Applied Energy 336 (2023) 120808

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy

Capacity estimation of lithium-ion batteries based on data aggregation and feature fusion via graph neural network

Zhe Wang [a,b], Fangfang Yang [c], Qiang Xu [b,d,*], Yongjian Wang [e,f,*], Hong Yan [b,d], Min Xie [a,b]

[a] *Department of Advanced Design and Systems Engineering, City University of Hong Kong, Hong Kong Special Administrative Region*
[b] *Centre for Intelligent Multidimensional Data Analysis, Hong Kong Science Park, Shatin, Hong Kong Special Administrative Region*
[c] *School of Intelligent Systems Engineering, Sun Yat-sen University, Shenzhen, China*
[d] *Department of Electrical Engineering, City University of Hong Kong, Hong Kong Special Administrative Region*
[e] *Southeast University, School of Automation, China*
[f] *Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, China*

ARTICLE INFO

Keywords:
Capacity estimation
Deep learning
Feature fusion
Graph neural network
Lithium-ion battery

ABSTRACT

Lithium-ion batteries in electrical devices face inevitable degradation along with the long-term usage. The accompanying battery capacity estimation is crucial for battery health management. However, the hand-crafted feature engineering in traditional methods and complicated network design followed by the laborious trial in data-driven methods hinder efficient capacity estimation. In this work, the battery measurements from different sensors are organized as the graph structure and comprehensively utilized based on graph neural network. The feature fusion is further designed to enhance the network capacity. The specific data aggregation and feature fusion operations are selected by neural architecture search, which relieves the network design and increases the adaptability. Two public datasets are adopted to verify the effectiveness of the proposed scheme. Additional discussions are conducted to emphasize the capability of the graph neural network and the necessity of architecture searching. The comparison analysis and the performance under noisy environment further demonstrate the superiority of proposed scheme.

1. Introduction

The lithium-ion batteries, shared the advantages such as high energy density, have achieved extensive applications in diverse energy storage scenarios [1,2]. However, battery degradation is inevitable with the charging and discharging processes, which leads to decreased capacity and final end-of-life. The timely and accurate capacity estimation is a critical objective for the battery prognostics and management [3,4].

Traditionally, the capacity estimation is conducted by hand-crafted features and machine learning methods [5–7], such as support vectors regression [8], random forest regression [9], sparse Bayesian learning [10], particle filter [11]. Zhu et al. extracted the statistical features from voltage relaxation and then selected the feature combination according to cross validation [12]. Richardson et al. introduced Gaussian process regression to estimate battery capacity based on the voltage measurements from periods of galvanostatic operation [13]. She et al. combined the radial basis function neural networks and modified random forest regression to improve estimation accuracy [14]. Most of these methods attempted to extract the health indicators from the statistic perspective, the performance of which depends heavily on the expert experience. Some mathematical models based on battery dynamics also have been developed to facilitate the prognostics, such as electrochemical model [15], open circuit voltage model [16], and fused model [17]. However, these established models require extensive expert knowledge. Besides, some of the model parameters are difficult to be decided and the models usually lack the generalization ability.

With the accumulation of accessible public datasets and rapid growth of computational power, deep learning based schemes attract more and more attention [18–20]. Tan et al. extracted features based on the geometrical analysis of the voltage curve and selected the reasonable features according to gray relational analysis [21]. Li et al. used the voltage measurements as input to the long short-term memory (LSTM) networks and realized remaining capacity estimation [22]. However, these researches neglected the current and temperature information. Chen et al. applied the extreme learning machine to predict the complete temperature variation profile and then extracted health features from the temperature curve [23]. As few works incorporate all the voltage, current, and temperature information, Hong et al. adopted three charge–discharge cycle data and applied the dilated convolutional neural network (CNN) to swiftly predict the remaining useful life [24].

[*] Corresponding authors.
E-mail addresses: qiangxu027@gmail.com (Q. Xu), yjwan@seu.edu.cn (Y. Wang).

https://doi.org/10.1016/j.apenergy.2023.120808
Received 18 October 2022; Received in revised form 24 December 2022; Accepted 3 February 2023
Available online 13 February 2023

Li et al. organized the continuous data into a three dimensional image and used a pruning convolutional neural network to predict the capacity [25]. However, these networks usually need to be designed deliberately, which leads to high costs from laborious experiments. Besides, the measurements from different sensors are simply stacked and then input to the network. The research community still lacks the trial of comprehensive utilization of the battery measurements. The relations between the measurements are not explored and further utilized. The commonly used CNN can extract features from the Euclidean data and LSTM is suitable for sequence, but they face challenges when processing graph data. The graph neural network (GNN) enables the process of non-Euclidean data structure and attracts more and more attention [26,27]. The outputs from diverse sensors can be modeled by the GNN, which provides potential scheme to accomplish the data aggregation.

The recent deep learning techniques involve complicated architectural design [28–30]. Jiao et al. embedded the conditional variational autoencoder into particle filter framework and introduced a re-weighting strategy [31]. Tang et al. designed hybrid model to combine the CNN and LSTM, which could extract the spatial and temporal features, respectively [32]. He et al. incorporated the quantum genetic algorithm, attention mechanism, and mogrifier-LSTM to construct the entire model [33]. Yang fused three-dimensional CNN and two-dimensional CNN with an auxiliary feature attention algorithm to heighten the prediction performance [34]. Ma et al. proposed the deep belief network - LSTM hybrid network and multiple health indicators fusion strategy to mine the hidden information [35]. Although the deliberately designed networks improve the performance, they also lack the necessary adaptability when facing new scenarios. Therefore, a uniform estimation framework, which can simultaneously choose the optimized architecture and accomplish the prediction of capacity, is required for the prognostics of battery.

In view of the deficiency in measurements exploration and the complexity in network design, a data aggregation and feature fusion scheme is proposed to estimate the capacity of lithium-ion battery. The monitoring data of voltage, current and temperature is organized in a graph structure. In the charging process, partial measurements are adopted to construct the graph nodes. Then, the data aggregation is accomplished by GNN to comprehensively utilize the measurements. The GNN layers constitute the Supernet. Each GNN layer is learned flexibly and a feature fusion strategy is further proposed to fuse the output features of different GNN layers. These designs will improve the representation capacity and alleviate the over-smoothing problem existed in GNN. The neural architecture search (NAS) is introduced to obtain the network from Supernet, which will get rid of the complicated operation caused by feature engineering or network design. To the best of our knowledge, this is the first attempt to adopt graph neural network for the health status estimation of battery. The main contributions of this work are briefly presented as follows:

(1) The data aggregation scheme is designed to comprehensively utilize the measurements.

(2) The specific aggregation and fusion operations are selected automatically, which avoids the feature engineering.

(3) Only partial charging data is adopted, which can be applied under incomplete charging process.

The remainder of this article is organized as follows. Section 2 introduces the proposed capacity estimation scheme in detail. The experimental investigations on two cases, MIT-Stanford dataset and Oxford dataset, are performed in Section 3. Section 4 conducted more analysis and discussions to demonstrate the effectiveness and superiority of the proposed scheme. Conclusions are drawn in Section 5.

2. Capacity estimation scheme based on graph neural network

In the battery management system, the basic monitoring data mainly consist of the voltage V, current I, and temperature T. Further analysis can be performed using these measurements. Considering the graph structure provides an efficient way to organize the outputs from multiple sensors, this work proposes the data aggregation and feature fusion scheme based on graph neural network. The overall architecture is depicted in Fig. 1. The graph structure is established based on the relation modeling of the measurements. Then, the NAS strategy is conducted to select optimal data aggregation and feature fusion scheme. The searched scheme will be deployed for model training and online capacity estimation.

2.1. Graph construction

The graph structure is formulated as $G = (V, E)$, where V and E denote the nodes and edges, respectively. The voltage V, current I, and temperature T naturally form the nodes in the graph network. The initial node attributes can be derived from the measurements. To accomplish the graph construction, maximum information coefficient (MIC) is adopted here to measure the relationship between nodes and further form the edge attributes.

Originally, MIC is proposed to describe the dependence relationship of different variables [36]. In this case, MIC is used to calculate the correlation coefficient between the measurements. Taking the voltage and temperature as an example, the variable pair (V_i, T_i) from time i constitutes the points in the $X-Y$ coordinates. The mutual information is firstly obtained by:

$$I([V,T],x,y) = \int p(x,y)\log_2 \frac{p(x,y)}{p(x)p(y)} dxdy, \tag{1}$$

where $p(x,y)$ denotes the joint probability, $p(x)$ and $p(y)$ are the edge probabilities. Based on mutual information, the $MIC([V,T])$ is further developed:

$$MIC([V,T]) = \max_{ab<B} \frac{\max\ I\,\{[V,T],x,y\}}{\log_2 \min\{a,b\}}, \tag{2}$$

where a and b are the grid numbers in the x and y axis, respectively. The mutual information is calculated under different meshing grids. The maximum value will be found to determine the MIC. The precondition is also applied to the meshing by giving the upper limit of B, according to [36].

The calculated $MIC([V,T])$ value will be adopted as the edge attributes between voltage node and temperature node. Similarly, other edges can be obtained and the graph structure is established for the battery measurements. Then, the GNN will be introduced to aggregate the information in the whole graph.

2.2. Data aggregation and feature fusion

The GNN can conduct the computation on the graph structure, which will accomplish the measurements aggregation. Multiple layers of GNN are adopted and each layer will output corresponding extracted features. These features from different GNN layers will be further fused based on certain fusion operations.

