Journal Pre-proofs

Research article

ISSN 2095-4956
CN 10-1287/O6
JEC
能源化学
JOURNAL OF ENERGY CHEMISTRY
Vol. 45 • June • 2020
Science Press
Dalian Institute of Chemical Physics, Chinese Academy of Sciences

Stochastic state of health estimation for lithium-ion batteries with automated feature fusion using quantum convolutional neural network

Chen Liang, Shengyu Tao, Xinghao Huang, Yezhen Wang, Bizhong Xia, Xuan Zhang

PII: S2095-4956(25)00178-0
DOI: https://doi.org/10.1016/j.jechem.2025.02.030
Reference: JECHEM 4502

To appear in: *Journal of Energy Chemistry*

Received Date: 12 January 2025
Revised Date: 13 February 2025
Accepted Date: 21 February 2025

Please cite this article as: C. Liang, S. Tao, X. Huang, Y. Wang, B. Xia, X. Zhang, Stochastic state of health estimation for lithium-ion batteries with automated feature fusion using quantum convolutional neural network, *Journal of Energy Chemistry* (2025), doi: https://doi.org/10.1016/j.jechem.2025.02.030

Stochastic state of health estimation for lithium-ion batteries with automated feature fusion using quantum convolutional neural network

Chen Liang[a,1], Shengyu Tao [a,b,1], Xinghao Huang[a], Yezhen Wang[a], Bizhong Xia[a,*], Xuan Zhang[a,*]

[a] *Institute of Data and Information, Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, Guangdong, China*

[b]*Department of Civil and Environmental Engineering, University of California Berkeley, Davis Hall, Berkeley, CA 94720, USA*

[1] These authors contributed equally to this work.

*Corresponding authors.

E-mail addresses: xiabz@sz.tsinghua.edu.cn (B. Xia), xuanzhang@sz.tsinghua.edu.cn (X. Zhang).

ABSTRACT

The accurate state of health (SOH) estimation of lithium-ion batteries is crucial for efficient, healthy, and safe operation of battery systems. Extracting meaningful aging information from highly stochastic and noisy data segments while designing SOH estimation algorithms that efficiently handle the large-scale computational demands of cloud-based battery management systems presents a substantial challenge. In this work, we propose a quantum convolutional neural network (QCNN) model designed for accurate, robust, and generalizable SOH estimation with minimal data and parameter requirements and is compatible with quantum computing cloud platforms in the Noisy Intermediate-Scale Quantum. First, we utilize data from 4 datasets comprising 272 cells, covering 5 chemical compositions, 4 rated parameters, and 73 operating conditions. We design 5 voltage windows as small as 0.3 V for each cell from incremental capacity peaks for stochastic SOH estimation scenarios generation. We extract 3 effective health indicators (HIs) sequences and develop an automated feature fusion method using quantum rotation gate encoding, achieving an R^2 of 96%. Subsequently, we design a QCNN whose convolutional layer, constructed with variational quantum circuits, comprises merely 39 parameters. Additionally, we explore the impact of training set size, using strategies, and battery materials on the model's accuracy. Finally, the QCNN with quantum convolutional layers reduces root mean squared error by 28% and achieves an R^2 exceeding 96% compared to other three commonly used algorithms. This work demonstrates the effectiveness of quantum encoding for automated feature fusion of HIs extracted from limited discharge data. It highlights the potential of QCNN in improving the accuracy, robustness, and generalization of SOH estimation while dealing with stochastic and noisy data with few parameters and simple structure. It also suggests a new paradigm for leveraging quantum computational power in SOH estimation.

Keywords: Lithium-ion battery; State of health; Feature fusion; Quantum convolutional neural network; Quantum machine learning

1. Introduction

With the increasing severity of climate change and environmental pollution, electric vehicles (EVs) have gained widespread attention and rapid development as a key means of energy structure transformation and carbon reduction [1–4]. In 2023, EVs helped reduce global greenhouse gas emissions by 220 million tons, a significant increase compared to 80 million tons in 2022. This figure is expected to reach 2 billion tons by 2035 [5]. Lithium-ion batteries (LIBs), due to their high energy density, long cycle life, and environmental friendliness, are widely used in EVs power batteries and energy storage applications [6–9]. The vast market has accelerated the technological development of LIBs, posing stricter demands on their efficiency, health, safety, and environmental friendliness [10–12].

The state of health (SOH) is a critical indicator for evaluating the health status of LIBs, reflecting the degradation of the battery's current state relative to its initial state [13,14]. SOH estimation is an essential function of the battery management system (BMS) [15]. However, the BMS can only access highly stochastic and noisy data segments during actual use [16]. Extracting health indicators (HIs) that reflect battery aging from partial cycle data and integrating multiple HIs to derive adequate information forms the foundation of SOH estimation [14]. SOH estimation often involves processing large-scale, high-dimensional, and noisy data, making precise SOH estimation challenging and computationally intensive. The vehicle-road-cloud integration system is considered to have the potential to advance autonomous driving to an advanced stage [17]. In this framework, cloud BMS has become an important direction for intelligent battery management systems. Therefore, there is an urgent need to develop models that meet the large data processing requirements of future intelligent connected vehicle cloud BMS [18–21]. Extracting effective HIs from partial cycle data and developing SOH estimation methods with high accuracy, robustness, generalization, and scalability for massive data remain prominent challenges.

SOH estimation methods can be classified into three categories: measurement-based [6,22], model-based methods [23–25], and data-driven methods [3,13,26–28]. Compared to other methods, data-driven methods require no prior knowledge and extract battery HIs from extensive experimental data to construct models for SOH estimation. HIs include two categories: statistical features [6,29,30] and raw features. Statistical HIs have low dimensionality and data volume, making them suitable for traditional machine learning (ML). However, the extraction process involves substantial computation, and identifying features suitable for diverse operating conditions remains challenging. Raw feature HIs are sampled or calculated directly from raw data such as current, voltage, and temperature. Common examples include incremental capacity (IC) curves [6] and differential voltage (DV) curves [29]. These HIs are characterized by high dimensionality and data volume, making them suitable for deep learning like multilayer perceptron (MLP), convolutional neural network (CNN), and long short-term memory (LSTM) [22,31,32]. Previous studies have primarily relied on complete charge/discharge data, rarely used in practical scenarios [27,33]. Recent research emphasizes using partial charge/discharge segments for SOH estimation [29,30,34–39]. Currently, there are two main approaches for SOH estimation based on segment data: directly using multi-dimensional data and using processed data. Shen et al. directly use voltage, current, and capacity as HIs sequences, forming a 25×3 matrix as input for a 5-layer convolutional neural network, which results in a high computational load for the neural network [37]. Wang et al. use current and voltage data as inputs for the proposed physics-guided neural network; while the input data is simple, it requires augmentation through a 2RC model [38]. When directly using multi-dimensional data, the data is highly stochastic and noisy, requiring the model to rely on the neural network or additional information for processing. Some researchers have used deeper neural networks [37] or physics-based neural networks to supplement additional physical information [38] in order to fully extract the effective information from the features, which undoubtedly increases the complexity of the neural network. However, using processed data can help extract effective information with limited raw data, improving the model's learning performance. Li et al. use current, voltage, and temperature as HIs sequences, mapping the three variables in a 15×3 matrix to a 3D image and forming a 15×15×3 matrix as input for a 3-layer neural network [34]. Peng et al. use a weighted allocation method to fuse multiple HIs [39]. Lu et al. design a separate neural network for each HIs and use an element-wise addition operation to obtain the model's input [13]. While processed data can effectively extract information and simplify storage and computation in subsequent neural network processes, it also increases computational complexity and does not fully consider the relationships between the features across different dimensions. Quantum computing demonstrates potential beyond foreseeable classical computing in certain computational tasks [40]. Quantum computers output wavefunctions through quantum gate operations, generating statistical distributions for automated dependency relationship exploration and exploitation according to the type and order of quantum gates [41]. Thanks to the superposition principle in quantum mechanics, variational quantum circuits (VQC) based quantum machine learning (QML) offers significant advantages over traditional methods: more substantial expressive power and computational efficiency [41], making it suitable for extracting adequate aging information from small datasets, it further automatedly integrates HIs from aging information to obtain a more simplified neural network input. Existing neural networks may reach their performance limits in specific tasks, requiring deeper models to improve accuracy, increasing training costs and overfitting risks. Furthermore, for future intelligent connected vehicles' cloud BMS, SOH estimation involves large-scale, high-dimensional data, where deep learning models may face computational bottlenecks and energy efficiency challenges [41]. Quantum computing can process large datasets simultaneously in a single operation, but its application in critical tasks of SOH estimation with high data volumes, stochastic conditions, and noisy measurements has yet to be fully studied.

As shown in Fig. 1, we investigate using quantum convolutional neural network (QCNN) for SOH estimation of LIBs, emphasizing the extraction of effective HIs from limited but highly stochastic discharge cycle data and automatically fusing features. This model features high accuracy, reliability, and generalization while using minimal parameters and a straightforward structure and being compatible with novel quantum computing cloud platforms, offering a solution for intelligent connected vehicle cloud BMS. To address this, we use quantum rotation gates for quantum encoding and, considering the relationships between the three HIs sequences, we automatedly fuse the three HIs sequences extracted from small voltage windows not exceeding 0.3 V into a single qubit. We utilize VQC to construct a quantum convolutional layer, which does not require activation functions and has excellent nonlinear expression capabilities. By combining classical neural networks and using only 4 reusable qubits, we implement feature extraction and SOH estimation with a simple quantum circuit containing 39 parameters. This approach is suitable for Noisy Intermediate-Scale Quantum (NISQ) cloud platforms and upgrades the computational framework. To evaluate model performance, we used data from 4 datasets comprising 272 cells, 5 chemical compositions, 4 rated parameters, and 73 operating conditions. We designed 6 experiments to validate the QCNN model's high accuracy, reliability, and generalization from multiple perspectives.

