Title:

BMSFormer: An efficient deep learning model for online state-of-health estimation of lithium-ion batteries under high-frequency early SOC data with strong correlated single health indicator

Author name(s):

Xiaopeng Li[a], Minghang Zhao[a,c,]*, Shisheng Zhong[a,b,c], Junfu Li[d], Song Fu[b], Zhiqi Yan[e]

Affiliation(s):

[a] Department of Mechanical Engineering, Harbin Institute of Technology, Weihai, Shandong 264209, China

[b] School of Mechatronics Engineering, Harbin Institute of Technology, Harbin, Heilongjiang 150000, China

[c] Weihai Key Laboratory of Intelligent Operation and Maintenance, Harbin Institute of Technology, Weihai, Shandong 264209, China

[d] School of Automotive Engineering, Harbin Institute of Technology, Weihai, Shandong 264209, China

[e] College of Aeronautical Engineering, Civil Aviation University of China, Tianjin 300300, China

Abstract:

The efficient and accurate state-of-health (SOH) estimation is crucial for reducing risks and ensuring effective application in battery management systems (BMS) of resource-limited devices. However, many recent state-of-the-art SOH estimation approaches rely on resource-consuming structures to obtain good performance. In this paper, an efficient deep learning model for SOH estimation, namely BMSFormer, is constructed. BMSFormer mainly integrates a Local-Global Fusion Attention module to capture both long-term and short-term dependencies while reducing computational complexity compared to traditional Softmax-based attention. Additionally, two kinds of depthwise separable convolution are embedded to fuse multi-scale and multi-channel features, enhancing feature diversity with fewer parameters than standard convolution. Three widely used battery datasets, each with different chemistries and operating conditions, are employed to evaluate the performance of BMSFormer. The experiments results illustrate that the proposed model achieves higher accuracy, lower computational consumption, and stabler performance across various hyperparameter combinations compared to alternative models.

Keywords: State of health; Lithium-ion batteries; Efficiency estimation; Local-Global Fusion Attention; Depthwise feature fusion;

Acknowledgments:

This work is supported by National Key R&D Program of China (2023YFB4302400).

corresponding author e-mail address: zhaomh@hit.edu.cn

1. Introduction

Nowadays, with the advantages of high energy density, low self-discharge rate and long service life, lithium-ion batteries have become one of the main energy storage devices in civil airlines, electric vehicles, mobile devices, aerospace and other domains [1]. Although, lithium-ion batteries bring convenience to people`s lives, they also pose latent safety hazards. These hazards primarily arise from the complicated degradation processes brought by complex internal chemical structure, different working conditions and environment. Therefore, precise monitoring and state estimation are essential to enhance battery safety, reliability, and performance.

As an essential guard of battery safe and efficient operation, battery management system (BMS) can record real-time data, manage cell balance, estimate internal status and provide emergency protection [2–4]. The internal states, such as state-of-health (SOH), cannot be measured directly. At the same time, due to the limited storage space of embedded devices, they often rely on online estimations using simple approaches, such as Kalman filter (KF) and its variants [5–7], classical physical models [8], equivalent circuit models (ECM) [9] and traditional machine learning (ML) algorithms like support vector machines (SVM) [10] and random forest (RF) [11]. However, on one hand, when faced with nonlinear and unstable data from online monitoring and historical cycles, traditional models struggle to provide high performance due to their structural constraints [12,13]; On the other hand, high-performance deep models are generally not feasible enough for convenient deployment on BMS due to their computationally expensive structures. Actually, since the emergence of AlexNet [14], the predominant trend in model architecture has been towards increasing depth and size, but many real-world applications need to be performed in real-time and/or on limited resource mobile devices [15]. Thereby, the high-performance deep models have a certain demand to be compact and exhibit low computational cost for practical applications in the future.

The recent SOH estimation approaches could be divided into two main categories: (1) model-based approaches, including electrochemical models and empirical models, (2) data-driven approaches, including machine learning and deep learning methods [16–19].

(1) Model-based approaches simulate the electrochemical principles in batteries using mathematical equations or circuit components [20]. Electrochemical models (*e.g.*, single-particle model and pseudo-two-dimensional model) utilize a series of differential equations to represent the electrochemical reaction processes in the battery. While these models may offer high accuracy, they often require extensive

computational resources, making online estimation challenging; and many parameters, such as particle radius and diffusion coefficients, are difficult to obtain [21,22]. In contrast, as a widely employed empirical model, equivalent circuit model (ECM), simulates battery internal behavior with circuit components such as resistors and capacitors [9,23]. ECMs (*e.g.*, Thevenin, PNGV, GNL and RC models) have relatively low computational complexity and easy parameter identification. However, they cannot fully capture battery internal changes, probably resulting in lower accuracy compared to electrochemical models.

(2) Data-driven approaches can be trained using historical data and mostly have lower dependencies on the internal degradation mechanism of batteries compared to model-based approaches [23]. They employ various algorithms to establish the relationships between input (health indicators, HIs) and output (state-of-health, SOH), then use new data for SOH estimation. Thereby, since a few years ago, many researchers tend to focus on establishing stronger data-driven models. For instance, Fei et al. [24] crafted 42 HIs from five kinds of main recorded data including voltage, current, temperature, internal resistance and capacity, and then fed them into five representative machine learning (ML) models including the Gaussian process regression (GPR) [25], support vector machine (SVM) [10], random forest (RF) [26], gradient boosting regression tree (GBRT) [27] and neural network (NN) [28]. Benefiting from the prosperity of deep learning (DL) methods, in recent years, many works employ convolutional neural networks (CNNs), recurrent neural networks (RNNs), and Transformers to estimate the states of batteries. For example, depending on the powerful local feature extraction ability of CNNs, Yang et al [29] utilized 1D-CNN to handle capacity degradation datasets, while Lee et al [30] transformed the data into a two-dimensional format and employed 2D-CNN. Additionally, dense neural network [31] was also attempted in similar research.

However, traditional CNNs are not adept at dealing with long time-series data, as this requires numerous convolutional layers to fuse local features that are distant from each other. RNNs and their variants such as gated recurrent unit (GRU) and long short-term memory (LSTM) attempt to overcome this challenge with hidden layers, state cells and various gates. These components function like a “nerve cell” that can storage information and pass it from one time step to the next for computation. Nonetheless, due to the sequential dependence, it is not convenient enough for traditional RNNs to make full use of modern multi-threaded GPU resources, even though recent works have achieved some improvements in computational efficiency like bidirectional long short-term memory (Bi-LSTM) [32,33], Attention-based Spatial-Temporal LSTM (ASTLSTM) [34], CNN-LSTM [35], CNN-ASTLSTM [36], and CNN-GRU [37], while the fundamental constraint of limited-sequential computation still remains.

To address the above problem, Vaswani [38] proposed the Transformer, a model architecture avoiding recurrence and instead relying on a self-attention mechanism to capture global dependencies between input and output. Due to its significant advantage in parallel computing and global modeling, Transformer has already achieved immense success across multiple domains [39–41]. Since Hannan et al. [42] applied the Transformer to estimate SOH of batteries in 2021, subsequent models have tended to increase in complexity. For instance, Gomez et al. [43] introduced an improved temporal fusion Transformer, integrating a Bi-LSTM encoder-decoder layer to enhance performance. Jia et al. [44] proposed a hybrid prediction model by combining bidirectional gated recurrent unit and Transformer. In another study, Gu et al. [45] suggested a novel approach using data pre-processing methods and a CNN-Transformer framework. Bai et al. [46] introduced a 2D-CNN-Transformer model using matrixing data, while Chen et al. [47] applied a vision-Transformer network (VIT) for SOH estimation, involving extra data flattening operations. Most existing methods do not attempt to simplify the structures of the Transformer-based networks, instead, they integrated it directly with other modules or models, resulting in computationally complicated approaches.

Three main problems faced by many recent approaches for battery SOH estimation, can be summarized as follows:

(1) **Computational Complexity Limitations:** In the field of battery state estimation, many existing deep learning models attempt to achieve higher accuracy by increasing the complexity of model structures or/and combining different classical models. Those approaches introduce numerous unnecessary parameter updates, thereby increasing computational resource demands. Such complexity may hinder future applications in resource-constrained devices, such as BMS.

(2) **Estimation Accuracy Challenges:** Many current approaches face accuracy challenges due to inherent computational constraints within the model itself, insufficient representational capabilities of the extracted health indicators, or the input of many weakly correlated health indicators, which together negatively impact the accuracy of SOH estimation results.

(3) **Performance Stability Issues:** The training and optimization process often require extensive hyperparameter tuning to achieve higher performance, which consumes a lot of time and computational resources before reaching the desired goals. Such consumption can be effectively reduced if the model is able to maintain stable and high performance across various hyperparameter configurations. However, most existing methods only present their optimal results without demonstrating the model's ability to solve this problem.

To address these challenges, this article develops an efficient deep learning model, namely BMSFormer, for SOH estimation of lithium-ion batteries, with the goal of optimizing computational resource usage, balancing accuracy and efficiency, and enhancing model stability for reliable performance. The main contributions are summarized as follows:

(1) By progressively shortening the window size and step size, health indicators (HIs) with higher correlations to the battery health state are identified within two high-frequency SOC charge and discharge segments. This approach successfully achieves Pearson correlation coefficient (PCC) values averaging over 0.99 across three different kinds of battery.

(2) A Local-Global Fusion Attention module is constructed to effectively capture both short-term and long-term features while reducing computational complexity compared to traditional Softmax-based attention mechanisms.

(3) Two kinds of depthwise separable convolution module with small kernel size (DSConv-S) and large kernel size (DSConv-L) are embedded to fuse multi-scale and multi-channel features, enhancing feature diversity with fewer parameters than standard convolution.

The remaining sections of this paper are arranged as follows. Section2 describes the major steps of the developed SOH estimation approach and the basic information of three different kinds of battery datasets. Section 3 introduces the structure and modules of BMSFormer. Section 4 presents comprehensive validations and compares BMSFormer with four prevailing deep learning models. Finally, the conclusions are summarized in Section 5.

2. Preliminaries

2.1. Overview of the developed SOH estimation approach

The SOH is a critical metric for evaluating health status of batteries. While there is still no standardized definition for SOH, battery capacity remains the most utilized representation [31], as illustrated below:

$$\mathrm{SOH}=\frac{C_{\mathrm{current}}}{C_{\mathrm{rated}}}\times 100\%, \tag{1}$$

where C_{current} and C_{rated} are the current capacity and rated capacity, respectively.

As illustrated in Fig. 1, the developed SOH estimation approach consists of four steps:

(1) Data acquisition. Three types of batteries with different chemical materials were subjected to full lifecycle aging tests under various charge/discharge protocols to evaluate the proposed model. The

high frequency of the intermediate SOC during daily battery operation, charge segment from 3.8 V to 4.2 V and discharge segment from 3.8 V to 3.4 V are selected for HI extraction.

(2) Feature engineering. The constant current charge and discharge times for each cycle are extracted as health indicators (HIs). Starting from the selected segments, the HI search process is refined using progressively smaller windows and steps until a 0.01V window is reached or no higher Pearson correlation coefficient (PCC) value is found. A sliding window is then used to divide the time series of HIs into subsets, moving forward one step at a time. The true SOH at the next step of each window serves as the label for its corresponding subset.

(3) Model training. For convenience, Cell1, B0005, and CS2-35 are used as training sets, with the first 30% of the data employed for training across 384 hyperparameter combinations including epochs, learning rate, number of blocks and layers, and dimension of dense and embedding layer. The remaining 70% data is used for validation and comparison to select the best-performing model. Then the best perform model is tested directly on the entire data of other batteries in the corresponding dataset to evaluate the model generalization and compare the performance of BMSFormer with four different deep learning models.

(4) Model evaluation. The accuracy, efficiency, and stability of different models are evaluated. Four typical evaluation metrics are used to measure the accuracy, four common computational complexity metrics are used to evaluate the training efficiency, and the R^2 results of 384 hyperparameter combinations are used to assess the stability of training results.

