance.

5. Results and Discussion

The proposed quantum machine learning methods were applied to detect anomalies in FFF and LPBF processes. The results are shown and compared in this section.

5.1. Results of Quantum Support Vector Machine

The results of QSVM are shown in Table 1. When four features are used, classical SVM with RBF kernel shows the best classification performance of 85.9% accuracy. QSVM has a comparable performance of 85.7% accuracy. When six features are used, QSVM shows 89.5% accuracy, outperforming both classical SVMs with accuracies of 84.0% and 86.7% for linear and RBF kernels, respectively.

Overall, classification accuracy is improved as the number of features increases. It is also observed that the accuracy of QSVM increases by 3.8% from the four-feature to six-feature case. This improvement is significantly larger than those of classical SVMs, which are 0.4% and 0.8% for the linear and RBF kernels, respectively. The results suggest that mapping data into exponentially high-dimensional Hilbert space results in more accurate classification. As the dimensionality of data increases, the QSVM becomes more advantageous in finding the optimal hyperplane for classification.

Table 1. Classification accuracies (%) of FFF machine states with different kernels.

# of features	Kernels		
	Linear	RBF	Quantum
4	83.6	**85.9**	85.7
6	84.0	86.7	**89.5**

5.2. Results of Quantum Convolutional Neural Network

The loss function values of QCNN during the training are plotted in Fig. 8. Overall, the loss was gradually reduced during the training, although some fluctuations are observed. The fluctuations are mostly caused by the probabilistic nature of measurements and single batch training. The predictions of QCNN are shown in Fig. 9. The color scale represents the intensities of the pixels. Convolution layers are used to extract information from images such as contours and brightness. 63 parameters are used for QCNN to classify the images as either 'Spatter' or 'No Spatter'. Fig. 9a and Fig. 9b show two instances of 'Spatter', and Fig. 9c and Fig. 9d show two instances of 'No Spatter'.

Table 2 shows the classification accuracies of QCNN and three NNs, where each NN is trained with a different number of parameters. The accuracy of QCNN is compared with the best accuracies of NNs, which resulted from a learning rate of 10^{-5}. With 8 qubits, QCNN results in both higher training and test accuracies than classical NN with 63 parameters. Furthermore, only the NN with 5,553 parameters outperforms QCNN. This suggests that QCNN can achieve the similar accuracy of classical NN but with exponentially fewer parameters.

Table 2. Comparisions of accuracies for QCNN and three NNs with different number of parameters.

	# of Parameters	Train Accuracy	Test Accuracy
QCNN	**63**	**75.0**	**64.6**
NN	63	60.1	59.2
	1,361	74.3	63.6
	5,553	78.6	69.0

5.3. Discussion

The architecture of QCNN can be represented as an n-level binary tree structure, where each node consists of two convolution circuits and one pooling circuit. Each convolution circuit and pooling circuit has three rotation angles as the parameters. The number of nodes in the binary tree is $N - 1$, where $N = 2^n$ is the number of qubits. Hence, the total number of parameters in QCNN is $9(N - 1)$, which is in the linear order $O(N)$.

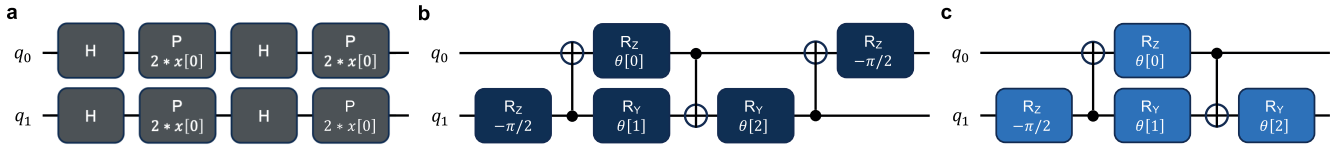


Figure 7. Quantum circuits for QCNN (a) Quantum Zfeature map for two qubits. (b) Quantum convolutional layer detects features between qubits. (c) Quantum pooling layer reduces dimensionality of the QCNN.

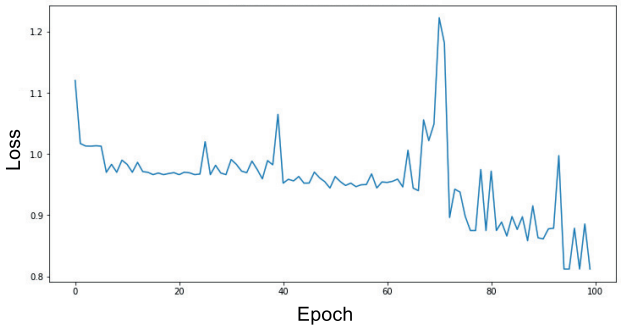


Figure 8. The loss function value of the quantum convolutional neural network during the training.