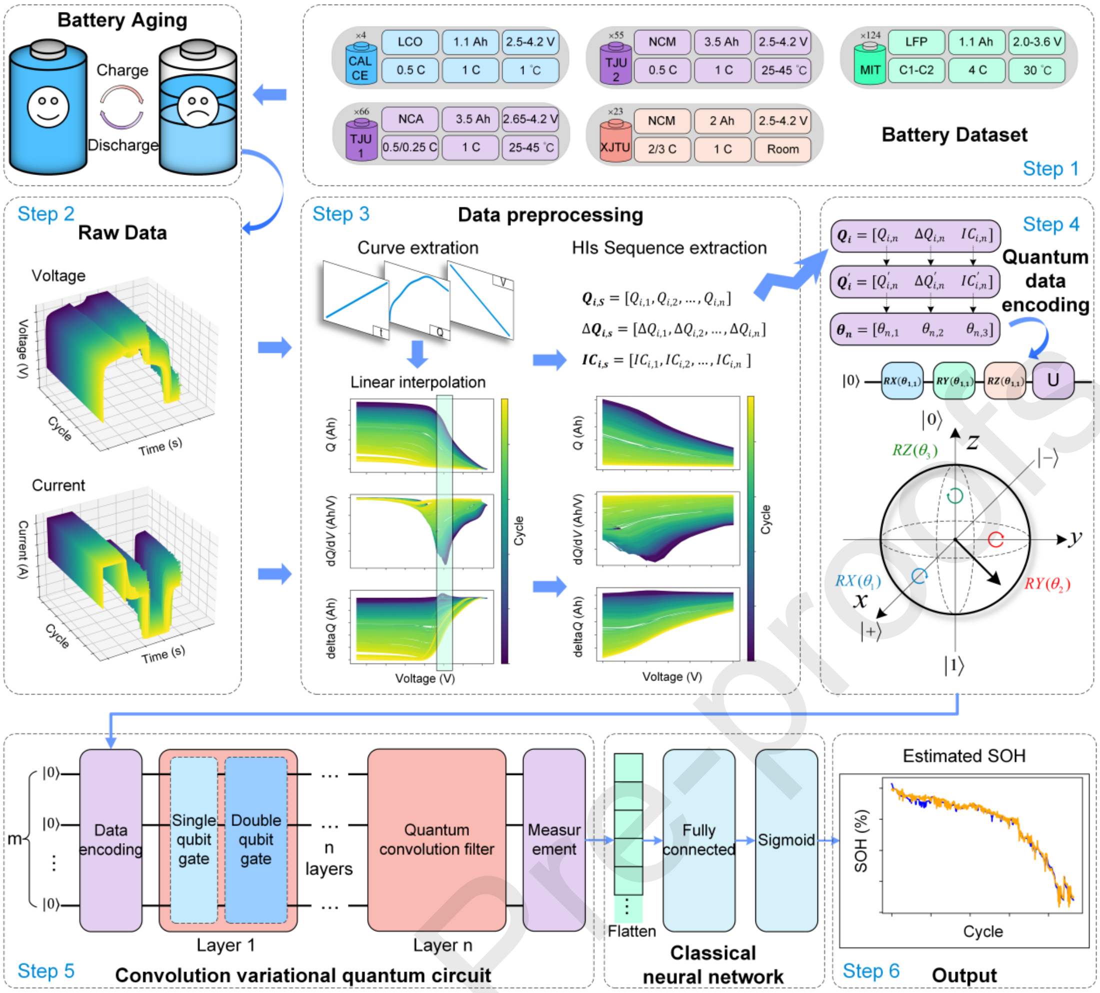

Battery Aging
Charge
Discharge
×4 CAL CE LCO 1.1 Ah 2.5-4.2 V 0.5 C 1 C 1 °C
×55 TJU 2 NCM 3.5 Ah 2.5-4.2 V 0.5 C 1 C 25-45 °C
×124 MIT LFP 1.1 Ah 2.0-3.6 V C1-C2 4 C 30 °C
×66 TJU 1 NCA 3.5 Ah 2.65-4.2 V 0.5/0.25 C 1 C 25-45 °C
×23 XJTU NCM 2 Ah 2.5-4.2 V 2/3 C 1 C Room
Battery Dataset
Step 1
Step 2
Raw Data
Voltage
Voltage (V)
Cycle
Time (s)
Current
Current (A)
Step 3
Data preprocessing
Curve extration
HIs Sequence extraction
$Q_{i,s} = [Q_{i,1}, Q_{i,2}, ..., Q_{i,n}]$
$\Delta Q_{i,s} = [\Delta Q_{i,1}, \Delta Q_{i,2}, ..., \Delta Q_{i,n}]$
$IC_{i,s} = [IC_{i,1}, IC_{i,2}, ..., IC_{i,n}]$
Linear interpolation
Q (Ah)
dQ/dV (Ah/V)
deltaQ (Ah)
Voltage (V)
Cycle
Step 4
Quantum data encoding
$Q_i = [Q_{i,n} \quad \Delta Q_{i,n} \quad IC_{i,n}]$
$Q_i' = [Q_{i,n}' \quad \Delta Q_{i,n}' \quad IC_{i,n}']$
$\theta_n = [\theta_{n,1} \quad \theta_{n,2} \quad \theta_{n,3}]$
$|0\rangle$ $RX(\theta_{1,1})$ $RY(\theta_{1,1})$ $RZ(\theta_{1,1})$ U
$|0\rangle$
Z
$RZ(\theta_3)$
$|-\rangle$
y
$RX(\theta_1)$
x
$|+\rangle$
$RY(\theta_2)$
$|1\rangle$
m
$|0\rangle$
$|0\rangle$
$|0\rangle$
Data encoding
Single qubit gate
Double qubit gate
n layers
Quantum convolution filter
Measurement
Layer 1
Layer n
Step 5
Convolution variational quantum circuit
Flatten
Fully connected
Sigmoid
Classical neural network
Estimated SOH
SOH (%)
Cycle
Step 6
Output

Fig. 1. The flowchart for SOH estimation using QCNN. Step 1: Conduct experiments using 5 different batteries from 4 datasets, dividing the data into training, validation, and test sets in an 8:1:1 ratio. Step 2: Obtain the voltage and current curves from the raw data. Step 3: Preprocess the raw data by extracting t, Q, and V data. Apply linear interpolation to obtain the feature sequences $Q(V)$, $\Delta Q(V)$, IC, and select discharge voltage window to derive HIs sequences $Q_{i,s}$, $\Delta Q_{i,s}$, $IC_{i,s}$. Step 4: Normalize the three HIs sequences, then apply quantum rotation gates for quantum encoding and feature fusion, using them as inputs for the QCNN model. Step 5: Build the quantum convolutional layer using VQC, where each quantum convolutional layer consists of a single qubit gate and a double qubit gate. After quantum measurement, the data is flattened and input into the fully connected layer. Step 6: Use a Sigmoid function to map the data to the SOH.

The remainder of this work is structured as follows: firstly, Section 2 introduces the five types of batteries from four datasets used in this work. Next, Section 3 outlines the data preprocessing process. Subsequently, Section 4 details the architecture of the proposed QCNN model. Then, Section 5 analyzes and discusses the results. Finally, Section 6 summarizes the overall findings and contributions of the work.

2. Experiment data

This work considers the materials and operating conditions of commonly used LIBs and selects 4 datasets, with 5 different chemical compositions, rated parameters, and operating conditions to validate the proposed QCNN model for SOH estimation of LIBs. Since battery aging manifests in multiple aspects, such as capacity reduction, internal resistance increase, and decreased charging efficiency [42], SOH can be defined from various perspectives [43]. Most studies define SOH based on the battery's maximum usable capacity [27,44–47], expressed as

$$SOH = \frac{C_{max}}{C_{norm}} \times 100\% \qquad (1)$$

where $C_{\max}$ represents the maximum usable capacity of the battery in its current state, and C_{norm} represents the nominal capacity of the battery. This work also adopts this definition of SOH. The capacity degradation process over the cycle count is illustrated in Fig. 2, while the parameters and operating conditions of the batteries in the four datasets are summarized in Table 1. The details of the datasets are shown in Note S1.

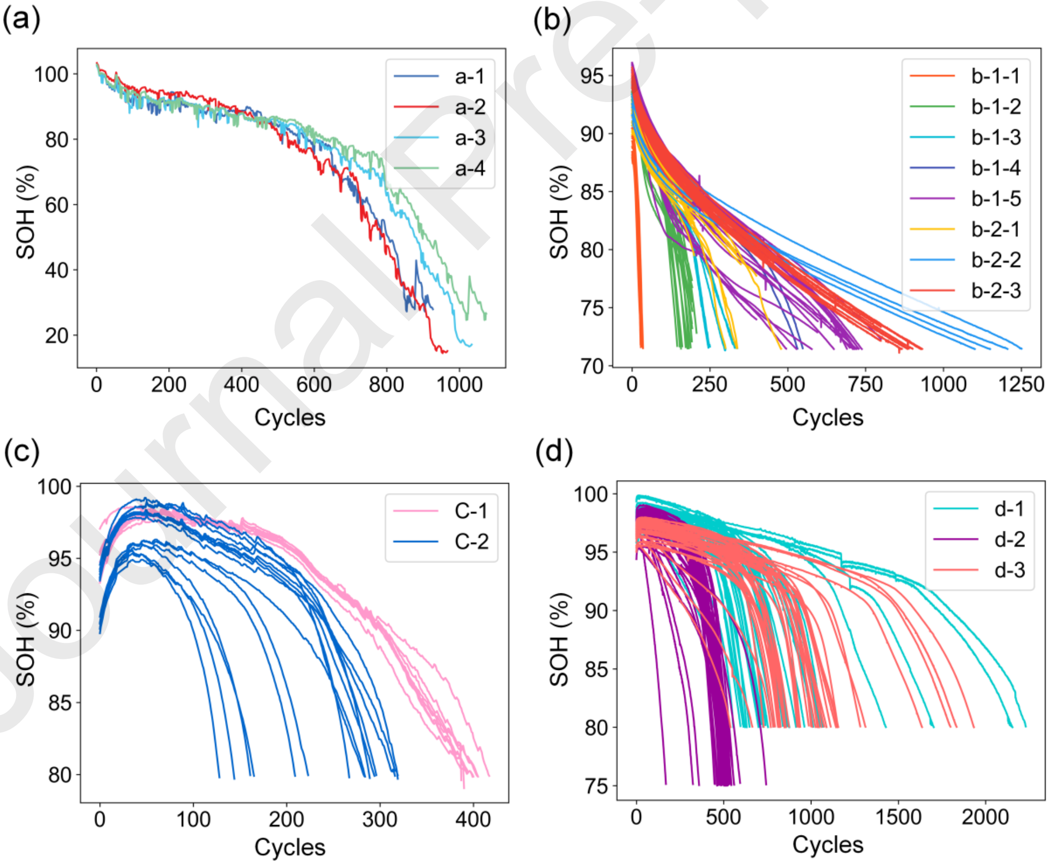


Fig. 2. SOH degradation curves of LIBs. (a–d) The SOH degradation curves of the four datasets: CALCE, TJU, XJTU, and MIT, respectively. In the CALCE dataset, each color represents a battery with the same charging and discharging strategy. In the

TJU and XJTU datasets, each color represents a group of batteries with a specific charging and discharging strategy. In the MIT dataset, each color represents a batch of batteries with different charging and discharging strategies.

Table 1. Main battery parameters at 25 ℃.

Dataset	Batch	Chemical component	Nominal capacity (mAh)	Cut-off-voltage (V)	Experiment temperature (℃)	Number of cell
CALCE	-	$LiCoO_2$	1100	2.7–4.2	1	4
TJU	1	$Li_{0.86}Ni_{0.86}Co_{0.11}Al_{0.03}O_2$	3500	2.65–4.2	25, 35, 45	66
	2	$Li_{0.84}Ni_{0.83}Co_{0.11}Mn_{0.07}O_2$	3500	2.5–4.2	25, 35, 45	55
XJTU	1,2	$LiNi_{0.5}Co_{0.2}Mn_{0.3}O_2$	2000	2.5–4.2	Room temperature	23
MIT	-	$LiFePO_4$	1100	2.0–3.6	30	124

The CALCE battery dataset (abbreviated as CALCE) was released by the University of Maryland [48]. The SOH degradation curve exhibits more fluctuations compared to other datasets, making it suitable for verifying the accuracy of the SOH estimation algorithm proposed in this work, particularly when the SOH is significantly lower than the typical 100%–80% range and when dealing with considerable SOH fluctuations.