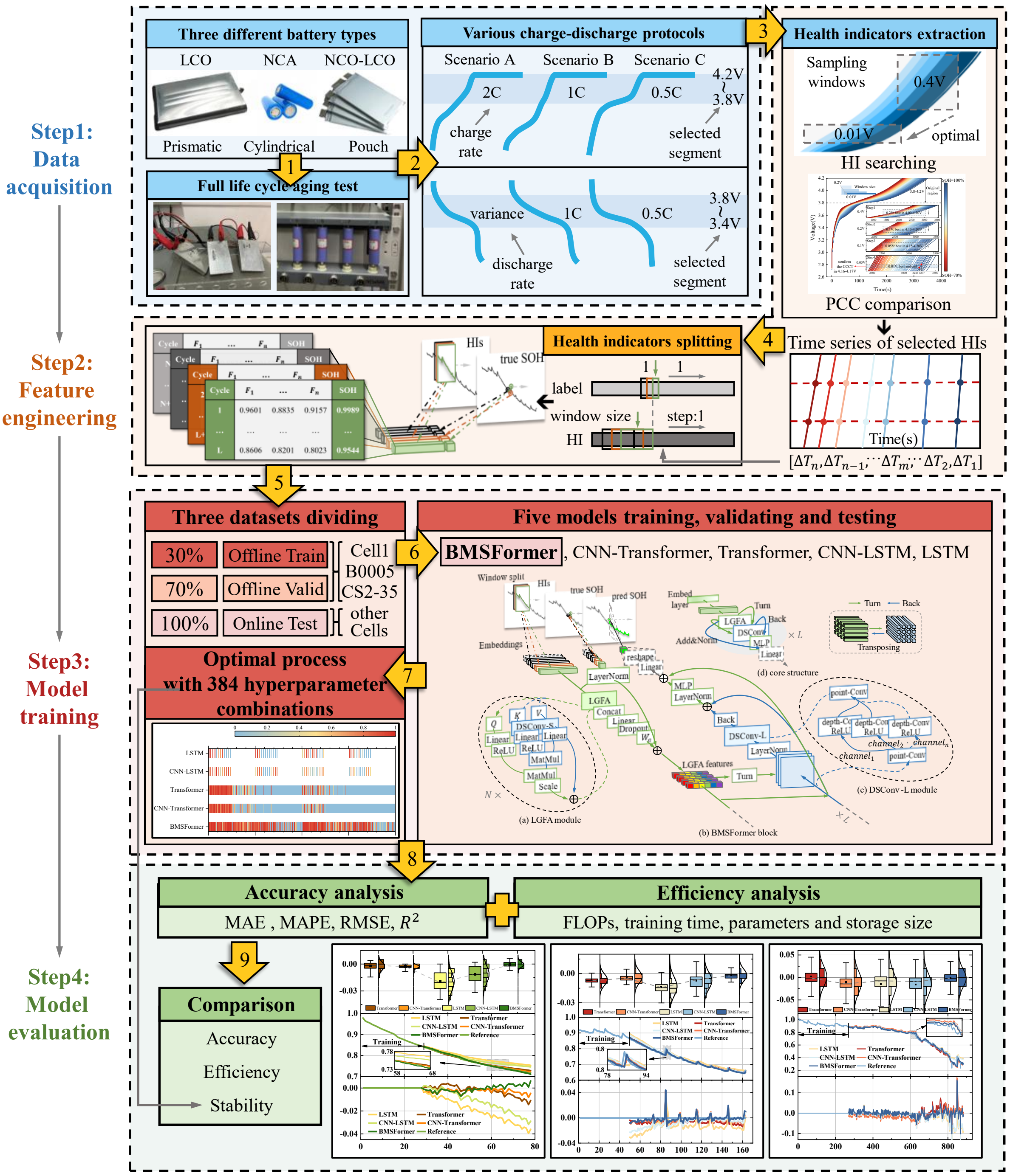


Fig 1. Flowchart of developed SOH estimation approach

2.2. Typical battery datasets acquisition

Three kinds of prevailing battery datasets from Oxford, NASA and CALCE are used in this paper, which are widely recognized and originate from leading battery research institutions, ensuring standardized experimental protocols and reliable data. They encompass various battery materials, specifications, and work conditions. Hence, the applicability and precision of the proposed approach can be evaluated through evaluation on a diverse range of batteries with different materials and subjected to various operating conditions. The main characteristics of battery datasets used in this work are summarized in Table 1; Figs. 2(a)-(c) illustrates the capacity degradation and increment phenomenon of three different batteries under various work conditions, while Figs. 2(d)-(f) displays the constant-current charge voltage curves in several cycle of the first battery from each dataset. Similar trends are observed within each battery type and distinct differences are noticeable between different types of batteries.

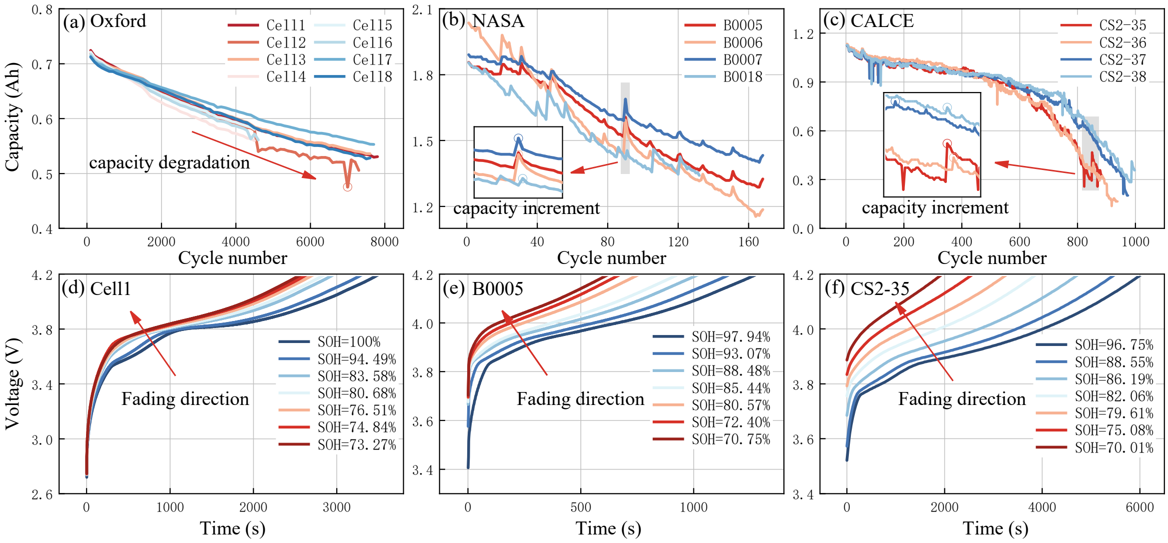


Fig. 2. Three different kinds of battery datasets: (a)-(c) capacity degradation and temporary increment curves of the Oxford, NASA and CALCE cells, respectively. (d)-(f) constant current charge voltage curves of Oxford Cell1, NASA B0005 and CALCE CS2-35, respectively.

2.2.1. Oxford lithium battery dataset

The Oxford lithium-ion battery dataset is provided by the Battery Intelligence Laboratory at the University of Oxford including degradation data from eight Kokam SLPB533459H4 lithium-ion pouch cells, each with a 0.74 Ah nominal capacity and test at 40℃. The negative electrode material of the Kokam pouch cells is graphite, the positive electrode material is a blend of lithium cobalt oxide (LCO) and lithium nickel cobalt oxide (NCO). The eight batteries are subjected to a constant-current and then constant-voltage charging profile up to 4.2 V, followed by a variance discharge current rate from the urban Artemis-derived profile process down to 2.7 V.

2.2.2. NASA lithium battery dataset.

The NASA lithium battery dataset is provided by Prognostics Center of Excellence Data Set Repository-NASA. Four Lithium-ion 18650 cylindrical batteries (#5, 6, 7, and 18) from LG Chem, each with 2Ah nominal capacity, undergo three operational profiles at room temperature: charging at 1.5A (0.75C) in CC mode to 4.2V, followed by CV mode until the current fell to 20mA; discharging at 2A in CC mode down to 2.7V, 2.5V, 2.2V and 2.5V, respectively. The aging process is accelerated through repeated cycles, with EOL defined as a 30% capacity loss to 1.4Ahr. The positive electrode material is lithium nickel cobalt aluminum oxide (NCA).

2.2.3. CALCE lithium battery dataset.

The CALCE lithium-ion battery dataset originates from the Center for Advanced Life Cycle Engineering (CALCE) at the University of Maryland. These batteries have a rated capacity of 1.1 Ah and utilize lithium cobalt oxide (LiCoO2) as the cathode material. The aging tests for battery life are performed at room temperature. The charge strategy is to charge at a constant-current rate of 0.5 C (0.55A) until the voltage reaches 4.2 V and then hold at 4.2 V until the charging current drops below 0.02 A, then the batteries are discharged with a constant current of 2 A until the voltage decreases to 2.7 V.

Table 1. Description of three battery datasets.

Data Sources	Oxford [48]	NASA[49]	CALCE [50]
Manufacturer	Kokam	LG Chem	LG Chem
Cell types	Pouch	18650 Cylindrical	Prismatic
Cathode material	NCO-LCO	NCA	LCO
Selected battery series	Cell (1 to 8)	B00 (05 to 07 and 18)	CS2 (35 to 38)
Battery number	8	4	4
Rated capacity (Ah)	0.74	2	1.1
Capacity failure threshold (Ah)	70%	1.4 (70%)	0.84 (76.4%)
Charge protocol (rate)	CC (2C)	CC-CV (0.75C,4.2V)	CC-CV (0.5C,4.2V)
Discharge protocol (rate)	variance	CC (1C)	CC (1C)
Cut-off voltage (V)	charge to 4.2 discharge to 2.7	charge to 4.2 discharge to 2.7/2.5/2.2/2.5	charge to 4.2 discharge to 2.7
Cut-off current (A)	-	charge to 0.02	charge to 0.05
Experiment temperature (℃)	40	24	24
Cycle numbers	8200/7700/8100/5100/5000/5000/8100/8100	168/168/168/132	881/935/971/995

2.3. Health indicators extraction

The selected health indicators (HIs) are inspired by a range of references [51–58] serving various research purposes. Considering the high frequency of the intermediate SOC during daily battery operation, charge segment from 3.8 V to 4.2 V and discharge segment from 3.8 V to 3.4 V are selected for HI extraction.

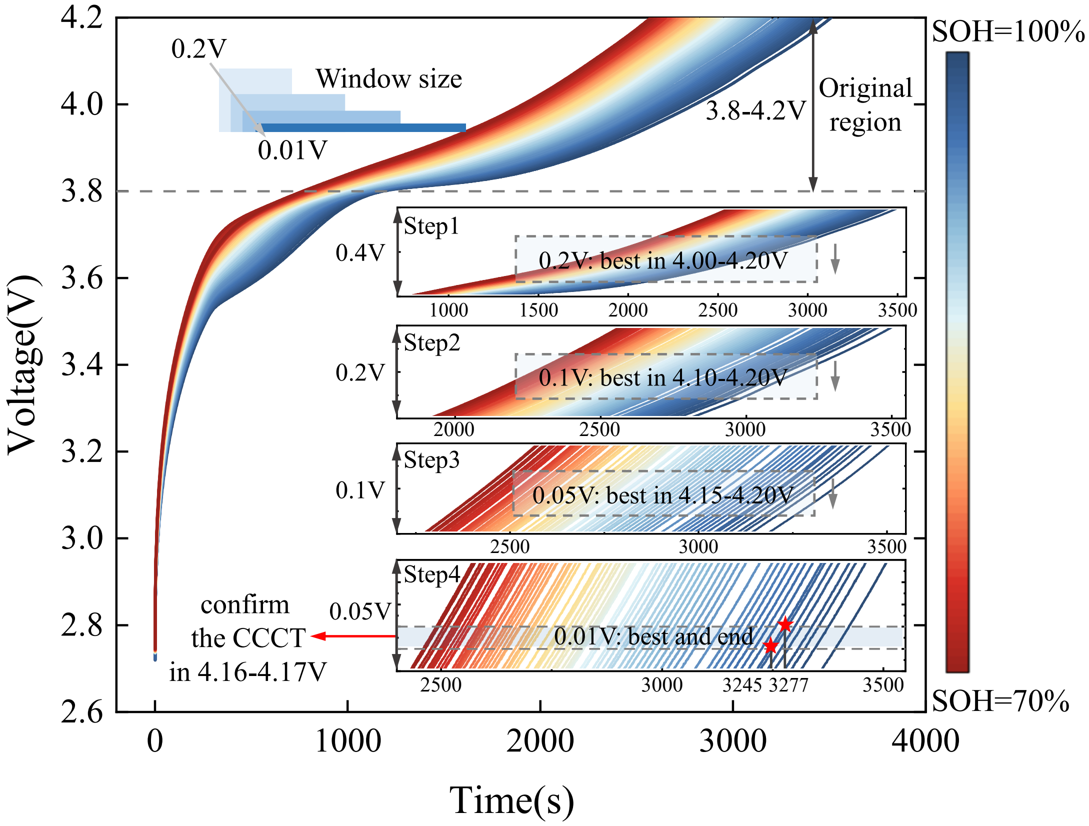


Fig. 3. The extraction procedures of the selected HI in the selected charge segment.

As shown in Fig. 3, in the first step of our HIs extraction procedures, a 0.2V window size with a 0.2V moving step size is utilized to divide the selected charge or discharge segment into shorter segments: 3.8-4.0V, 3.9-4.1V, and 4.0-4.2V in charge segment (3.8-3.6V, 3.7-3.5V, 3.6-3.4V in discharge segment). Then the minimum Pearson correlation coefficient (PCC) value across all batteries of each dataset is taken as the baseline, and the segment with the highest baseline value is selected as the search segment for the next step. In the 2nd, 3rd, and 4th steps, the search procedures are repeated with progressively smaller windows and steps until a 0.01V window is reached or no higher baseline is identified than in the previous step. Finally, the constant current charge time (CCCT) series from 4.16V to 4.17V and the constant current discharge time (CCDT) series from 3.8V to 3.4V are respectively confirmed as the selected HIs.