2.2.1. Data aggregation by graph neural network

To utilize both the node attribute and edge attribute in the graph structure, four typical variants of GNN are considered in this work. These models are graph convolutional network (GCN) [37], Chebyshev network (ChebyNet) [38], graph attention network (GAT) [39] and GraphConv [40].

The GCN can be formulated as:

$$\boldsymbol{X}' = \hat{\boldsymbol{D}}^{-1/2}\hat{\boldsymbol{A}}\hat{\boldsymbol{D}}^{-1/2}\boldsymbol{X}\boldsymbol{\Theta}, \tag{3}$$

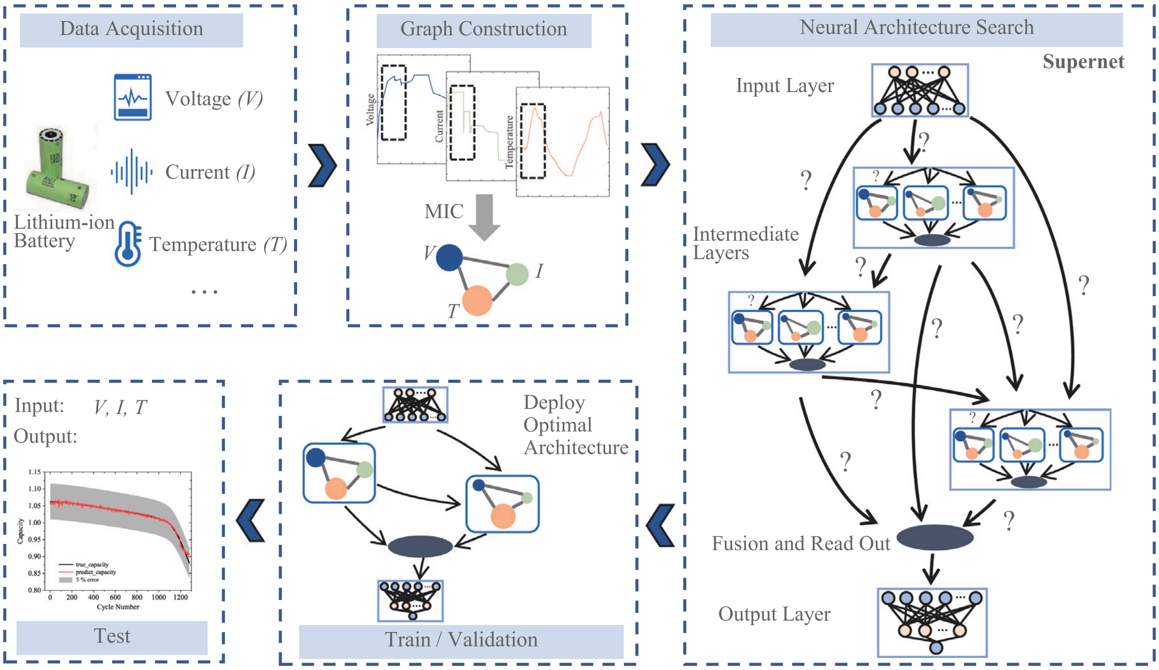


Fig. 1. The overall architecture of the proposed framework for battery capacity estimation.

where $\boldsymbol{X}$ is the node feature matrix from previous layer, $\hat{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$ is the adjacency matrix $\boldsymbol{A}$ with added self-connections, $\hat{D}_{ii} = \sum_{j=0} \hat{A}_{ij}$, $\boldsymbol{\Theta}$ is the weight matrix.

The ChebyNet is expressed as:

$$\boldsymbol{X}' = \sum_{k=1}^{K} \boldsymbol{Z}^{(k)} \boldsymbol{\Theta}^{(k)} \tag{4}$$

where $\boldsymbol{Z}^{(k)}$ is defined in the recursive formulation using the form of Chebyshev polynomial:

$$\begin{aligned} \boldsymbol{Z}^{(1)} &= \boldsymbol{X} \\ \boldsymbol{Z}^{(2)} &= \hat{\boldsymbol{L}} \cdot \boldsymbol{X} \\ \boldsymbol{Z}^{(k)} &= 2 \cdot \hat{\boldsymbol{L}} \cdot \boldsymbol{Z}^{(k-1)} - \boldsymbol{Z}^{(k-2)} \end{aligned} \tag{5}$$

where $\hat{\boldsymbol{L}}$ denotes the scaled Laplacian expressed as $\hat{\boldsymbol{L}} = 2\boldsymbol{L}/\lambda_{max} - \boldsymbol{I}$, and λ_{max} is from the diagonalized value after the matrix factorization.

The GAT adopts the idea of attention mechanism:

$$\boldsymbol{x}_i' = \alpha_{i,i} \boldsymbol{\Theta} \boldsymbol{x}_i + \sum_{j \in \mathcal{N}(i)} \alpha_{i,j} \boldsymbol{\Theta} \boldsymbol{x}_j \tag{6}$$

where $\mathcal{N}(i)$ denotes the neighbors of node i, $\alpha_{i,j}$ is the attention coefficients. The attention mechanism can be expressed as:

$$\alpha_{i,j} = \frac{\exp(LeakyReLU(\boldsymbol{a}^T [\boldsymbol{\Theta} \boldsymbol{x}_i \| \boldsymbol{\Theta} \boldsymbol{x}_j \| \boldsymbol{\Theta}_e \boldsymbol{e}_{i,j}]))}{\sum_{k \in \mathcal{N}(i) \bigcup \{i\}} \exp(LeakyReLU(\boldsymbol{a}^T [\boldsymbol{\Theta} \boldsymbol{x}_i \| \boldsymbol{\Theta} \boldsymbol{x}_k \| \boldsymbol{\Theta}_e \boldsymbol{e}_{i,k}]))} \tag{7}$$

where $\boldsymbol{a}$ represents the weight vector to represent the attention, $LeakyReLU$ is the commonly used nonlinearity function, $\|$ denotes the concatenation operation, $\boldsymbol{e}_{i,j}$ is the edge attribute.

The GraphConv can be formulated as:

$$\boldsymbol{x}_i' = \boldsymbol{W}_1 \boldsymbol{x}_i + \boldsymbol{W}_2 \sum_{j \in \mathcal{N}(i)} e_{j,i} \cdot \boldsymbol{x}_j \tag{8}$$

where $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are the learnable parameter matrices, and $e_{j,i}$ is the edge weight from source node j to target node i.

To illustrate the measurements aggregation, the process of ChebyNet is described in Fig. 2. The notation of n means the node number, F_{in} and F_{out} denote the input feature dimension and output feature dimension, respectively. Through the computation contained in GNN, the information in the nodes will interact and message passing will occur among the nodes.

Before the GNN layers, one fully connected layer will perform as the input layer. This layer will transform the input data dimension into a fixed feature dimension. After the final GNN layer, the global average pooling is added as the graph read out. Following that, the output layer of two fully connected layers will map the feature to the estimated capacity.

2.2.2. Feature fusion operation

The outputs of GNN layers will be further combined using the feature fusion operation. The fusion operation includes two steps: selection and fusion, as suggested by [41]. The selection step is used to determine whether to choose the feature or not, while the fusion process is to gather the features from different layers.

The detailed feature fusion operation is depicted in Fig. 3. A stack of three GNN layers is adopted. The output of one layer will connect to the following layers. The selection step of feature fusion is executed first to decide if the output will be incorporated or not.

In the selection step, "N" means to exclude the feature, while "Y" is the opposite. For layer v, this selection step can be expressed as:

$$\boldsymbol{X}^v = \alpha_1 N(\boldsymbol{X}^u) + \alpha_2 Y(\boldsymbol{X}^u) \tag{9}$$

where u is the previous layer, α_1 and α_2 are the weights. $N(\boldsymbol{X}^u) = 0$ means the feature is not chosen, and $Y(\boldsymbol{X}^u) = \boldsymbol{X}^u$ is to choose the feature from layer u.

The input of layer v is from all the previous layers. To fuse the multiple input features, the fusion operation is applied. The operation options include: sum, max, mean, and concatenation. They can be expressed as:

$$\text{SUM}: \quad \boldsymbol{x}_i^{(1)} + \cdots + \boldsymbol{x}_i^{(u)} \tag{10a}$$

$$\text{MAX}: \quad \max\left(\boldsymbol{x}_i^{(1)}, \ldots, \boldsymbol{x}_i^{(u)}\right) \tag{10b}$$

$$\text{MEAN}: \quad \frac{1}{u}\left(\boldsymbol{x}_i^{(1)} + \cdots + \boldsymbol{x}_i^{(u)}\right) \tag{10c}$$

$$\text{CONCAT}: \quad \boldsymbol{x}_i^{(1)} \| \cdots \| \boldsymbol{x}_i^{(u)} \tag{10d}$$

where $\boldsymbol{x}_i$ is the attributes of node i. The layers from 1 to $u(u < v)$ are fused in layer v. The fusion operation is conducted in a node-wise way, which means the same node from different layers will go through the fusion computation.