The TJU battery dataset (abbreviated as TJU) was released by Tongji University [14]. The batteries in this dataset were tested at different temperatures and with two types of materials, making it suitable for verifying the effectiveness of the SOH estimation algorithm proposed in this work under varying temperatures and battery materials.

The XJTU battery dataset (abbreviated as XJTU) was released by Xi'an Jiaotong University [8]. This dataset was selected to verify the effectiveness of the proposed algorithm under different battery materials and non-fixed charging strategy conditions.

The MIT-Stanford battery dataset (abbreviated as MIT) was released by the Massachusetts Institute of Technology and Stanford University [49]. The characteristic of this dataset is that the charging process involves fast charging, making it suitable for studying the impact of fast charging on battery life degradation.

3. Data preprocessing

This work estimates SOH based on neural network. Before training the neural network, the raw data from the batteries' full lifecycle need to be processed to extract data that is suitable for input into the neural network and contains battery aging information. Researchers generally believe that $Q(V)$ contains aging information about the battery [35,50,51], and the $Q(V)$ sequence is used for SOH estimation. Moreover, $\Delta Q(V)$ and IC are both derived from $Q(V)$ and are closely related to battery aging. Therefore, statistical features calculated from $Q(V)$ are also widely used. Since some sequences better reflect the incomplete charge and discharge cycles that batteries go through during use [40,52], this work uses the discharge part of the $Q(V)$, $\Delta Q(V)$, IC sequences from the battery usage process as inputs to the neural network, as shown in Fig. 3.

$Q(V)$ is a function of capacity and voltage, where Q represents discharge capacity and reflects the process of discharge capacity increasing as voltage decreases. $Q(V)$ can easily be calculated by the battery management system (BMS), which can directly measure voltage V, current I, and time t, and calculate the discharge capacity Q through ampere-hour integration. Some researchers use statistical features of the $Q(V)$ sequence as inputs for ML [27], which significantly reduces the data volume but results in information loss. To maintain the integrity of the features, and given the small dataset and the powerful feature extraction ability of neural network, this work directly uses the $Q(V)$ sequence as input to the neural network. A voltage window is selected, within the upper and lower cutoff voltages, where all three features show significant changes. The voltage sequence $V_{i,S} = [V_{i,1}, V_{i,2}, V_{i,3},\ldots, V_{i,n}]$ is extracted, where i represents the current cycle count, there are n sampling points, and the sampling interval is constant. Since some sampling points may not exist in the dataset, linear interpolation is used to obtain the feature sequence, which results in the corresponding discharge capacity sequence $Q_{i,S} = [Q_{i,1}, Q_{i,2}, Q_{i,3},\ldots, Q_{i,n}]$.

$\Delta Q(V)$ represents the change in Q corresponding to the same voltage sampling points across different cycles. The sequence is given by $\Delta Q_{i,S} = [Q_{i,1} - Q_{r,1}, Q_{i,2} - Q_{r,2}, Q_{i,3} - Q_{r,3},\ldots, Q_{i,n} - Q_{r,n}]$, where $Q_{r,n}$ is the discharge capacity at the n_{th} sampling point of the selected reference cycle r. The reference cycle used in this work is the first cycle, so $\Delta Q_{i,S} = [Q_{i,1} - Q_{0,1}, Q_{i,2} - Q_{0,2}, Q_{i,3} - Q_{0,3},\ldots, Q_{i,n} - Q_{0,n}]$. This feature sequence reflects the change in the $Q(V)$ sequence at the current cycle compared to the initial cycle as the battery ages.

The IC sequence is the sampling sequence of the incremental capacity curve, which reflects the rate of change in $Q(V)$ at the same voltage sampling points as the battery ages. The IC sequence is obtained by taking the derivative of adjacent $Q_{i,S}$ and $V_{i,S}$, as shown by $IC_{i,s} = [0, \frac{Q_{i,2}-Q_{i,1}}{\Delta V}, \frac{Q_{i,3}-Q_{i,2}}{\Delta V}, \frac{Q_{i,4}-Q_{i,3}}{\Delta V}, ..., \frac{Q_{i,n}-Q_{i,1}}{\Delta V}]$. Numerous studies have shown that the peaks of the IC curve are closely related to the battery aging mechanism. As the battery ages, its capacity decreases, and both the position and the area under the peak of the IC curve change [2]. Therefore, in this work, the voltage window is selected within the IC peak range.

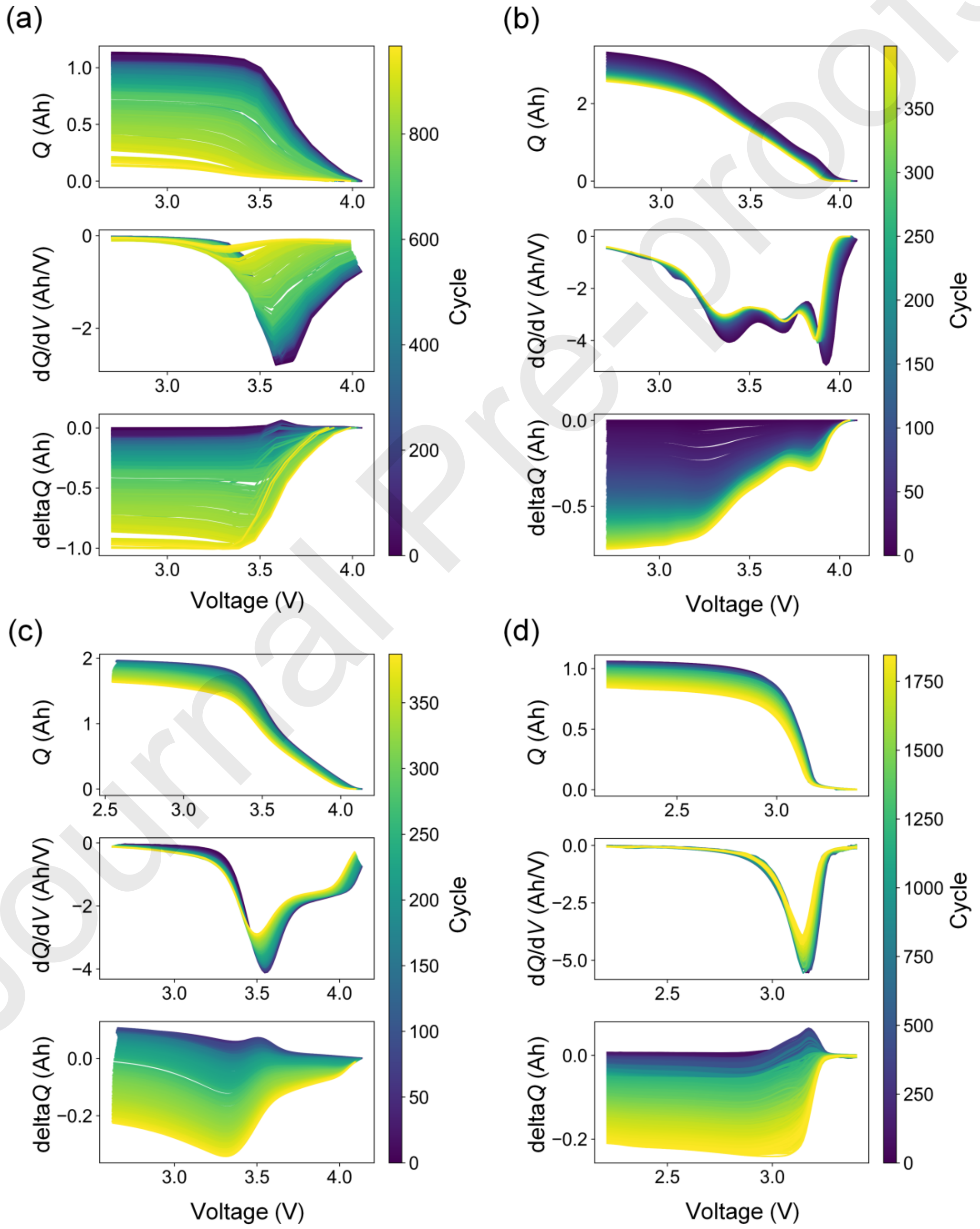


Fig. 3. $Q(V)$, $\Delta Q(V)$, IC sequences. (a–d) The three HIs sequences extracted from the four datasets proposed in this work. As the battery cycle count increases, the color in the images gradually transitions from dark blue to light green.

4. Quantum convolutional neural networks

This section proposes a QCNN for estimating SOH, which uses quantum angular encoding to encode the three HIs sequences $Q_{i,s}$, $\Delta Q_{i,s}$, $IC_{i,s}$ and uses these encoded sequences as input to the QCNN. SOH is taken as the output of the QCNN. The QCNN architecture is the same as the classical CNN, consisting of convolutional layers, pooling layers, and fully connected layers. The difference is that the QCNN constructs the convolutional layers using quantum circuits, which not only enhances the expressive power of the neural network but also takes advantage of the current capabilities of NISQ computers. The detail of the structure of CNN is shown in Note S2.

4.1. Quantum data encoding

In ML, feature mapping is commonly used to map input data to a new feature space, i.e., $\phi\colon \mathcal{X}\rightarrow\mathcal{X}'$, to facilitate subsequent tasks such as classification, regression, or other objectives. Similarly, in QML, quantum encoding [53] is required to map classical data to a Hilbert space, i.e., $\phi\colon\mathcal{X}\rightarrow\mathcal{H}$. This allows quantum transformations to be performed on qubits in subsequent steps. The quantum encoding operation $x\in\mathcal{X}\rightarrow|\phi(x)\rangle\in\mathcal{H}$ is achieved by applying a unitary transformation $|0\rangle^{\otimes n}$ to the initial quantum state $U_{\phi}(x)$, such that $U_{\phi}(x)|0\rangle^{\otimes n}=|\phi(x)\rangle$, where n is the number of qubits.