The PCC is a widely utilized statistical metric for evaluating the strength of linear association between HIs and true SOH [59,60]. The value of the PCC ranges from -1 to 1, with the absolute value closer to 1 indicating a stronger relationship. The PCC formula is shown as follows:

$$\mathrm{PCC}=\frac{\sum_{k}^{n}(x_k-\bar{x})(y_k-\bar{y})}{\sqrt{\sum_{k}^{n}(x_k-\bar{x})^2\sum_{k}^{n}(y_k-\bar{y})^2}} \tag{2}$$

where x, y, $\bar{x}$, $\bar{y}$ and n represent the value of the selected HI sequence and the battery SOH sequence, the average value of x and y, the number of samples, respectively.

As shown in Table 2, the selected HI, which is the constant current charging time from 4.16 to 4.17V, has the highest PCC compared to others of recent methods. Similarly, as shown in Table 3, the selected HI, which is the constant current discharge time from 3.8 to 3.4V, has the highest PCC compared to others.

Table 2. PCC comparison of different HIs on Oxford datasets.

Battery series	selected HI	CCCT from ICA [55]	CCCT from DTA [56]	CCCQ [57]	CCDQ [55]	IC peak [58]	sample entropy [55]	Discharge voltage curve slope [58]
Cell1	**0.99996**	0.9999	0.9989	0.99947	0.9998	0.9444	-0.9853	0.9999
Cell2	**0.99999**	0.9999	0.9959	0.98677	0.9999	-	-0.9839	-
Cell3	**0.99996**	0.9999	0.9993	0.99948	0.9989	0.9375	-0.9876	0.9998
Cell4	**0.99998**	0.9104	-	*0.99981*	0.9999	-	-0.9885	-
Cell5	**0.99992**	0.9999	-	0.99960	0.9999	-	-0.9133	-
Cell6	**0.99998**	0.9999	-	0.99961	0.9991	-	-0.9848	-
Cell7	**0.99999**	0.9999	-	0.99973	0.9999	0.9296	-0.9896	0.9998
Cell8	**0.99999**	0.9999	-	0.99970	0.9999	0.9378	-0.9908	0.9998

Table 3. PCC comparison of different HIs on NASA and CALCE datasets.

Battery series	selected HI	CCCT [61]	CCCQ [57]	Max projection distance of start and end charge voltage [62]	IC peak [58]	Relax time between charge and discharge cycle [63]
B0005	**0.9934**	0.9703	0.94268	0.95	0.9910	-0.8941
B0006	**0.9858**	0.9366	0.94518	0.97	0.9800	-0.9373
B0007	**0.9983**	0.9745	0.92037	0.99	0.9251	-0.9049
B0018	**0.9987**	-	0.81248	-	-	-
CS2-35	**0.9960**	0.9799	0.90290	0.95	-	-0.9843
CS2-36	**0.9968**	0.9827	0.93370	0.99	-	-0.9829
CS2-37	**0.9943**	0.9789	0.93920	0.98	-	-
CS2-38	**0.9952**	0.9748	-	0.96	-	-

3. The developed BMSFormer

In this section, the overall architecture and workflow of BMSFormer are first introduced. Then, the designed Multi-scale Depthwise Separable Convolution modules, including DSConv-L and DSConv-S, are described. Following that, the proposed Local-Global Fusion Attention (LGFA) module is detailed. Finally, several foundational modules are briefly introduced.

3.1. Architecture overview

In this research, to solve the problem that the accuracy and efficiency of the current models often cannot be optimal together, a lightweight network termed BMSFormer with both global features and local information capturing capabilities is proposed.

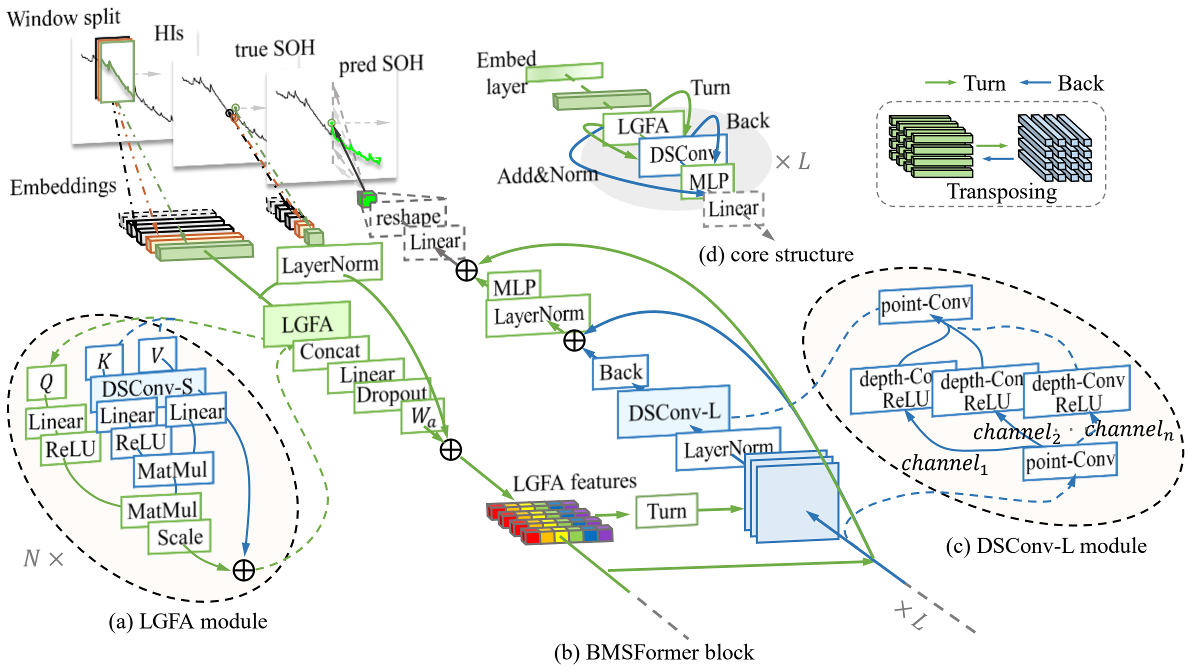


Fig. 4. Framework of BMSFormer. (a) LGFA module, (b) BMSFormer block, (c) DSConv-L module, (d) BMSFormer core structure.

The entire flowchart is illustrated in Fig. 4 and described as follows: HIs are taken as input, segmented into fragments via window splitting, embedded in a high-dimensional space, and then fed into a BMSFormer block, which includes the LGFA module and DSConv-L module. The output from LGFA module is transposed, passed through a DSConv-L module, and transposed back. Finally, the output from all BMSFormer blocks is fed into a Multilayer Perceptron (MLP) layer before outputting the result. During the training process, the true SOH value right following each window segment is utilized as a label, which guides the model to adjust the gradient descent optimization.

Mathematically, this means, for the input x_i of BMSFormer block i, the corresponding output y_i of the block is calculated using

$$x_i' = W_a \cdot \mathrm{LGFA}(x_i) + \mathrm{LN}(x_i), \tag{3}$$

$$x'' = \mathrm{Back}\left(\text{DSConv-L}\left(\mathrm{LN}\left(\mathrm{Turn}\left(x_i'\right)\right)\right)\right) + x_i', \tag{4}$$

$$y_i = \mathrm{MLP}\left(\mathrm{LN}\left(x_i''\right)\right) + x_i', \tag{5}$$

where LGFA, LN, DSConv-L, MLP respectively represent the Local-Global Fusion Attention, layer normalization, depthwise separable convolution with large kernel size, and multilayer perceptron. Turn and Back refer to the transposing operation of tensor, and W_a is a learnable weight parameter.

3.2. The designed multi-scale Depthwise Separable Convolutional modules

In this section, the fundamental structure of DSConv is introduced and compared to standard convolution, then the multi-scale properties of DSConv-S and DSConv-L are elaborated subsequently.

3.2.1. Fundamental structure of DSConv-S and DSConv-L modules

Convolutions operations are adept in extracting local information through sliding filters [30,46,64]. Standard convolution operates across all input channels, with each convolutional kernel generating a feature map as one output channel. Therefore, the consumption of computational resources and training time will significantly increase when dealing with large time-series datasets. The standard convolutions have the computational cost of:

$$k \times k \times N_{\mathrm{in}} \times N_{\mathrm{out}} \times N_F \times N_F \tag{6}$$

where the k, N_{in}, N_{out}, N_F represent the kernel size, input channels, output channels, and the feature map size, respectively.

Compared to standard convolution, firstly, in general, depthwise separable convolution utilizes smaller filter than standard convolution [15,65–68], as illustrated in the upper left area of Fig. 5. Secondly, depthwise convolution separates the features map into multiple channels, and each channel is convolved with its own set of filters. These two operations not only allow for the extraction of a variety of features but also significantly decreases the computational load. Thirdly, the outputs from the depthwise step are then combined using 1×1 filter of pointwise convolution. This step mixes the features across all channels, allowing for learning interactions between them. Finally, the residual connection helps in mitigating the vanishing gradient problem and allows the network to learn more complex functions.

DSConv separates the spatial and channel-wise operations, reducing computational complexity by first applying a depthwise convolution across spatial dimensions, followed by a pointwise convolution to adjust the number of output channels:

$$k \times k \times N_{\text{in}} \times N_F \times N_F + N_{\text{in}} \times N_{\text{out}} \times N_F \times N_F \tag{7}$$

where k, N_{in}, N_{out}, N_F represent the kernel size, input channels, output channels, and the feature map size, respectively.

By breaking convolution operation into a two-step process of depthwise filtering followed by pointwise combination, a reduction is achieved in computation of:

$$\begin{aligned} &\frac{k \times k \times N_{\text{in}} \times N_F \times N_F + N_{\text{in}} \times N_{\text{out}} \times N_F \times N_F}{k \times k \times N_{\text{in}} \times N_{\text{out}} \times N_F \times N_F} \\ &= \frac{1}{N_{\text{out}}} + \frac{1}{k^2} \end{aligned} \tag{8}$$

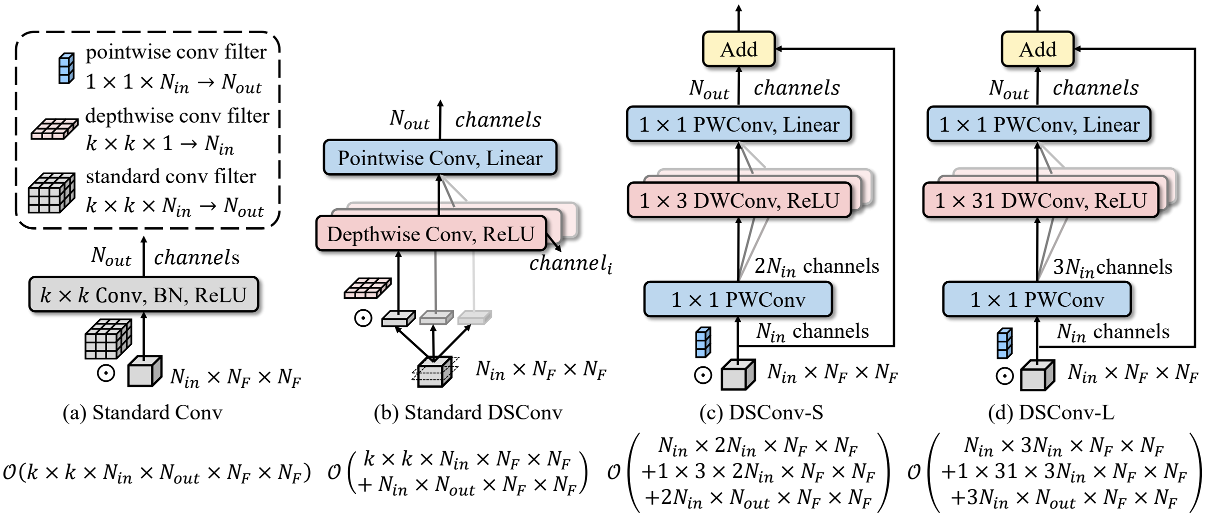

pointwise conv filter
$1 \times 1 \times N_{in} \rightarrow N_{out}$
depthwise conv filter
$k \times k \times 1 \rightarrow N_{in}$
standard conv filter
$k \times k \times N_{in} \rightarrow N_{out}$
N_{out} channels
$k \times k$ Conv, BN, ReLU
$\odot$
$N_{in} \times N_F \times N_F$
(a) Standard Conv
$O(k \times k \times N_{in} \times N_{out} \times N_F \times N_F)$
N_{out} channels
Pointwise Conv, Linear
Depthwise Conv, ReLU
$channel_i$
$\odot$
$N_{in} \times N_F \times N_F$
(b) Standard DSConv
$O\begin{pmatrix} k \times k \times N_{in} \times N_F \times N_F \\ + N_{in} \times N_{out} \times N_F \times N_F \end{pmatrix}$
Add
N_{out} channels
1×1 PWConv, Linear
1×3 DWConv, ReLU
$2N_{in}$ channels
1×1 PWConv
N_{in} channels
$\odot$
$N_{in} \times N_F \times N_F$
(c) DSConv-S
$O\begin{pmatrix} N_{in} \times 2N_{in} \times N_F \times N_F \\ +1 \times 3 \times 2N_{in} \times N_F \times N_F \\ +2N_{in} \times N_{out} \times N_F \times N_F \end{pmatrix}$
Add
N_{out} channels
1×1 PWConv, Linear
1×31 DWConv, ReLU
$3N_{in}$channels
1×1 PWConv
N_{in} channels
$\odot$
$N_{in} \times N_F \times N_F$
(d) DSConv-L
$O\begin{pmatrix} N_{in} \times 3N_{in} \times N_F \times N_F \\ +1 \times 31 \times 3N_{in} \times N_F \times N_F \\ +3N_{in} \times N_{out} \times N_F \times N_F \end{pmatrix}$

Fig. 5. The fundamental structure of: (a) Standard convolution; (b) Standard Depthwise separable convolution; (c) DSConv-S module; (d) DSConv-L module.

