



OPEN Quantum machine learning enhanced laser speckle analysis for precise speed prediction

YiXiong Chen^{2,4}, WeiLu Han^{1,4}, GuangYu Bin¹, ShuiCai Wu¹, Stephen Peter Morgan³ & Shen Sun¹✉

Laser speckle contrast imaging (LSCI) is an optical technique used to assess blood flow perfusion by modeling **changes in speckle intensity**, but it is generally limited to qualitative analysis due to difficulties in absolute quantification. **Three-dimensional convolutional neural networks (3D CNNs) enhance the quantitative performance of LSCI by excelling at extracting spatiotemporal features** from speckle data. However, **excessive downsampling techniques can lead to significant information loss**. To address this, we propose a hybrid quantum–classical 3D CNN framework that leverages variational quantum algorithms (VQAs) to enhance the performance of classical models. The proposed framework employs variational quantum circuits (VQCs) to replace the 3D global pooling layer, enabling the model to **utilize the complete 3D information extracted by the convolutional layers for feature integration**, thereby **enhancing velocity prediction performance**. We perform cross-validation on experimental LSCI speckle data and demonstrate the superiority of the hybrid models over their classical counterparts in terms of prediction accuracy and learning stability. Furthermore, we evaluate the models on an unseen test set and observe that the hybrid models outperform the classical models with up to **14.8% improvement in mean squared error (MSE) and up to 26.1% improvement in mean absolute percentage error (MAPE) evaluation metrics**. Finally, our qualitative analysis shows that the hybrid models offer substantial improvements over classical models in predicting blood flow at both low and high velocities. These results indicate that the hybrid models possess more powerful learning and generalization capabilities.

Keywords Laser speckle contrast imaging, Velocity prediction, Quantum machine learning, Variational quantum algorithms, Hybrid model, Blood flow imaging

Laser speckle contrast imaging (LSCI) has established itself as a valuable tool for **non-invasively assessing microcirculation blood flow perfusion**. Renowned for its broad field of view and the elimination of the need for contrast agents, LSCI offers several distinct advantages¹. Given its versatility, LSCI has been extensively utilized in **both preclinical research and clinical settings, playing a crucial role in determining disease status and evaluating treatment effectiveness**². For example, cerebral blood flow imaging^{3,4}, retinal assessments^{5,6}, trauma and burn evaluations^{7–10}, and the examination of diabetic foot ulcers¹¹ are some of the applications included. LSCI can be categorized into qualitative and quantitative approaches. **Qualitative analysis utilizes speckle contrast to delineate vascular structures, but does not provide detailed blood flow information**¹². Conversely, quantitative analysis utilizes principles of dynamic light scattering to derive metrics like blood flow velocity under specific assumptions and with the use of approximate models. However, the presence of static scatterers, variations in speckle size, exposure time discrepancies, and the inherent complexities of dynamic light scattering methods can compromise the precision of quantitative measurements^{13–15}. These limitations curtail the broader clinical deployment of LSCI for in-depth hemodynamic evaluations¹⁶.

Driven by the growing interest in applying machine learning technologies to medical image processing^{17,18}, researchers have increasingly explored the integration of these techniques into the analysis of LSCI. Lopez-Tiro et al. employed **K-means clustering for automated vessel localization in raw speckle images**¹⁹. Cheng et al. introduced a deep residual learning network with skip connections (DRSNet) for real-time denoising of blood flow images²⁰. Hultman et al. constructed an artificial neural network model using multi-exposure LSCI

¹Department of Biomedical Engineering, Beijing University of Technology, Beijing 100124, China. ²Beijing Science and Technology Project Manager Management Corporation Ltd, Beijing 100083, China. ³University of Nottingham, Nottingham NG7 2RD, UK. ⁴These authors contributed equally: YiXiong Chen and WeiLu Han. ✉email: sunshen@bjut.edu.cn

(MELSCI) images, achieving blood perfusion prediction accuracy comparable to laser doppler flowmetry (LDF)²¹. They subsequently developed an integrated system based on this model, enabling blood perfusion estimation from multiple exposure data at the speed of a single exposure system²². Furthermore, Jain et al. utilized various machine learning techniques to classify speckle data at 10 velocities, finding that quadratic discriminant analysis achieved the highest accuracy at 96%²³. Yu et al. utilized microfluidic phantoms and MELSCI setup to acquire the dataset and built a model, which can extract relative flow velocity changes²⁴. These researches demonstrate the potential of combining LSCI with machine learning to improve accuracy in blood flow assessment.

Three-dimensional convolutional neural networks (3D CNNs) utilize a hierarchical deep learning framework, demonstrating an advantage for handling three-dimensional data, making them widely used in medical image analysis^{25,26}. Unlike their two-dimensional counterparts (2D CNNs), 3D CNNs excel in integrating spatiotemporal information from video data, enabling a more comprehensive understanding of the data^{27,28}. This feature makes 3D CNNs particularly valuable for applications like LSCI, where the dynamic fluctuations of speckle intensity over time and space contain valuable hemodynamic information. Hao et al. applied 3D CNNs to develop an LSCI regression model trained using sequential video frames chronicling speckle variations. This model successfully predicted flow velocity values, achieving a mean squared error (MSE) of 0.33 and a mean absolute percentage error (MAPE) of 0.34²⁹. These results underscore the efficiency of machine learning methodologies in extracting speckle features for determining blood flow velocity, significantly enhancing quantitative analytical methods in LSCI.

3D convolutional neural network models generally consist of two modules: feature extraction and feature integration. First, a series of 3D convolutional layers are applied to the 3D input data for feature extraction. Then, the extracted features are integrated through fully connected layers, producing the prediction output. However, the feature vectors extracted by the convolutional layers are often high-dimensional. Directly using fully connected layers for feature integration significantly increases the number of parameters in these layers, which can lead to overfitting, especially with limited training data. Therefore, 3D CNNs often employ a 3D global pooling layer before the fully connected layers to compress the features, and then input them into the fully connected layers for feature integration. This effectively reduces the number of parameters in the fully connected layers and helps prevent overfitting.

However, 3D global pooling compresses the entire feature map into a single value for each feature channel, thereby significantly sacrificing spatial and temporal information. As a result, the model is restricted to using only channel-wise representations for feature integration, neglecting the vital spatiotemporal correlations initially captured by the convolutional layers. LSCI speckle data is a three-dimensional dataset that combines spatial information from two-dimensional image frames with temporal information from sequential captures. This unique structure provides a comprehensive representation of blood flow dynamics across both time and space. The accurate prediction of blood flow velocity from these LSCI speckle patterns heavily depends on preserving these intricate spatial and temporal relationships. The loss of this vital information due to 3D global pooling directly compromises the model's ability to capture the dynamic nature of blood flow, thus significantly impacting the prediction accuracy.

To address this issue, we explore quantum machine learning technology to refine the 3D CNN framework, aiming to preserve critical information without imposing additional computational demands. Quantum machine learning (QML) is an emerging interdisciplinary field that combines quantum computing with machine learning algorithms³⁰. The premise of QML is to utilize the principles of quantum mechanics, such as superposition and entanglement, to process information in ways that classical computers cannot and enhance models for machine learning tasks. In recent years, QML has already achieved continuous progress and demonstrated quantum advantages in various classical machine learning problems, including support vector machines³¹, clustering³², and principal component analysis (PCA)³³.

In this paper, we propose a novel methodology for predicting phantom velocity based on variational quantum algorithms (VQAs)^{34–36} in quantum machine learning. VQAs represent a class of hybrid quantum–classical algorithms that optimize a parameterized quantum circuit by using classical computers. They are particularly suitable for the current noisy intermediate-scale quantum (NISQ) era due to their adaptability to the limitations of noisy and imperfect quantum hardware³⁴. As a result, VQAs have been extensively explored in quantum machine learning tasks. Kulkarni et al. conducted a study on pneumonia detection through chest X-rays, where they developed a hybrid neural network model. This model innovated by integrating a quantum layer in place of the traditional dense layer within a classical convolutional neural network. The resulting hybrid model demonstrated superior prediction accuracy compared to its classical counterpart³⁷. Shi et al. enhanced quantum neurons and employed them to develop a quantum-inspired convolutional neural network (QICNN). This approach resulted in QICNN outperforming traditional CNNs in terms of accuracy and convergence speed when evaluated on the MNIST dataset³⁸.

Our proposed quantum–classical framework leverages the power of quantum computing to address the information loss problem in the classical 3D CNN framework. Specifically, we replace the 3D global pooling layer in the classical models with a variational quantum circuit (VQC)^{35,39–41} which possesses remarkable data storage capacity. In this way, our framework utilizes the full feature maps from the 3D convolutional layers for integration, without losing any spatial and temporal information, thus enhancing the model's predictive accuracy. Moreover, unlike classical deep learning models, our framework does not suffer from the issues of excessive parameters and overfitting, even without the global pooling layer. This is attributed to the robust data encoding and expressive power of the VQCs utilized in our approach. In fact, our quantum framework requires fewer parameters than its classical counterpart.

In summary, we introduce a novel LSCI velocities prediction framework based on VQAs. This framework eliminates the need for global pooling prior to the regression layer, allowing models to effectively utilize more

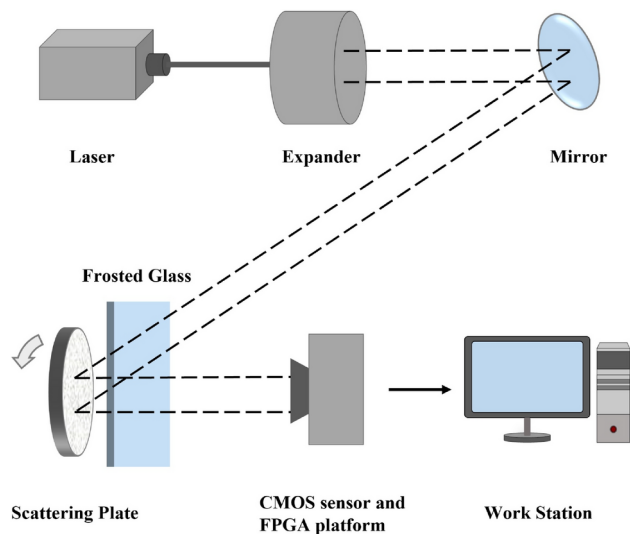


Fig. 1. Experimental Setup. The laser beam is expanded and reflected before reaching the scattering plate, then the CMOS sensor captures the scattered light and sends the data to the FPGA platform and work station for processing.