By selecting $|0\rangle$ and $|1\rangle$ as the basis states, a single qubit $|\psi\rangle$ can be represented as a superposition state

$$|\psi\rangle=\alpha|0\rangle+\beta|q\rangle \tag{2}$$

where α and β are complex numbers, and $\alpha^2+\beta^2=1$. Thus

$$|\psi\rangle=e^{i\gamma}\left(cos\ \frac{\theta}{2}|0\rangle+e^{i\varphi}sin\ \frac{\theta}{2}|1\rangle\right) \tag{3}$$

where $\theta,\varphi,\gamma\in[0,2\pi]$ represent angles. Since $e^{i\gamma}$ has no observable effect, it can be omitted, so

$$|\psi\rangle = e^{i\gamma}\left(cos\ \frac{\theta}{2}|0\rangle + e^{i\varphi}sin\ \frac{\theta}{2}|1\rangle\right) \tag{4}$$

The process of quantum encoding is shown in Fig. 4. First, the three feature sequences are normalized to the range [0, 2π] to facilitate the application of the operators.

$$Q'_{i,S} = [Q'_{i,1}, Q'_{i,2}, Q'_{i,3}, \ldots, Q'_{i,n}] \tag{5}$$

$$\Delta Q'_{i,s} = [Q'_{i,1} - Q'_{r,1}, Q'_{i,2} - Q'_{r,2}, Q'_{i,3} - Q'_{r,\,3}, \ldots, Q'_{i,n} - Q'_{r,n} \tag{6}$$

$$IC'_{i,s} = [0, \frac{Q'_{i,2}-Q'_{i,1}}{\Delta V}, \frac{Q'_{i,3}-Q'_{i,2}}{\Delta V}, \frac{Q'_{i,4}-Q'_{i,3}}{\Delta V}, \ldots, \frac{Q'_{i,n}-Q'_{i,1}}{\Delta V}] \tag{7}$$

Then, for the three feature points corresponding to the same voltage sampling point $\left[Q'_{i,n} \quad Q'_{i,n} - Q'_{r,n} \quad \frac{Q'_{i,n}-Q'_{i,1}}{\Delta V}\right]$ operators are applied. Therefore, each feature point corresponds to a point on the surface of the Bloch sphere.

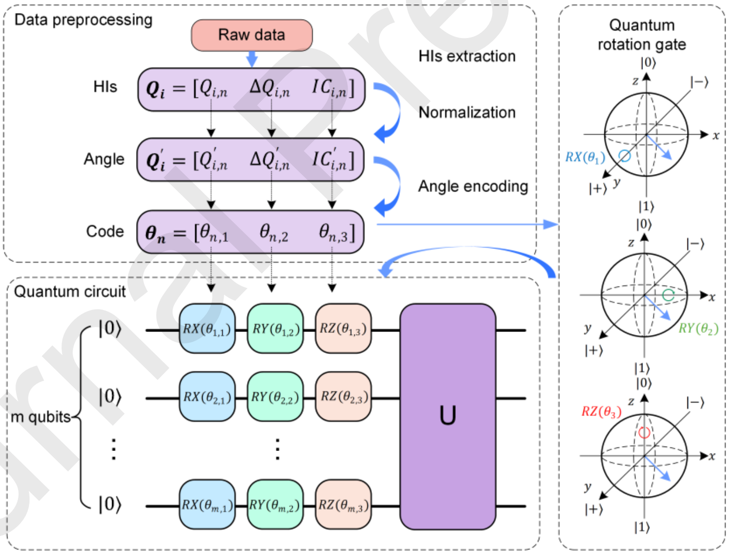


Fig. 4. The quantum encoding process. First, the raw data is preprocessed to extract three HIs sequences and normalized to the range [0, 2π]. Next, quantum rotation gates are applied to encode these three features as angular values, mapping them to points on the surface of the Bloch sphere in Hilbert space. Finally, a quantum circuit is constructed, where each qubit simultaneously encodes the three features from the same cycle, serving as the input to the QCNN.

As described in Section 3, $\Delta Q_{i,s}$ and $IC_{i,s}$ are both derived from $Q_{i,s}$. Therefore, we consider the relationships between the 3 HIs sequences and use a single qubit to represent the 3 HIs sequences by rotating along 3 directions on the Bloch sphere, reducing storage and computation space while performing feature fusion. The details of the quantum rotation gates are shown in Note S3.

4.2 QCNN architecture design

As shown in Fig. 5, the basic structure includes the input layer, convolutional layers, pooling layers, and fully connected layers. Unlike classical CNNs, the convolutional layers of QCNN are built using VQC, which also exhibits local connectivity and parameter sharing. In this work, each quantum convolutional layer consists of a single-qubit layer and a two-qubit gate layer. In the quantum convolutional layer, two-qubit gates are applied to adjacent qubits, reflecting the local connectivity. All single-qubit and two-qubit gates in a convolutional layer share the same parameters, reflecting parameter sharing.

As shown in Fig. 5, the quantum convolutional layer utilizes a quantum circuit with m qubits. First, m feature points $A_{p:(p+m-1),3}$ are extracted from the 3 normalized feature sequences, where p is the starting index of the current local receptive field. Then, as described in Section 4.1, the features are encoded using the RX, RY, and RZ gates to obtain the quantum state $|\psi\rangle$.

$$|\psi\rangle = |\psi_0 A_{p:(p+m-1),3}\rangle \tag{8}$$

where $|\psi_0\rangle$ represents the quantum state associated with the RX, RY, and RZ gates, which correspond to unitary operators. By applying n convolutional layers to $|\psi\rangle$, the quantum state evolves to $|\psi'\rangle$.

$$|\psi'\rangle = U_i^{\otimes n}|\psi\rangle = U_i^{\otimes n}|\psi_0 A_{p:(p+m-1),3}\rangle \tag{9}$$

After the quantum evolution, a measurement is performed using the $Z^{\otimes m}$ operator, leading to the feature map

$$A_{p:(p+m-1),3} \rightarrow \langle\psi|Z^{\otimes m}|\psi'\rangle \tag{10}$$

In this work, the number of qubits is set to 4. The single-qubit gate used is the RX gate, while the two-qubit gate is the CNOT gate, represented as

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{11}$$

Quantum evolution has a nonlinear nature, so there is no explicit need for activation functions in the convolutional layers, which reduces the network's complexity. Due to quantum measurement-induced collapse, the quantum state collapses irreversibly from the superposition state to a definite state. Additionally, quantum gates can be arranged sequentially within the quantum circuit, allowing multiple convolutional layers to be constructed within the same quantum circuit.

Since the feature sequences are input sequentially into the quantum convolutional layers, the quantum convolutional layers can be reused, requiring only *m* qubits. The structure of the quantum circuit remains the same, allowing it to be repeated efficiently. This quantum convolutional layer requires only a small number of qubits, resulting in low computational costs and making it suitable for the NISQ regime.

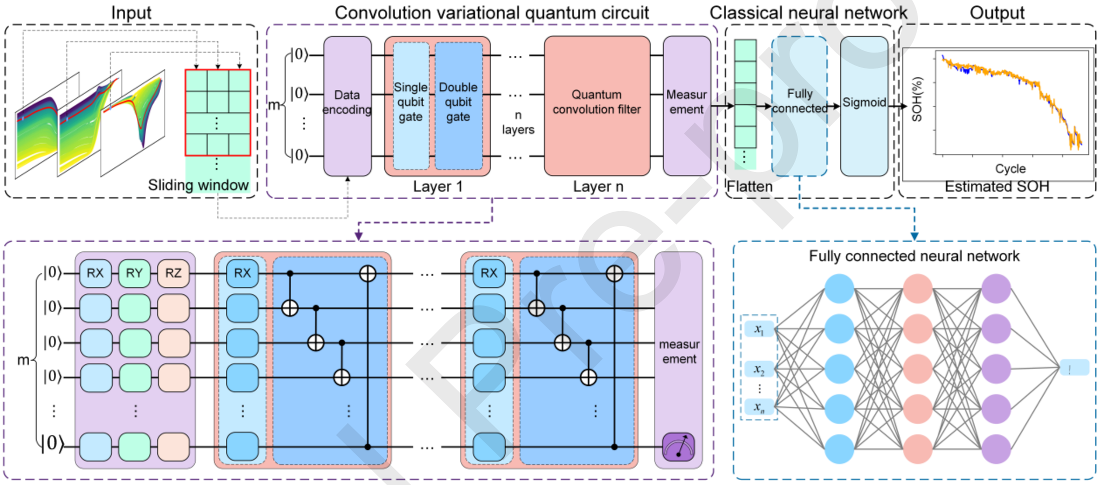


Fig. 5. The basic structure of QCNN. QCNN consists of four main components: the input layer, quantum convolutional layers, classical network layers, and output layer. Unlike classical CNN, the data encoding and convolutional layers are built using quantum circuits. The input layer feeds the raw data into the quantum data encoding layer. The quantum convolutional layer consists of RX and CNOT gates, performing quantum operations on the data. After quantum measurement of the final qubit, the data is passed to the classical network layer, which includes a flattening layer, fully connected layers, and a sigmoid function. The output layer then provides the estimated SOH value. This hybrid structure leverages both quantum and classical models for enhanced performance.

4.3. Complexity analysis of QCNN

In this work, the quantum-related components were simulated using Pennylane. Pennylane is a cross-platform Python library for programming quantum computers, which integrates quantum hardware with powerful machine learning frameworks such as Pytorch, TensorFlow, and NumPy, supporting hybrid quantum-classical models [54].

Since the quantum circuit construction in this work is based on the cross-platform Python library Pennylane, simulating quantum computing on a classical computer, the state space of the quantum system grows exponentially with the number of qubits. For n qubits, the number of states in the system is 2^n. Classical computers must store and process all possible states, while quantum computing can perform parallel computation. Quantum computing leverages quantum superposition and entanglement for parallel computation, while classical computers must process each state sequentially during simulation. Although multithreading or distributed computing can speed up the process to some extent, they still cannot match the inherent parallelism of quantum computing. Additionally, simulating quantum computing on a classical computer requires a large amount of matrix multiplication and vector transformation, which further increases computation time. As a result, the speed of simulating quantum computing on a classical computer cannot meet expectations. Therefore, this work only analyzes the complexity of the QCNN model in terms of storage and computation from a theoretical perspective.

First, this work uses quantum data encoding to encode data, converting classical bits into qubit patterns that meet the needs of the subsequent quantum neural network and perform feature fusion. By using only one qubit to fuse 3 HIs, it effectively reduces the storage and computation costs of features, providing an advantage for processing large data. Second, the QCNN model uses variational quantum circuits to build the convolution layer, where single-qubit and two-qubit gates are used to achieve local connectivity and parameter sharing, similar to classical CNNs. To accurately estimate the SOH, classical neural networks require nonlinear activation functions to introduce nonlinearity. In contrast, since quantum gates inherently perform nonlinear quantum evolution, there is no need to introduce activation functions in the convolution layers, which reduces computational costs. Moreover, quantum circuits themselves have strong expressive abilities, making them promising for applications in mining data nonlinearity [42,55] and enhancing the accuracy of large data estimation. Lastly, in terms of parameters, the QCNN model has an advantage over other classical neural networks. A convolution kernel of the QCNN consists of 4 qubits, and the 3 convolution layers each use 1 single-qubit gate and 1 two-qubit gate. Only the single-qubit gates have hyperparameters, and the hyperparameters for the four qubits are the same. During operation, the sliding window extracts information from a 16×3 matrix corresponding to each cycle, using a 4×3 window size and a stride of 1. For a 16×3 HI sequence in one cycle, the VQC has 39 hyperparameters. Compared to classical neural networks, the QCNN model has significantly fewer parameters. In conclusion, the QCNN model fuses three features using only one qubit, does not require activation functions, and has very few parameters in the VQC-based convolution layers, all of which effectively reduce the computation and storage costs of features.

5. Results and discussion

In this section, six sets of experiments are conducted to evaluate the performance of the proposed QCNN for estimating the SOH of LIBs. The experiments focus on: voltage window selection, effectiveness of feature fusion, the impact of training set size on model accuracy, the effect of different charging and discharging strategies on model performance, the influence of different battery materials on model accuracy, and a comparison of QCNN with other methods. The experiments are carried out on four datasets, with each experiment dividing the data into training, testing, and validation sets in an 8:1:1 ratio, and each condition is trained 10 times.