3.2.2. Computational analysis of DSConv-S and DSConv-L modules

The depthwise separable convolution with small kernel size (DSConv-S) and large kernel size (DSConv-L) are designed to extract multi-scale and multi-channel feature, aiming to addresses the expressiveness limitations of ReLU attention, which is constrained by its linear attention mechanism, while maintaining high computational efficiency.

As illustrated in Fig. 5, with a double expansion factor for input channels, the DSConv-S module applies a 1×3 depthwise convolutional kernel size to enhance the quality of input information and the sensitivity of attention for local variety. This module further incorporates two layers of 1×1 pointwise convolutions to integrate multi-channel features, further improving feature representation. The computational cost of DSConv-S can be expressed as:

$$N_{\text{in}} \times 2N_{\text{in}} \times N_F \times N_F + 1 \times 3 \times 2N_{\text{in}} \times N_F \times N_F + 2N_{\text{in}} \times N_{\text{out}} \times N_F \times N_F \tag{9}$$

where $2N_{\text{in}}$ and $N_F \times N_F$ represents the input channels after expanding and the feature map size, respectively.

With a triple expansion factor for input channels, the DSConv-L module applies a 1×31 depthwise convolutional kernel size to extract long-term features, aiming to further enhance the feature diversity and model generalization. This module also incorporates two layers of 1×1 pointwise convolutions. Therefore, the computational cost of DSConv-L can be expressed as:

$$N_{\text{in}} \times 3N_{\text{in}} \times N_F \times N_F + 1 \times 31 \times 3N_{\text{in}} \times N_F \times N_F + 3N_{\text{in}} \times N_{\text{out}} \times N_F \times N_F \tag{10}$$

where $3N_{\text{in}}$ and $N_F \times N_F$ represents the input channels after expanding and the feature map size, respectively.

3.3. The proposed Local-Global Fusion Attention module

In this section, the general form of self-attention mechanism in Transformers is described, the advantages and disadvantages of traditional Softmax and linear attention are briefly analyzed, and two main superiorities of proposed Local-Global Fusion Attention (LGFA) module, including global receptive field and local information sensitivity are introduced subsequently.

Most existing SOH estimation methods still rely on traditional Softmax attention [45–47], suffering from excessive computational complexity. Meanwhile, although the linear attention reduces the complexity, it continues to struggle with insufficient model expressiveness. Therefore, the Local-Global Fusion Attention (LGFA) module is proposed to integrate linear complexity and high expressiveness, which are potentially beneficial for efficient and accurate SOH estimation.

3.3.1. General form of multi-head self-attention

To improve the model's representational capacity, multi-head self-attention processes the input through several parallel "heads", each representing an independent self-attention mechanism. Each head calculates its own attention weights and generates a unique attention representation. The self-attention mechanism computes the relationships (attention weights) between each element in the input and all other elements. It then produces a weighted sum of these relationships to create new representations. This allows the model to focus on important information in the sequence, especially long-range dependencies.

In $i-\text{th}$ head, given an input $X \in \mathbb{R}^{N\times C}$, the general form of self-attention can be formulated as follows:

$$Q = XW_Q, \tag{11}$$

$$K = XW_K, \tag{12}$$

$$V = XW_V, \tag{13}$$

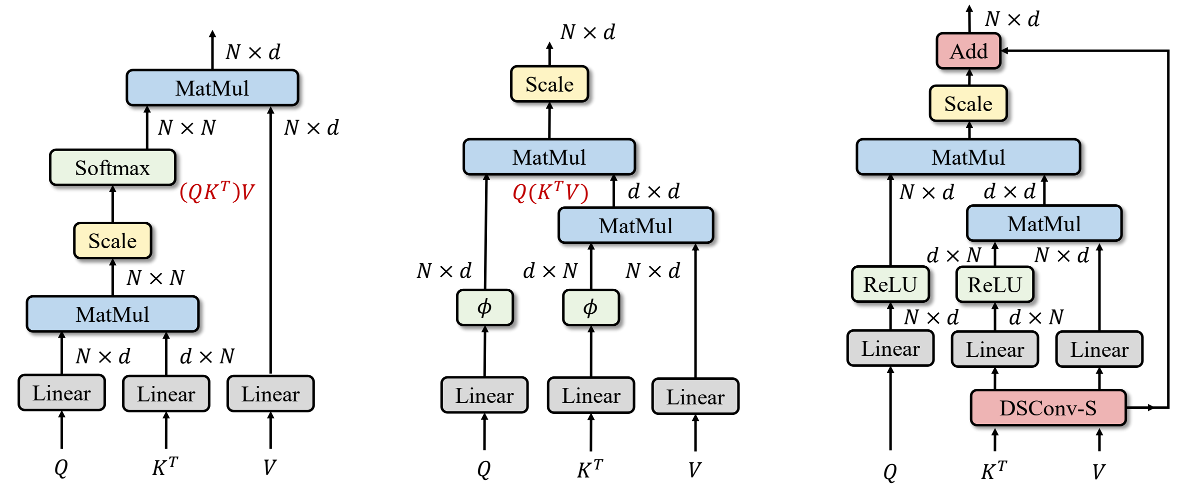
$$O_i = \sum_{j=1}^{N} \frac{Sim\left(Q_i, K_j\right)}{\sum_{j=1}^{N} Sim\left(Q_i, K_j\right)} V_j, \tag{14}$$

where W_Q, W_K, and $W_V \in \mathbb{R}^{C\times d}$ are the learnable linear projection matrices, C and d are the channel dimensions of module and each head. $Sim(\cdot,\cdot)$ is the similarity function of self-attention mechanism.

3.3.2. Softmax attention mechanism and linear attention mechanism

Transformers mainly utilize Softmax Attention [38] with the $Sim\left(Q, K\right) = \exp(QK^T / \sqrt{d})$ in Eq. (14). However, the attention map is obtained by computing the similarity between all query-key pairs, leading to the computation complexity of $\mathcal{O}\left(N^2\right)$ as illustrated in Fig. 6(a). In contrast, linear attention [69], as shown in Fig. 6(b), efficiently reduces the computation complexity by carefully designed similarity functions with $Sim\left(Q, K\right) = \phi\left(Q\right)\phi\left(K\right)^T$, which changes the computation order from $\left(\phi(Q)\phi(K)^T\right)V$ to $\phi(Q)\left(\phi(K)^T V\right)$ base on the associative property of matrix multiplication. The computation complexity with respect to the number of tokens in the sequence is reduced to $\mathcal{O}(N)$, where N represents the sequence length. This linear complexity allows the model to efficiently handle long sequences, which is particularly beneficial for time series analysis.

Traditional Softmax attention amplifies the differences between all query-key pairs using the exponential function, significantly enhancing larger similarity values when calculating attention weights. Although Softmax attention provides stronger expressive power, its computational cost is a bit unbeneficial to its application in large-scale sequences. In contrast, linear attention uses different kernel functions $\phi(\cdot)$ to map Q and K into a new space, thereby enhancing the differences. While this may result in some loss of expressiveness, the new calculation order of the weight matrix in linear attention reduces computational complexity, which offers an advantage in processing long time-series sequences like battery SOH estimation.



(a) Softmax Global Attention $O(N^2d)$ (b) Linear Attention $O(Nd^2)$ (c) Local-Global Fusion Attention $O(Nd^2)$

Fig. 6. Difference between (a) traditional Softmax attention mechanism, (b) traditional Linear Attention mechanism, and (3) the proposed Local-Global Fusion attention module.

3.3.3. The proposed attention module

As illustrated in Fig. 6(c), the proposed Local-Global Fusion Attention (LGFA) module utilizes the Rectified Linear Unit attention mechanism and enhances feature diversity through the DSConv-S module introduced in Section 3.2.1. The similarity function Eq. (14) of this attention mechanism is defined as:

$$Sim(Q,K) = \mathrm{ReLU}(Q)\mathrm{ReLU}\left(\text{DSConv-S}(K)^{T}\right) \tag{15}$$

where the self-attention module can be rewritten as:

$$O_i = \frac{\sum_{j=1}^{N}\left[\mathrm{ReLU}(Q_i)\mathrm{ReLU}\left(\text{DSConv-S}(K_j)^{T}\right)\right]\text{DSConv-S}(V_j)}{\mathrm{ReLU}(Q_i)\sum_{j=1}^{N}\mathrm{ReLU}\left(\text{DSConv-S}(K_j)^{T}\right)} \tag{16}$$

Then, by using the associative property of matrix multiplication, the computational complexity can be reduced from quadratic to linear while maintaining the same functionality:

$$\begin{aligned} O_i &= \frac{\sum_{j=1}^{N}\left\{\mathrm{ReLU}(Q_i)\mathrm{ReLU}\left[\text{DSConv-S}(K_j)^{T}\right]\right\}\text{DSConv-S}(V_j)}{\mathrm{ReLU}(Q_i)\sum_{j=1}^{N}\mathrm{ReLU}\left[\text{DSConv-S}(K_j)^{T}\right]} \\ &= \frac{\mathrm{ReLU}(Q_i)\left\{\sum_{j=1}^{N}\mathrm{ReLU}\left[\text{DSConv-S}(K_j)^{T}\right]\text{DSConv-S}(V_j)\right\}}{\mathrm{ReLU}(Q_i)\sum_{j=1}^{N}\mathrm{ReLU}\left[\text{DSConv-S}(K_j)^{T}\right]} \end{aligned} \tag{17}$$

As shown in Eq. (17), the computation of $\sum_{j=1}^{N}\mathrm{ReLU}(K_j)^{T}V_j \in \mathbb{R}^{d\times d}$ and $\sum_{j=1}^{N}\mathrm{ReLU}(K_j)^{T} \in \mathbb{R}^{d\times 1}$ are performed only once, after which these results can be reused for each query. This approach thereby requires only $\mathcal{O}(N)$ computational cost and $\mathcal{O}(N)$ storage memory.

Another key advantage of ReLU-based global attention is that it prevents the vanishing gradient problem and involves simple operation $\max(0, x)$, making it faster to compute than Softmax, which requires exponentiation and normalization. For instance, some previous latency comparisons [70] show that, with similar computational demands, ReLU-based linear attention outperforms Softmax attention, particularly on mobile devices.

4. Experiments and analysis

This section presents a comprehensive evaluation of five models (BMSFormer, CNN-Transformer, CNN-LSTM, Transformer and LSTM) across three public battery datasets. The Cell1, B0005, and CS2-35 are employed as training sets, with the first 30% of the data utilized for training across 384 hyperparameter combinations, including variations in epochs, learning rate, number of blocks and layers, and the dimensions of the dense and embedding layers. The remaining 70% of the data is reserved for validation and comparison to identify the best-performing model parameters. The selected model is then directly tested on the entire datasets of other batteries to assess its generalization capability and to compare the performance of BMSFormer against the alternative models. Moreover, the robustness of the five models is compared through a detailed analysis of the 384 training results, without any further hyperparameter refinement.

The training process is conducted on an ASUS TUF A15 computer, which is equipped with an AMD Ryzen R7-4800H (2.90GHz), an NVIDA GeForce RTX 2060 and 24GB of RAM. The operation system on this machine is Windows 11 Professional. The prediction models are built using PyTorch 2.0.0, utilizing Python 3.9.16 as the programming language.

4.1. The evaluation criteria

Several widely adopted evaluation metrics are selected to assess the models' performance: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), the coefficient of determination R^2, and the Average Root Mean Squared Error (ARMSE). MAE measures the average magnitude of the errors in predictions without considering their direction, which refers to whether the predicted values are above or below the actual values, providing a straightforward measure of prediction accuracy. MAPE expresses accuracy as a percentage, making it useful for comparing performance across different datasets. RMSE is the square root of the average squared differences between predicted and actual values, giving larger weight to larger errors. The R^2 score indicates the proportion of variance in the dependent variable predictable from the independent variables, with larger values showing better performance. ARMSE, the average of RMSE values over several subsets or folds, provides a stable measure of model performance across different data partitions. The formulas for these metrics are as follows:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{18}$$

$$\text{MAPE}=\frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i-\hat{y}_i}{y_i}\right| \tag{19}$$

$$\text{RMSE}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2} \tag{20}$$

$$R^2=1-\frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{\sum_{i=1}^{n}(y_i-\overline{y})^2} \tag{21}$$

$$\text{ARMSE}=\frac{1}{m}\sum_{j=1}^{m}\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i-\hat{y}_i\right)^2} \tag{22}$$

where y, $\hat{y}$, $\overline{y}$, n and m are the actual SOH, predicted SOH, the average value of y, the number of testing samples, and the number of batteries, respectively [71].