	1	2	3	4	5	6	7	8	9	10
Angular speed (rad/s)	0.047	0.105	0.210	0.317	0.424	0.530	0.636	0.741	0.848	0.952

Table 1. Angular speed.

spatial and temporal information for feature integration with fewer parameters. To the best of our knowledge, this is the **first QML framework proposed for the task of predicting phantom velocity in the field of LSCI.**

Materials and methods
Dataset

Experimental equipment

This study employs a dataset derived from experiments conducted with a specialized setup that includes a **tissue phantom and a full-field LSCI system.** The tissue phantom is engineered to replicate the light scattering characteristics of human tissue, thereby enabling the simulation of blood flow at various controlled speeds. The full-field LSCI system is used to illuminate the phantom and captures scattered light, and data are processed to obtain **speckle pixel value matrices.** These matrices are then used as the training data. The experiment setup and the position of each component are shown in Fig. 1.

The phantom setup consists of a servo motor, a scattering plate designed to mimic light scattering in tissue, and a frosted glass. The servo motor precisely controls the rotation of the scattering plate, simulating the movement of red blood cells under the skin at various velocities. The **scattering plate acts as a proxy for red blood cells in tissue, while the frosted glass scatters light similarly to human skin.** During the experiment, the LSCI system illuminates the phantom and captures the resulting speckle pattern generated by the interaction of light with the moving scattering plate. This approach offers the advantage of precisely controlled rotation speeds, enabling the simulation of blood flow across a range of velocities relevant to human tissue.

The LSCI system setup consists of four main components: a laser unit emitting a 532 nm, 50 mW green laser beam (OXXIUS S.A. 532 S-50-COL-PP) for illumination, a high-resolution megapixel CMOS sensor (MT9M413) for image capture, a DDR3 SDRAM memory unit for data storage supporting high data transfer rates necessary for processing high-resolution images, and an FPGA platform (Virtex 6 XC6VLXC240T) as the processing unit.

Data acquisition and processing

The motor's angular velocity varied from 0.047 to 0.952 rad/s in increments of 0.106 rad/s, the specific angular velocity values are shown in Table. 1. We captured 1024 consecutive image frames at 320 × 32 pixel resolution for each velocity. This resulted in **320 × 32 × 1024 pixels per measurement.** To ensure reliable data, each velocity setting was repeated 16 times, resulting in a total of 160 data points.

We need to select an appropriate **number of video frames for model training.** A small number of frames does not provide sufficient information and having more frames as the input is helpful for capturing more information. However, due to the computational resource limitation, we select **15 video frames for training.**

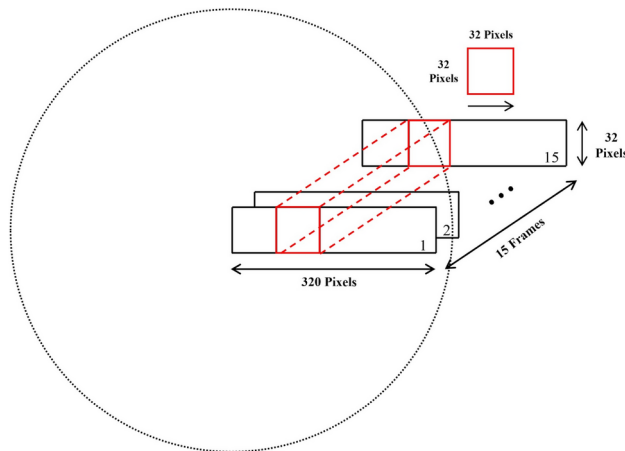


Fig. 2. Data augmentation method.

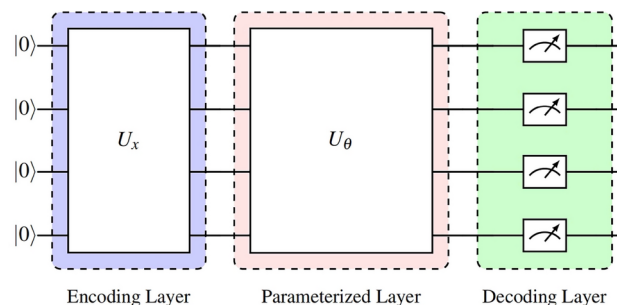


Fig. 3. The architecture of the VQC. The classical input x is encoded into a quantum state by the **encoding layer implemented by the operator U_x** . Subsequently, this encoded quantum state is transformed by the parameterized layer implemented by the operator U_θ where $\theta \in [0, 2\pi)$ are learnable parameters. Finally, the decoding layer retrieves the classical output by performing quantum measurements.

To obtain more data, we perform data augmentation spatially, the specific process is as shown in Fig. 2. For each data, from the 32nd pixel to the 320th pixel, a square window of size 32×32 is used, sliding with a step size of 32. Each square window represents a different radius, where the radius multiplied by the preceding angular velocity yields the label - linear velocity. This approach expands each original data point with 320×32 pixels into 9 new augmented data points each of which contains 32×32 pixels. In this way, we obtain a total number of $160 \times 9 = 1440$ data points, each with a size of $15 \times 32 \times 32$. The label is obtained by multiplying the radius corresponding to each data by the corresponding angular velocity.

Quantum-classical hybrid neural network

Variational quantum algorithms

Variational quantum algorithms (VQAs) are a class of quantum algorithms that use a hybrid quantum-classical approach to solve computational problems. In contrast to traditional quantum algorithms that heavily depend on precise and exact quantum operations, VQAs employ Variational Quantum Circuits (VQCs) as the central component, which provide a more flexible and adaptable approach to quantum computation, particularly in the context of noisy or imperfect quantum systems. A VQC consists of quantum gates with adjustable parameters and can be trained through classical optimization algorithms with the objective of minimizing a loss function. This training process involves the continuous adjustment of the parameters of quantum gates to enhance the model performance⁴¹. As depicted in Fig. 3, a VQC is typically comprised of three distinct modules: the encoding layer, the parameterized layer, and the decoding layer.

The encoding layer is responsible for quantum encoding, employing quantum gate operations to leverage quantum parallelism and the inherent efficiency of quantum systems. This allows for transforming classical input data into a quantum state, marking an initial step for further quantum computations. The quantum encoding process can be represented as follows:

$$|\psi_x\rangle = U_x|0\rangle \quad (1)$$

where the input feature vector x is encoded into a quantum state $|\psi_x\rangle$ via the encoding operator U_x . Encoding methods include basis encoding, amplitude encoding, and angle encoding^{42,43}. Basis encoding uses the basis state

of a quantum system to represent information. Amplitude encoding encodes classical data onto the amplitudes of qubits. One advantage is that it can represent a large amount of classical data using a small number of qubits. Angle encoding represents information through the rotation angles of qubits. In some cases, it can be highly efficient, but it requires a large number of qubits and quantum gates to represent complex data. In this work, we employ amplitude encoding for our proposed framework. The details of this encoding technique will be introduced in the next subsection.

Following the encoding process, the parameterized layer operates on the encoded quantum state using a series of quantum gates (e.g., rotation gates) that are adjustable and controlled by a set of parameters. By adjusting these parameters, the circuit can explore a wide range of quantum states and transformations, enabling it to adapt to different computational tasks. The parameterized layer plays a crucial role in harnessing the power of variational techniques in quantum computation. It allows the VQC to learn and adapt from data, making it a versatile tool for solving complex problems. Let us denote all quantum operations in the parameterized layer by U_θ for simplicity, where θ represents all adjustable parameters. After applying this layer onto the initial encoded state $|\psi_x\rangle$, the resulting quantum state is:

$$|\psi_{x,\theta}\rangle = U_\theta|\psi_x\rangle \quad (2)$$

Finally, the decoding layer extract classical information by transforming the quantum state $|\psi_{x,\theta}\rangle$ into a classical vector $f(x, \theta)$ via a map:

$$\mathcal{M}: |x, \theta\rangle \rightarrow f(x, \theta) \quad (3)$$

The vector $f(x, \theta)$ represents the expected value of certain local observables \hat{O}

$$f(x, \theta) = \langle \psi_{x,\theta} | \hat{O} | \psi_{x,\theta} \rangle \quad (4)$$

which can be estimated from repeated quantum measurements. As we can see, the output $f(x, \theta)$ contains the information of the input x and the adjustable parameters θ in the VQC, and this classical data can be processed by subsequent classical layers in the model.

Amplitude encoding

Amplitude encoding is a technique utilized in quantum computing to represent classical data by assigning amplitudes to quantum states. In this approach, a normalized classical vector x with N dimensions is encoded into the amplitudes of an n -qubit quantum state, where n is determined by $n = \lceil \log_2 N \rceil$. The encoded state $|x\rangle$ is obtained by summing over all elements of the classical vector x , each multiplied by the corresponding computational basis state $|i\rangle$:

$$|x\rangle = \sum_i^N x_i |i\rangle \quad (5)$$

To ensure a valid quantum state, the classical vector x must satisfy the normalization condition $|x|^2 = 1$, representing the amplitudes of the quantum state.