Four evaluation metrics are used to comprehensively assess the performance: root mean squared error (RMSE), coefficient of determination (R^2), mean absolute error (MAE), and mean absolute percentage error (MAPE). The details of the evaluation metrics calculation are presented in Note S4. Additionally, the impact of various charging and discharging strategies, battery materials, and dataset split ratios on model accuracy are discussed in Notes S5–S7.

5.1. Selection of voltage windows

The *IC* peak V_{peak} is calculated for five battery models from four datasets, and five interval increments $\Delta V = [0.1, 0.15, 0.2, 0.25, 0.3]$ are used to define five voltage ranges: $[V_{\text{peak}}-\Delta V_k, V_{\text{peak}}-\Delta V_k\ ]$, where k represents the k_{th} increment. The V_{peak} values for the five batteries are 3.5, 3.3, 3.4, 3.55, and 3.15 V.

Since the voltage windows are small and their differences are minor, the sampling consistency is maintained by uniformly sampling 16 voltage points from each window. Each voltage window is tested 10 times to obtain the expected values of the four evaluation metrics. The expected values of the four evaluation metrics are presented in Table 2 and the error distribution boxplots for different voltage windows are shown in Fig. 6.

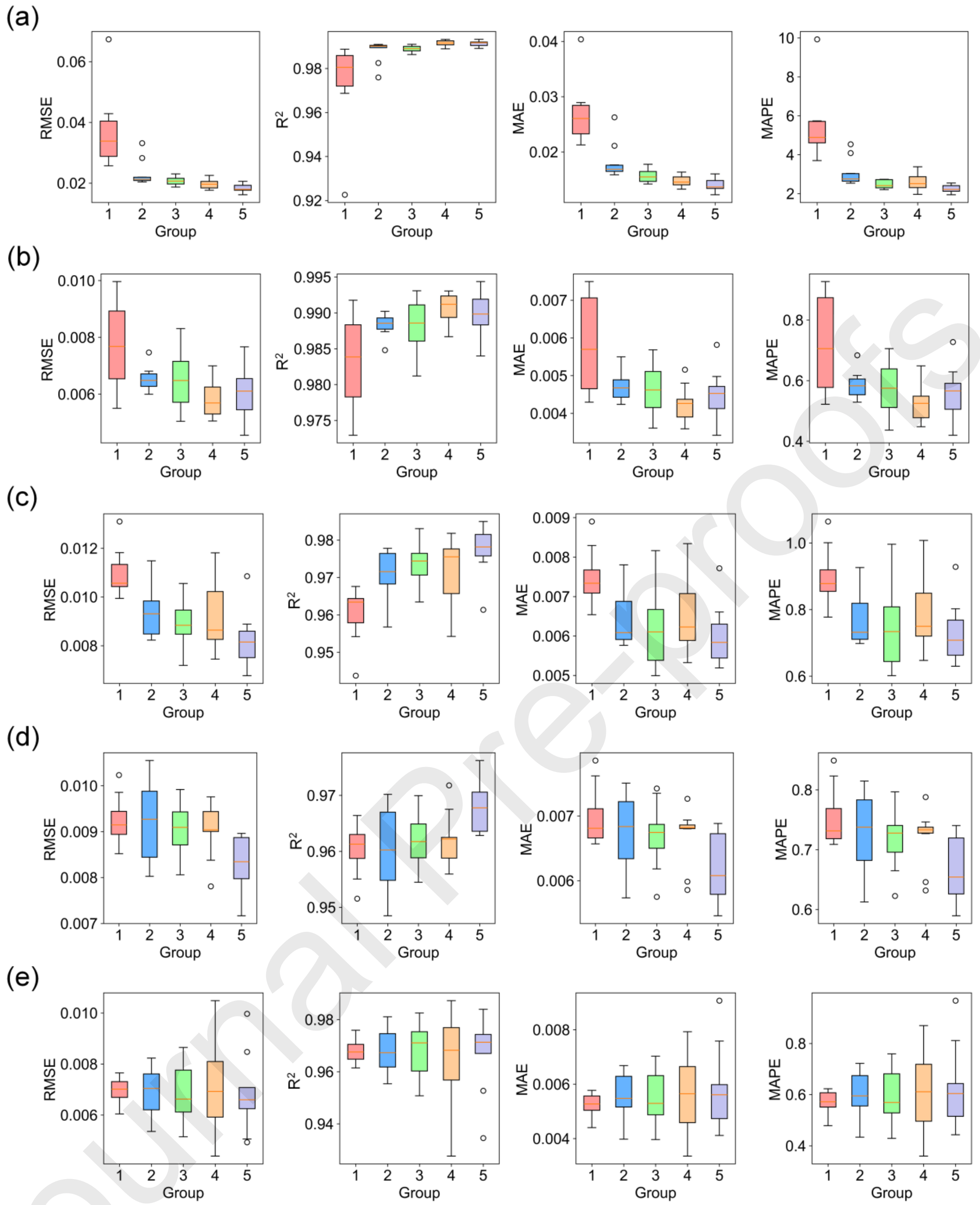


Fig. 6. Model error distribution for different voltage windows. Panels (a–e) The error distribution boxplots for the five types of batteries. Each plot consists of four subplots, each corresponding to one of the four evaluation metrics: RMSE, R^2, MAE, and MAPE. The five colors in each plot represent the corresponding five voltage windows.

Table 2. Expected SOH estimation results for different voltage windows.

Dataset	Number	1	2	3	4	5

CALCE		Voltage	3.6–3.7 V	3.575–3.725 V	3.55–3.75 V	3.525–3.775 V	**3.5–3.8 V**
		SOC	34%–61%	27%v66%	20%–70%	15%–74%	**11%–77%**
		RMSE (%)	3.66	2.31	2.06	1.97	**1.83**
		R^2 (%)	97.48	98.80	98.90	99.15	**99.13**
		MAE (%)	2.68	1.81	1.56	1.47	**1.40**
		MAPE (%)	5.43	3.06	2.48	2.58	**2.24**
TJU	NCA	Voltage	3.25–3.35 V	3.225–3.375 V	3.2–3.4 V	**3.175–3.425 V**	3.15–3.45 V
		SOC	23%–33%	21%–35%	19%–38%	**18%–41%**	17%–44%
		RMSE (%)	0.77	0.65	0.65	**0.58**	0.61
		R^2 (%)	98.32	98.83	98.81	**99.07**	98.97
		MAE (%)	0.58	0.47	0.46	**0.42**	0.44
		MAPE (%)	0.72	0.58	0.58	**0.52**	0.56
	NCM	Voltage	3.35–3.45 V	3.325–3.475 V	3.3–3.5 V	3.275–3.525 V	**3.25–3.55 V**
		SOC	39%–50%	36%–53%	33%–55%	30%–57%	**28%–59%**

	RMSE (%)	1.09	0.93	0.89	0.92	**0.82**
	R^2 (%)	96.06	97.09	97.38	97.18	**97.74**
	MAE (%)	0.75	0.64	0.63	0.65	**0.60**
	MAPE (%)	0.90	0.77	0.75	0.79	**0.73**
XJTU	Voltage	3.5–3.6 V	3.475–3.625 V	3.45–3.65V	3.425–3.675 V	**3.4–3.7V**
	SOC	35%–54%	30%–58%	26%–61%	22%–64%	**18%–66%**
	RMSE (%)	0.93	0.93	0.91	0.90	**0.83**
	R^2 (%)	96.02	95.98	96.18	96.21	**96.82**
	MAE (%)	0.70	0.68	0.67	0.67	**0.62**
	MAPE (%)	0.75	0.73	0.72	0.72	**0.66**
MIT	Voltage	**3.1–3.2 V**	3.075–3.225 V	3.05–3.25 V	3.025–3.275 V	3.0–3.3 V
	DOD	**53%–96%**	43%–97%	35%–98%	29%–98.5%	24%–99%
	RMSE (%)	**0.69**	0.70	0.68	0.70	0.70
	R^2 (%)	**96.81**	96.75	96.83	96.55	96.79

MAE (%)	**0.52**	0.56	0.55	0.56	0.58
MAPE (%)	**0.57**	0.60	0.59	0.61	0.62

As shown in Fig. 6, with the exception of the CALCE dataset, the RMSE for the other four battery datasets is less than 0.012, MAE is less than 0.0085, MAPE is less than 1.0, and R^2 is generally above 0.95. The higher RMSE, MAE, and MAPE values for the CALCE dataset are due to its larger SOH range 15%–100% and more significant fluctuations in the SOH degradation curve, compared to the other datasets, which have SOH ranges between 70% and 100% with smaller fluctuations.

Theoretically, a larger voltage window provides a broader data range, allowing for the extraction of more comprehensive features, which helps train more robust models. However, this also presents challenges in data processing, especially in handling noise and outliers. Moreover, the batteries used in this work have different materials and distinct OCV platforms, which may cause voltage windows to be distributed across platform regions due to the differing SOC ranges. When the voltage window falls within the platform region, even if the voltage window is very small, the corresponding SOC range is large, leading to inaccurate voltage window measurements and longer measurement times. As shown in Fig. 6, with the increase in voltage window size, the model performance for the CALCE, TJU-NCA, TJU-NCM, and XJTU datasets improved. For the CALCE dataset, where the material is LCO and a clear platform region exists, a 0.3 V voltage window centered around the IC curve peak was chosen to improve RMSE, MAE, and MAPE metrics. Notably, the performance of the TJU-NCA dataset was better with the 0.25 V window than with the 0.3 V window. As shown in Fig. 3(b), the IC curve for the two batteries in the TJU dataset has multiple peaks. As the voltage window increases, it may overlap with adjacent IC peaks, causing interference in feature extraction for the current IC peak. For the NCA and NCM batteries in the TJU and XJTU datasets, the SOC range corresponding to the voltage windows is small due to the voltage platform, so a slightly larger voltage range can be chosen. For the MIT dataset, as shown in Fig. 3(d), the IC peaks are relatively narrow, so a smaller voltage window is sufficient to cover the peaks, while a larger window introduces more noise. Moreover, for the LFP battery, where the voltage platform is significant, the 0.3 V voltage window corresponds to a large SOC range. Considering estimation accuracy, a 0.1 V voltage window, which corresponds to a smaller SOC range and higher estimation accuracy, was selected.

5.2. Feature fusion effectiveness

To validate the effectiveness of the three HIs sequences for SOH estimation and the feature fusion via quantum angle encoding, this section tests various combinations of the three HIs sequences. For encoding, each HIs sequence is mapped using a single quantum rotation operator. The error distribution density plots for the estimation results across four datasets and five batteries are shown in Fig. 7. The details of the single estimation results are shown in Note S8.