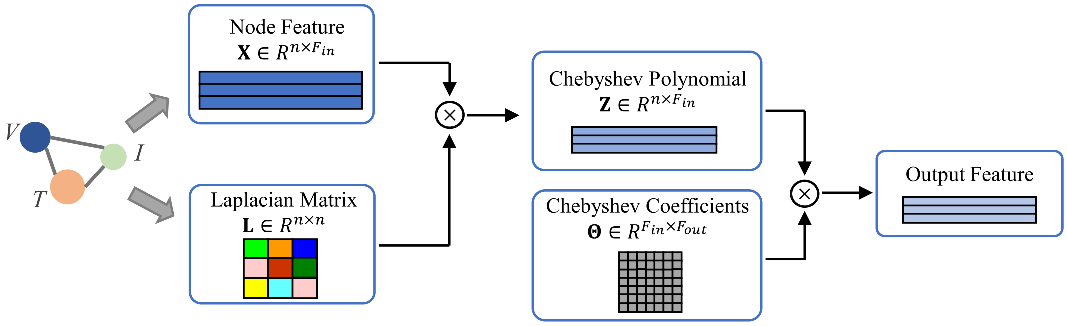

Node Feature
$\mathbf{X} \in R^{n\times F_{in}}$
Laplacian Matrix
$\mathbf{L} \in R^{n\times n}$
V
I
T
Chebyshev Polynomial
$\mathbf{Z} \in R^{n\times F_{in}}$
Chebyshev Coefficients
$\mathbf{\Theta} \in R^{F_{in}\times F_{out}}$
Output Feature

Fig. 2. The illustration of ChebyNet.

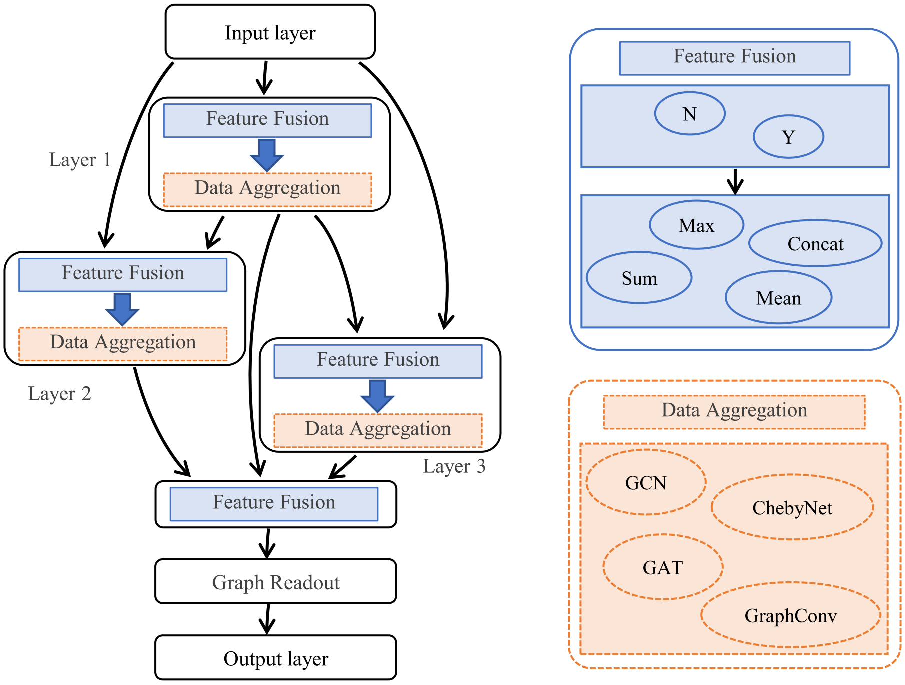


Fig. 3. The data aggregation and feature fusion operation.

The selection step and fusion step will realize the automatic feature filtering and optimization. Besides, the selection step will also perform the function of skipping connection. If the “N” is adopted for the final searched architecture, the corresponding GNN layers will be removed.

2.2.3. Architecture based on NAS

The different operations in the last section actually constitute the search space. To form the specific architecture, the NAS strategy is needed. The differentiable architecture search (DARTS) provides an executable method [42]. In detail, each layer will process the features introduced from predecessors and it can be expressed as:

$$\boldsymbol{X}^{v} = \sum_{u<v} Op^{u\to v}(\boldsymbol{X}^{u}), \tag{11}$$

where $Op^{u\to v}$ means the operation from layer u to layer v. The total combination of all the operations forms one Supernet. The relaxation technique is further applied to make the searching process continuous. The computations in each layer become a mixture of possible operations:

$$\overline{Op}(\boldsymbol{X}) = \sum_{k=1}^{\mathcal{O}} \alpha_k Op(\boldsymbol{X}), \tag{12}$$

where α_k is the weight, which is calculated by the relaxation function using the common softmax:

$$\alpha_k = \frac{\exp(c_k/\lambda)}{\sum_{i=1}^{\mathcal{O}} \exp(c_i/\lambda)}, \tag{13}$$

where λ is the temperature parameter, c_k is learnable Supernet parameter.

The above procedures create a continuous search space and it can also be conducted using the gradient descent method. Thus, the architecture and the weights can be jointly optimized. The overall objective is to find the parameter c which makes the network achieve the minimal validation loss. This bi-level optimization can be expressed as:

$$\begin{aligned} &\min_{c} \quad \mathcal{L}_{val}\left(\omega^*(c), c\right) \\ &\text{s.t.} \quad \omega^*(c) = \operatorname{argmin}_{\omega}\ \mathcal{L}_{train}(\omega, c) \end{aligned} \tag{14}$$

where $\mathcal{L}_{train}$ and $\mathcal{L}_{val}$ are the training loss and validation loss, respectively. To solve the above-mentioned optimization problem, the approximation scheme can be adopted according to DARTS [42].

After the NAS process, the operations with the largest weight α will be chosen to constitute the eventual architecture. This derived architecture will be deployed for further training and testing.

Algorithm 1: Data aggregation and feature fusion for capacity estimation

```
Data: Voltage $V$, Current $I$, Temperature $T$. Construct the graph
  $G = (V, E)$ by MIC and form the training and validation
  datasets.
Automatic architecture searching:
Create the Supernet with mixed operation
Initialize the weights $\omega$ and $c_k$ randomly
while not converged do
    (1) Conduct approximation scheme based on DARTS
    (2) Update architecture weights $c_k$
    (3) Update network weights $\omega$
Derive the optimal architecture according to the largest
  architecture weights $c_k$
Training and Validation:
Initialize the network parameters $\omega$ randomly
Input: Optimizer; Learning rate; Maximum epoch
for $i$ in training epochs do
    (1) Calculate the training loss $\mathcal{L}_{train}$
    (2) Backpropagation
    (3) Conduct the validation and preserve the better network
      parameters
Testing:
Input: Measurements from unknown test battery
Output: Capacity
```

2.3. Implementation procedure

The pseudo-code of the proposed method is shown in Algorithm 1. The main procedures are the architecture searching and followed by the architecture usage. The designed Supernet with mixed operations is established. The DARTS provides the method to update the architecture parameters and network parameters simultaneously. From the search space, the largest architecture weights c_k obtained in the iterations will be used to choose the specific operations. Thus, the final architecture can be determined. The architecture will be deployed as the optimal network. This network will be trained just like the standard process in deep learning. The measurements from test battery will be input to the network to obtain estimated capacity. Due to the diverse distributions of different datasets, the obtained architecture will also be different.

The experimental hyper-parameters are presented in Table 1. The overall framework contains three GNN layers. The trials demonstrate that three layers are efficient and the obtained results are satisfactory. The feature fusion also benefits to alleviate the over-smoothing problem in GNN. The learning rates are borrowed from the DARTS. The learning rate for architecture updating is smaller than that of network parameter updating. The architecture searching optimizer for parameter c is Adam optimizer, while the network optimizer for parameter ω is stochastic gradient descent (SGD). For the GNN model, $K = 2$ is adopted for ChebyNet. The head number of 4 is adopted for GAT model. The maximum epoch of 400 is adopted. The temperature used in Eq. (13) is 0.01, which will improve the output of softmax function. The gradient clipping value is set as 5 to prevent exploding gradients. The window length is set as 350 and the value will be further analyzed in the discussion section.

To quantitatively compare the results, the capacity prediction accuracy can be evaluated by the following metrics. In these equations, $\hat{y}_i$ denotes the predicted capacity, while y_i denotes the measured capacity or true value.

(1) Root mean square error (RMSE)

This index is usually applied to measure the difference between the predicted capacity and true value. The lower RMSE means the better prediction.

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}, \tag{15}$$

(2) Mean absolute error (MAE)

The MAE averages the absolute difference:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|, \tag{16}$$

(3) Maximum absolute error (ME)

This index emphasizes the largest difference:

$$\mathrm{ME} = \underset{1\le i\le n}{max}\left|y_i - \hat{y}_i\right|, \tag{17}$$

(4) Goodness-of-fit: R-Squared (R^2)

This index is to measure the match between the predicted and true values. The ideal value is 1.

$$\mathrm{R}^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}, \tag{18}$$

3. Experimental results

3.1. Battery dataset

3.1.1. Case I: MIT-Stanford dataset

This battery dataset is from the MIT-Stanford, which consists of 124 commercial lithium-ion batteries (type of APR18650M1A) [43]. The nominal capacity of these batteries is 1.1 Ah. Different charging protocols were applied and the policy conforms to “C1(Q1)-C2”, where “C1” and “C2” indicate the charging rate of two constant-current stages, “Q1” is the state-of-charge after first stage. The charging policy of “5C(67%)-4C”, which contains six batteries, is chosen in this study.

3.1.2. Case II: Oxford dataset

The Oxford battery degradation dataset is also used to verify the proposed framework [44]. This dataset contains eight commercial Kokam batteries with a capacity of 0.74 Ah. The measurements are different from those of MIT-Stanford dataset. In detail, after 100 drive cycles, the characterization measurements, which contain the 1C cycle and pseudo-OCV cycle were conducted.