4.2. Comparison on Oxford dataset

4.2.1. Accuracy and generalization comparison

Comparative experiments are conducted between BMSFormer and four alternative models. The layer configurations of these models are presented in Table 4.

Table 4. The layer configurations of different deep learning models

BMSFormer	CNN-Transformer	Transformer	CNN-LSTM	LSTM
Input	Input	Input	Input	Input
Embed layer (32)	Embed layer (32)	Embed layer (32)	Conv1D (1×3, 1) ReLU	Linear
LGFA module	Positional Encoder	Positional Encoder	MaxPooling1D (2×2, 2)	LSTM layer (32)
DSConv-L module	Softmax Attention	Softmax Attention	Conv1D (1×3, 1) ReLU	LSTM layer (32)
Add&Norm	Conv1D (1×3, 1) ReLU	Add&Norm	LSTM layer (32)	LSTM layer (32)
MLP	MaxPooling1D (2, 2)	Softmax Attention	LSTM layer (32)	Linear
Add&Norm	Conv1D (1×3, 1) ReLU, Add&Norm	Add&Norm	LSTM layer (32)	
Linear	Softmax Attention	MLP	Linear	
	Add&Norm	Add&Norm		
	MLP			
	Add&Norm			
	Linear			

Fig. 7 and Table 5 show the estimation results and errors of BMSFormer, CNN-Transformer, CNN-LSTM, Transformer and LSTM on the Oxford dataset. As shown in the figure, in the validation results for Cell1, CNN-Transformer's estimation error IQR is close to zero. However, the asymmetry in the whiskers of the boxplot indicates a greater number of low outliers in the estimation results. This trend is also observed from Cell2 to Cell8, suggesting that the model's estimation stability may be insufficient. The boxplot of prediction errors and the normal distribution plot across the eight batteries clearly show that BMSFormer has smaller and more concentrated prediction errors, illustrating superior prediction accuracy and robustness.

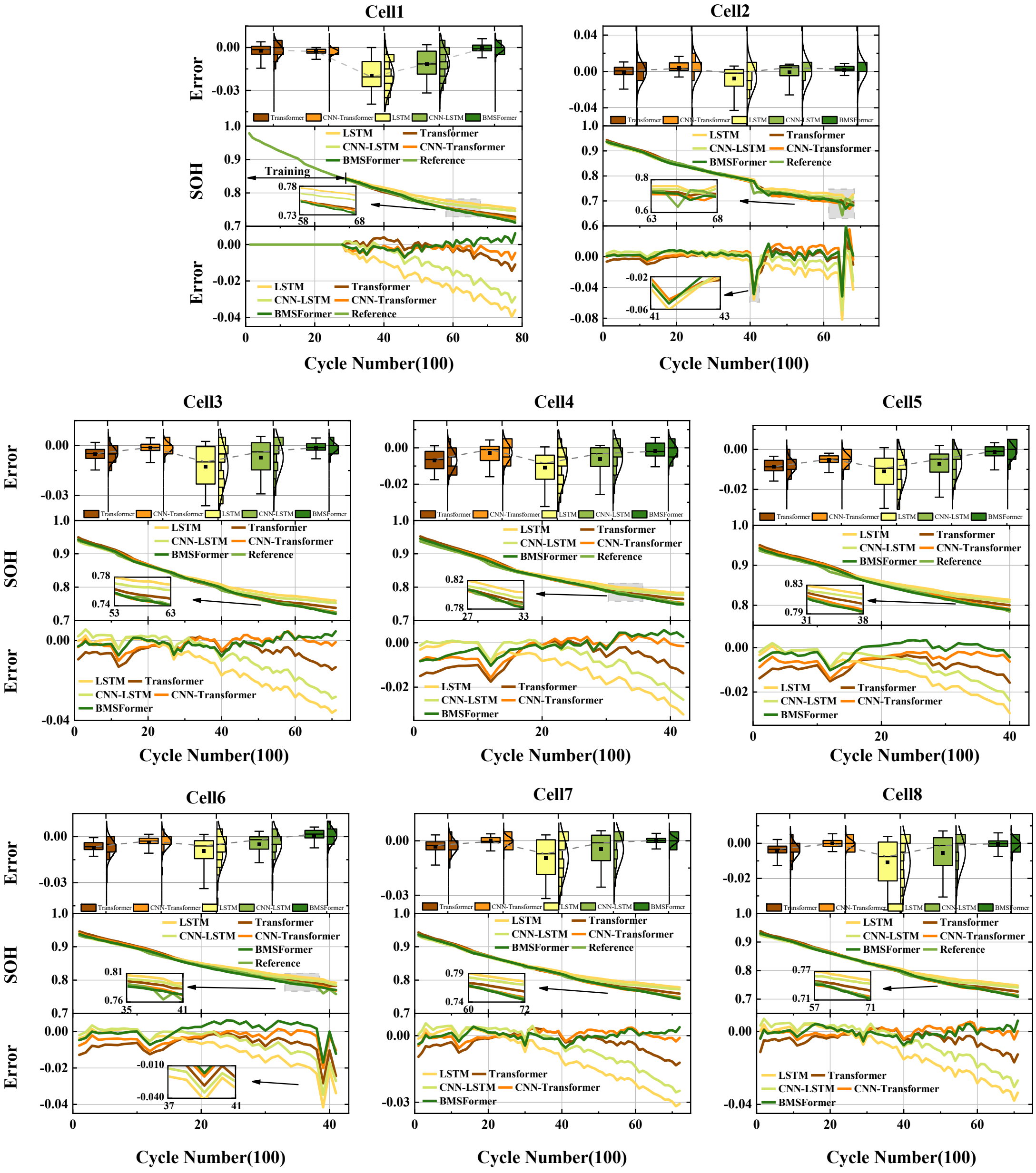


Fig. 7. The experimental results of different models on the Oxford dataset

Notably, in Cells 2 and 6, where there are several sudden drops, CNN-Transformer and BMSFormer were able to track the true values most accurately, with BMSFormer showing the smallest prediction errors in response to sudden changes in Cell6. For the other six batteries with relatively stable SOH changes (excluding Cell3 and Cell8), BMSFormer performed the best and ranked first in the mean prediction results across all eight batteries, showing improvements in the average MAE of 14.81%, 59.65%, 46.51%, 73.56%, the average RMSE of 16.21%, 62.19%, 44.64%, and 73.27%, and the average MAPE of 20.00%, 63.63%, 50.00%, 76.32% compared to CNN-Transformer, Transformer, CNN-LSTM, and LSTM, respectively.

Table 5. SOH estimation errors of different models on the Oxford dataset

Cells	Methods	MAE	MAPE	RMSE	R^2
Cell1*	BMSFormer	0.0032	**0.0022**	**0.0018**	**0.9937**
	CNN-Transformer	**0.0022**	0.0038	0.0025	0.9913
	CNN-LSTM	0.0085	0.0154	0.0109	0.8641
	Transformer	0.0027	0.0049	0.0037	0.9814
	LSTM	0.0140	0.0252	0.0160	0.7044
Cell2	BMSFormer	**0.0040**	**0.0074**	**0.0083**	**0.9802**
	CNN-Transformer	0.0058	0.0107	0.0091	0.9752
	CNN-LSTM	0.0059	0.0105	0.0095	0.9755
	Transformer	0.0047	0.0084	0.0082	0.9798
	LSTM	0.0075	0.0139	0.0124	0.9765
Cell3	BMSFormer	0.0019	0.0031	0.0023	0.9977
	CNN-Transformer	**0.0017**	**0.0028**	**0.0022**	**0.9978**
	CNN-LSTM	0.0062	0.0110	0.0087	0.9710
	Transformer	0.0040	0.0068	0.0049	0.9898
	LSTM	0.0093	0.0165	0.0126	0.9395
Cell4	BMSFormer	**0.0027**	**0.0044**	**0.0032**	**0.9936**
	CNN-Transformer	0.0029	0.0046	0.0041	0.9901
	CNN-LSTM	0.0045	0.0077	0.0069	0.9765
	Transformer	0.0052	0.0083	0.0064	0.9753
	LSTM	0.0077	0.0132	0.0102	0.9487
Cell5	BMSFormer	**0.0020**	**0.0031**	**0.0026**	**0.9940**
	CNN-Transformer	0.0039	0.0061	0.0043	0.9829
	CNN-LSTM	0.0051	0.0085	0.0069	0.9649
	Transformer	0.0064	0.0101	0.0068	0.9591
	LSTM	0.0078	0.0128	0.0098	0.9286
Cell6	BMSFormer	**0.0026**	**0.0043**	**0.0035**	**0.9905**
	CNN-Transformer	0.0028	0.0045	0.0039	0.9884
	CNN-LSTM	0.0041	0.0069	0.0067	0.9720
	Transformer	0.0054	0.0087	0.0067	0.9673
	LSTM	0.0068	0.0115	0.0095	0.9429
Cell7	BMSFormer	**0.0011**	0.0018	**0.0014**	**0.9987**
	CNN-Transformer	0.0012	**0.0015**	0.0019	**0.9987**
	CNN-LSTM	0.0053	0.0091	0.0072	0.9719
	Transformer	0.0029	0.0047	0.0037	0.9917

	LSTM	0.0076	0.0132	0.0104	0.9422
Cell8	BMSFormer	0.0015	0.0026	0.0020	0.9983
	CNN-Transformer	**0.0012**	**0.0021**	**0.0016**	**0.9989**
	CNN-LSTM	0.0060	0.0106	0.0082	0.9739
	Transformer	0.0033	0.0057	0.0043	0.9924
	LSTM	0.0087	0.0156	0.0119	0.9455
Average	BMSFormer	**0.0023**	**0.0036**	**0.0031**	**0.9934**
	CNN-Transformer	0.0027	0.0045	0.0037	0.9904
	CNN-LSTM	0.0057	0.0099	0.0082	0.9587
	Transformer	0.0043	0.0072	0.0056	0.9796
	LSTM	0.0087	0.0152	0.0116	0.9160

Cell1* denotes a training set of BMSFormer with 80% data, and its values are averaged among ten experiments. The best results are marked in bold and underlined.

4.2.2. Efficiency comparison

In practical application, it is important to consider training cost (floating-point operations, training time, parameters) and hardware cost (storage size) besides the accuracy [72]. To ensure a fair comparison, the same training and model hyperparameters are set to evaluate the comprehensive performance of different models. The definitions of these evaluation metrics and the settings of hyperparameters are described as follows:

1. Evaluation Metrics: (1) floating-point operations (FLOPs), which are measured using the profile function from the THOP library, quantify the numbers of FLOPs required for a single forward pass. (2) Training time, measured in seconds using Python's time module, represents the duration required to train a model on the training dataset. (3) Parameters, which are counted using the summary function from the torchsummary library, typically measured in millions (Million), indicate the total number of trainable parameters in the model. (4) Storage size, which is counted using the os.path.getsize function of Python, measured in kilobytes (KB), indicates the storage requirements of the model parameters on a device.

2. Training Hyperparameters: For the experiments, the same training hperparameters are used across all models. Each model is trained with a batch size of 128 for 1000 epochs. The learning rate is uniformly set to 0.01, and a dropout rate of 0.1 is applied.

3. Model Hyperparameters: To compare the computational complexity of different model structures under the same input data and training hyperparameters, the model hyperparameters are set as follows: For models based on the Transformer architecture, the parameters are set uniformly: an embed dimension of 16, a dense dimension (MLP) of 16, four attention head, and both the encoder and decoder comprise one layer each. Four LSTM layers and hidden dimension of 32 are also employed for models based on the LSTM architecture. Additionally, for the CNN-Transformer models, the CNN module corresponds to our depthwise separable convolutional module and is also placed after the multi-head self-attention module. The basic hyperparameters for the two convolutional layers are set as follows: the first convolutional layer has a kernel size of 3 and a stride of 1, the second pooling layer has a kernel size of 2 and a stride of 2, and the third convolutional layer has a kernel size of 3 and a stride of 1. The activation function used is ReLU.

Under the almost same configuration and Oxford dataset, as illustrated in Table 6, the LSTM-based networks result in lower training time than other three networks due to the inherently sequential structure; however, this architecture limits their ability to capture complex degradation patterns, leading to lower estimation accuracy. In contrast, compared to CNN-Transformer and Transformer, BMSFormer achieves the shortest training time. Specifically, despite having 63.04% more FLOPs and 12.03% more parameters than Transformer, BMSFormer still reduces training time and storage capacity by 21.37% and 7.88%, respectively.