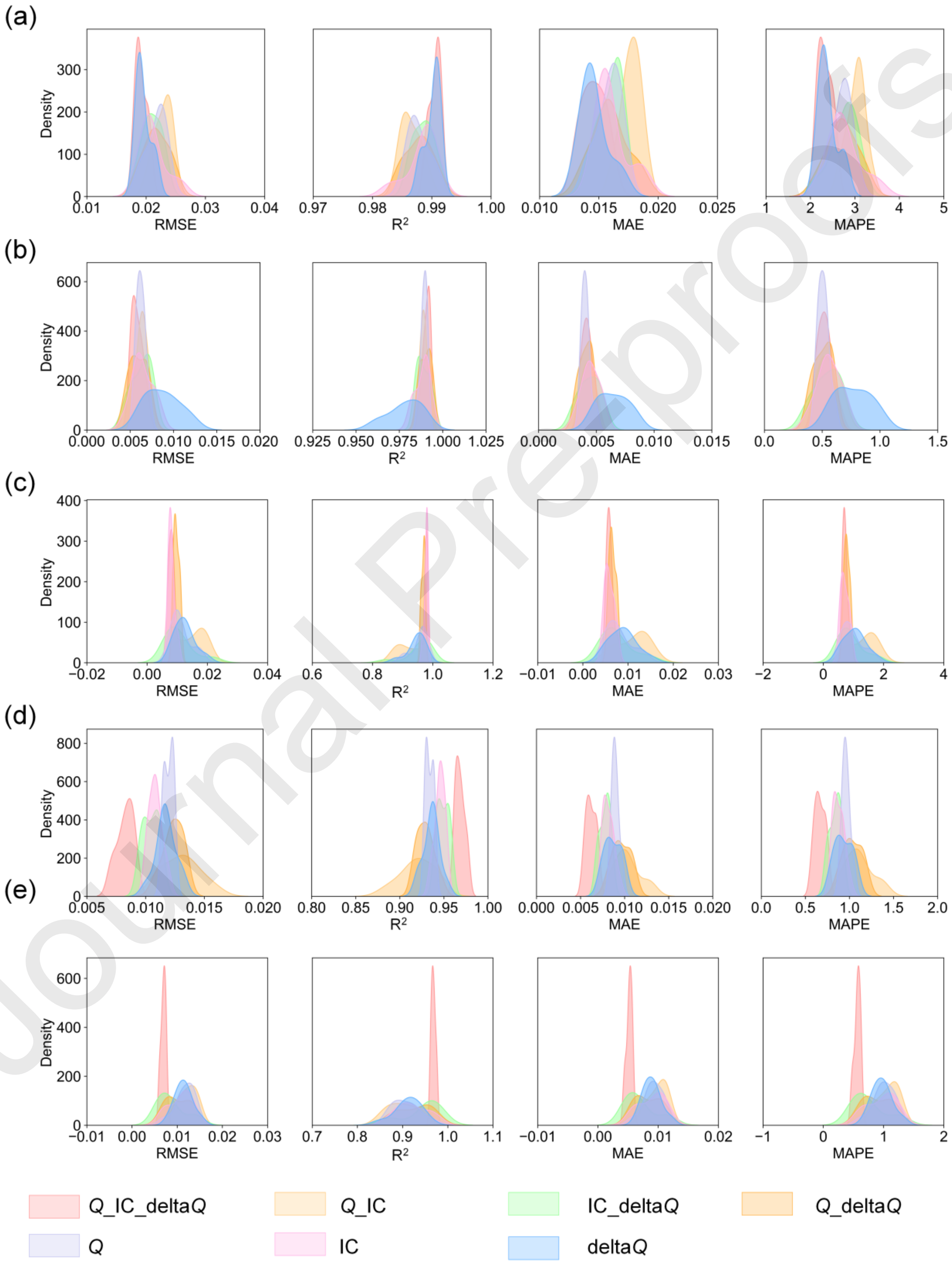


Fig. 7. Error distribution density plot for different feature combinations. (a–e) The error distribution probability density plots for five different batteries. Each plot consists of four subplots representing four evaluation metrics: RMSE, R^2, MAE, and MAPE. The seven colors in each plot represent the seven unordered combinations of the three features, showing the model's performance and error distribution under different feature combinations.

From Fig. 7, it is evident that even when using only one HIs sequence, the minimum R^2 is above 0.8, demonstrating the effectiveness of each HIs sequence extracted in this work.

For each battery, the effectiveness of the three HIs sequences varies. Specifically, the ΔQ sequence is most effective for the CALCE dataset, the Q sequence is more effective for the TJU-NCA dataset, and the IC sequence is more effective for the TJU-NCM dataset. After performing feature fusion using quantum data encoding, some datasets, such as the TJU-NCA, XJTU, and MIT datasets, show improved evaluation metrics. For the CALCE and TJU-NCM datasets, the improvements are not as significant but are still comparable to the performance of the optimal individual HIs sequence. Therefore, the feature fusion method based on quantum data encoding proposed in this work is effective for enhancing the SOH estimation by combining HIs sequences.

5.3. The effect of training set size on model accuracy

In order to investigate the impact of the training dataset size on the accuracy of the QCNN model, we used data from two types of batteries, NCA and NCM, each with four different dataset sizes, as shown in Table 3. The SOH aging curves of the selected datasets are shown in Fig. 2. The training, validation, and test sets were divided in an 8:1:1 ratio, and the models were trained 10 times for each dataset size. The estimation errors for the four dataset sizes were compared, as shown in Fig. 8.

Table 3. Dataset grouping for different training set sizes.

Dataset	TJU-NCA			TJU-NCM		
Group	Batch	Number of cells	Sum	Batch	Number of cells	Sum
1	b-1-1	9	9	b-2-1	10	10
2	b-1-2	19	19	b-2-1	20	20

3	b-1-1	9	28	b-2-1	20	30
	b-1-2	19		b-2-3	10	
4	b-1-2	19	39	b-2-1	20	40
	b-1-5	20		b-2-3	20	

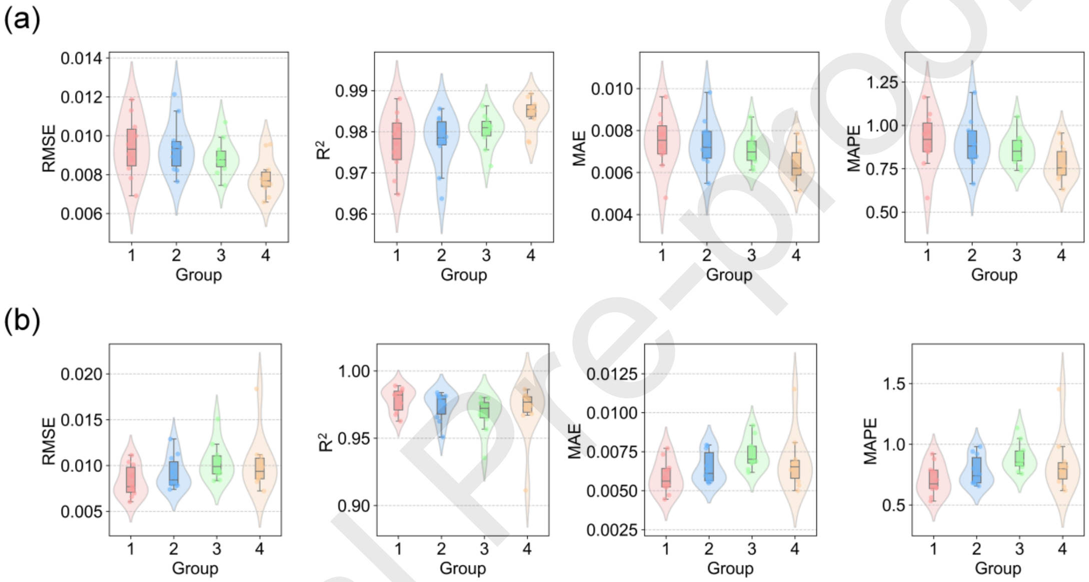


Fig. 8. Estimation error distribution when training with different training set sizes. (a and b) The estimation error distribution for the TJU-NCA and TJU-NCM datasets. Each figure consists of four subplots, representing the four evaluation metrics: RMSE, R^2, MAE, and MAPE. The four colors in each figure correspond to the four dataset size groupings.

As shown in Fig. 8, it can be observed that even when using a small sample dataset containing around 10 batteries, the trained model, except for a few outliers, maintains an R^2 above 0.95, and the RMSE, MAE, and MAPE do not exceed 0.013, 0.01, and 1.2%, respectively. This demonstrates that the proposed QCNN model performs well even with small sample sizes, indicating that the QCNN model can maintain high accuracy and stability under data-scarce conditions.

Theoretically, as the size of the dataset increases, the model's accuracy should gradually improve. As shown in Fig. 8(a), with the increasing size of the NCA dataset, the model's performance improves across all four evaluation metrics. However, as shown in Fig. 8(b), with the growing dataset size, the model's performance decreases, and outliers appear in the fourth group. The analysis reveals that, as shown in Fig. 2, the datasets used in this section did not distinguish between charging and discharging strategies. In Groups 3 and 4, batteries with different charging strategies were mixed. This demonstrates that the QCNN model has good generalization ability when the charging and discharging strategies are similar. The impact of the battery charging and discharging strategy on model accuracy will be further analyzed in the following sections. The dataset batches selected for the fourth group were b-2-1 and b-2-3. As shown in Fig. 2(b), the SOH degradation curves corresponding to these two batches exhibit irregular shapes, and the SOH at the endpoint differs. The majority of the batteries' SOH degradation curves end within the 70%–75% SOH range, but a few curves terminate in the 80%–85% range. Since the dataset was randomly divided proportionally, outliers may occur if the validation set happens to include several higher termination points. 10 experiments were conducted using different random seeds, and one outlier occurred, with the 4 evaluation metrics deviating significantly from the expected values. However, the R^2 value remained above 90%, indicating that the QCNN model has the ability to handle data from mixed usage conditions.

5.4. Comparison with other methods

This section compares the proposed QCNN model with commonly used data-driven methods, including MLP, CNN, and LSTM. The datasets used are CALCE, TJU, XITU, and MIT, where the TJU dataset is a mixture of NCA and NCM battery data. The layer structure of the QCNN model is shown in the Table 4 below. To ensure a fair comparison between the QCNN model and other classical neural networks, we set the structure of the MLP to be the same as that of the QCNN model after removing the quantum convolutional layer. The first fully connected layer is set to 48×128 to ensure matching between the nodes and the input data. The CNN model consists of 3 convolutional layers followed by 2 fully connected layers. The number of convolutional layers is the same, and the number of fully connected layers and their parameters are the same, maintaining consistency with the overall structure of the QCNN to compare the quantum convolutional layer and the classical convolutional layer. Considering the complexity of LSTM compared to other models, we reduced the number of layers to 1, followed by 2 fully connected layers. This adjustment was made to reduce the complexity of the model to match that of the other models. Overall, the architecture of the comparison models is kept consistent with the QCNN, with only minor adjustments made when necessary. MLP, CNN, LSTM, and QCNN models are trained on the four battery datasets, with each model undergoing 10 tests. The results are shown in Table 5 and Fig. 9.

Table 4. QCNN model layer structure.