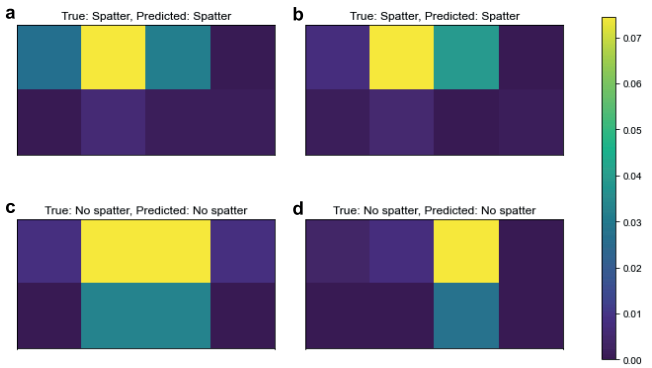


Figure 9. Quantum convolutional neural network predictions. Grayscale images were visualized with color scale. (a),(b) 'Spatter' and (c),(d) 'No Spatter' were predicted by QCNN.

In contrast, the space complexity of neural network is typically $O(N^n)$, where N is the size of inputs and n is the depth of layers. Thus, QCNN is exponentially more efficient than NN, which is also demonstrated in our results.

All data-driven models, including quantum machine learning, are data-specific and difficult for extrapolation. Yet the major challenge in machine learning for manufacturing applications is overfitting of parameters caused by the lack of training data. Our results show that the exponentially fewer parameters in QCNN and exponentially higher-dimensional feature space in QSVM can potentially overcome the overfitting issue.

In quantum machine learning, information is encoded in quantum states. Retrieving the information requires repeated samplings or measurements. In QSVM, the quantum kernel in Eq. (11) is estimated with the sampling complexity of $O(\epsilon^{-2}T^2)$, where ϵ is the sampling error and T is the training dataset size [11]. In QCNN, with E epochs of training, the sampling complexity is $O(\epsilon^{-2}TE)$. Sampling complexity is a unique characteristic of quantum algorithms, in addition to time and space complexities that need to be considered as in classical algorithms.

6. Conclusion

This paper demonstrates the feasibility of quantum machine learning in manufacturing applications. QSVM was applied to identify FFF machine states based on AE sensor data. QSVM is a hybrid strategy which combines quantum kernel with classical SVM. The results show that quantum kernel is more effective than classical kernels to map data to high-dimensional feature space. When data are mapped to the exponentially high-dimensional Hilbert space, the optimal hyperplane for classification can be found more easily.

QCNN was used to detect spatters in LPBF process. For an N-qubit circuit, QCNN requires $O(N)$ parameters. Compared to classical neural networks, QCNN can provide the same level of prediction accuracy, but with exponentially fewer parameters. Fewer parameters can prevent overfitting and the training is less costly with a smaller training dataset. Fewer parameters will also lead to better convergence during the training. The results in this study show the quantum advantage of QCNN over classical neural networks.

Although current quantum computers are limited in the number of qubits and stability, our small-scale demonstrations show that quantum computing has the great potential to tackle the curse-of-dimensionality challenge in solving large-scale problems in manufacturing and other applications. Developing new models and algorithms for application-specific problems also helps overcome the scalability limitation. For instance, quantum machine learning tailored for small and imbalanced training datasets can provide the flexibility for manufacturing applications. The capability of dynamic learning for evolving manufacturing systems is also attractive. The predictions of quantum neural networks are probabilistic. Improving the sampling efficiency for predictions is beneficial. The hybrid quantum machine learning approaches rely on classical optimizers for training. Efficient global optimization algorithms will allow us to build and train better models.

Acknowledgements

Authors would like to thank Dr. Dehao Liu for providing the data of laser powder bed fusion in this study.

References

[1] Abdulsalam, G., Meshoul, S., Shaiba, H., 2023. Explainable heart disease prediction using ensemble-quantum machine learning approach. Intelligent Automation & Soft Computing 36, 761–779.

[2] Benedetti, M., Coyle, B., Fiorentini, M., Lubasch, M., Rosenkranz, M., 2021. Variational inference with a quantum computer. Physical Review Applied 16, 044057.

[3] Bhatia, A.S., Saggi, M.K., Kais, S., 2023. Quantum machine learning predicting adme-tox properties in drug discovery. Journal of Chemical Information and Modeling 63, 6476–6486.

[4] Bhatta, S., Dang, J., 2024. Multiclass seismic damage detection of buildings using quantum convolutional neural network. Computer-Aided Civil and Infrastructure Engineering 39, 406–423.