Amplitude encoding offers an efficient method for processing large amounts of data in quantum computers, leveraging the principles of superposition and interference. By encoding high-dimensional data vectors into quantum states, amplitude encoding grants quantum algorithms the capacity to manipulate complex data in potentially more effective manners than classical algorithms.

3D global pooling

3D global pooling is a technique employed in deep learning, particularly in 3D CNNs, to aggregate feature maps across the spatial and temporal dimensions. Specifically, this technique compresses an input feature map of dimensions $D \times H \times W \times C$ into a vector of size $1 \times 1 \times 1 \times C$, where D , H , W , and C represent depth, height, width, and number of channels, respectively. By doing this, it reduces the computational complexity and memory requirements of the network, enhancing efficiency for tasks involving volumetric data, such as video analysis or 3D medical imaging.

Two common forms of 3D global pooling are 3D global average pooling (GAP) and 3D global maximum pooling (GMP), as shown in Fig. 4. In 3D GMP, the average value of all elements within the feature map is computed, resulting in a single value per feature channel. On the other hand, 3D GMP selects the maximum value from each feature map, capturing the most dominant feature across the entire spatial and temporal extent. While these operations effectively reduce dimensionality, they often result in significant information loss by discarding detailed spatial-temporal relationships and fine-grained features that could be crucial for tasks requiring precise temporal and spatial understanding, such as video analysis and motion recognition.

3D CNN framework based on variational quantum algorithms

A typical 3D CNN framework, as shown in Fig. 5a, consists of convolutional layers and fully connected layers. The convolutional layers are used for feature extraction, while the fully connected layers are used for feature

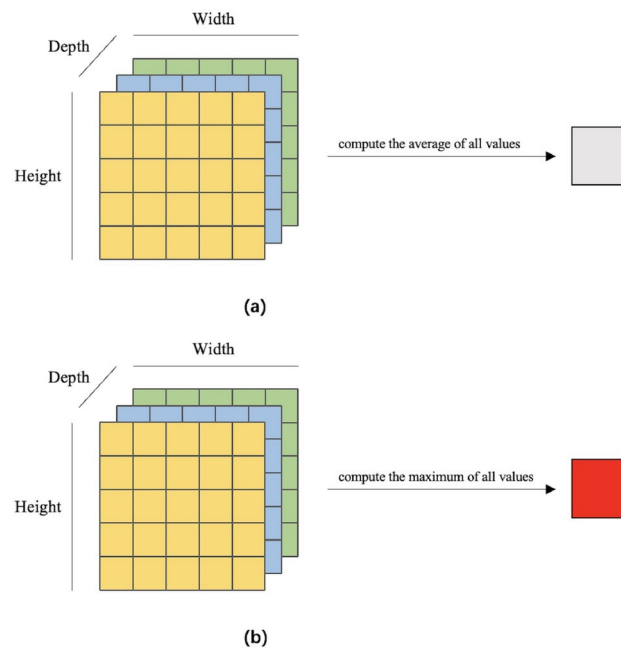


Fig. 4. Examples of two type of 3D global pooling operations. The input is a 3D data with a size of depth \times height \times width, where depth = 3, height = 5, and width = 5. The output is a scalar value. **(a)** 3D global average pooling. **(b)** 3D global maximum pooling.

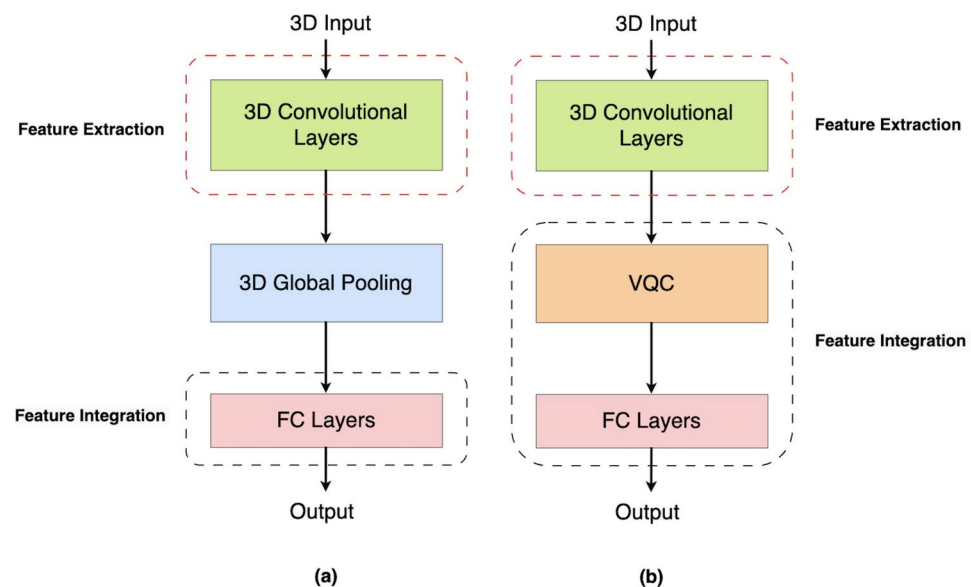


Fig. 5. Comparison of 3D CNN framework and 3D QCNN framework. The red dashed box represents the feature extraction module, while the black dashed box denotes the feature integration module. **(a)** 3D CNN framework. **(b)** 3D QCNN framework.

integration and output prediction. 3D convolutional neural network models often introduce a 3D global pooling layer between the feature extraction module and the feature integration module. **The purpose is to compress the features extracted by the convolutional layers, reduce the number of parameters in the fully connected layers, and avoid model overfitting.** However, as discussed earlier, the 3D global pooling operation **loses a significant amount of spatial and temporal information**, allowing only channel information to enter the feature integration module. This loss can negatively impact the model's predictive accuracy, particularly for tasks such as blood flow velocity prediction, which rely on capturing both spatial and temporal information.

To address this issue, we propose a hybrid quantum-classical 3D CNN framework based on variational quantum algorithms, as shown in Fig. 5b. We refer to this framework as 3D QCNN framework. It is a departure

from the classical framework introduced above since it is based on VQAs. In the proposed framework, we substitute the global pooling layer in the classical network with a quantum layer implemented by a VQC. This allows the full sets of features extracted by the convolutional layers to be input into the feature integration module without any loss, thereby improving the model's predictive performance. In this new framework, the VQC is also part of the feature integration module. It's worth noting that since the features extracted by the convolutional layers are often large, amplitude encoding is used in the encoding layer of the VQC to handle this high-dimensional information more efficiently. This approach allows the VQC to operate with only a small number of qubits while managing the high-dimensional output from the feature extraction module. The structure of the parameterized layer in the VQC is flexible and can be chosen arbitrarily. For the decoding layer, the Pauli-Z operator is generally chosen as the observable.

Quantum LSCI velocities prediction models

We select two classic models to validate our proposed method. The first model is the classical 3D CNN model proposed in the work²⁹ for LSCI velocities prediction. We refer to this classical model as CNN-LSCI and show its architecture in Fig. 6a. The feature extraction module of this network employs three convolutional layers with kernel sizes of $3 \times 3 \times 5$, $3 \times 3 \times 5$, and $3 \times 3 \times 3$, respectively. Following each convolutional layer, batch normalization (BN) and ReLU activation functions are applied to enhance training stability. To reduce the computational complexity, two max pooling layers with a pooling size of $2 \times 2 \times 2$ are placed after the first two convolutional layers respectively. The second model is a 3D residual neural network (ResNet) which we refer to as ResNet-LSCI. This model is a state-of-the-art deep learning architecture designed for 3D computer vision tasks, particularly in the fields of video analysis, 3D object recognition, and medical imaging. Fig. 7a illustrates the structure of the ResNet-LSCI model. For feature extraction, it uses a convolutional layer with a $2 \times 3 \times 3$ kernel size, followed by a max pooling layer with a $2 \times 2 \times 2$ kernel size, and two residual blocks. As shown in Fig. 8, each residual block consists of two convolutional layers with a shortcut connection that bypasses these layers. The shortcut connection, also known as a skip connection or residual connection, allows the input to be added to the output of the two convolutional layers. Both models CNN-LSCI and ResNet-LSCI use a GAP layer after the feature extraction module to downsample the feature maps obtained by the last convolutional layer and output a feature vector of fixed length. Finally, the fully connected layer learns higher-level representations from this fixed-length vector and predicts a single phantom velocity value. As previously mentioned, the global pooling operation helps reduce the parameter count in the fully connected layer, but it results in a significant loss of information, potentially degrading network performance. For example, in the CNN-LSCI model, the global pooling layer reduces the size of the feature maps outputted by the last convolutional layer from $128 \times 5 \times 9 \times 8$ to 128, leading to the loss of 45952 features which contain rich spatial and temporal information.

To tackle this problem, we apply our proposed method to CNN-LSCI and ResNet-LSCI models. Specifically, we substitute the global pooling layer in these two models with a quantum layer implemented by a VQC. This results in two hybrid quantum-classical neural networks, which we refer to as QCNN-LSCI and QResNet-LSCI respectively. Fig. 6b illustrates the architecture of QCNN-LSCI, and Fig. 7b shows the structure of QResNet-LSCI. Due to the removal of the global pooling layer, the input sizes for the VQC in these two models are 46080 and 51840 respectively. Considering the substantially large sizes of these feature vectors, amplitude encoding is utilized to efficiently encode the data into quantum states, as other encodings would demand impractical qubit resources (e.g., 51840 qubits required by angle encoding for the case of QResNet-LSCI) for NISQ

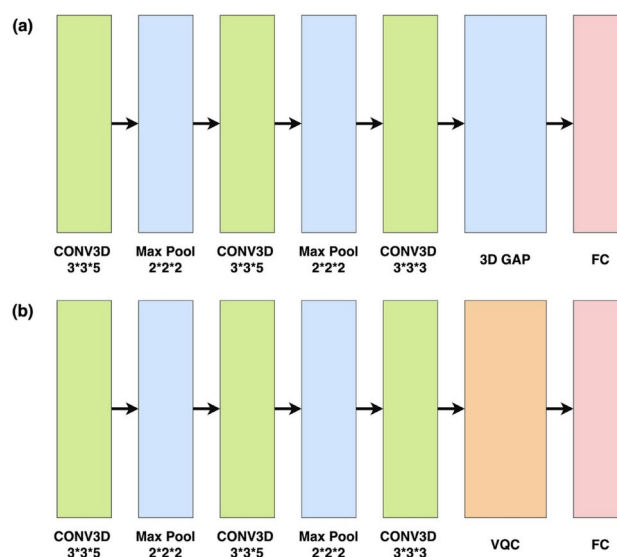


Fig. 6. Comparison of CNN-LSCI and QCNN-LSCI Architectures. CONV3D and Max Pool represent a 3D convolutional layer and a max pooling layer, respectively. 3D GAP and FC denote a 3D global average pooling layer and a fully connected layer, respectively. (a) CNN-LSCI. (b) QCNN-LSCI.

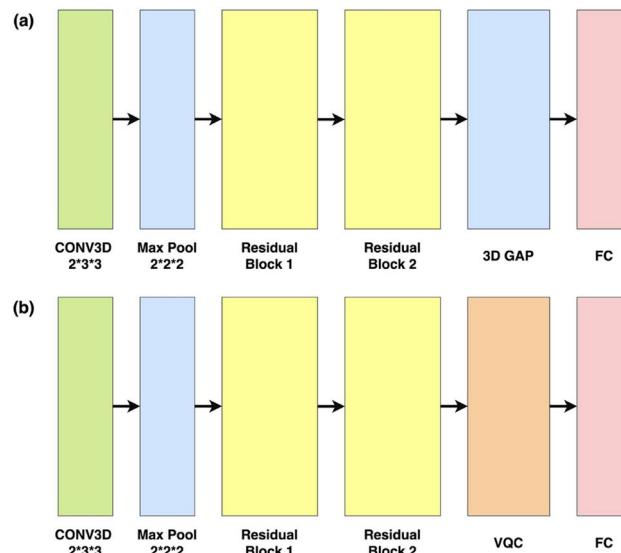


Fig. 7. Comparison of ResNet-LSCI and QResNet-LSCI Architectures. **(a)** ResNet-LSCI. **(b)** QResNet-LSCI.

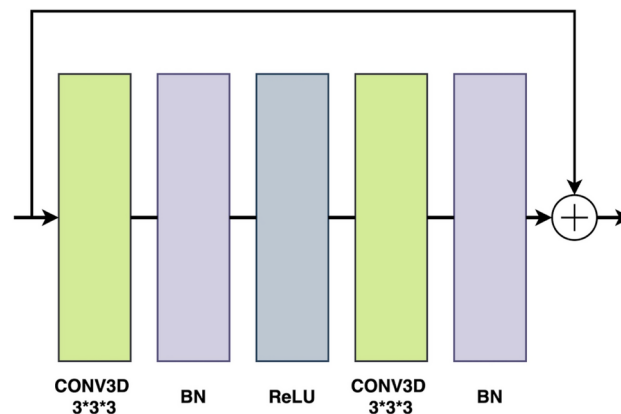


Fig. 8. Residual block. BN denotes a batch normalization layer and ReLU represents the Rectified Linear Unit activation function.

devices. Although the VQCs in QCNN-LSCI and QResNet-LSCI have different input sizes, they both require $\lceil \log_2 46080 \rceil = \lceil \log_2 51840 \rceil = 16$ qubits. Following the data encoding process, a R_x gate is applied to each qubit to transform the encoded quantum state. The R_x gate is formulated as

$$R_x(\theta) = \begin{pmatrix} \cos \frac{\theta}{2} & -i \sin \frac{\theta}{2} \\ -i \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix} \quad (6)$$

where the trainable parameter $\theta \in [0, 2\pi)$ represents the angle of clockwise rotation around the x-axis of the Bloch sphere⁴⁴. Finally, measurements are performed on the Pauli-Z operators, and the corresponding expectation values are used as the output of the VQC. The structure of the VQC in both QCNN-LSCI and QResNet-LSCI is illustrated in Fig. 9.

Our quantum models have three main advantages. Firstly, due to the absence of global pooling, the feature maps outputted by the last convolutional layer are fully preserved for the following feature integration process. This is highly beneficial for improving model prediction accuracy. Secondly, our models have fewer parameters. By employing amplitude encoding in the VQC, the number of qubits required is exponentially smaller than the size of the input data. This is attributed to amplitude encoding leveraging the principles of superposition and interference in quantum mechanics. Additionally, the R_x gates used in the VQC act on only one qubit and have a single parameter. This locality ensures a small parameter count for the VQC. In contrast, in classical fully connected neural networks, each input neuron is connected to all output neurons, resulting in an extensive amount of parameters. Specifically, the parameter count in our quantum models after the last convolutional layer

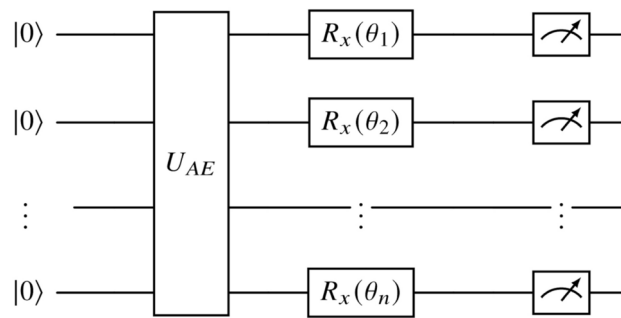


Fig. 9. The general architecture of the VQC used for both QCNN-LSCI and QResNet-LSCI models. The quantum circuit contains n qubits. U_{AE} represents the amplitude encoding operator which encodes the input classical data into a quantum state. $R_x(\theta_i)$ is a single rotation gate around the X-axis of the Bloch sphere by an angle of θ_i , where $i = 1, \dots, n$. For both QCNN-LSCI and QResNet-LSCI, $n = 16$.

is $16 + 16 \times 1 + 1 = 33$. Here, 16 refers to the number of quantum parameters in the VQC, and $16 \times 1 + 1 = 17$ represents the number of classical parameter in the fully connected layer after the VQC, where 16×1 and 1 correspond to the weights and bias respectively. In contrast, the classical models have a parameter count of $128 \times 1 + 1$ after the last convolutional layer, which is significantly higher than that of quantum models. Thirdly, by virtue of searching for the optimal quantum state within a vast Hilbert space, the VQC provides greater expressive power and learning capacity compared to classical network layers^{39,45,46}. In Appendix 1, we present a derivation of the mathematical expression of the VQC in our model, which illustrates a more complex non-linear representation than that of a classical fully connected layer.

Experiments and results

Experimental environment

In the experiment, we construct classical models CNN-LSCI and ResNet-LSCI, as well as their quantum counterparts QCNN-LSCI and QResNet-LSCI. The performances of these classical and quantum models are then compared on the velocity prediction task. We conduct the experiments on a computer equipped with an Intel Core i7 processor, 32GB of RAM, and an NVIDIA GeForce RTX 2060 GPU. We implement quantum layers using PennyLane⁴⁷ and classical layers with PyTorch⁴⁸. PennyLane is a quantum machine learning open-source library which enables users to easily build, optimize and deploy quantum-classical applications. We utilize PennyLane's *default.qubit* simulator, which is a standard state-vector simulator, as the backend for executing quantum circuits in our hybrid models. This simulator is compatible with both backpropagation^{49,50} and adjoint⁵¹ differentiation techniques, which are essential for computing quantum gradients in the VQC.

Cross-validation results

We employed the k-fold cross-validation method to evaluate the performances of both quantum and classical models. This approach involves dividing the dataset into k equal-sized subsets (i.e., folds) and iteratively training the model on $k-1$ folds while using the remaining fold for validation. This process is repeated k times, ensuring a comprehensive assessment and reducing the impact of uneven data distribution. We randomly extract 20% of our experimental data as the test set and perform k-fold cross-validation on the remaining data with k set to 5. Considering we have a total of 1440 samples, the dataset is split into a training set of 922 samples, a validation set of 230 samples, and a test set of 288 samples. At each fold, both quantum and classical models are trained for 100 epochs using a batch size of 16 and the Adam optimizer with a learning rate of 0.001. The MSE is employed as the loss function:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (A_i - F_i)^2 \quad (7)$$

where N indicates the total number of samples, A_i is the actual value, and F_i is the predicted value. After each epoch of training, the model performance is evaluated on the validation set. The evaluation metrics include the MSE loss, and MAPE which is defined as follows:

$$\text{MAPE} = \frac{1}{N} \sum_{i=1}^N \left| \frac{A_i - F_i}{A_i} \right| \quad (8)$$

MAPE is a metric commonly used to evaluate the performance of predictive models. Unlike MSE, which measures absolute errors, MAPE offers the advantage of scale-independent error comparison, making it easier to interpret across different datasets.

The 5-fold cross-validation results for QCNN-LSCI and CNN-LSCI are shown in Table 2, while those for QResNet-LSCI and ResNet-LSCI are presented in Table 3. It is evident that both quantum models consistently

Model	Train loss	Val loss	Val MAPE	Test loss	Test MAPE
QCNN-LSCI	0.4283 ± 0.0865	0.9059 ± 0.1699	$19.67\% \pm 1.89\%$	0.7420 ± 0.0822	$22.30\% \pm 1.24\%$
CNN-LSCI	1.1118 ± 0.1330	0.9556 ± 0.1831	$25.07\% \pm 1.86\%$	0.8704 ± 0.1227	$30.16\% \pm 2.71\%$

Table 2. Results of cross-validation for QCNN-LSCI and CNN-LSCI . The loss is calculated by MSE.

Model	Train loss	Val loss	Val MAPE	Test loss	Test MAPE
QResNet-LSCI	0.6646 ± 0.1946	0.9986 ± 0.1455	$19.08\% \pm 0.86\%$	0.8909 ± 0.1335	$19.97\% \pm 1.21\%$
ResNet-LSCI	0.6883 ± 0.2116	1.1186 ± 0.1546	$21.86\% \pm 1.39\%$	0.9447 ± 0.1469	$24.50\% \pm 1.39\%$

Table 3. Results of cross-validation for QResNet-LSCI and ResNet-LSCI. The loss is calculated by MSE.

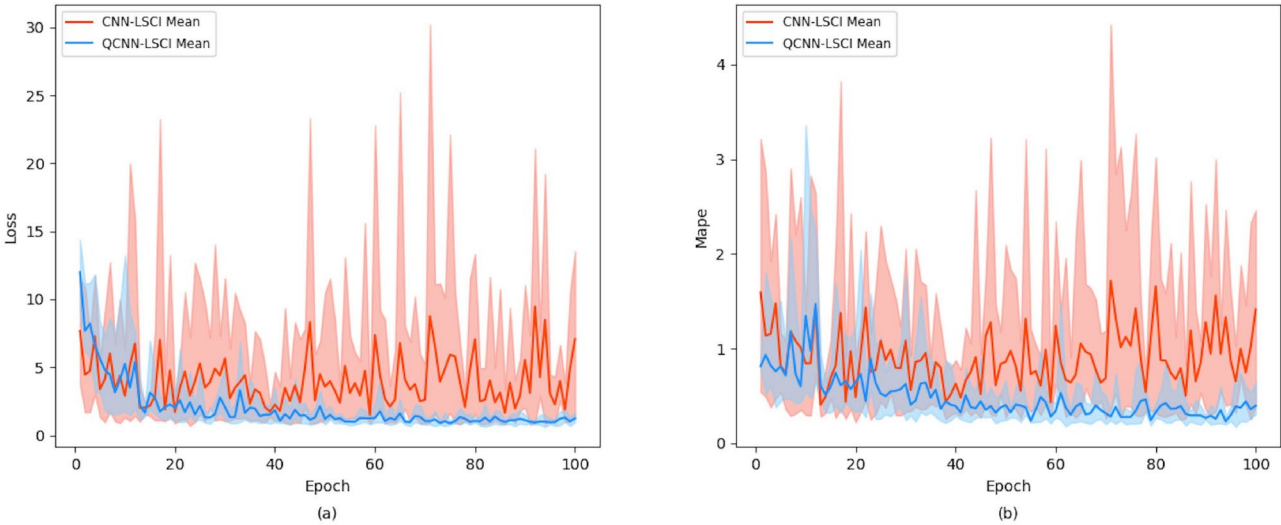


Fig. 10. Comparison of learning curves between QCNN-LSCI and CNN-LSCI models on the validation set. The blue and red curves depict the mean values of the evaluation metrics for the QCNN-LSCI and CNN-LSCI models, respectively. The shadow areas of the corresponding color denote the fluctuation which indicates the maximum and minimum values of the metric. a) Learning curves of loss. b) Learning curves of MAPE.

outperform their classical counterparts respectively across all performance metrics. As shown in the table, the train loss for the CNN-LSCI is 1.1118 ± 0.1330 , while the QCNN-LSCI achieves a lower and more stable train loss of 0.4283 ± 0.0865 , indicating a better fit during the training phase. Similarly, ResNet-LSCI shows a train loss of 0.6883 ± 0.2116 , while QResNet-LSCI achieves a better train loss of 0.6646 ± 0.1946 . The same trend is observed in the validation phase, where QCNN-LSCI demonstrates a relative enhancement of 5.2% and 21.5% respectively compared to CNN-LSCI in terms of the average validation loss and validation MAPE. Meanwhile, QResNet-LSCI shows improvements of 10.7% and 12.7% over a in validation loss and validation MAPE respectively. We next evaluate the generalization ability of the models on unseen data using the test set. QCNN-LSCI outperforms CNN-LSCI by a substantial margin, obtaining 14.8% and 26.1% relative improvement in average test loss and test MAPE respectively. Similarly, QResNet-LSCI achieves 5.7% and 18.5% relative improvement over ResNet-LSCI in test loss and test MAPE. These findings highlight the enhanced generalization capability of our quantum models on unseen data. In addition, the quantum models have exhibited greater model robustness and stability. Compared to their classical counterparts, QCNN-LSCI shows a 33% reduction in the standard deviation of test loss and a 54% reduction in test MAPE, while QResNet-LSCI demonstrates reductions of 9% and 13% in these respective metrics.

Furthermore, the validation loss and MAPE learning curves comparing QCNN-LSCI with CNN-LSCI and QResNet-LSCI with ResNet-LSCI are presented in Figs. 10 and 11, respectively. It can be clearly observed that the learning curves for both metrics of the classical models, especially CNN-LSCI, exhibit significant fluctuations and fail to converge. This is primarily due to the global pooling layer, which causes the models to lose a substantial amount of 3D information, especially in cases with limited sample size. In contrast, our quantum models demonstrates smooth convergence to the optimal values in both the loss and MAPE curves. This can be attributed to the absence of a global pooling layer in the quantum models, allowing for comprehensive learning of the 3D information present in the samples.

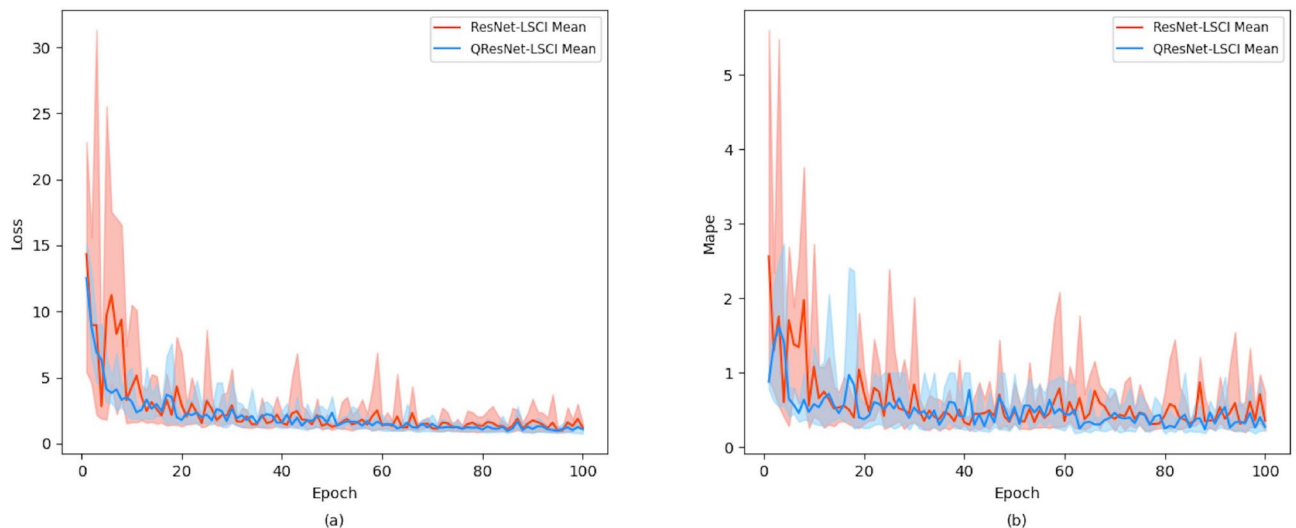


Fig. 11. Comparison of learning curves between QResNet-LSCI and ResNet-LSCI models on the validation set. (a) Learning curves of loss. (b) Learning curves of MAPE.

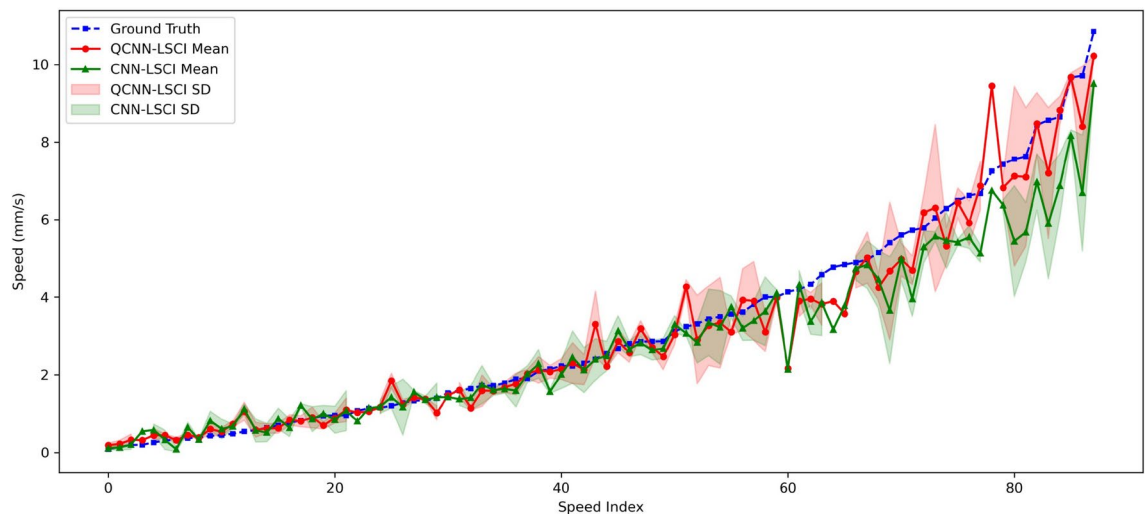


Fig. 12. Comparison of predicted speeds between QCNN-LSCI and CNN-LSCI models on the test set. Blue dash line is the actual velocity value. The red and green curves represent the mean values of the prediction results based on $n \geq 1$ samples for the QCNN-LSCI and CNN-LSCI models, respectively. The shadow areas of the corresponding color denote the standard deviation (SD).