Table 6. The comprehensive performance of models under the same and optimal configurations

Models	Training hyperparameters			Model hyperparameters			Performance indicator			
	e	lr	b	d_e	d_d	n	FLOPs (Million)	Training time (s)	Parameters	Storage size(KB)
BMSFormer	1000	0.01	128	16	16	4	0.46	19.83	5330	**36.37**
CNN-Transformer	1000	0.01	128	16	16	4	0.27	26.57	6257	47.10
Transformer	1000	0.01	128	16	16	4	**0.17**	25.22	**4689**	39.48
CNN-LSTM	1000	0.01	128	-	16	4	0.48	11.84	12719	53.72
LSTM	1000	0.01	128	-	16	4	0.83	**8.07**	8753	37.85

Here, the e, lr, b, d_e, d_d and n represent the epochs, learning rate, batch size, embedding dimension, dense dimension of MLP or hidden dimension of LSTM, and layer number, respectively. The best results are marked in bold and underlined.

When adapting to different battery types or tasks, models usually require adjustments in hyperparameter combinations to match data patterns efficiently. Among the five adjustable hyperparameters (including e, lr, d_e, d_d and n), the LSTM network lacks embed dimension (d_e), and storage size depends only on model hyperparameters (including d_d and n). Therefore, to comprehensively evaluate storage memory consumption across different hyperparameter combinations, three hidden layer dimensions (16, 32, 64) and four model layers (1, 2, 3, 4) are configured, resulting in a total of 12 combinations applied to five models, as shown in Table 7.

Table 7. Grouping of models hyperparameter

Group number	Dense dimension	Layer number
G1	16	1
G2	32	1
G3	64	1
G4	16	2
G5	32	2
G6	64	2
G7	16	3
G8	32	3
G9	64	3
G10	16	4
G11	32	4
G12	64	4

As shown in Fig. 8(a), the bar chart for the 12 combinations indicates that BMSFormer consistently exhibits a lower storage size compared to CNN-Transformer and Transformer across the G1-G12 configurations, making it more suitable for deployment on resource-constrained devices. Moreover, LSTM and CNN-LSTM models experience a significant increase in memory usage when dimensions are raised to 64 (as shown in G3, G6, G9, G12), far exceeding the memory usage of the other three models. Although lower dimensions and fewer layers reduce the storage size of LSTM and CNN-LSTM, these models may suffer from a decline in estimation accuracy, failing to achieve an optimal balance between precision and efficiency.

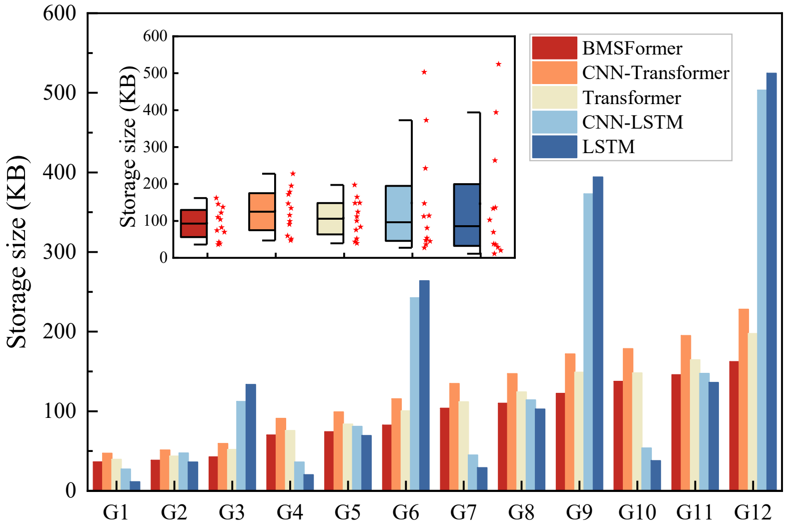


Fig. 8. Storage size of five models across G1-G12 hyperparameter combinations: (a) Bar chart comparing storage sizes for each group; (b) Box plot showing overall distribution

As illustrated in Fig. 8(b), the box plot shows that the proposed model has the shortest box and whiskers, indicating the lowest average memory consumption with minimal fluctuation across the 12 hyperparameter configurations. This characteristic is helpful when handling more complex data patterns, as the model maintains stable storage size while increasing dimensions or depth to enhance estimation accuracy.

In summary, the experimental results indicate that the proposed model exhibits low and stable memory usage across 384 hyperparameter combinations, making it well-suited for usage on resource-limited devices.

4.3. Comparison on NASA and CALCE datasets

This section discusses the performance of BMSFormer and four mainstream deep learning models on the NASA and CALCE datasets, which are different from the Oxford dataset in battery materials and charge/discharge protocols. For both datasets, 30% of the data from the first cell (B0005 for NASA and CS2-35 for CALCE) is used as the training set, while the remaining 70% is reserved for validation. The full datasets from the remaining six cells (B0006, B0007, B0018 for NASA and CS2-36, CS2-37, CS2-38 for CALCE) are used as the test set.

The five models' evaluation process is described as follows. First, the five models are respectively trained on their training sets using 384 distinct hyperparameter combinations and validated. All the validation results are then used to evaluate the stability of each model and the best-performance models, with highest validation accuracy, are selected as the final trained models. Second, the selected five final models are then directly tested on the unseen test sets to further evaluate the model generalization.

4.3.1. Stability evaluation across 384 hyperparameter combinations

Table 8 presents the comprehensive set of hyperparameter combinations. Each column represents an adjustable hyperparameter category, including the train epochs, the learning rate, the number of attention block, the number of LSTM layer (or attention head), the dense layer dimension of MLP (or the hidden dimension of LSTM), and the dimension of embedding layer. This can help identify the sets of hyperparameter combinations that yield the high validation accuracies and reflects the model's stability across different combinations.

To ensure a comprehensive exploration of training hyperparameters, "epochs" are chosen as 1000 and 200, and "learning rates" are set to 0.001 and 0.01, resulting in a total of $2 \times 2 = 4$ basic combinations, which correspond to the four segments on the x-axis of Fig. 9. Meanwhile, various model hyperparameter settings are also considered: BMSFormer, CNN-Transformer, and Transformer are based on attention mechanism and each have $2 \times 3 \times 4 \times 4 = 96$ possible hyperparameter groups, in which 2, 3, 4 and 4 represent the "block number", "head number", "dense dimension" of MLP layer, and "embed dimension" of embedding layer, respectively. In contrast, LSTM and CNN-LSTM models just have two additional adjustable hyperparameters for the "layer number" of LSTM and the "dense dimension" of hidden layer, resulting in $3 \times 4 = 12$ combinations.

Table 8. The combination of hyperparameter

Hyperparameter	Epochs	Learning rate	Block number	Layer number	Dense dimension	Embedding dimension
Values	$[200,1000]$	$[0.001,0.01]$	$[1,2]$	$[1,2,4]$	$[16,32,64,128]$	$[16,32,64,128]$

Consequently, the total number of hyperparameter combinations of the models represented on the y-axis of the figure is $4 \times 12 = 48$ for the first two models and $4 \times 96 = 384$ for the latter three models. Each color bar corresponds to a fitting score, with redder colors indicating higher accuracy and bluer colors indicating worser. The red vertical line on the far right represents the number of combinations with a fitting score greater than 0.98, with a higher number of groups indicating better performance.

Overall, compared to recent mainstream models, BMSFormer shows robust performance and accuracy across different hyperparameter combinations. Specifically, BMSFormer has 41 combinations where the prediction model's R^2 value exceeds 0.98, representing 10.67% of the total combinations. Additionally, 43.23% of the combinations have an R^2 value greater than 0.9. BMSFormer also achieves the highest R^2 of 0.9892 than others.

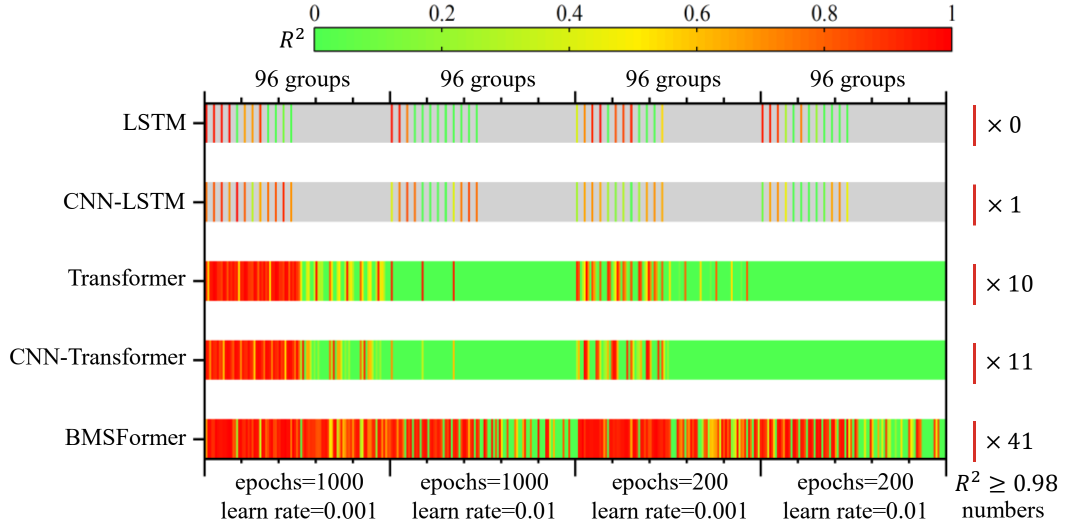


Fig. 9. The whole validating results when using different models and hyperparameter combinations.

4.3.2. Generalization evaluation

This section presents and compares the testing results obtained through the process of directly testing on the unseen test sets using the five optimal models with highest validation accuracies.

As illustrated in Fig. 10 (a)-(c), the estimated SOH of BMSFormer is generally closer to the true SOH compared to other models, and the estimation error is also closer to zero, both indicating higher accuracy. The localized magnifications in the ‘SOH’ plots further illustrate the better stability of BMSFormer when encountering sudden SOH changes in the degradation process of ten batteries.

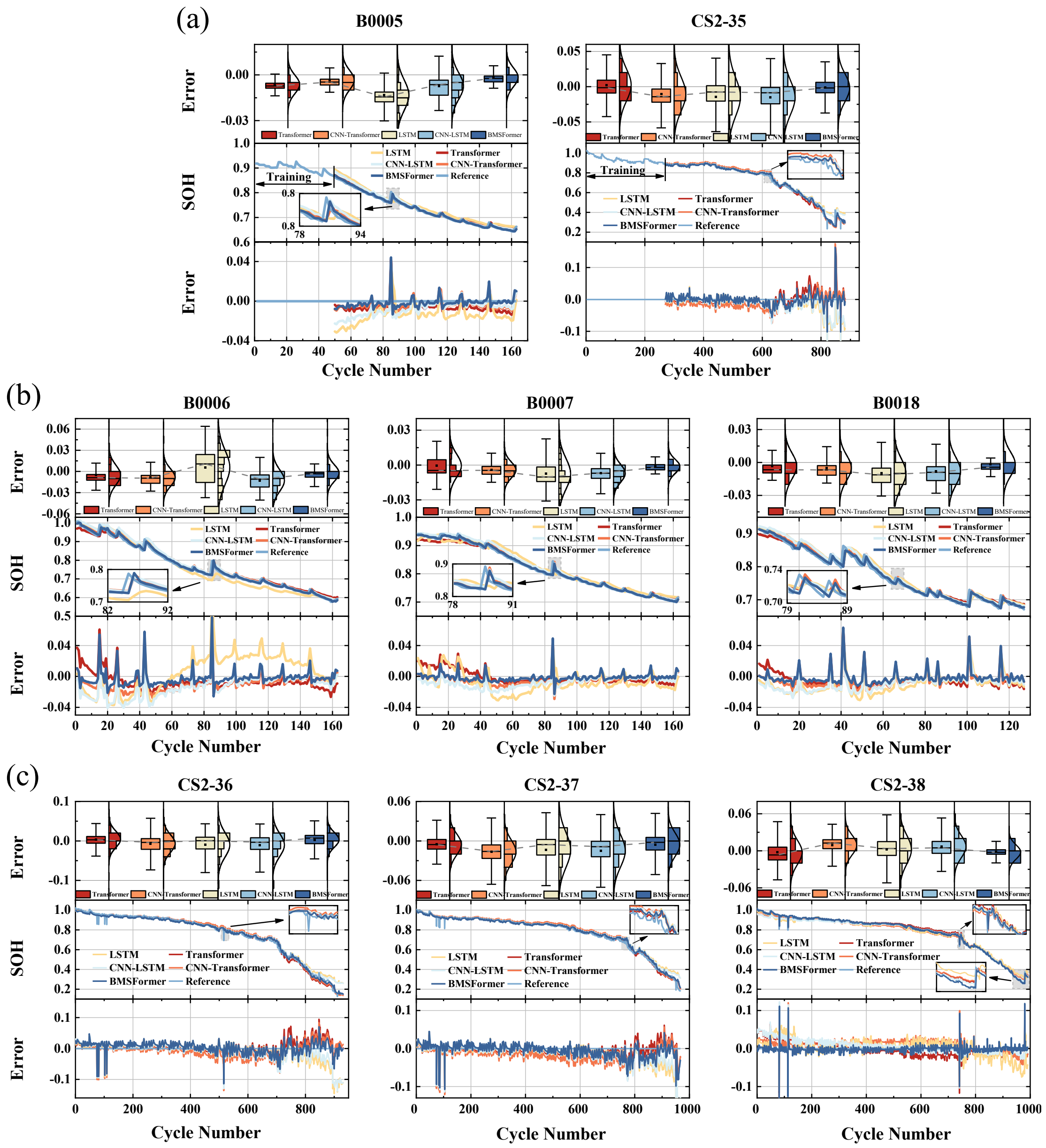


Fig. 10. (a) The validation results of five models on B0005 and CS2-35. (b) The test results using five weighted models trained on B0005. (c) The test results using five weighted models trained on CS2-35.

As detailed in Table 9**,** BMSFormer achieves improvements of 37.15%, 42.65%, 23.67%, and 47.90% in average MAE, 27.30%, 48.20%, 20.42%, and 53.71% in average MAPE, and 34.29%, 40.69%, 18.84%, and 46.29% in average RMSE, compared to CNN-Transformer, CNN-LSTM, Transformer, and LSTM, respectively, across eight battery cells from NASA and CALCE datasets. Notably, the zoomed-in prediction curves reveal that BMSFormer performs exceptionally well even in scenarios with the most severe SOH fluctuations, such as those observed in B0018 and CS2-38.

LSTM still exhibits the lowest prediction accuracy due to insufficient sensitivity to local features. Although CNN-LSTM partially addresses this issue, it still suffers from structural limitations, resulting in the loss of important information after convolution processing. Transformer also struggles with local feature recognition, leading to larger fluctuations in error curves and particularly noticeable deviations in B0005. CNN-Transformer and BMSFormer show similar performance, but CNN-Transformer's small convolution kernel size of 1×3 and the lack of feature diversity constrain its ability to identify complex features pattern, resulting in slightly lower prediction accuracy compared to BMSFormer. Increasing the convolution kernel size in CNN-Transformer might improve performance but would also increase the computational cost.

Table 9. Estimation results on the NASA and CALCE datasets

Cell	Methods	MAE	MAPE	RMSE	R^2
B0005*	BMSFormer	**0.0042**	**0.0053**	**0.0065**	**0.9950**
	CNN-Transformer	0.0071	0.0089	0.0089	0.9907
	CNN-LSTM	0.0104	0.0129	0.0121	0.9830
	Transformer	0.0091	0.0115	0.0109	0.9861
	LSTM	0.0140	0.0181	0.0157	0.9716
B0006	BMSFormer	**0.0073**	**0.0117**	**0.0092**	**0.9905**
	CNN-Transformer	0.0123	0.0151	0.0155	0.9843
	CNN-LSTM	0.0158	0.0194	0.0197	0.9730
	Transformer	0.0118	0.0156	0.0148	0.9848
	LSTM	0.0209	0.0274	0.0246	0.9581
B0007	BMSFormer	**0.0037**	**0.0045**	**0.0061**	**0.9939**
	CNN-Transformer	0.0062	0.0074	0.0081	0.9893
	CNN-LSTM	0.0086	0.0102	0.0105	0.9821
	Transformer	0.0079	0.0095	0.0098	0.9841
	LSTM	0.0131	0.0159	0.0152	0.9626
B0018	BMSFormer	**0.0068**	**0.0111**	**0.0087**	**0.9771**
	CNN-Transformer	0.0093	0.0119	0.0123	0.9713
	CNN-LSTM	0.0122	0.0153	0.0148	0.9591
	Transformer	0.0094	0.0122	0.0120	0.9729
	LSTM	0.0139	0.0178	0.0163	0.9503
CS2-35*	BMSFormer	**0.0116**	**0.0208**	**0.0197**	**0.9892**

	CNN-Transformer	0.0198	0.0262	0.0324	0.9808
	CNN-LSTM	0.0191	0.0416	0.0289	0.9767
	Transformer	0.0140	0.0259	0.0227	0.9856
	LSTM	0.0193	0.0439	0.0304	0.9742
CS2-36	BMSFormer	**0.0144**	**0.0242**	**0.0188**	**0.9936**
	CNN-Transformer	0.0169	0.0306	0.0233	0.9902
	CNN-LSTM	0.0197	0.0536	0.0306	0.9831
	Transformer	0.0138	0.0278	0.0209	0.9921
	LSTM	0.0204	0.0582	0.0320	0.9815
CS2-37	BMSFormer	0.0125	0.0221	0.0184	0.9904
	CNN-Transformer	0.0196	0.0245	0.0304	0.9831
	CNN-LSTM	0.0185	0.0386	0.0286	0.9770
	Transformer	**0.0117**	**0.0202**	**0.0168**	**0.9920**
	LSTM	0.0179	0.0402	0.0299	0.9749
CS2-38	BMSFormer	**0.0214**	**0.0356**	**0.0276**	**0.9778**
	CNN-Transformer	0.0391	0.0615	0.0441	0.9432
	CNN-LSTM	0.0385	0.0696	0.0487	0.9309
	Transformer	0.0296	0.0473	0.0338	0.9667
	LSTM	0.0377	0.0708	0.0500	0.9271
Average	BMSFormer	**0.0102**	**0.0169**	**0.0143**	**0.9884**
	CNN-Transformer	0.0163	0.0233	0.0219	0.9791
	CNN-LSTM	0.0179	0.0327	0.0242	0.9706
	Transformer	0.0134	0.0213	0.0177	0.9830
	LSTM	0.0197	0.0365	0.0268	0.9625

B0005* and CS2-35* denote that they are the training sets with 30% of the early data. The best results are marked in bold and underlined.

5. Conclusion

This paper presents BMSFormer, an innovative and efficient deep learning model for online state-of-health (SOH) estimation of lithium-ion batteries, alleviating the limitations of traditional approaches that cannot keep well balance between accuracy and efficiency. By incorporating the constructed Local-Global Fusion Attention module, BMSFormer can not only capture both short-term and long-term features, but also reduce computational complexity compared to traditional Softmax-based attention methods. Meanwhile, the integration of multi-scale Depthwise Separable Convolutions enhances feature extraction across multiple scales and channels, contributing to a more diverse representation of battery degradation characteristics. These advancements collectively provide technical advantages in tackling challenges related to battery behaviors, such as random capacity increments and gradual performance degradation.

The comprehensive experiments on the three different kinds of public battery datasets have shown the advantages of BMSFormer in estimation accuracy, efficiency, and generalization. The validation results also illustrate the stability of BMSFormer across 384 hyperparameter combinations. These advancements make BMSFormer a potentially robust solution for SOH estimation, particularly in some resource-constrained environments.

Finally, there is still room for further optimization of the model architecture, particularly in enhancing computational efficiency and reducing model complexity. Future research will focus on refining the structure of BMSFormer, expanding its applicability to a wider range of battery types and use cases, and integrating the model into Battery Management Systems (BMS) for real-world deployment and evaluation. These efforts aim to extend the practical impact of BMSFormer in advancing the field of battery health management.

Reference

[1] F.C. Krause, J.P. Ruiz, S.C. Jones, E.J. Brandon, E.C. Darcy, C.J. Iannello, R.V. Bugga, Performance of Commercial Li-Ion Cells for Future NASA Missions and Aerospace Applications, J. Electrochem. Soc. 168 (2021) 040504. https://doi.org/10.1149/1945-7111/abf05f.

[2] A. Barré, B. Deguilhem, S. Grolleau, M. Gérard, F. Suard, D. Riu, A review on lithium-ion battery ageing mechanisms and estimations for automotive applications, Journal of Power Sources 241 (2013) 680–689. https://doi.org/10.1016/j.jpowsour.2013.05.040.

[3] S. Ma, M. Jiang, P. Tao, C. Song, J. Wu, J. Wang, T. Deng, W. Shang, Temperature effect and thermal impact in lithium-ion batteries: A review, Progress in Natural Science: Materials International 28 (2018) 653–666. https://doi.org/10.1016/j.pnsc.2018.11.002.

[4] Y. Xia, G. Chen, Q. Zhou, X. Shi, F. Shi, Failure behaviours of 100% SOC lithium-ion battery modules under different impact loading conditions, Engineering Failure Analysis 82 (2017). https://doi.org/10.1016/j.engfailanal.2017.09.003.

[5] G.L. Plett, Sigma-point Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 2: Simultaneous state and parameter estimation, Journal of Power Sources 161 (2006) 1369–1384. https://doi.org/10.1016/j.jpowsour.2006.06.004.

[6] G.L. Plett, Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 1. Background, Journal of Power Sources 134 (2004) 252–261. https://doi.org/10.1016/j.jpowsour.2004.02.031.

[7] S. Peng, D. Zhang, G. Dai, L. Wang, Y. Jiang, F. Zhou, State of charge estimation for LiFePO4 batteries joint by PID observer and improved EKF in various OCV ranges, Applied Energy 377 (2025) 124435. https://doi.org/10.1016/j.apenergy.2024.124435.

[8] N.A. Chaturvedi, R. Klein, J. Christensen, J. Ahmed, A. Kojic, Algorithms for Advanced Battery-Management Systems, IEEE Control Systems Magazine 30 (2010) 49–68. https://doi.org/10.1109/MCS.2010.936293.

[9] C. Zhang, W. Allafi, Q. Dinh, P. Ascencio, J. Marco, Online estimation of battery equivalent circuit

model parameters and state of charge using decoupled least squares technique, Energy 142 (2018) 678–688. https://doi.org/10.1016/j.energy.2017.10.043.

[10] M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, B. Scholkopf, Support vector machines, IEEE Intelligent Systems and Their Applications 13 (1998) 18–28. https://doi.org/10.1109/5254.708428.

[11] Q.-J. Zhang, K.C. Gupta, V.K. Devabhaktuni, Artificial neural networks for RF and microwave design - from theory to practice, IEEE Transactions on Microwave Theory and Techniques 51 (2003) 1339–1350. https://doi.org/10.1109/TMTT.2003.809179.

[12] J. Kim, J. Oh, H. Lee, Review on battery thermal management system for electric vehicles, Applied Thermal Engineering 149 (2019) 192–212. https://doi.org/10.1016/j.applthermaleng.2018.12.020.

[13] S. Sista, A. Sista, Intelligent BMS Solution Using AI and Prognostic SPA, in: Proceedings of the FISITA 2012 World Automotive Congress, Springer, Berlin, Heidelberg, 2013: pp. 755–764. https://doi.org/10.1007/978-3-642-33741-3_4.

[14] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Commun. ACM 60 (2017) 84–90. https://doi.org/10.1145/3065386.

[15] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, (2017). http://arxiv.org/abs/1704.04861 (accessed August 22, 2024).

[16] W. Duan, S. Song, F. Xiao, Y. Chen, S. Peng, C. Song, Battery SOH estimation and RUL prediction framework based on variable forgetting factor online sequential extreme learning machine and particle filter, Journal of Energy Storage 65 (2023) 107322. https://doi.org/10.1016/j.est.2023.107322.

[17] P. Yang, H.D. Yang, X.B. Meng, C.R. Song, T.L. He, J.Y. Cai, Y.Y. Xie, K.K. Xu, Joint evaluation and prediction of SOH and RUL for lithium batteries based on a GBLS booster multi-task model, Journal of Energy Storage 75 (2024) 109741. https://doi.org/10.1016/j.est.2023.109741.

[18] F. Guo, X. Wu, L. Liu, J. Ye, T. Wang, L. Fu, Y. Wu, Prediction of remaining useful life and state of health of lithium batteries based on time series feature and Savitzky-Golay filter combined with gated recurrent unit neural network, Energy 270 (2023) 126880. https://doi.org/10.1016/j.energy.2023.126880.

[19] M.-F. Ge, Y. Liu, X. Jiang, J. Liu, A review on state of health estimations and remaining useful life prognostics of lithium-ion batteries, Measurement 174 (2021) 109057. https://doi.org/10.1016/j.measurement.2021.109057.

[20] Degradation diagnostics for lithium ion cells, Journal of Power Sources 341 (2017) 373–386. https://doi.org/10.1016/j.jpowsour.2016.12.011.

[21] L. Zhang, C. Sun, T. He, Y. Jiang, J. Wei, Y. Huang, M. Yuan, High-performance quasi-2D perovskite light-emitting diodes: from materials to devices, Light Sci Appl 10 (2021) 61. https://doi.org/10.1038/s41377-021-00501-0.

[22] S.K. Rechkemmer, X. Zang, W. Zhang, O. Sawodny, Empirical Li-ion aging model derived from single particle model, Journal of Energy Storage 21 (2019) 773–786. https://doi.org/10.1016/j.est.2019.01.005.

[23] K. Luo, X. Chen, H. Zheng, Z. Shi, A review of deep learning approach to predicting the state of health and state of charge of lithium-ion batteries, Journal of Energy Chemistry 74 (2022) 159–173. https://doi.org/10.1016/j.jechem.2022.06.049.

[24] Z. Fei, F. Yang, K.-L. Tsui, L. Li, Z. Zhang, Early prediction of battery lifetime via a machine learning based framework, Energy 225 (2021) 120205. https://doi.org/10.1016/j.energy.2021.120205.

[25] R.R. Richardson, M.A. Osborne, D.A. Howey, Gaussian process regression for forecasting battery state of health, Journal of Power Sources 357 (2017) 209–219. https://doi.org/10.1016/j.jpowsour.2017.05.004.

[26] Y. Li, C. Zou, M. Berecibar, E. Nanini-Maury, J.C.-W. Chan, P. van den Bossche, J. Van Mierlo, N. Omar, Random forest regression for online capacity estimation of lithium-ion batteries, Applied Energy 232 (2018) 197–210. https://doi.org/10.1016/j.apenergy.2018.09.182.

[27] Y. Shi, J. Li, Z. Li, Gradient Boosting With Piece-Wise Linear Regression Trees, (2019). https://doi.org/10.48550/arXiv.1802.05640.

[28] Z. Ye, M.K. Kim, Predicting electricity consumption in a building using an optimized back-propagation and Levenberg–Marquardt back-propagation neural network: Case study of a shopping mall in China, Sustainable Cities and Society 42 (2018) 176–183. https://doi.org/10.1016/j.scs.2018.05.050.

[29] N. Yang, Z. Song, H. Hofmann, J. Sun, Robust State of Health estimation of lithium-ion batteries using convolutional neural network and random forest, Journal of Energy Storage 48 (2022) 103857. https://doi.org/10.1016/j.est.2021.103857.

[30] G. Lee, D. Kwon, C. Lee, A convolutional neural network model for SOH estimation of Li-ion batteries with physical interpretability, Mechanical Systems and Signal Processing 188 (2023) 110004. https://doi.org/10.1016/j.ymssp.2022.110004.

[31] J. Lu, R. Xiong, J. Tian, C. Wang, F. Sun, Deep learning to estimate lithium-ion battery state of health without additional degradation experiments, Nat Commun 14 (2023) 2760. https://doi.org/10.1038/s41467-023-38458-w.

[32] T. Luo, M. Liu, P. Shi, G. Duan, X. Cao, A Hybrid Data Preprocessing-Based Hierarchical Attention BiLSTM Network for Remaining Useful Life Prediction of Spacecraft Lithium-Ion Batteries, IEEE Trans. Neural Netw. Learning Syst. (2024) 1–14. https://doi.org/10.1109/TNNLS.2023.3311443.

[33] S. Peng, J. Zhu, T. Wu, C. Yuan, J. Cang, K. Zhang, M. Pecht, Prediction of wind and PV power by fusing the multi-stage feature extraction and a PSO-BiLSTM model, Energy 298 (2024) 131345. https://doi.org/10.1016/j.energy.2024.131345.

[34] Y. Zhang, Y. Wang, C. Zhang, X. Qiao, Y. Ge, X. Li, T. Peng, M.S. Nazir, State-of-health estimation for lithium-ion battery via an evolutionary Stacking ensemble learning paradigm of random vector functional link and active-state-tracking long–short-term memory neural network, Applied Energy 356 (2024) 122417. https://doi.org/10.1016/j.apenergy.2023.122417.

[35] G. Ma, Y. Zhang, C. Cheng, B. Zhou, P. Hu, Y. Yuan, Remaining useful life prediction of lithium-ion batteries based on false nearest neighbors and a hybrid neural network, Applied Energy 253 (2019) 113626. https://doi.org/10.1016/j.apenergy.2019.113626.

[36] H. Xu, L. Wu, S. Xiong, W. Li, A. Garg, L. Gao, An improved CNN-LSTM model-based state-of-health estimation approach for lithium-ion batteries, Energy 276 (2023) 127585. https://doi.org/10.1016/j.energy.2023.127585.

[37] Y. Zheng, J. Hu, J. Chen, H. Deng, W. Hu, State of health estimation for lithium battery random charging process based on CNN-GRU method, Energy Reports 9 (2023) 1–10.

https://doi.org/10.1016/j.egyr.2022.12.093.

[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention Is All You Need, (2023). http://arxiv.org/abs/1706.03762 (accessed August 22, 2024).

[39] L. Dong, S. Xu, B. Xu, Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, Calgary, AB, 2018: pp. 5884–5888. https://doi.org/10.1109/ICASSP.2018.8462506.

[40] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, (2019). https://doi.org/10.48550/arXiv.1810.04805.

[41] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, (2021). http://arxiv.org/abs/2010.11929 (accessed June 16, 2024).

[42] M.A. Hannan, D.N.T. How, M.S.H. Lipu, M. Mansor, P.J. Ker, Z.Y. Dong, K.S.M. Sahari, S.K. Tiong, K.M. Muttaqi, T.M.I. Mahlia, F. Blaabjerg, Deep learning approach towards accurate state of charge estimation for lithium-ion batteries using self-supervised transformer model, Sci Rep 11 (2021) 19541. https://doi.org/10.1038/s41598-021-98915-8.

[43] W. Gomez, F.-K. Wang, J.-H. Chou, Li-ion battery capacity prediction using improved temporal fusion transformer model, Energy 296 (2024) 131114. https://doi.org/10.1016/j.energy.2024.131114.

[44] C. Jia, Y. Tian, Y. Shi, J. Jia, J. Wen, J. Zeng, State of health prediction of lithium-ion batteries based on bidirectional gated recurrent unit and transformer, Energy 285 (2023) 129401. https://doi.org/10.1016/j.energy.2023.129401.

[45] X. Gu, K.W. See, P. Li, K. Shan, Y. Wang, L. Zhao, K.C. Lim, N. Zhang, A novel state-of-health estimation for the lithium-ion battery using a convolutional neural network and transformer model, Energy 262 (2023) 125501. https://doi.org/10.1016/j.energy.2022.125501.

[46] T. Bai, H. Wang, Convolutional Transformer-Based Multiview Information Perception Framework for Lithium-Ion Battery State-of-Health Estimation, IEEE Transactions on Instrumentation and

Measurement 72 (2023) 1–12. https://doi.org/10.1109/TIM.2023.3300451.

[47] L. Chen, S. Xie, A.M. Lopes, X. Bao, A vision transformer-based deep neural network for state of health estimation of lithium-ion batteries, International Journal of Electrical Power & Energy Systems 152 (2023) 109233. https://doi.org/10.1016/j.ijepes.2023.109233.

[48] C.R. Birkl, M.R. Roberts, E. McTurk, P.G. Bruce, D.A. Howey, Degradation diagnostics for lithium ion cells, Journal of Power Sources 341 (2017) 373–386. https://doi.org/10.1016/j.jpowsour.2016.12.011.

[49] I. Obisakin, C.V. Ekeanyanwu, State of Health Estimation of Lithium-Ion Batteries Using Support Vector Regression and Long Short-Term Memory, Open Journal of Applied Sciences 12 (2022) 1366–1382. https://doi.org/10.4236/ojapps.2022.128094.

[50] W. He, N. Williard, M. Osterman, M. Pecht, Prognostics of lithium-ion batteries based on Dempster–Shafer theory and the Bayesian Monte Carlo method, Journal of Power Sources 196 (2011) 10314–10321. https://doi.org/10.1016/j.jpowsour.2011.08.040.

[51] J. Chen, Y. Hu, Q. Zhu, H. Rashid, H. Li, A novel battery health indicator and PSO-LSSVR for LiFePO4 battery SOH estimation during constant current charging, Energy 282 (2023) 128782. https://doi.org/10.1016/j.energy.2023.128782.

[52] F. Wang, Z. Zhai, Z. Zhao, Y. Di, X. Chen, Physics-informed neural network for lithium-ion battery degradation stable modeling and prognosis, Nat Commun 15 (2024) 4332. https://doi.org/10.1038/s41467-024-48779-z.

[53] C. Lin, J. Xu, M. Shi, X. Mei, Constant current charging time based fast state-of-health estimation for lithium-ion batteries, Energy 247 (2022) 123556. https://doi.org/10.1016/j.energy.2022.123556.

[54] S. Peng, J. Zhu, T. Wu, A. Tang, J. Kan, M. Pecht, SOH early prediction of lithium-ion batteries based on voltage interval selection and features fusion, Energy 308 (2024) 132993. https://doi.org/10.1016/j.energy.2024.132993.

[55] J. Bai, J. Huang, K. Luo, F. Yang, Y. Xian, A feature reuse based multi-model fusion method for state of health estimation of lithium-ion batteries, Journal of Energy Storage 70 (2023) 107965. https://doi.org/10.1016/j.est.2023.107965.

[56] H. Dai, J. Wang, Y. Huang, Y. Lai, L. Zhu, Lightweight state-of-health estimation of lithium-ion batteries based on statistical feature optimization, Renewable Energy 222 (2024) 119907. https://doi.org/10.1016/j.renene.2023.119907.

[57] J. Wu, Z. Liu, Y. Zhang, D. Lei, B. Zhang, W. Cao, Data-driven state of health estimation for lithium-ion battery based on voltage variation curves, Journal of Energy Storage 73 (2023) 109191. https://doi.org/10.1016/j.est.2023.109191.

[58] H. Feng, L. Zhang, A heterogeneous learner fusion method with supplementary feature for lithium-ion batteries state of health estimation, Journal of Energy Storage 92 (2024) 111896. https://doi.org/10.1016/j.est.2024.111896.

[59] D. Roman, S. Saxena, V. Robu, M. Pecht, D. Flynn, Machine learning pipeline for battery state of health estimation, (2021). http://arxiv.org/abs/2102.00837 (accessed June 5, 2024).

[60] M. Lin, L. Ke, W. Wang, J. Meng, Y. Guan, J. Wu, Health prognosis via feature optimization and convolutional neural network for lithium-ion batteries, Engineering Applications of Artificial Intelligence 133 (2024) 108666. https://doi.org/10.1016/j.engappai.2024.108666.

[61] Y. Tan, G. Zhao, Transfer Learning With Long Short-Term Memory Network for State-of-Health Prediction of Lithium-Ion Batteries, IEEE Trans. Ind. Electron. 67 (2020) 8723–8731. https://doi.org/10.1109/TIE.2019.2946551.

[62] H. Jin, N. Cui, L. Cai, J. Meng, J. Li, J. Peng, X. Zhao, State-of-health estimation for lithium-ion batteries with hierarchical feature construction and auto-configurable Gaussian process regression, Energy 262 (2023) 125503. https://doi.org/10.1016/j.energy.2022.125503.

[63] K. Huang, K. Yao, Y. Guo, Z. Lv, State of health estimation of lithium-ion batteries based on fine-tuning or rebuilding transfer learning strategies combined with new features mining, Energy 282 (2023) 128739. https://doi.org/10.1016/j.energy.2023.128739.

[64] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, D.J. Inman, 1D convolutional neural networks and applications: A survey, Mechanical Systems and Signal Processing 151 (2021) 107398. https://doi.org/10.1016/j.ymssp.2020.107398.

[65] F. Chollet, Xception: Deep Learning with Depthwise Separable Convolutions, in: 2017 IEEE

Conference on Computer Vision and Pattern Recognition (CVPR), 2017: pp. 1800–1807. https://doi.org/10.1109/CVPR.2017.195.

[66] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, MobileNetV2: Inverted Residuals and Linear Bottlenecks, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018: pp. 4510–4520. https://doi.org/10.1109/CVPR.2018.00474.

[67] X. Zhang, X. Zhou, M. Lin, J. Sun, ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices, (2017). https://doi.org/10.48550/arXiv.1707.01083.

[68] N. Ma, X. Zhang, H.-T. Zheng, J. Sun, ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design, in: V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss (Eds.), Computer Vision – ECCV 2018, Springer International Publishing, Cham, 2018: pp. 122–138. https://doi.org/10.1007/978-3-030-01264-9_8.

[69] D. Han, X. Pan, Y. Han, S. Song, G. Huang, FLatten Transformer: Vision Transformer using Focused Linear Attention, in: 2023 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, Paris, France, 2023: pp. 5938–5948. https://doi.org/10.1109/ICCV51070.2023.00548.

[70] H. Cai, J. Li, M. Hu, C. Gan, S. Han, EfficientViT: Lightweight Multi-Scale Attention for High-Resolution Dense Prediction, in: 2023 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, Paris, France, 2023: pp. 17256–17267. https://doi.org/10.1109/ICCV51070.2023.01587.

[71] D. Chicco, M. Warrens, G. Jurman, The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation, PeerJ Computer Science 7 (2021) e623. https://doi.org/10.7717/peerj-cs.623.

[72] P. Li, Z. Zhang, R. Grosu, Z. Deng, J. Hou, Y. Rong, R. Wu, An end-to-end neural network framework for state-of-health estimation and remaining useful life prediction of electric vehicle lithium batteries, Renewable and Sustainable Energy Reviews 156 (2022) 111843. https://doi.org/10.1016/j.rser.2021.111843.