Level	Layer name	Parameter	Value	Output shape

L1	Input Layer	Input feature size	16×3	
		Number of qubits	4	
L2		Sliding window size	4	
		Stride	1	
L3, L4, L5	QCL[a] ×3	Number of qubits	4	13×1
		RX Gates	4	
		CNOT Gates	4	
		Number of QCL	3	
L6	FCL[b] 1	Number of nodes	13×128	128×1
		Activation function	ReLU	
		Dropout ratio	0.5	
L7	FCL 2	Number of nodes	128×256	256×1
		Activation function	ReLU	
		Dropout ratio	0.5	
L8	Output Layer	Number of nodes	256×1	1
		Activation function	Sigmoid	

[a]QCL: quantum convolutional layer

[b]FCL: fully connected layers

Table 5. Expected errors of QCNN and other models.

Dataset	Algorithms	RMSE (%)	R^2 (%)	MAE (%)	MAPE (%)
CALCE	MLP	2.65	98.15	2.11	3.30
	CNN	1.67	99.27	1.26	2.13
	LSTM	1.77	99.19	1.29	2.16
	QCNN	1.83	99.13	1.40	2.24
TJU	MLP	1.23	95.11	1.03	1.23
	CNN	0.94	97.29	0.81	0.98
	LSTM	0.90	97.55	0.76	0.91
	QCNN	0.89	97.59	0.65	0.80
XJTU	MLP	2.25	76.08	1.82	1.98
	CNN	0.91	96.13	0.75	0.79
	LSTM	0.88	96.40	0.72	0.76
	QCNN	0.83	96.82	0.62	0.66
MIT	MLP	1.01	93.26	0.83	0.90
	CNN	0.77	95.97	0.65	0.70

LSTM	1.07	92.45	0.84	0.91
QCNN	0.67	97.00	0.51	0.55

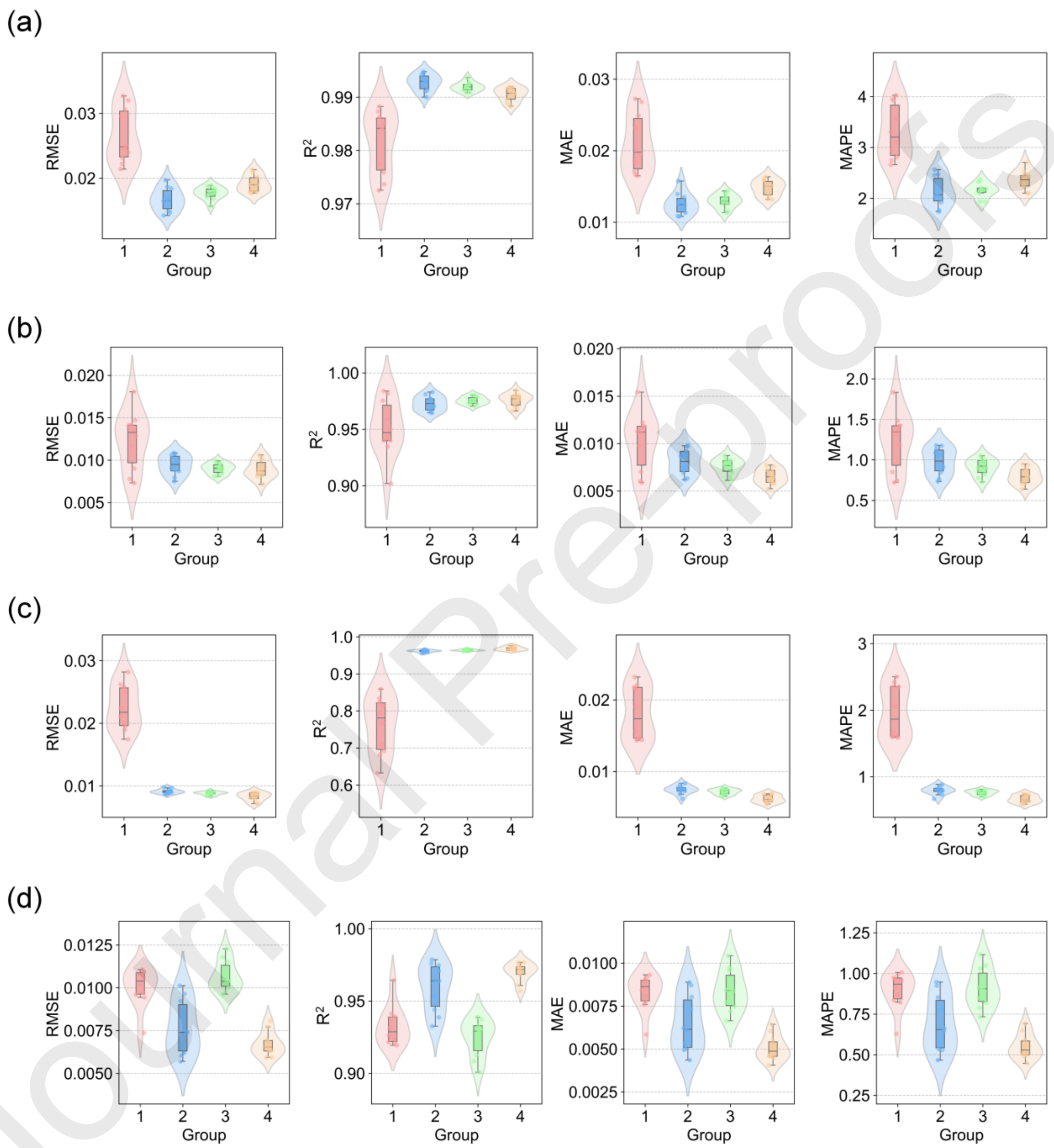


Fig. 9. Error distribution chart of QCNN and other models. (a–d) The error distribution probability density plots for four battery datasets. Each plot consists of four subplots representing the four evaluation metrics RMSE, R^2, MAE, and MAPE. The seven colors in each plot represent the four SOH estimation models: MLP, CNN, LSTM, and QCNN.

Table 5 lists the expected errors for each model over 10 tests, with the best model for each dataset highlighted in bold. From Fig. 9, it can be seen that, except for the CALCE dataset, QCNN outperforms the other three models on the remaining three datasets. QCNN's performance on the CALCE dataset is also comparable to the best-performing CNN and LSTM models. The MLP model, which is QCNN without the quantum convolutional layer, shows a significant improvement when compared to QCNN, as shown in Fig. 9 and Table 5, confirming the effectiveness of using quantum convolutional layers for feature extraction. Overall, the proposed method can accurately estimate SOH.

Using VQC to construct the convolutional layer eliminates the need for activation functions, effectively reducing computational complexity while maintaining strong nonlinear expressive ability, enabling precise SOH estimation. At the same time, the convolutional kernel uses only 4 reusable qubits, suitable for NISQ cloud platforms, and contains only 39 parameters. This results in a much lower complexity for the convolutional layer compared to classical CNNs. As shown in Table 5 and Fig. 9, the QCNN model outperforms the fully connected layer without the quantum convolutional layer, with an RMSE improvement of at least 28%, and R^2 consistently above 96%, demonstrating the effectiveness of the quantum convolutional layer in data mining. Compared to a similar CNN with the same number of convolutional layers, the MAE is reduced by 18%.

6. Conclusions

This work proposes a QCNN model for SOH estimation of LIBs. Earlier studies have employed partial but highly stochastic charge-discharge data to extract multiple HIs. When the input segments are short and the feature dimensions are small, directly using multi-dimensional data results in highly stochastic and noisy data. This requires relying on neural networks [1] or additional physical information [2] to mine aging information in order to complete SOH estimation, which undoubtedly increases the complexity of the neural network. Process multi-dimensional data before inputting it into a neural network, additional neural networks [5] or higher-dimensional data [4] may be introduced in the processing step. Although the estimation effect may improve, this increases the computational complexity. Furthermore, most studies have not fully considered the relationships between the features across dimensions [5]. Moreover, existing neural networks have reached their limits in SOH estimation. Further improvements in accuracy require deeper models, which lead to higher manual demands and energy efficiency bottlenecks when dealing with the large-scale, high-dimensional data of cloud-based BMS in intelligent connected vehicles. However, no models tailored for NISQ machines have yet been proposed in the field of SOH estimation for LIBs.

To address this, the work introduces an innovative QCNN model designed for reliable, and generalized SOH estimation while being compatible with NISQ machines. Firstly, recognizing the diversity of batteries in real-world scenarios, this work selected highly stochastic and noisy data from 4 datasets, including 272 batteries across 5 chemical compositions, 4 rated parameters, and 73 operating conditions. Secondly, considering the scarcity of full lifecycle data in practical applications, the model extracts 3 HIs sequences $Q_{i,s}$, $\Delta Q_{i,s}$, $IC_{i,s}$ from small voltage segments as short as 0.3 V. Based on the relevance of IC peaks to battery aging mechanisms, 5 voltage windows were designed for each battery, with the optimal one selected. Additionally, automated feature fusion was applied to integrate multiple HIs, enhancing the input for the neural network. Under the chosen voltage windows, R^2 values exceeded 95% across all 4 datasets. Subsequently, the work explored the impacts of training dataset size, charging/discharging strategies, and battery materials on model accuracy. QCNN demonstrates exceptional accuracy, making it well-suited to handle highly random and noisy data under various conditions. Finally, the QCNN model was compared with MLP, CNN, and LSTM under stochastic operation conditions. The QCNN reduced RMSE by at least 28% compared to MLP, with R^2 values consistently exceeding 96%. Except when trained on the CALCE dataset, where SOH estimation accuracy was comparable to CNN and LSTM, QCNN achieved at least 18% and 13% lower MAE than CNN and LSTM, respectively, across the other three datasets. In this work, the convolutional layer of the QCNN is constructed using VQC, which utilizes only 39 parameters. Compared to classical neural networks, this design is significantly more parameter-efficient and structurally simpler, making it well-suited for large-scale computations in cloud BMS. Additionally, the QCNN's quantum circuit was constructed using only a few qubits and VQC, making it suitable for NISQ computers.

To further enhance the QCNN model's practicality, future plans include incorporating more charging/discharging strategies and data from different materials, as well as exploring transfer learning to address the scarcity of training samples in real-world applications. Additionally, dynamic operating condition data and interactions with quantum computing cloud platforms during in-vehicle operations will be considered. In summary, this work demonstrates the feasibility of using QML for SOH estimation by extracting effective HIs from highly stochastic, noisy and limited cycle data and performing automatedly feature fusion with a QCNN model. This is the first attempt to utilize NISQ quantum computing power in the SOX estimation field. The results highlight the significant potential of quantum computing for SOH estimation in future cloud BMS for intelligent connected vehicles.

Acknowledgments

This work was funded by the Research on SOC/SOH Joint Estimation Technology of Electric Vehicle Battery System State Based on Online Parameter Identification Project (2019) and the National Natural Science Foundation of China (Grant No. 51877120).

CRediT authorship contribution statement

Chen Liang: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Formal analysis, Conceptualization. Shengyu Tao: Writing – review & editing, Writing – original draft, Visualization, Conceptualization. Xinghao Huang: Writing – review & editing, Visualization. Yezhen Wang: Writing – review & editing. Bizhong Xia: Resources, Supervision, Project administration, Funding acquisition, Conceptualization. Xuan Zhang: Writing – review & editing, Supervision, Conceptualization.