Qualitative analysis

To gain a better understanding of the performance improvement exhibited by the proposed framework, we perform a qualitative analysis on the test set. The test dataset consists of 288 samples, covering 88 different velocities ranging from 0.0846 mm/s to 10.8528 mm/s. Most velocities have two or more corresponding samples. These 88 velocities are sorted in ascending order and indexed from 0 to 87. Each velocity corresponds to n samples, where $n \geq 1$. Subsequently, the mean and standard deviation of the predicted results from both quantum and classical models are calculated for each velocity based on the n samples. We present these statistical results in Fig. 12 for QCNN-LSCI and CNN-LSCI, and in Fig. 13 for QResNet-LSCI and ResNet-LSCI. It can be observed from the figure that the quantum models generally outperform the classical models in terms of predictive performance across most velocity ranges. Particularly, in the high-speed range with velocities greater than 5.41 mm/s, the classical models exhibit noticeable underestimation, which can be attributed to the limited number of learning samples in this range. However, even in this case of data scarcity, the quantum models demonstrate superior performance by producing predictions that are closer to the true velocities. This advantage stems from the powerful learning and generalization capability of the quantum models. This finding provides insights into predicting high-speed blood flow velocities from LSCI images, which is a particularly challenging task due to the significantly reduced speckle contrast and blurred speckle patterns associated with high flow

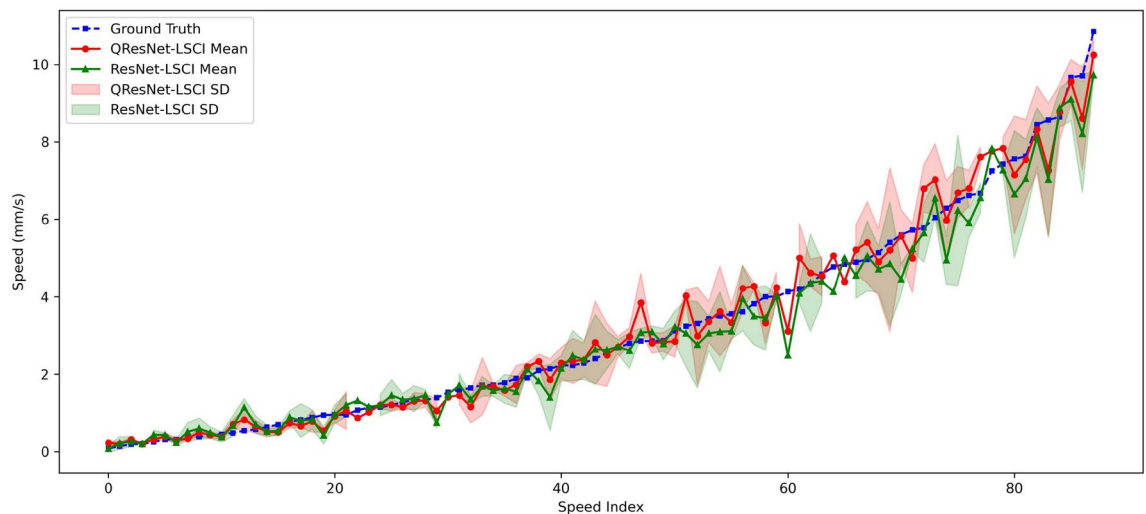


Fig. 13. Comparison of predicted speeds between QResNet-LSCI and ResNet-LSCI models on the test set.

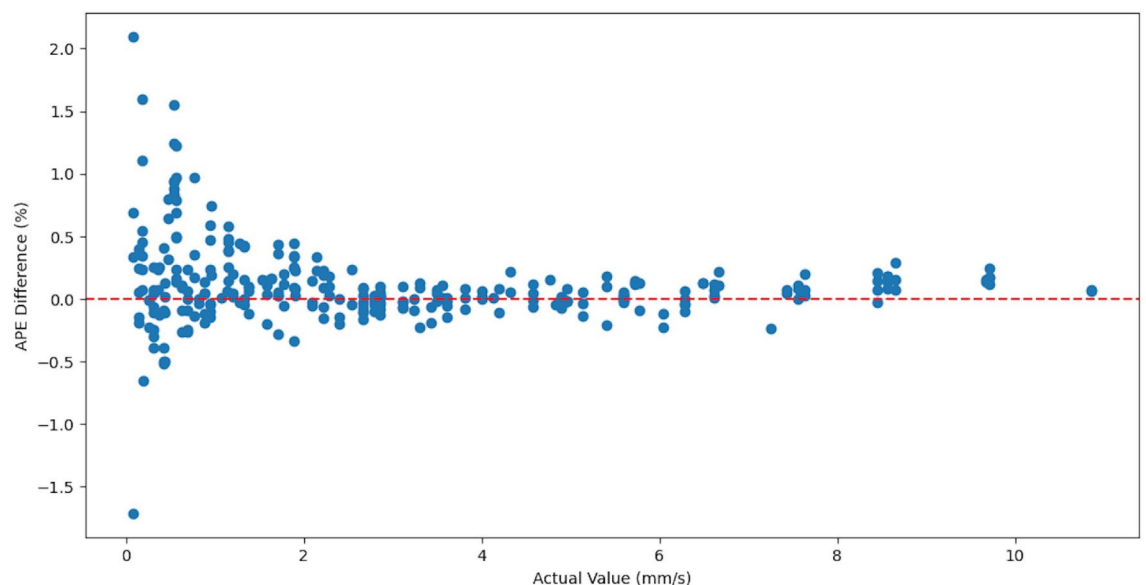


Fig. 14. Relationship between the APE difference of CNN-LSCI and QCNN-LSCI with the actual value. The y-axis represents the difference in APE calculated for each sample in the test set between models CNN-LSCI and QCNN-LSCI, while the x-axis represents the actual speed value of the sample. Multiple samples may correspond to each speed value. The x-axis is denoted by a red dashed line. Points above the x-axis indicate that the CNN-LSCI model has a higher APE than the QCNN-LSCI model, while points below the x-axis indicate that the CNN-LSCI model has a lower APE than the QCNN-LSCI model.

speeds. It should be noted that all models exhibit some fluctuations in their predictions within the high-speed range, which is also due to the lack of training data.

To further understand the performance of classical and quantum models in predicting different velocity ranges, we calculate the absolute percentage error (APE) for both models at each data point. The APE is defined as follows:

$$\text{APE} = \left| \frac{A - F}{A} \right|, \quad (9)$$

where A and F represent the actual value and predicted value respectively. We then present the relationship between the APE difference of CNN-LSCI and QCNN-LSCI with the actual value in Fig. 14, and shows a similar plot for ResNet-LSCI and QResNet-LSCI in Fig. 15. In each of these figures, points above the x-axis indicate that the classical model has higher APE than the quantum model, while those below the x-axis indicate a lower APE

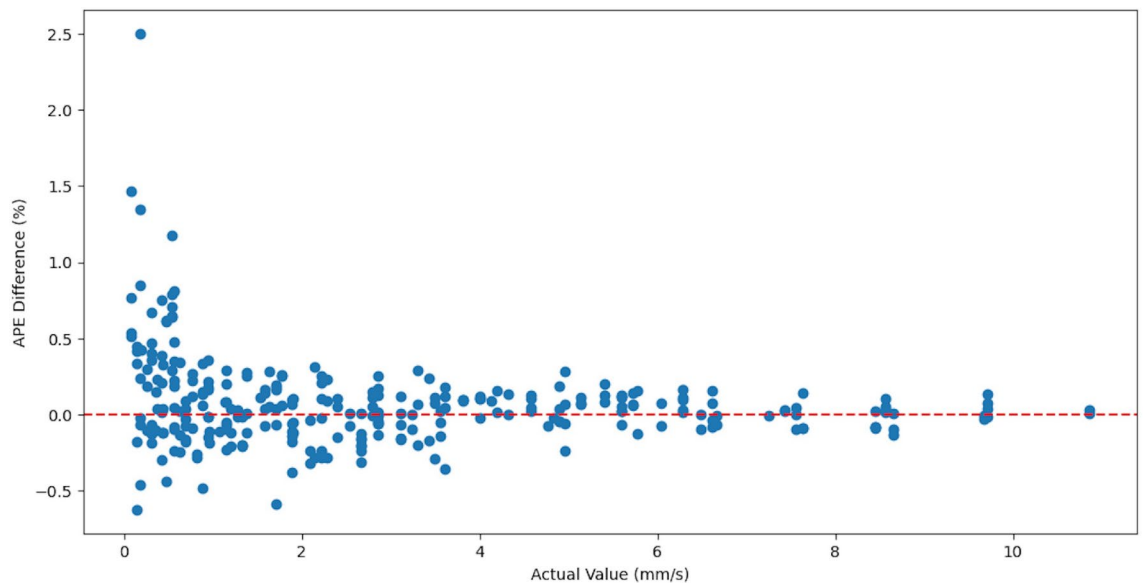


Fig. 15. Relationship between the APE difference of ResNet-LSCI and QResNet-LSCI with the actual value.

for the classical model. It can be observed from the figures that both quantum models outperform their classical counterparts respectively in predicting blood flow velocities, with a majority of the points lying above the x axis. Notably, the quantum models demonstrate a significant advantage over the classical models in predicting low-speed blood flow velocities (e.g. velocities less than 1 mm/s). Predicting low-speed blood flow velocities from LSCI images is also challenging. Even though red blood cells at low speeds yield images with high speckle contrast and clear speckle patterns, they exhibit minimal positional changes between consecutive frames, resulting in subtle speckle pattern variations that are difficult to detect. Additionally, the signals generated by low-speed blood flow are weak and easily overwhelmed by noise. In this context, our quantum models' superior performances in predicting low-speed blood flow velocities is particularly noteworthy. The advantage of quantum models in this regard stems from their ability to utilize all information obtained from the feature extraction module and capture through the VQC the complex and subtle patterns present in LSCI images that are characteristic of low-speed blood flow velocities. In contrast, the classical models' poorer performance in predicting low-speed blood flow velocities is mainly due to the significant loss of fine-grained information in the global pooling layer.

Conclusion and discussion

In this study, we propose a hybrid quantum-classical 3D CNN framework for predicting phantom velocity. The advantage of this framework lies in no longer relying on the global pooling operation applied at the later stage of the classical framework to reduce computational complexity and parameters, thus avoiding the significant loss of information incurred by global pooling. Therefore, our framework can fully utilize the temporal and spatial information in the 3D data for feature integration and hence improve the performance in predicting phantom velocity. Through comparative analysis from cross-validation, we demonstrate the superior prediction accuracy and learning stability of quantum models constructed within this framework. Additionally, we exhibit the stronger generalization capability of these quantum models on an unseen test set, and discover through qualitative analysis that they still possess a powerful learning ability even on small sample data. The robust generalization and feature learning capability arise from the highly expressive and efficient representation power of the VQC used in the proposed framework. Furthermore, we find these quantum models achieve substantial improvements in predicting blood flow at both low and high velocities compared to classical models. This is particularly noteworthy and valuable, as LSCI-based prediction at these extremes is very challenging. Overall, our research has improved the accuracy of phantom velocity prediction without increasing system complexity, thereby advancing the clinical application of LSCI. As far as we know, it is the first time that quantum machine learning techniques are applied to the field of LSCI. Our research also demonstrates the advantages of quantum machine learning.

In spite of a series of results achieved in this work, there are some issues that need to be addressed in the future research. First, the present work is based on experimental speckle contrast data. The performance of our framework can be further validated through in vivo experiments in the future. Second, due to computational resource constraints, we have selected only the first 15 frames from the original speckle contrast data to train our models in this work. In the future, we can seek additional computational resources to increase the number of frames in the data. These resources include both classical resources and quantum resources (e.g., real quantum hardware). Nevertheless, we expect that the advantages of the quantum models will become increasingly apparent as the number of frames grows, as global pooling in classical models tends to result in more information loss.

Third, our method is highly flexible and can be extended to other application scenarios of LSCI. For example, we can transform our regression model into a classification model by increasing the number of neurons in the final fully connected layer and adding a softmax activation function. This model can be employed for blood flow velocity classification task introduced in the paper²³. Other possible applications include denoising of blood flow images and blood perfusion prediction. Beyond LSCI applications, our approach has the potential to tackle a variety of challenges in other fields^{52–57}. For instance, the 3D CNN-based video retrieval method in the work⁵² incorporates a 3D global pooling operation between the convolutional and hash layers. This operation sacrifices crucial spatiotemporal information, hindering retrieval effectiveness. Similarly, the 2D CNN-based distracted driver detection model from the literature⁵⁵, designed for intelligent transportation systems, employs 2D global pooling before the fully connected layer. This feature compression loses significant discriminative detail essential for recognizing driving behaviors. Substituting VQCs for the global pooling layers could potentially enhance the performance of these models. Finally, our current study employs a single type of parameterized layer within the VQC. Future research could investigate a wider variety of quantum circuit architectures.

Data availability

The datasets used and/or analysed during the current study available from the corresponding author on reasonable request.

Appendix 1: Mathematical formulation of the VQC used in this study

In this appendix, we use a simple example to analytically derive the mathematical expression of the VQC (see Fig. 9) employed in the current work. Let's assume the feature extraction module in the 3D QCNN framework outputs a feature map F with dimensions $T \times H \times W \times C$, where T , H , W , and C represent time, height, width, and number of channels, respectively. Without loss of generality, let $T = 1$, $H = 2$, $W = 2$, $C = 1$. Therefore, the feature map F contains a total of 4 features, denoted as c_0 , c_1 , c_2 and c_3 . We use the VQC from the paper to process these features.

We begin by encoding the features c_0 , c_1 , c_2 , and c_3 into a two-qubit quantum state $|\psi\rangle$ using amplitude encoding. This state is a superposition of computational basis states:

$$|\psi\rangle = c_0|00\rangle + c_1|01\rangle + c_2|10\rangle + c_3|11\rangle,$$

which can be represented as the vector

$$|\psi\rangle = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}.$$

Next, we apply single-qubit R_x gates to each qubit, where the R_x gate is defined in equation (6). The combined operation on the two-qubit system is given by the tensor product $R_x(\theta_1) \otimes R_x(\theta_2)$, which results in a 4×4 matrix, where rotation angles θ_1 and θ_2 are trainable weight parameters. Applying this transformation to the initial state $|\psi\rangle$ produces a new state $|\psi'\rangle$:

$$|\psi'\rangle = (R_x(\theta_1) \otimes R_x(\theta_2))|\psi\rangle = \begin{bmatrix} c'_0 \\ c'_1 \\ c'_2 \\ c'_3 \end{bmatrix},$$

where the coefficients c_0 to c_3 are calculated as follows::

$$\begin{aligned} c'_0 &= c_0 \cos(\theta_1/2) \cos(\theta_2/2) - ic_1 \cos(\theta_1/2) \sin(\theta_2/2) \\ &\quad - ic_2 \sin(\theta_1/2) \cos(\theta_2/2) - c_3 \sin(\theta_1/2) \sin(\theta_2/2), \\ c'_1 &= -ic_0 \cos(\theta_1/2) \sin(\theta_2/2) + c_1 \cos(\theta_1/2) \cos(\theta_2/2) \\ &\quad + c_2 \sin(\theta_1/2) \sin(\theta_2/2) - ic_3 \sin(\theta_1/2) \cos(\theta_2/2), \\ c'_2 &= -ic_0 \sin(\theta_1/2) \cos(\theta_2/2) + c_1 \sin(\theta_1/2) \sin(\theta_2/2) \\ &\quad + c_2 \cos(\theta_1/2) \cos(\theta_2/2) - ic_3 \cos(\theta_1/2) \sin(\theta_2/2), \\ c'_3 &= -c_0 \sin(\theta_1/2) \sin(\theta_2/2) - ic_1 \sin(\theta_1/2) \cos(\theta_2/2) \\ &\quad - ic_2 \cos(\theta_1/2) \sin(\theta_2/2) + c_3 \cos(\theta_1/2) \cos(\theta_2/2). \end{aligned}$$

Finally, we perform measurements of the Pauli-Z operator on each qubit. For the first qubit, this corresponds to the operator $Z \otimes I$:

$$Z \otimes I = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}.$$

Applying this operator to $|\psi'\rangle$ gives:

$$(Z \otimes I)|\psi'\rangle = \begin{bmatrix} c'_0 \\ c'_1 \\ -c'_2 \\ -c'_3 \end{bmatrix}$$

The expectation value of the Pauli-Z operator measured on the first qubit is calculated as:

$$\langle \psi' | (Z \otimes I) | \psi' \rangle = [c'_0 \quad c'_1 \quad c'_2 \quad c'_3] \begin{bmatrix} c'_0 \\ c'_1 \\ -c'_2 \\ -c'_3 \end{bmatrix} = |c'_0|^2 + |c'_1|^2 - |c'_2|^2 - |c'_3|^2. \quad (10)$$

Similarly, the expectation value of the Pauli-Z operator on the second qubit is

$$\langle \psi' | (I \otimes Z) | \psi' \rangle = [c'_0 \quad c'_1 \quad c'_2 \quad c'_3] \begin{bmatrix} c'_0 \\ -c'_1 \\ c'_2 \\ -c'_3 \end{bmatrix} = |c'_0|^2 - |c'_1|^2 + |c'_2|^2 - |c'_3|^2. \quad (11)$$

We can see that the outputs (10) and (11) of the VQC are complex nonlinear functions of the inputs and weight parameters, demonstrating strong expressive capacity. As the dimensionality of the input features increases, the VQC will utilize more qubits, leading to even more complex outputs.

Received: 10 July 2024; Accepted: 5 November 2024

Published online: 12 November 2024

References

1. Senarathna, J., Rege, A., Li, N. & Thakor, N. V. Laser speckle contrast imaging: Theory, instrumentation and applications. *IEEE Rev. Biomed. Eng.* **6**, 99–110 (2013).
2. Heeman, W., Steenbergen, W., van Dam, G. M. & Boerma, E. C. Clinical applications of laser speckle contrast imaging: A review. *J. Biomed. Opt.* **24**, 080901–080901 (2019).
3. Hong, S.-H., Doan, A. & Marrelli, S. P. Measurement of uninterrupted cerebral blood flow by laser speckle contrast imaging (lsci) during the mouse middle cerebral artery occlusion model by an inverted lsci setup. In *Neural Repair: Methods and Protocols* 83–96 (Springer, 2023).
4. Mangraviti, A. et al. Intraoperative laser speckle contrast imaging for real-time visualization of cerebral blood flow in cerebrovascular surgery: Results from pre-clinical studies. *Sci. Rep.* **10**, 7614 (2020).
5. Feng, X. et al. Functional imaging of human retina using integrated multispectral and laser speckle contrast imaging. *J. Biophotonics* **15**, e202100285 (2022).
6. Patel, D. D. & Lipinski, D. M. Validating a low-cost laser speckle contrast imaging system as a quantitative tool for assessing retinal vascular function. *Sci. Rep.* **10**, 7177 (2020).
7. Couturier, A., Bouvet, R., Cracowski, J.-L. & Roustit, M. Reproducibility of high-resolution laser speckle contrast imaging to assess cutaneous microcirculation for wound healing monitoring in mice. *Microvasc. Res.* **141**, 104319 (2022).
8. Dijkstra, A. et al. Laser speckle contrast imaging, an alternative to laser doppler imaging in clinical practice of burn wound care derivation of a color code. *Burns* (2023).
9. Zheng, K., Middelkoop, E., Stoop, M., van Zuijlen, P. & Pijpe, A. Validity of laser speckle contrast imaging for the prediction of burn wound healing potential. *Burns* **48**, 319–327 (2022).
10. Mirdell, R., Farnbo, S., Sjöberg, F. & Tesselaar, E. Interobserver reliability of laser speckle contrast imaging in the assessment of burns. *Burns* **45**, 1325–1335 (2019).
11. Zharkikh, E., Dremine, V., Zherebtsov, E., Dunaev, A. & Meglinski, I. Biophotonics methods for functional monitoring of complications of diabetes mellitus. *J. Biophotonics* **13**, e202000203 (2020).
12. Qureshi, M. M. et al. Advances in laser speckle imaging: From qualitative to quantitative hemodynamic assessment. *J. Biophotonics* e202300126 (2023).
13. Vaz, P. G., Humeau-Heurtier, A., Figueiras, E., Correia, C. & Cardoso, J. Effect of static scatterers in laser speckle contrast imaging: An experimental study on correlation and contrast. *Phys. Med. Biol.* **63**, 015024 (2017).
14. Földes, P., Siket, M., Nagy, Á. & Jánoki, I. Correction of overexposure in laser speckle contrast imaging. *Opt. Express* **30**, 21523–21534 (2022).
15. Wang, C. et al. Robust quantitative single-exposure laser speckle imaging with true flow speckle contrast in the temporal and spatial domains. *Biomed. Opt. Express* **10**, 4097–4114 (2019).
16. Vaz, P. G., Humeau-Heurtier, A., Figueiras, E., Correia, C. & Cardoso, J. Laser speckle imaging to monitor microvascular blood flow: A review. *IEEE Rev. Biomed. Eng.* **9**, 106–120 (2016).
17. Barragán-Montero, A. et al. Artificial intelligence and machine learning for medical imaging: A technology review. *Physica Med.* **83**, 242–256 (2021).
18. Castiglioni, I. et al. Ai applications to medical images: From machine learning to deep learning. *Physica Med.* **83**, 9–24 (2021).

19. Lopez-Tiro, F., Peregrina-Barreto, H., Rangel-Magdaleno, J. & Ramirez-San-Juan, J. C. Localization of blood vessels in in-vitro lsc images with k-means. In *2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)* 1–5 (IEEE, 2021).
20. Cheng, W. et al. Dilated residual learning with skip connections for real-time denoising of laser speckle imaging of blood flow in a log-transformed domain. *IEEE Trans. Med. Imaging* **39**, 1582–1593 (2019).
21. Fredriksson, I., Hultman, M., Strömberg, T. & Larsson, M. Machine learning in multiexposure laser speckle contrast imaging can replace conventional laser doppler flowmetry. *J. Biomed. Opt.* **24**, 016001–016001 (2019).
22. Hultman, M., Larsson, M., Strömberg, T. & Fredriksson, I. Real-time video-rate perfusion imaging using multi-exposure laser speckle contrast imaging and machine learning. *J. Biomed. Opt.* **25**, 116007–116007 (2020).
23. Jain, P. & Gupta, S. Quadratic discriminant analysis performance for blood flow prediction in speckle imaging. In *2023 IEEE 8th International Conference for Convergence in Technology (I2CT)* 1–5 (IEEE, 2023).
24. Yu, C.-Y., Chammas, M., Gurden, H., Lin, H.-H. & Pain, F. Design and validation of a convolutional neural network for fast, model-free blood flow imaging with multiple exposure speckle imaging. *Biomed. Opt. Express* **14**, 4439–4454 (2023).
25. Gu, Y. et al. Automatic lung nodule detection using a 3d deep convolutional neural network combined with a multi-scale prediction strategy in chest cts. *Comput. Biol. Med.* **103**, 220–231 (2018).
26. Vesal, S., Maier, A. & Ravikumar, N. Fully automated 3d cardiac MRI localisation and segmentation using deep neural networks. *J. Imaging* **6**, 65 (2020).
27. Tran, D., Bourdev, L., Fergus, R., Torresani, L. & Paluri, M. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision* 4489–4497 (2015).
28. Kanagaraj, K. & Priya, G. L. A new 3d convolutional neural network (3d-cnn) framework for multimedia event detection. *SIVIP* **15**, 779–787 (2021).
29. Hao, X. et al. A quantitative laser speckle-based velocity prediction approach using machine learning. *Opt. Lasers Eng.* **166**, 107587 (2023).
30. Khan, T. M. & Robles-Kelly, A. Machine learning: Quantum vs classical. *IEEE Access* **8**, 219275–219294 (2020).
31. Kusumoto, T., Mitarai, K., Fujii, K., Kitagawa, M. & Negoro, M. Experimental quantum kernel trick with nuclear spins in a solid. *npj Quantum Inf.* **7**, 94 (2021).
32. DiAdamo, S., O'Meara, C., Cortiana, G. & Bernabé-Moreno, J. Practical quantum k-means clustering: Performance analysis and applications in energy grid classification. *IEEE Trans. Quantum Eng.* **3**, 1–16 (2022).
33. Xin, T. et al. Experimental quantum principal component analysis via parametrized quantum circuits. *Phys. Rev. Lett.* **126**, 110502 (2021).
34. McClean, J. R., Romero, J., Babbush, R. & Aspuru-Guzik, A. The theory of variational hybrid quantum-classical algorithms. *New J. Phys.* **18**, 023023 (2016).
35. Cerezo, M. et al. Variational quantum algorithms. *Nat. Rev. Phys.* **3**, 625–644 (2021).
36. Bharti, K. et al. Noisy intermediate-scale quantum algorithms. *Rev. Mod. Phys.* **94**, 015004 (2022).
37. Kulkarni, V., Pawale, S. S. & Kharat, A. A classical-quantum convolutional neural network for detecting pneumonia from chest radiographs. *Neural Comput. Appl.* **35**, 15503–15510 (2022).
38. Shi, S. et al. Quantum-inspired complex convolutional neural networks. *Appl. Intell.* **52**, 17912–17921 (2022).
39. Mitarai, K., Negoro, M., Kitagawa, M. & Fujii, K. Quantum circuit learning. *Phys. Rev. A* **98**, 032309 (2018).
40. Farhi, E. & Neven, H. Classification with quantum neural networks on near term processors. Preprint at [arXiv:1802.06002](https://arxiv.org/abs/1802.06002) (2018).
41. Benedetti, M., Lloyd, E., Sack, S. & Fiorentini, M. Parameterized quantum circuits as machine learning models. *Quantum Sci. Technol.* **4**, 043001 (2019).
42. Schuld, M. & Petruccione, F. *Supervised learning with quantum computers*, vol. 17 (Springer, 2018).
43. Weigold, M., Barzen, J., Leymann, F. & Salm, M. Data encoding patterns for quantum computing. In *HILLSIDE Proc. of Conf. on Pattern Lang. of Prog.* 22 (2019).
44. Bloch, F. Nuclear induction. *Phys. Rev.* **70**, 460 (1946).
45. Henderson, M., Shakyia, S., Pradhan, S. & Cook, T. Quantum convolutional neural networks: Powering image recognition with quantum circuits. *Quantum Mach. Intell.* **2**, 2 (2020).
46. Havlíček, V. et al. Supervised learning with quantum-enhanced feature spaces. *Nature* **567**, 209–212 (2019).
47. Bergholm, V. et al. PennyLane: Automatic differentiation of hybrid quantum-classical computations. Preprint at [arXiv:1811.04968](https://arxiv.org/abs/1811.04968) (2018).
48. Paszke, A. et al. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural. Inf. Process. Syst.* **32**, 8026–8037 (2019).
49. Linnainmaa, S. Taylor expansion of the accumulated rounding error. *BIT Numer. Math.* **16**, 146–160 (1976).
50. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning representations by back-propagating errors. *Nature* **323**, 533–536. <https://doi.org/10.1038/323533a0> (1986).
51. Jones, T. & Gacon, J. Efficient calculation of gradients in classical simulations of variational quantum algorithms. Preprint at [arXiv:2009.02823](https://arxiv.org/abs/2009.02823) (2020).
52. Chen, H. et al. A supervised video hashing method based on a deep 3d convolutional neural network for large-scale video retrieval. *Sensors* **21**, 3094 (2021).
53. Solov'yev, R., Kalinin, A. A. & Gabruseva, T. 3d convolutional neural networks for stalled brain capillary detection. *Comput. Biol. Med.* **141**, 105089 (2022).
54. Liu, J., Zhao, Y., Wu, H. & Jiang, D. Positional-spectral-temporal attention in 3d convolutional neural networks for EEG emotion recognition. In *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)* 305–312 (IEEE, 2021).
55. Liu, D., Yamasaki, T., Wang, Y., Mase, K. & Kato, J. Toward extremely lightweight distracted driver recognition with distillation-based neural architecture search and knowledge transfer. *IEEE Trans. Intell. Transp. Syst.* **24**, 764–777 (2022).
56. Sun, G., Cholakkal, H., Khan, S., Khan, F. & Shao, L. Fine-grained recognition: Accounting for subtle differences between similar classes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 12047–12054 (2020).
57. Li, Z. et al. Teeth category classification via seven-layer deep convolutional neural network with max pooling and global average pooling. *Int. J. Imaging Syst. Technol.* **29**, 577–583 (2019).

Acknowledgements

The authors acknowledge the support from the program of China Scholarship Council (CXXM20210004).

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to S.S.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2024