3.2. Compared methods

The adopted comparison methods includes: LSTM, CNN-LSTM, AD-TCN, I-PCNN, GCN and GAT. LSTM is the model based on recurrent neural network. It is often deployed to model the time-series and achieves good predictive results. In this work, the one-layer LSTM with hidden dimension of 512, same with that of the proposed method, is adopted. CNN-LSTM is one hybrid network which combines the convolutional operation and timescale computation [45]. The LSTM layer is connected following the CNN layer. AD-TCN is the attention depthwise temporal convolutional network, which is an improved version of temporal convolutional network, which is suggested by [46]. The convolutional block attention module is incorporated to increase the weights of important features. Usually, these methods are for one sensor output modeling. To conduct an effective comparison, the methods are applied to the three measurements V, I, T independently. After that, the global averaging pooling is supplemented to obtain summarized features. Then the commonly used output layer maps the features to the predicted capacity value. I-PCNN firstly converts the three measurements into 3 dimensional image and then uses the pruned convolutional neural network, which is suggested by [25]. In the GNN-based methods, the typical graph aggregation methods of GCN, GAT are chosen as the comparison methods. Different from the proposed scheme, just one layer of GCN or GAT composes the model to realize the same function.

Table 1
Hyper-parameters in the proposed method.

Parameter	Value	Parameter	Value
Temperature λ	0.01	Architecture Optimizer	Adam
Maximum Epoch	400	Architecture Learning Rate	3e−4
Feature Dimension	512	Network Optimizer	SGD
Gradient Clipping	5	Network Learning Rate	0.01
Window Length	350	Network Momentum	0.9

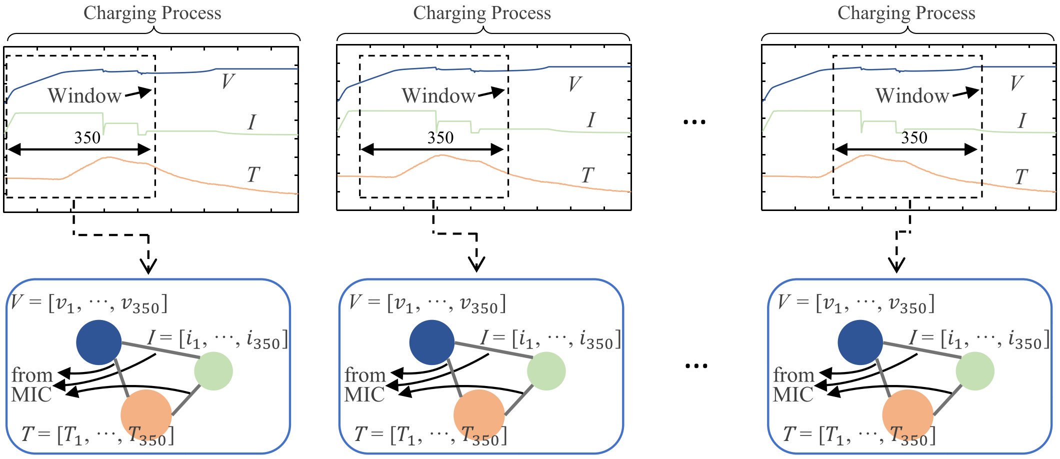

Charging Process
Window
350
V
I
T
···
$V = [v_1, \cdots, v_{350}]$
$I = [i_1, \cdots, i_{350}]$
$T = [T_1, \cdots, T_{350}]$
from MIC

Fig. 4. The sample generation process from the battery measurements.

3.3. Implementation details and results

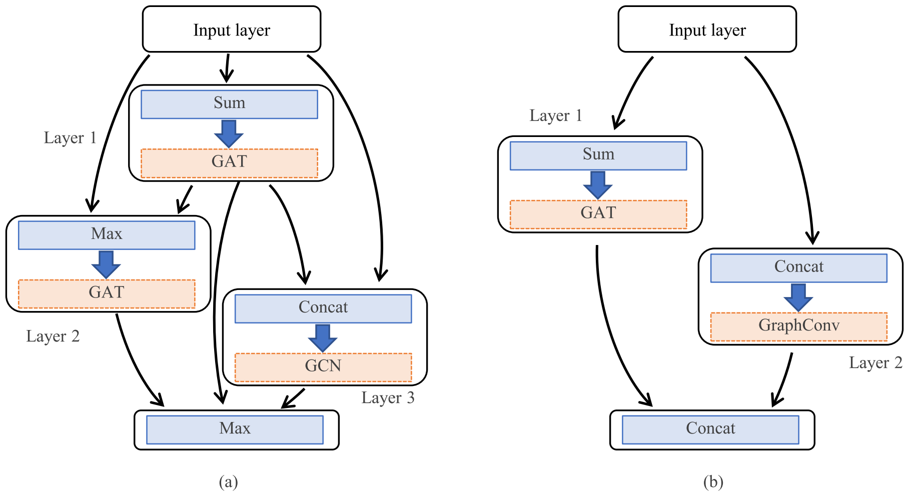
Before constructing the graph structure, the outliers in the capacity values are removed, based on the statistics of standard deviation around mean. The sample generation process from one battery cycle is illustrated in Fig. 4. The window size 350 is chosen to split the time series of measurements and form the nodes. The window moves along the time dimension to construct more graphs, which can be regarded as the data augmentation. This process is conducted on all the battery cycles along with the degradation. It needs to note that no specific principle is applied to choose the starting point of the window. For convenience, the first window can follow the time when the charging process begins, but no strict limitation is needed. Due to the different data amount in the two datasets, the sliding window numbers of 3 and 6 are adopted for MIT-Stanford dataset and Oxford dataset, respectively. In the test stage, the sample from the first window is used to estimate the capacity. Since the network is trained by multiple windowed data, the samples from other windows can also be applied for test. The differences of the estimated capacities from different samples will be slight. The following will just use the first window for network test.

In the proposed scheme, the architecture searching is conducted first. This process is repeated 5 times and the searching result with the lowest RMSE is chosen. The searched architecture for MIT-Stanford dataset and Oxford dataset are depicted in Figs. 5 (a) and (b), respectively. It can be seen that the two architectures own different layer numbers. The selection operation will automatically decide whether to choose the corresponding layer. The two datasets adopt different batteries and they present distinguished degradation characteristics. The NAS process will decide the specific architecture from the search space. Rather than using fixed network, this method involves the architecture searching and then architecture usage. It benefits to the flexible adaptation to different datasets. Besides, the automatic process will get rid of the manual design according to experience or trial experiments.

The leave-one-out validation is carried out to assess the performance of searched architecture. One battery is picked to test the performance in each dataset, while the rest batteries are used for training and validation with a proportion of 70% and 30%. All the measured capacity curves of the batteries in the two datasets are provided in Figs. 6(a) and (d), respectively. It can be found the two datasets present distinguish degradation processes. Following these figures, the capacity estimation results are given. For MIT-Stanford dataset, the best prediction is achieved in battery ID 5 with RMSE of 0.004. The worst is in ID 4 with RMSE of 0.009. The results are provided in Figs. 6(b) and (c), respectively. For Oxford dataset, Fig. 6(e) presents the best prediction from ID 5 with RMSE of 0.0064. Fig. 6(f) presents the worst prediction from ID 2 with RMSE of 0.0108.

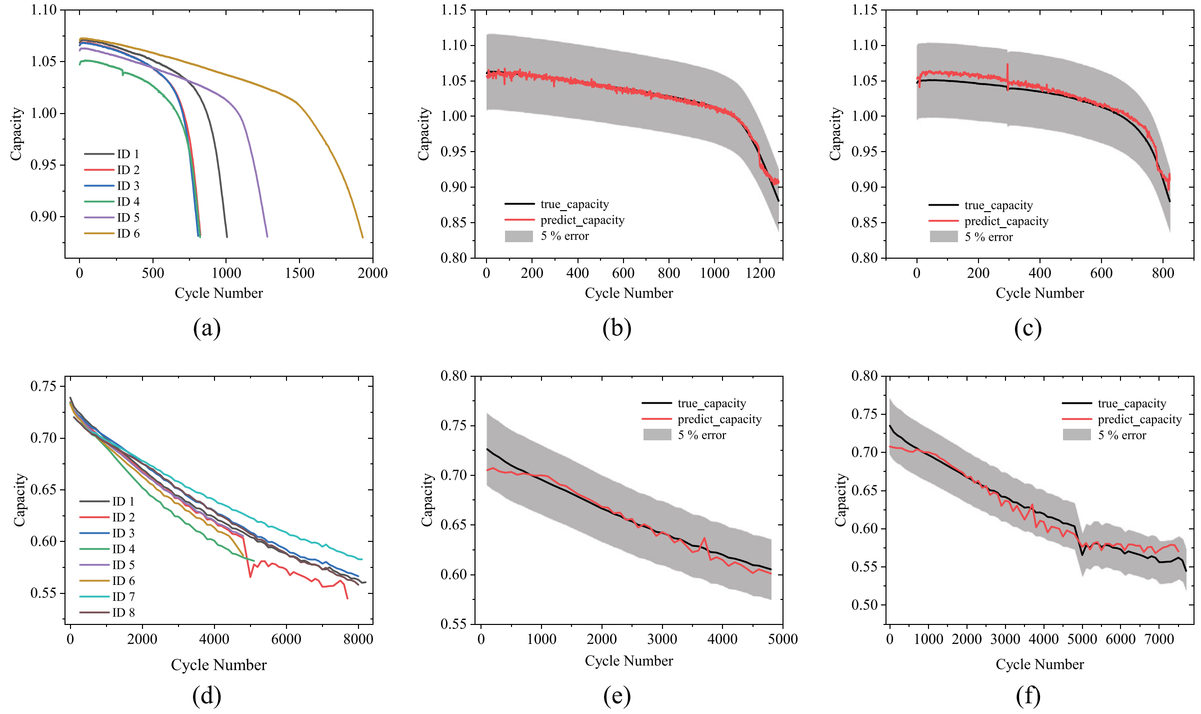
Four metrics are calculated to evaluate the performance of proposed scheme for capacity estimation. To quantitatively demonstrate the superiority, the compared methods are also conducted in the same dataset setting. The results of the above-mentioned batteries in the two datasets are summarized in Tables 2 and 3, respectively. For the best results, the fonts are marked as bold. It can be found that the searched architecture performs well except for the battery ID 4 in MIT-Stanford dataset. The GCN model achieves a slightly better result in the RMSE metric, however, the ME metric is worse than the other methods. It needs to note that ID 4 is the worst estimation in the proposed method. However, it still maintains enough competitiveness with other methods.

For clear comparison of these methods, the RMSE values of all the batteries are presented in Fig. 7. Compared with other methods, the proposed scheme obtains more stable and superior estimation performance in all cases. For LSTM and CNN-LSTM models, the errors are rather large in MIT-Stanford dataset, among the methods. In Oxford dataset, these two models present close results to the proposed method. AD-TCN and I-PCNN models present middle accuracy in both two datasets. The temporal convolutional module with attention module benefits to the relatively steady performance. While for I-PCNN, the results also show distinct fluctuations among different battery IDs in each dataset. The GCN model behaves in an opposite trend and the errors in the second dataset have difficulty meeting the capacity estimation need in the actual scenario. On the contrary, the proposed scheme will search suitable architecture for different datasets. The architecture after training also performs well in the target battery. Therefore, the proposed scheme presents competitive results for battery capacity estimation.



(a) (b)

Fig. 5. The searched architectures. (a) For MIT-Stanford dataset. (b) For Oxford dataset.



(a) (b) (c) (d) (e) (f)

Fig. 6. The capacity estimation results for MIT-Stanford dataset (a)–(c) and Oxford dataset (d)–(e). (a) The measured capacity in the MIT-Stanford dataset. (b) The best estimation. (b) The worst estimation. (d) The measured capacity in the Oxford dataset. (e) The best estimation. (f) The worst estimation.

Table 2
Comparison results of two test batteries in MIT-Stanford dataset.

Method	Test Battery ID 5				Test Battery ID 4			
	RMSE	MAE	ME	R^2	RMSE	MAE	ME	R^2
LSTM	0.0152	0.0128	0.0583	0.8439	0.0094	0.0067	0.0469	0.9348
CNN-LSTM	0.0152	0.0122	0.0570	0.8444	0.0091	0.0057	0.0884	0.9382
AD-TCN	0.0070	0.0048	0.0488	0.9668	0.0111	0.0073	0.0778	0.9088
I-PCNN	0.0125	0.0100	0.0454	0.8938	0.0094	0.0070	0.0666	0.9337
GCN	0.0069	0.0054	0.0401	0.9676	**0.0086**	**0.0039**	0.1160	**0.9452**
GAT	0.0101	0.0083	0.0403	0.9315	0.0097	0.0073	0.0830	0.9300
Proposed	**0.0040**	**0.0025**	**0.0278**	**0.9894**	0.0090	0.0078	**0.0377**	0.9399

4. Further analysis and discussions

To verify the effectiveness and superiority of proposed scheme, more analysis are conducted in this section.

4.1. Influence of window length

The window length is an important hyper-parameter for time-series modeling. In this work, the windowed data is used to construct the

Table 3
Comparison results of two test batteries in Oxford dataset.

Method	Test Battery ID 5				Test Battery ID 2			
	RMSE	MAE	ME	R^2	RMSE	MAE	ME	R^2
LSTM	0.0142	0.0072	0.0766	0.8405	0.0159	0.0105	0.0798	0.9145
CNN-LSTM	0.0083	0.0066	0.0225	0.9458	0.0129	0.0097	0.0350	0.9443
AD-TCN	0.0129	0.0084	0.0593	0.8691	0.0164	0.0111	0.0793	0.9099
I-PCNN	0.0093	0.0061	0.0330	0.9312	0.0141	0.0105	0.0327	0.9332
GCN	0.0214	0.0162	0.0679	0.6385	0.0410	0.0297	0.1257	0.4323
GAT	0.0155	0.0132	0.0401	0.8113	0.0128	0.0104	0.0279	0.9449
Proposed	**0.0064**	**0.0049**	**0.0212**	**0.9674**	**0.0108**	**0.0086**	**0.0272**	**0.9606**

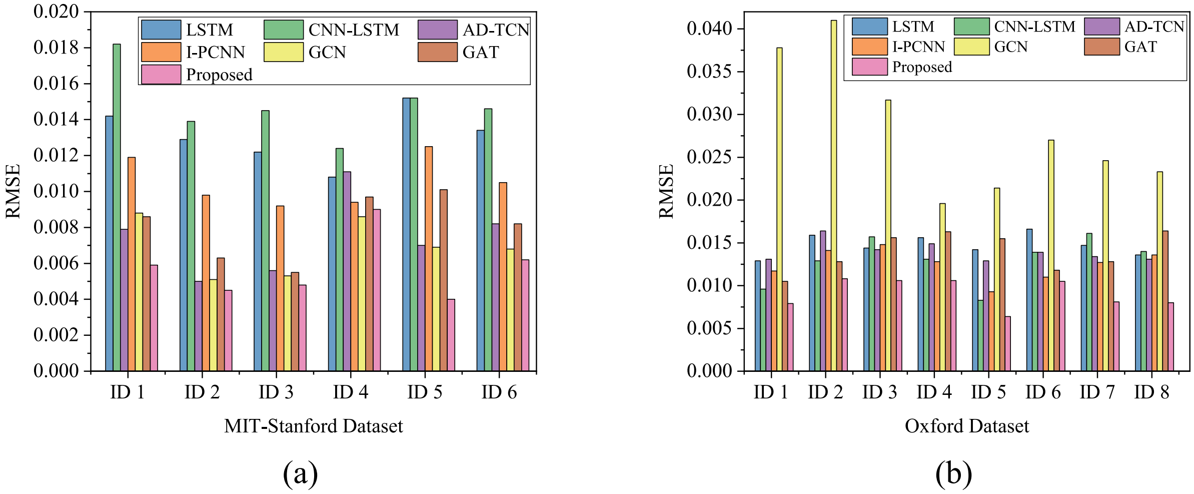


Fig. 7. The RMSE values for the two datasets. (a) The results of the MIT-Stanford dataset. (b) The results of the Oxford dataset.

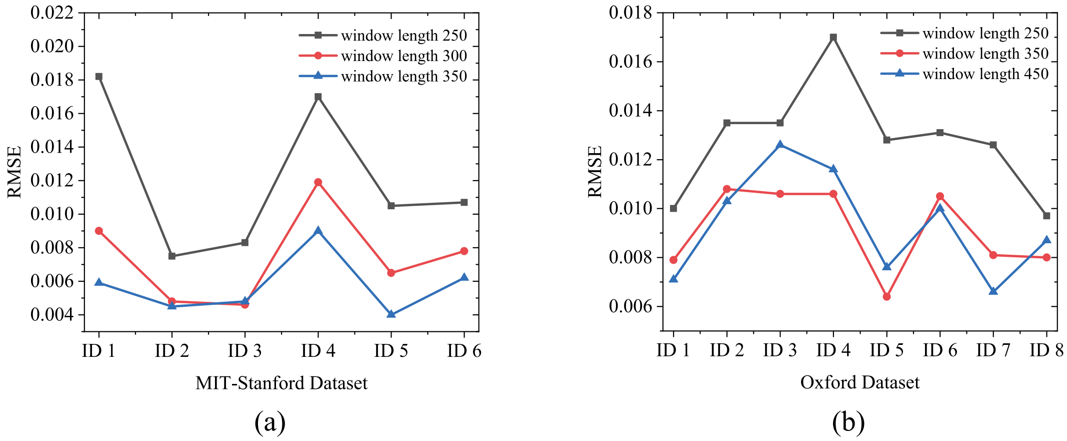


Fig. 8. The influence of different window lengths. (a) The RMSE value in the MIT-Stanford dataset. (b) The RMSE value in the Oxford dataset.

graph. To analyze its influence, different window lengths are compared in this work, and the results are depicted in Fig. 8. As the provided charging data in MIT-Stanford dataset is shorter than that of Oxford dataset, the window lengths adopted in the two datasets are different.

The results present the general trend that the RMSE drops with the increase of window length. In the Oxford dataset, from window length 350 to 450, the improvement of capacity estimation is not distinct. Overall, the chosen window length of 350 as the hyper-parameter seems to be a reasonable choice according to the quantified results.

In the literature, some works used the charging data in a fixed voltage range to predict the capacity. While in this work, the windowed data with fixed length is considered. Thus, the input length will also be fixed for the network, which is convenient for the network training and following application. The key point is that the network input should own enough information for further feature extraction.

4.2. Ablation study on GNN layer

The designed network incorporates both the GNN layers and final output layers. Since the output layers and initial embedding layer have the function of mapping the input to the capacity, the ablation study is carried out to verify the effectiveness of GNN layers. After removing the GNN layers, the architectures will be like the multi-layer perceptron. It still remains the modeling capability of neural network.

The results of models with and without GNN layers are shown in Tables 4 and 5. Without GNN layers, the apparent increase exists in the RMSE values for both two datasets. For some battery IDs, the increased error reaches more than 100%. These ablation results indicate the important efforts of GNN layers in the estimation.

In summary, the GNN layers aggregate the measurements effectively and accomplish the modeling from the measurements to the features.

Table 4
RMSE of MIT-Stanford dataset from the ablation study.

Ablation Study	ID 1	ID 2	ID 3	ID 4	ID 5	ID 6
Without GNN Layers	0.0148	0.0138	0.0127	0.0106	0.0162	0.0138
With GNN Layers	**0.0059**	**0.0045**	**0.0048**	**0.009**	**0.004**	**0.0062**

Table 5
RMSE of Oxford dataset from the ablation study.

Ablation Study	ID 1	ID 2	ID 3	ID 4	ID 5	ID 6	ID 7	ID 8
Without GNN Layers	0.0440	0.0502	0.0400	0.0334	0.0486	0.0451	0.0349	0.0382
With GNN Layers	**0.0079**	**0.0108**	**0.0106**	**0.0106**	**0.0064**	**0.0105**	**0.0081**	**0.0080**

Table 6
The alternative models for the two datasets.

Dataset	Model	Layer	Data Aggregation	Feature fusion
MIT-Stanford	Alternative Model 1	3	ChebyNet, GAT, GCN	Sum, Max, Concat, Max
	Alternative Model 2	2	GAT, ChebyNet	Concat, Concat, Concat
	Alternative Model 3	3	ChebyNet, ChebyNet, ChebyNet	Concat, Max, Concat, Max
Oxford	Alternative Model 1	3	ChebyNet, ChebyNet, GCN	Max, Max, Concat, Max
	Alternative Model 2	2	ChebyNet, GraphConv	Max, Concat, Concat
	Alternative Model 3	2	ChebyNet, GAT	Sum, Concat, Concat

The other layers mainly worked as auxiliary structures. Combined with the superior results shown in Fig. 7, it concludes that the GNN layers present the capability to learn discriminant features automatically.

It is worth mentioning that, the features from different GNN layers are also fused to improve the network learning ability. The proposed method is compared with the cases of one GNN layer, such as GCN, GAT. From the enhancement of performance, the feature fusion from multiple GNN layers benefits to the efficient mapping from measurements to capacity. Especially for Oxford dataset, the final deployed network contains two GNN layers. According to the comparisons with one layer of GCN and GAT in Table 3, the four metrics all achieve distinct improvements.

4.3. Ablation study on NAS

The NAS strategy is used in this work to automatically find the optimal network architecture, from all combinations of different operations in data aggregation and feature fusion. To verify the effectiveness of NAS, other architectures are adopted here for comparison. Three alternative models produced in the searching process of NAS are chosen. These models are the intermediate output before the ending of NAS. The alternative models are presented in Table 6. These models present different layer numbers and diverse module compositions.

These models are trained using the same setting to observe the results. The validation loss curves in the model validation process are extracted. Figs. 9 (a) and (b) present the loss curves for the MIT-Stanford dataset and Oxford dataset, respectively. In the MIT-Stanford dataset, the loss curve of finally searched model is nearly in a steady decreasing trend. The other three alternative models present apparent rise and fall at the beginning. At last, they converge to a larger loss compared with the searched model. In the Oxford dataset, the increasing loss occurs in the initial stage for these models. However, the alternative models decrease slower in the following epochs. Alternative models 1 and 2 also show severe fluctuations when reaching the ending of the training/validation process.

For the testing stage of these models, the four metrics are calculated for final capacity estimation. The results are shown in Figs. 9 (c) and (d) for the two datasets, respectively. In the metric of R-Squared, since it is used to measure the fitness, the values reach near 1 and the differences in these models are not distinct. Regarding the accuracy related RMSE metric, the searched architecture outperforms the other models. It can also be found that some of the alternative models can achieve better ME metrics. In the overall view, the NAS strategy shows the effectiveness in model searching.

4.4. Influence of noise

The sensors may produce noisy data and this will influence the performance of data-driven methods. To measure such kind of influence, the white Gaussian noise is added to the measurements of V, I, T. The addictive noise has a mean of zero and a mean standard deviation of 5%. The model is trained and tested again under such a noisy environment.

The RMSE values with and without the Gaussian noise are shown in Fig. 10. It can be found the RMSE metrics present the increases in different levels, which indicates the performance of models both face challenges. For the Oxford dataset, the larger capacity estimation errors occur compared with those in the MIT-Stanford dataset.

To evaluate the performance, the absolute error of $|y_i - \hat{y}_i|$ is calculated for the estimated capacity in each cycle. The statistical results are shown in the form of box plot. The comparison methods are also conducted. Figs. 11 (a) and (b) present the error statistics for the two datasets. For the MIT-Stanford dataset, the proposed method has the fewest outliers. Besides, the error ranges are also limited. For the Oxford dataset, in most cases, the proposed method presents lower errors and also more concentrated error ranges. In addition, the comparison methods obtain inconsistent performances in the two datasets. AD-TCN achieves not prominent but stable results in MIT-Stanford dataset. However, the error ranges become larger and also present evident variations in different battery IDs of Oxford dataset. I-PCNN can obtain relatively better results in Oxford dataset, but its performances drop in the other dataset. The GCN model can achieve competitive results for MIT-Stanford dataset, while the error ranges become the largest for another dataset. Besides, some literature uses the incremental capacity analysis which needs differential operation conducted on voltage information. These categories of methods will be severely affected since the differential operation tends to amplify measurement noise.

Overall, the proposed scheme presents a certain degree of advantage when facing noisy data. Nonetheless, the noise worsens the capacity estimation. Necessary filtering pre-processing steps can be applied before data input to the model.

In this section, the hyper-parameter of window length is investigated. The chosen value is reasonable to obtain satisfactory results. The components of the proposed framework, namely the GNN layer and NAS, are discussed to demonstrate their necessity and efficiency. The performances under noisy environments are also analyzed to verify

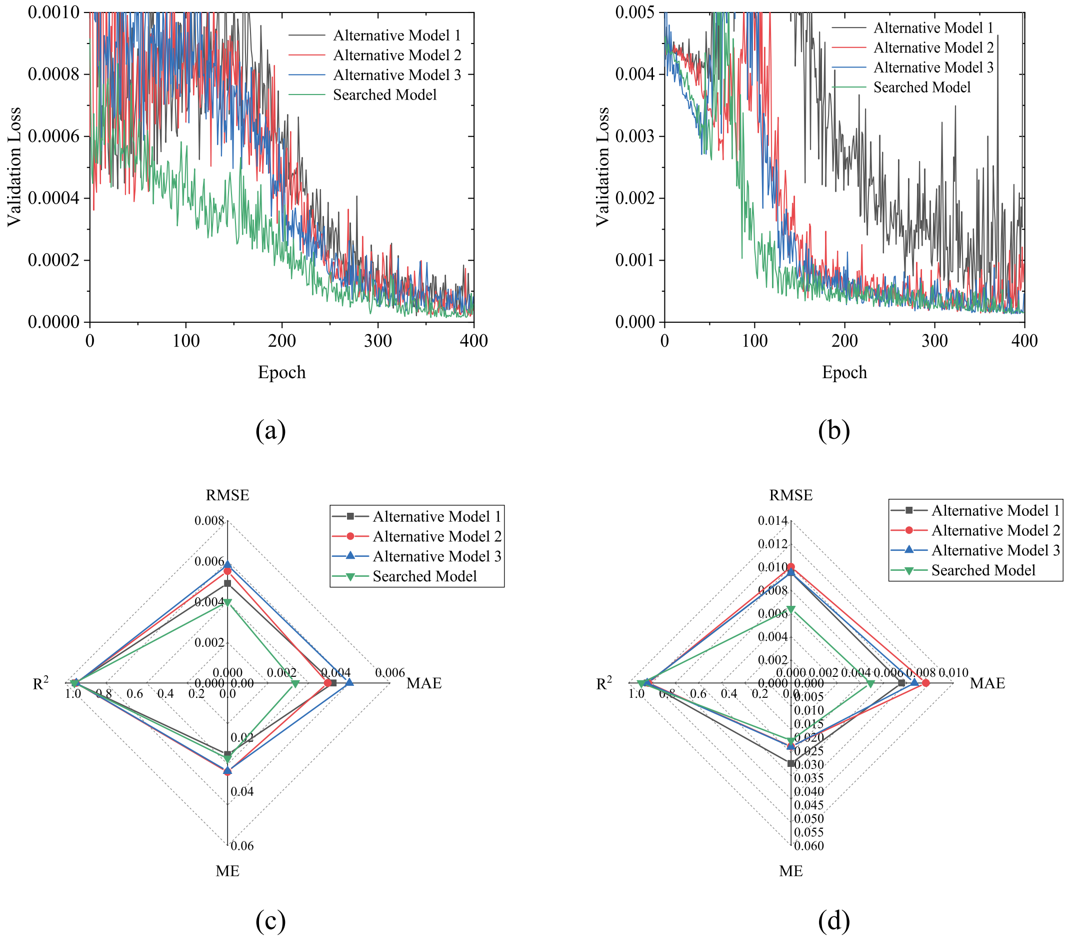


Fig. 9. The verification of NAS. (a) The validation loss curve in the MIT-Stanford dataset. (b) The validation loss curve in the Oxford dataset. (c) The metrics of capacity estimation in the MIT-Stanford dataset. (d) The metrics of capacity estimation in the Oxford dataset.

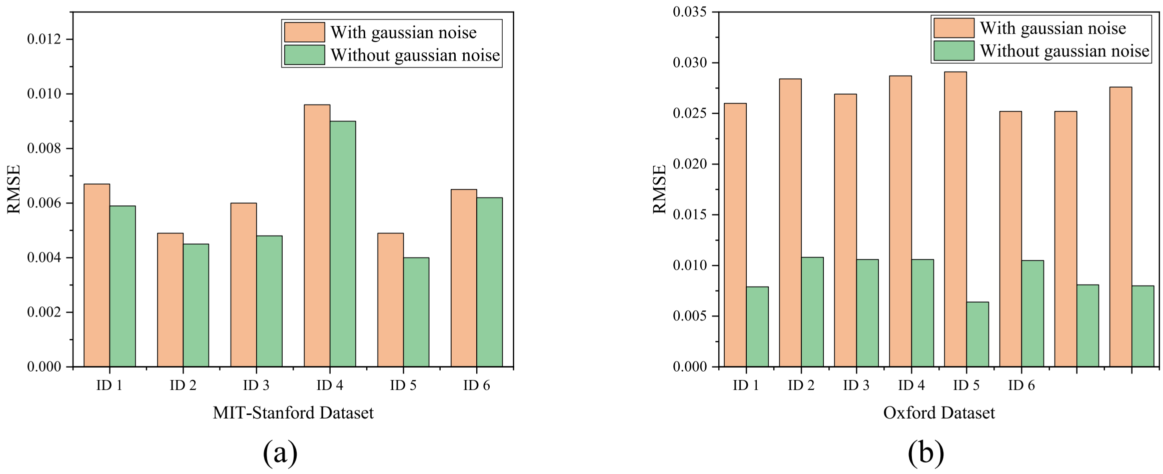


Fig. 10. The error caused by noise. (a) The RMSE value in the MIT-Stanford dataset. (b) The RMSE value in the Oxford dataset.

the advantage. However, some drawbacks need to be mentioned. The implementation of NAS requires the computation of all the possible operations and the process is time-consuming. The GNN model adopts the measurements from multiple sensors. If one of the sensors fails, it will influence the normal function of this model. Overall, the proposed method presents great potential for the battery management system. With the real-time monitoring data of voltage, current, and temperature, the method can be deployed online to predict the capacity, which provides the prospect for practical engineering application.

5. Conclusions

This work proposes the measurements aggregation and feature fusion scheme to estimate the capacity of lithium-ion batteries. The MIC is introduced to organize the measurements into graph structure. Different GNN layers are designed to accomplish the data aggregation and further feature fusion. To demonstrate the effectiveness of proposed scheme, the MIT-Stanford and Oxford public datasets are applied to

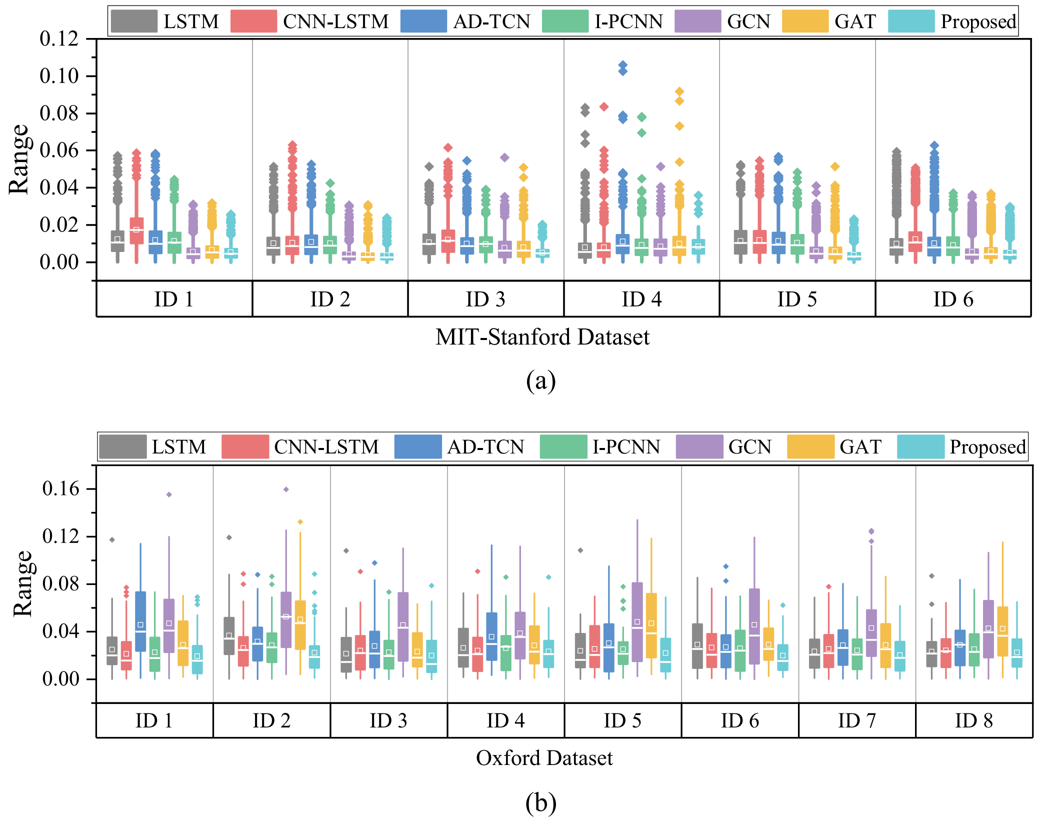


Fig. 11. The results after the introduction of Gaussian noise. (a) The estimation errors in the MIT-Stanford dataset. (b) The estimation errors in the Oxford dataset.

conduct the algorithm. The proposed scheme presents superior and consistent performance for capacity estimation. The GNN layers provide the effective modeling capability and aggregate the data efficiently. The adopted NAS successfully obtain the optimal architectures for the two datasets. The experiments under noise further indicate the superiority of the proposed scheme. In all, this work provides a potential avenue towards accurate capacity estimation. The future works include the enhancement of computational efficiency, improvement for lightweight model and investigation of robustness under sensor failure.

CRediT authorship contribution statement

Zhe Wang: Conceptualization, Methodology, Experiments, Writing – original draft. **Fangfang Yang:** Conceptualization, Methodology, Writing – review & editing. **Qiang Xu:** Validation, Investigation, Formal analysis, Writing – review & editing. **Yongjian Wang:** Validation, Investigation, Writing – review & editing. **Hong Yan:** Supervision, Project administration, Writing – review & editing. **Min Xie:** Supervision, Project administration, Writing – review & editing.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by National Natural Science Foundation of China (71971181, 72032005 and 62203482), the Research Grant Council of Hong Kong (11203519, 11200621), the Hong Kong Innovation and Technology Commission (InnoHK Project CIMDA), and Guangdong Basic and Applied Basic Research Foundation (2021A1515110354).

References

[1] Roman D, Saxena S, Robu V, Pecht M, Flynn D. Machine learning pipeline for battery state-of-health estimation. Nat Mach Intell 2021;3(5):447–56. http://dx.doi.org/10.1038/s42256-021-00312-3.

[2] Ng M-F, Zhao J, Yan Q, Conduit GJ, Seh ZW. Predicting the state of charge and health of batteries using data-driven machine learning. Nat Mach Intell 2020;2(3):161–70. http://dx.doi.org/10.1038/s42256-020-0156-7.

[3] Hu X, Xu L, Lin X, Pecht M. Battery lifetime prognostics. Joule 2020;4(2):310–46. http://dx.doi.org/10.1016/j.joule.2019.11.018.

[4] Wei Z, Hu J, He H, Yu Y, Marco J. Embedded distributed temperature sensing enabled multistate joint observation of smart lithium-ion battery, IEEE Trans Ind Electron 70:555–65. https://doi.org/10.1109/TIE.2022.3146503.

[5] Zhang Y, Xiong R, He H, Pecht MG. Lithium-ion battery remaining useful life prediction with Box–Cox transformation and Monte Carlo simulation. IEEE Trans Ind Electron 2019;66(2):1585–97. http://dx.doi.org/10.1109/TIE.2018.2808918.

[6] He W, Williard N, Osterman M, Pecht M. Prognostics of lithium-ion batteries based on Dempster–Shafer theory and the Bayesian Monte Carlo method. J Power Sources 2011;196(23):10314–21. http://dx.doi.org/10.1016/j.jpowsour.2011.08.040.

[7] Dai H, Zhao G, Lin M, Wu J, Zheng G. A novel estimation method for the state of health of lithium-ion battery using prior knowledge-based neural network and Markov chain. IEEE Trans Ind Electron 2019;66(10):7706–16. http://dx.doi.org/10.1109/TIE.2018.2880703.

[8] Li X, Yuan C, Wang Z. State of health estimation for Li-ion battery via partial incremental capacity analysis based on support vector regression. Energy 2020;203:117852. http://dx.doi.org/10.1016/j.energy.2020.117852.

[9] Li Y, Zou C, Berecibar M, Nanini-Maury E, Chan JCW, van den Bossche P, et al. Random forest regression for online capacity estimation of lithium-ion batteries. Appl Energy 2018;232:197–210. http://dx.doi.org/10.1016/j.apenergy.2018.09.182.

[10] Hu C, Jain G, Schmidt C, Strief C, Sullivan M. Online estimation of lithium-ion battery capacity using sparse Bayesian learning. J Power Sources 2015;289:105–13. http://dx.doi.org/10.1016/j.jpowsour.2015.04.166.

[11] Wang D, Yang F, Tsui K-L, Zhou Q, Bae SJ. Remaining useful life prediction of lithium-ion batteries based on spherical cubature particle filter, IEEE Trans Instrum Meas 65 (6):1282–91. https://doi.org/10.1109/TIM.2016.2534258.

[12] Zhu J, Wang Y, Huang Y, Bhushan Gopaluni R, Cao Y, Heere M, et al. Data-driven capacity estimation of commercial lithium-ion batteries from voltage relaxation. Nature Commun 2022;13(1). http://dx.doi.org/10.1038/s41467-022-29837-w.

[13] Richardson RR, Birkl CR, Osborne MA, Howey DA. Gaussian process regression for *in situ* capacity estimation of lithium-ion batteries. IEEE Trans Ind Inf 2019;15(1):127–38. http://dx.doi.org/10.1109/TII.2018.2794997.

[14] She C, Li Y, Zou C, Wik T, Wang Z, Sun F. Offline and online blended machine learning for lithium-ion battery health state estimation. IEEE Trans Transp Electrif 2022;8(2):1604–18. http://dx.doi.org/10.1109/TTE.2021.3129479.

[15] Lyu C, Lai Q, Ge T, Yu H, Wang L, Ma N. A lead-acid battery's remaining useful life prediction by using electrochemical model in the particle filtering framework. Energy 2017;120:975–84. http://dx.doi.org/10.1016/j.energy.2016.12.004.

[16] Bian X, Liu L, Yan J, Zou Z, Zhao R. An open circuit voltage-based model for state-of-health estimation of lithium-ion batteries: Model development and validation. J Power Sources 2020;448:227401. http://dx.doi.org/10.1016/j.jpowsour.2019.227401.

[17] Guha A, Patra A. State of health estimation of lithium-ion batteries using capacity fade and internal resistance growth models, IEEE Trans Transp Electrif 4 (1):135–46. https://doi.org/10.1109/TTE.2017.2776558.

[18] Zhang Y, Li Y-F. Prognostics and health management of lithium-ion battery using deep learning methods: A review. Renew Sustain Energy Rev 2022;161. http://dx.doi.org/10.1016/j.rser.2022.112282.

[19] Han T, Wang Z, Meng H. End-to-end capacity estimation of lithium-ion batteries with an enhanced long short-term memory network considering domain adaptation. J Power Sources 2022;520:230823. http://dx.doi.org/10.1016/j.jpowsour.2021.230823.

[20] Wei Z, Quan Z, Wu J, Li Y, Pou J, Zhong H. Deep deterministic policy gradient-drl enabled multiphysics-constrained fast charging of lithium-ion battery, IEEE Trans Ind Electron 69:2588–98. https://doi.org/10.1109/TIE.2021.3070514.

[21] Tan Y, Zhao G. Transfer learning with long short-term memory network for state-of-health prediction of lithium-ion batteries. IEEE Trans Ind Electron 2020;67(10):8723–31. http://dx.doi.org/10.1109/TIE.2019.2946551.

[22] Li W, Sengupta N, Dechent P, Howey D, Annaswamy A, Sauer DU. Online capacity estimation of lithium-ion batteries with deep long short-term memory networks. J Power Sources 2021;482:228863. http://dx.doi.org/10.1016/j.jpowsour.2020.228863.

[23] Chen Z, Zhao H, Zhang Y, Shen S, Shen J, Liu Y. State of health estimation for lithium-ion batteries based on temperature prediction and gated recurrent unit neural network. J Power Sources 2022;521:230892. http://dx.doi.org/10.1016/j.jpowsour.2021.230892.

[24] Hong J, Lee D, Jeong E-R, Yi Y. Towards the swift prediction of the remaining useful life of lithium-ion batteries with end-to-end deep learning. Appl Energy 2020;278:115646. http://dx.doi.org/10.1016/j.apenergy.2020.115646.

[25] Li Y, Li K, Liu X, Wang Y, Zhang L. Lithium-ion battery capacity estimation — A pruned convolutional neural network approach assisted with transfer learning. Appl Energy 2021;285:116410. http://dx.doi.org/10.1016/j.apenergy.2020.116410.

[26] Zhou J, Cui G, Hu S, Zhang Z, Yang C, Liu Z, et al. Graph neural networks: A review of methods and applications. AI Open 2020;1:57–81. http://dx.doi.org/10.1016/j.aiopen.2021.01.001.

[27] Wu Z, Pan S, Chen F, Long G, Zhang C, Yu PS. A comprehensive survey on graph neural networks. IEEE Trans Neural Netw Learn Syst 2021;32(1):4–24. http://dx.doi.org/10.1109/TNNLS.2020.2978386.

[28] Shen S, Sadoughi M, Li M, Wang Z, Hu C. Deep convolutional neural networks with ensemble learning and transfer learning for capacity estimation of lithium-ion batteries. Appl Energy 2020;260:114296. http://dx.doi.org/10.1016/j.apenergy.2019.114296.

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

[30] Zraibi B, Okar C, Chaoui H, Mansouri M. Remaining useful life assessment for lithium-ion batteries using CNN-LSTM-DNN hybrid method. IEEE Trans Veh Technol 2021;70(5):4252–61. http://dx.doi.org/10.1109/TVT.2021.3071622.

[31] Jiao R, Peng K, Dong J. Remaining useful life prediction of lithium-ion batteries based on conditional variational autoencoders-particle filter. IEEE Trans Instrum Meas 2020;69(11):8831–43. http://dx.doi.org/10.1109/TIM.2020.2996004.

[32] Tang Y, Yang K, Zheng H, Zhang S, Zhang Z. Early prediction of lithium-ion battery lifetime via a hybrid deep learning model. Measurement 2022;199:111530. http://dx.doi.org/10.1016/j.measurement.2022.111530.

[33] He J, Tian Y, Wu L. A hybrid data-driven method for rapid prediction of lithium-ion battery capacity. Reliab Eng Syst Saf 2022;108674. http://dx.doi.org/10.1016/j.ress.2022.108674.

[34] Yang Y. A machine-learning prediction method of lithium-ion battery life based on charge process for different applications. Appl Energy 2021;292:116897. http://dx.doi.org/10.1016/j.apenergy.2021.116897.

[35] Ma Y, Shan C, Gao J, Chen H. Multiple health indicators fusion-based health prognostic for lithium-ion battery using transfer learning and hybrid deep learning method. Reliab Eng Syst Saf 2022;108818. http://dx.doi.org/10.1016/j.ress.2022.108818.

[36] Reshef DN, Reshef YA, Finucane HK, Grossman SR, McVean G, Turnbaugh PJ, et al. Detecting novel associations in large data sets. Science 2011;334(6062):1518–24.

[37] Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. In: 5th international conference on learning representations, ICLR 2017, Toulon, France, April 24-26, 2017, conference track proceedings. OpenReview.net; 2017, URL https://openreview.net/forum?id=SJU4ayYgl.

[38] Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. In: Lee DD, Sugiyama M, von Luxburg U, Guyon I, Garnett R, editors. Advances in neural information processing systems 29: annual conference on neural information processing systems 2016, December 5-10, 2016, Barcelona, Spain. 2016, p. 3837–45, URL https://proceedings.neurips.cc/paper/2016/hash/04df4d434d481c5bb723be1b6df1ee65-Abstract.html.

[39] Velickovic P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph attention networks. In: 6th international conference on learning representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, conference track proceedings. OpenReview.net; 2018, URL https://openreview.net/forum?id=rJXMpikCZ.

[40] Morris C, Ritzert M, Fey M, Hamilton WL, Lenssen JE, Rattan G, et al. Weisfeiler and leman go neural: Higher-order graph neural networks. In: The thirty-third AAAI conference on artificial intelligence, AAAI 2019, the thirty-first innovative applications of artificial intelligence conference, IAAI 2019, the ninth AAAI symposium on educational advances in artificial intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019. AAAI Press; 2019, p. 4602–9. http://dx.doi.org/10.1609/aaai.v33i01.33014602.

[41] Wei L, Zhao H, He Z. Designing the topology of graph neural networks: A novel feature fusion perspective. In: Proceedings of the ACM web conference 2022. 2022, p. 1381–91.

[42] Liu H, Simonyan K, Yang Y. DARTS: differentiable architecture search. In: 7th international conference on learning representations, ICLR 2019, New Orleans, la, USA, May 6-9, 2019. OpenReview.net; 2019.

[43] Severson KA, Attia PM, Jin N, Perkins N, Jiang B, Yang Z, et al. Data-driven prediction of battery cycle life before capacity degradation. Nat Energy 2019;4(5):383–91. http://dx.doi.org/10.1038/s41560-019-0356-8.

[44] Birkl C. Diagnosis and prognosis of degradation in lithium-ion batteries (Ph.D. thesis), University of Oxford; 2017, Dissertation/Thesis ER.

[45] Toughzaoui Y, Toosi SB, Chaoui H, Louahlia H, Petrone R, Le Masson S, et al. State of health estimation and remaining useful life assessment of lithium-ion batteries: A comparative study. J Energy Storage 2022;51:104520. http://dx.doi.org/10.1016/j.est.2022.104520.

[46] Zhou D, Wang B. Battery health prognosis using improved temporal convolutional network modeling. J Energy Storage 2022;51:104480. http://dx.doi.org/10.1016/j.est.2022.104480.