[5] Cerezo, M., Verdon, G., Huang, H.Y., Cincio, L., Coles, P.J., 2022. Challenges and opportunities in quantum machine learning. Nature Computational Science 2, 567–576.

[6] Choi, E., An, K., Kang, K.T., 2022. Deep-learning-based microfluidic droplet classification for multijet monitoring. ACS Applied Materials & Interfaces 14, 15576–15586.

[7] Cong, I., Choi, S., Lukin, M.D., 2019. Quantum convolutional neural networks. Nature Physics 15, 1273–1278.

[8] Deutsch, D., 1985. Quantum theory, the church–turing principle and the universal quantum computer. Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences 400, 97–117.

[9] Farhi, E., Gutmann, S., 1998. Quantum computation and decision trees. Physical Review A 58, 915.

[10] Feynman, R.P., 1982. Simulating physics with computers. Int. j. Theor. phys 21.

[11] Havlíček, V., Córcoles, A.D., Temme, K., Harrow, A.W., Kandala, A., Chow, J.M., Gambetta, J.M., 2019. Supervised learning with quantum-enhanced feature spaces. Nature 567, 209–212.

[12] Horn, D., Gottlieb, A., 2001. Algorithm for data clustering in pattern recognition problems based on quantum mechanics. Physical Review Letters 88, 018702.

[13] Kim, J.E., Wang, Y., 2023. Quantum approximate bayesian optimization algorithms with two mixers and uncertainty quantification. IEEE Transactions on Quantum Engineering 4, 1–17.

[14] Kim, S.J., Choi, E., Won, D.Y., Han, G., An, K., Kang, K.T., Kim, S., 2023. Accelerated deep-learning-based process monitoring of microfluidic inkjet printing. CIRP Journal of Manufacturing Science and Technology 46, 65–73.

[15] Lamata, L., 2017. Basic protocols in quantum reinforcement learning with superconducting circuits. Scientific Reports 7, 1609.

[16] Liu, Y.J., Smith, A., Knap, M., Pollman, F., 2023. Model-independent learning of quantum phases of matter with quantum convolutional neural networks. Physical Review Letters 130, 220603.

[17] Lu, Y., Wang, Y., Pan, L., 2023. A feature-based physics-constrained active dictionary learning scheme for image-based additive manufacturing process monitoring. Journal of Manufacturing Processes 103, 261–273.

[18] Olson, M., Li, G., Huang, P.Y.H., Liu, D., 2024. Real-time melt pool monitoring and diagnostics in laser powder bed fusion based on single-camera two-wavelength imaging pyrometry and machine learning, in: 2024 ASME IDETC/CIE, ASME. p. (to be submitted).

[19] Powell, M.J., 1994. A direct search optimization method that models the objective and constraint functions by linear interpolation. Springer.

[20] Preskill, J., 2018. Quantum computing in the nisq era and beyond. Quantum 2, 79.

[21] Qin, J., Hu, F., Liu, Y., Witherell, P., Wang, C.C., Rosen, D.W., Simpson, T.W., Lu, Y., Tang, Q., 2022. Research and application of machine learning for additive manufacturing. Additive Manufacturing 52, 102691.

[22] Sathya, T., Sudha, S., 2023. Oqcnn: optimal quantum convolutional neural network for classification of facial expression. Neural Computing and Applications 35, 9017–9033.

[23] Shara, J., 2023. Quantum machine learning and cybersecurity. Quantum 12, 47–56.

[24] Wang, Y., 2013. Simulating stochastic diffusions by quantum walks, in: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers. p. V03BT03A053.

[25] Wang, Y., 2014. Global optimization with quantum walk enhanced grover search, in: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers. p. V02BT03A027.

[26] Wang, Y., 2016. Accelerating stochastic dynamics simulation with continuous-time quantum walks, in: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers. p. V006T09A053.

[27] Wang, Y., Kim, J.E., Suresh, K., 2023. Opportunities and challenges of quantum computing for engineering optimization. Journal of Computing and Information Science in Engineering 23, 060817.

[28] Wu, H., Wang, Y., Yu, Z., 2016. In situ monitoring of fdm machine condition via acoustic emission. The International Journal of Advanced Manufacturing Technology 84, 1483–1495.

[29] Wu, H., Yu, Z., Wang, Y., 2017. Real-time fdm machine condition monitoring and diagnosis based on acoustic emission and hidden semi-markov model. The International Journal of Advanced Manufacturing Technology 90, 2027–2036.

[30] Wuest, T., Weimer, D., Irgens, C., Thoben, K.D., 2016. Machine learning in manufacturing: advantages, challenges, and applications. Production & Manufacturing Research 4, 23–45.