Declaration of competing interest

The authors declare no competing interest.

Data availability

The data used in this work is a publicly available open dataset, as mentioned in the manuscript. The specific datasets include: CALCE dataset, TJU dataset, XJTU dataset, MIT dataset.

References

[1] J. Wu, L. Fang, G. Dong, M. Lin, Energy 262 (2023) 125380.

[2] K. He, S. Tao, S. Fu, H. Fan, Y. Tao, Y. Wang, Y. Sun, J. Electrochem. Soc. 170 (2023) 030514.

[3] S. Tao, R. Ma, Z. Zhao, G. Ma, L. Su, H. Chang, Y. Chen, H. Liu, Z. Liang, T. Cao, H. Ji, Z. Han, M. Lu, H. Yang, Z. Wen, J. Yao, R. Yu, G. Wei, Y. Li, X. Zhang, T. Xu, G. Zhou, Nat Commun 15 (2024) 10154.

[4] S. Tao, G. Ma, H. Yang, M. Lu, G. Wei, G. Zhou, X. Zhang, (2025). arXiv preprint arXiv:2502.16848, 2025.

[5] The Global Electric Vehicle Market In 2024, https://www.virta.global/global-electric-vehicle-market (accessed January 11, 2025).

[6] K. Peng, Z. Deng, Z. Bao, X. Hu, Journal of Energy Storage 67 (2023) 107549.

[7] A. Tang, Y. Xu, Y. Hu, J. Tian, Y. Nie, F. Yan, Y. Tan, Q. Yu, Applied Energy 370 (2024) 123632.

[8] F. Wang, Z. Zhai, Z. Zhao, Y. Di, X. Chen, Nat Commun 15 (2024) 4332.

[9] S. Tao, C. Sun, S. Fu, Y. Wang, R. Ma, Z. Han, Y. Sun, Y. Li, G. Wei, X. Zhang, G. Zhou, H. Sun, ACS Energy Lett. 8 (2023) 3269–3279.

[10] S. Tao, H. Liu, C. Sun, H. Ji, G. Ji, Z. Han, R. Gao, J. Ma, R. Ma, Y. Chen, S. Fu, Y. Wang, Y. Sun, Y. Rong, X. Zhang, G. Zhou, H. Sun, Nat Commun 14 (2023) 8032.

[11] R. Ma, S. Tao, X. Sun, Y. Ren, C. Sun, G. Ji, J. Xu, X. Wang, X. Zhang, Q. Wu, G. Zhou, Nat Commun 15 (2024) 7641.

[12] C. Lin, J. Xu, D. Jiang, J. Hou, Y. Liang, Z. Zou, X. Mei, Journal of Energy Chemistry 100 (2025) 739–759.

[13] Z. Lu, Z. Fei, B. Wang, F. Yang, Energy 288 (2024) 129690.

[14] J. Zhu, Y. Wang, Y. Huang, R. Bhushan Gopaluni, Y. Cao, M. Heere, M.J. Mühlbauer, L. Mereacre, H. Dai, X. Liu, A. Senyshyn, X. Wei, M. Knapp, H. Ehrenberg, Nat Commun 13 (2022) 2261.

[15] S. Fu, S. Tao, H. Fan, K. He, X. Liu, Y. Tao, J. Zuo, X. Zhang, Y. Wang, Y. Sun, Applied Energy 353 (2024) 121991.

[16] X. Li, D. Yu, S.B. Vilsen, D.I. Stroe, Journal of Energy Chemistry 92 (2024) 591–604.

[17] B. Gao, J. Liu, H. Zou, J. Chen, L. He, K. Li, IEEE Trans. Intell. Transport. Syst. 25 (2024) 19295–19318.

[18] B. Bose, S. Shaosen, W. Li, L. Gao, K. Wei, A. Garg, Sustainable Energy, Grids and Networks 36 (2023) 101197.

[19] Y. Wang, R. Xu, C. Zhou, X. Kang, Z. Chen, Journal of Manufacturing Systems 62 (2022) 124–134.

[20] W. Li, M. Rentemeister, J. Badeda, D. Jöst, D. Schulte, D.U. Sauer, Journal of Energy Storage 30 (2020) 101557.

[21] S. Yang, Z. Zhang, R. Cao, M. Wang, H. Cheng, L. Zhang, Y. Jiang, Y. Li, B. Chen, H. Ling, Y. Lian, B. Wu, X. Liu, Energy and AI 5 (2021) 100088.

[22] H. Xu, L. Wu, S. Xiong, W. Li, A. Garg, L. Gao, Energy 276 (2023) 127585.

[23] B. Liu, X. Tang, F. Gao, Electrochimica Acta 344 (2020) 136098.

[24] V. S, H.S. Che, J. Selvaraj, K.S. Tey, J.W. Lee, H. Shareef, R. Errouissi, Applied Energy 369 (2024) 123542.

[25] J. Li, K. Adewuyi, N. Lotfi, R.G. Landers, J. Park, Applied Energy 212 (2018) 1178–1190.

[26] J. Zhao, Z. Wang, Y. Wu, A.F. Burke, Applied Energy 377 (2025) 124746.

[27] Y. Che, Y. Zheng, Y. Wu, X. Sui, P. Bharadwaj, D.-I. Stroe, Y. Yang, X. Hu, R. Teodorescu, Applied Energy 323 (2022) 119663.

[28] J. Zhao, X. Han, Y. Wu, Z. Wang, A.F. Burke, Journal of Energy Chemistry 102 (2025) 463–496.

[29] Y. Che, A. Foley, M. El-Gindy, X. Lin, X. Hu, M. Pecht, Automot. Innov. 4 (2021) 103–116.

[30] X. Hu, Y. Che, X. Lin, Z. Deng, IEEE/ASME Transactions on Mechatronics 25 (2020) 2622–2632.

[31] Y. Che, Z. Deng, X. Lin, L. Hu, X. Hu, IEEE Transactions on Vehicular Technology 70 (2021) 1269–1277.

[32] Y. He, Z. Deng, J. Chen, W. Li, J. Zhou, F. Xiang, X. Hu, Journal of Energy Chemistry 98 (2024) 1–11.

[33] Z. Jiao, J. Ma, X. Zhao, K. Zhang, S. Li, Journal of Energy Storage 82 (2024) 110480.

[34] Y. Li, K. Li, X. Liu, L. Zhang, Transactions of the Institute of Measurement and Control (2020). DOI: 10.1177/0142331220966425.

[35] B. Ma, L. Zhang, H. Yu, B. Zou, W. Wang, C. Zhang, S. Yang, X. Liu, Journal of Energy Chemistry 82 (2023) 1–17.

[36] Y. Che, Y. Zheng, X. Sui, R. Teodorescu, Journal of Energy Chemistry 84 (2023) 335–346.

[37] S. Shen, M. Sadoughi, X. Chen, M. Hong, C. Hu, Journal of Energy Storage 25 (2019) 100817.

[38] F. Wang, Z. Wu, Z. Zhao, Z. Zhai, C. Wang, X. Chen, Reliability Engineering & System Safety 251 (2024) 110325.

[39] S. Peng, J. Zhu, T. Wu, A. Tang, J. Kan, M. Pecht, Energy 308 (2024) 132993.

[40] V. Dunjko, J.M. Taylor, H.J. Briegel, Phys. Rev. Lett. 117 (2016) 130501.

[41] M. Henderson, S. Shakya, S. Pradhan, T. Cook, (2019). http://arxiv.org/abs/1904.04767 (accessed October 30, 2024).

[42] S. Yang, C. Zhang, J. Jiang, W. Zhang, L. Zhang, Y. Wang, Journal of Cleaner Production 314 (2021) 128015.

[43] K. Luo, X. Chen, H. Zheng, Z. Shi, Journal of Energy Chemistry 74 (2022) 159–173.

[44] A. Tang, Y. Jiang, Q. Yu, Z. Zhang, Journal of Energy Storage 68 (2023) 107734.

[45] Y. Xie, S. Wang, G. Zhang, P. Takyi-Aninakwa, C. Fernandez, F. Blaabjerg, Journal of Energy Chemistry 97 (2024) 630–649.

[46] S. Tao, M. Zhang, Z. Zhao, H. Li, R. Ma, Y. Che, X. Sun, L. Su, C. Sun, X. Chen, H. Chang, S. Zhou, Z. Li, H. Lin, Y. Liu, W. Yu, Z. Xu, H. Hao, S. Moura, X. Zhang, Y. Li, X. Hu, G. Zhou, Energy Environ. Sci. 18 (2025) 1544–1559.

[47] S. Tao, R. Ma, Y. Chen, Z. Liang, H. Ji, Z. Han, G. Wei, X. Zhang, G. Zhou, Journal of Power Sources 597 (2024) 234156.

[48] N. Williard, W. He, M. Osterman, M. Pecht, IJPHM 4 (2020). https://doi.org/10.36001/ijphm.2013.v4i1.1437.

[49] K.A. Severson, P.M. Attia, N. Jin, N. Perkins, B. Jiang, Z. Yang, M.H. Chen, M. Aykol, P.K. Herring, D. Fraggedakis, M.Z. Bazant, S.J. Harris, W.C. Chueh, R.D. Braatz, Nat Energy 4 (2019) 383–391.

[50] Y. Li, C. Zou, M. Berecibar, E. Nanini-Maury, J.C.-W. Chan, P. van den Bossche, J. Van Mierlo, N. Omar, Applied Energy 232 (2018) 197–210.

[51] C. Lin, J. Xu, M. Shi, X. Mei, Energy 247 (2022) 123556.

[52] C. Zhang, L. Luo, Z. Yang, B. Du, Z. Zhou, J. Wu, L. Chen, Energy 295 (2024) 131009.

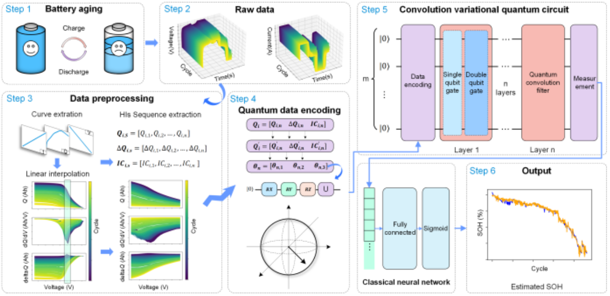
[53] I.F. Araujo, D.K. Park, T.B. Ludermir, W.R. Oliveira, F. Petruccione, A.J. da Silva, Quantum Inf Process 22 (2023) 123.

[54] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, V. Ajith, et al. (2018). https://doi.org/10.48550/arXiv.1811.04968.

[55] H. Yano, Y. Suzuki, K. Itoh, R. Raymond, N. Yamamoto, IEEE Trans. Quantum Eng. 2 (2021) 1–14.

Graphical Abstract

The segment cycle data is utilized to estimate the state of health of the lithium-ion batteries using a quantum convolutional neural network with automated feature fusion.



Declaration of Interest Statement

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The author is an Editorial Board Member/Editor-in-Chief/Associate Editor/Guest Editor for this journal and was not involved in the editorial review or the decision to publish this article.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: