Applied Energy 355 (2024) 122332

Contents lists available at ScienceDirect

Applied Energy

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

Lithium-ion battery state-of-health estimation: A self-supervised framework incorporating weak labels

Tianyu Wang [a], Zhongjing Ma [a], Suli Zou [a,*], Zhan Chen [a], Peng Wang [b]

[a] School of Automation, Beijing Institute of Technology, Beijing 100081, China
[b] Department of Electrical Engineering, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Keywords:
Lithium-ion battery
Self-supervised learning
State-of-health estimation
Vision transformer
Deep learning

ABSTRACT

The State-of-Health (SOH) estimation of Lithium-ion (Li-ion) batteries is critical for the safe and reliable operation of the batteries. Deep learning technologies are currently the popular methods for SOH estimation due to the advantages of no modeling and automatic feature extraction. However, existing methods require a large amount of annotated data to ensure model fitting, and the collection and labeling of battery aging data are time-consuming and laborious. Therefore, a self-supervised framework incorporating weak labels (SSF-WL) is proposed in this paper to obtain excellent estimation results on a small amount of annotated data. First, a novel data processing method based on the Gramian angular field, difference calculation, and raw data is proposed to enrich information and enhance features. Then, a five-layer Transformer encoder is constructed in SSF-WL for feature extraction. Finally, the model is pre-trained and fine-tuned on the proposed SSF-WL to obtain the estimated results of SOH. The proposed method is validated on the 124 commercial battery and Oxford databases. Experiments indicate that when using only 30% of the annotated training data, the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) obtained by SSF-WL are 0.5219%/0.6085% lower than traditional supervised learning on the 124 commercial battery database, respectively. Moreover, the SSF-WL pre-trained model on a large unannotated database can be transferred to different types of batteries with a small annotated database and obtains on-par or better estimation results than the model trained from scratch.

1. Introduction

Lithium-ion (Li-ion) batteries have been used as power sources in numerous fields due to their high energy density, long lifetime, and low self-discharge rate [1]. However, the internal resistance increases and capacity fading resulting from battery aging will significantly reduce battery performance and increase security risk [2]. Therefore, the online estimation of the battery State-of-Health (SOH) is crucial in managing battery life and eliminating potential failures [3].

Existing SOH estimation methods mainly include three categories: direct measurement methods, model-based methods, and data-driven methods. The Coulomb counting [4] and open circuit voltage methods [5] are typical direct measurement methods. These methods are simple and efficient but are limited to offline SOH estimation in the laboratory due to accumulated errors or measurement conditions. The model-based methods realize the online SOH estimation by establishing empirical, electrochemical, and equivalent circuit models. [6] realized the state estimation of batteries based on the equivalent circuit model, and the constrained ensemble Kalman filter was introduced to improve the model performance. The combination method can effectively improve the estimation accuracy, but the model may lack the ability to capture the aging characteristics of batteries. Therefore, Amir et al. [7] improved the equivalent circuit model to capture the degradation relationships of Li-ion batteries, including time and temperature factors. Further, [8,9], and [10] constructed electrochemical models to realize SOH estimation, addressing the limitations of the equivalent circuit model in inherent physical representations. However, these methods are limited by extensive expertise and complex mathematical calculations. Furthermore, Zheng et al. [11] proposed an empirical model combining segmented charging curves and the discrete Arrhenius aging model. Although most model-based methods provide intuitive physical insights, the accuracy of the estimate is heavily dependent on the model quality. In fact, establishing a reliable model relies on many factors, including extensive expertise, complex mathematical calculations, and

* Corresponding author.
E-mail addresses: 3120215487@bit.edu.cn (T. Wang), mazhongjing@bit.edu.cn (Z. Ma), sulizou@bit.edu.cn (S. Zou), 3220200729@bit.edu.cn (Z. Chen), wangpeng2020@mail.tsinghua.edu.cn (P. Wang).

https://doi.org/10.1016/j.apenergy.2023.122332
Received 27 August 2023; Received in revised form 17 October 2023; Accepted 11 November 2023
Available online 18 November 2023

Nomenclature

CC	Constant current
CNN	Convolutional neural network
CV	Constant voltage
GAF	Gramian angular field
GELU	Gaussian error linear unit
GRU	Gate recurrent unit
Li-ion	Lithium-ion
LSTM	Long short-term memory
MAE	Mean absolute error
MAPE	Mean absolute percentage error
MSE	Mean square error
RMSE	Root mean square error
SOC	State-of-charge
SOH	State-of-health
SSF-WL	Self-supervised framework incorporating weak labels
TCN	Temporal Convolutional Network
ViT	Vision Transformer

the researcher's personal experience. These all limit the widespread use of model-based methods.

Compared to model-based methods, data-driven methods only require establishing the nonlinear mapping relationship between battery charge–discharge aging data and SOH to achieve SOH estimation without considering internal reactions and aging mechanisms of the battery. They are mainly represented by machine learning-based and deep learning-based methods. For machine learning-based methods, [12,13] proposed Gaussian process regression models based on partial incremental capacity curves and multiple energy features, respectively. Further, [14,15] developed support vector regression models to estimate SOH. The input features were extracted from battery charging curves and charge–discharge curves, respectively. Furthermore, Driscoll et al. [16] extracted features from charging curves and realized SOH estimation based on the artificial neural network. Although machine learning methods do not require the construction of complex mathematical models, they rely on manual feature extraction, which is time-consuming and labor-intensive. Moreover, the limited expressive ability of such hand-crafted features leads to model application-specific.

The automatic feature extraction and better generalization performance give the potential to apply deep learning. Fan et al. [17] proposed a Gate Recurrent Unit-Convolutional Neural Network (GRU-CNN) for SOH estimation with charging voltage, current, and temperature sequences as input. This simple combination model may have a lot of room for improvement. Therefore, [18] used dilated CNN and bidirectional GRU to improve the above GRU-CNN framework. [19] proposed an encoder–decoder model consisting of four modules for SOH estimation. Further, Temporal Convolutional Networks (TCNs) have received significant attention for their remarkable capability in handling sequential tasks. [20] utilized TCN for SOH estimation on multiple charging segments and applied Bayesian hyperparameter tuning to optimize model parameters. Zhang et al. [21] developed a TCN model with the modified flower pollination algorithm and incorporated ohmic resistance trajectories as additional inputs to improve accuracy. However, due to the receptive field setting in TCN, this model has challenges when dealing with domain transfer. To extract important features, Xiong et al. [22] performed feature selection from measured and calculated parameters by combining four selection algorithms. Although this method can extract effective features, the expressive ability of manually extracted fixed-form features is limited. In fact, supervised training makes the above methods heavily reliant on the amount of annotated data. In addition, these methods have insufficient generalization ability when distribution differences exist between databases. Therefore, some scholars have introduced transfer learning to reduce the model's dependence on annotated data and improve the generalization ability. In paper [23], a seven-layer CNN was pre-trained on a large annotated source domain database. Then, the model was transferred to a small annotated target domain database for fine-tuning. [24,25] employed the maximum mean discrepancy technology to achieve domain adaptation between databases with distribution differences. The target domain data is unannotated in domain adaptation methods, but annotated source domain data is required for model training. Note that domain adaptation methods require access to target domain data during training, which can be challenging in practice. The achievements of existing deep learning-based methods are remarkable, but they still have some limitations, as follows:

1. Feature enhancement: Most current methods only perform simple processing on raw data without enhancing features or incorporating prior information, which is crucial for improving the performance of deep learning models.
2. Data dependency: The performance of existing methods heavily depends on the amount of annotated data. Acquiring annotated battery aging data is time-consuming and costly as it is indirectly measured under laboratory conditions by other methods.
3. Model generalization: Existing methods often overlook the generalization ability of the model. Although some transfer learning-based methods have been proposed, they still rely on large annotated databases for pre-training or supervised training.

According to the above limitations of existing methods, this paper aims to propose an effective data enhancement method, reduce the model's dependence on annotated data, and improve the generalization ability of the model. Therefore, a novel Li-ion battery SOH estimation method based on the constructed self-supervised framework incorporating weak labels (SSF-WL) is proposed in this paper. The main contributions of this paper are as follows:

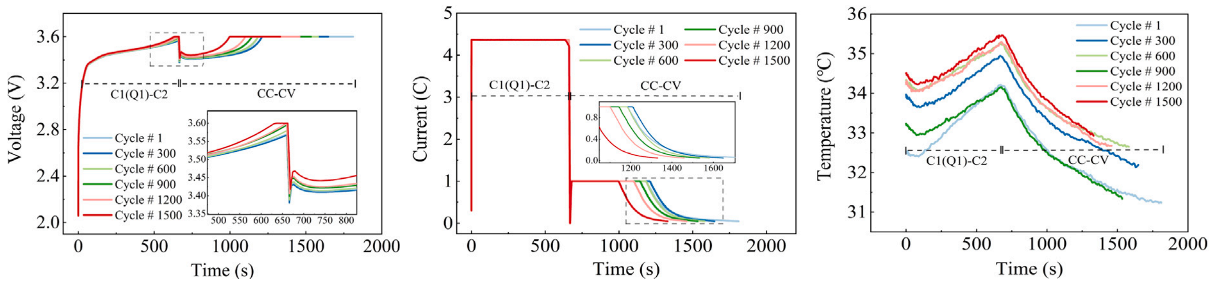
1. A data processing method combining the Gramian Angular Field (GAF) [26], difference calculation, and raw data is proposed to convert the raw sequence data into a 3D matrix. This method can effectively enrich original information and enhance features.
2. A simple Transformer-based five-layer encoder is constructed in the SSF-WL framework for feature extraction. The introduction contributes to the improvement of SOH estimation accuracy.
3. The SSF-WL is proposed to reduce the model's dependence on annotated data. The framework achieves competitive SOH estimation results on a small amount of annotated data.
4. The SSF-WL can realize performance transfer between batteries with different charge–discharge conditions or different materials on a small amount of annotated data and achieves on-par or better results compared to the model trained from scratch on a new battery.

The rest of this paper is organized as follows: Section 2 analyzes the aging characteristics of Li-ion batteries. Section 3 describes our proposed SOH estimation method in detail. Section 4 discusses our experimental results. Finally, the conclusions are presented in Section 5.

2. Aging characteristics analyses of lithium-ion batteries

Battery SOH reflects valuable information not only about the health condition but also about the reliability of a battery system. The short of the battery life is revealed as the capacity fading [17,27], and we apply a typical expression of SOH defined based on the capacity fading of a Li-ion battery [17,27] in this paper, as below:

$$\mathrm{SOH} = \frac{C_{\mathrm{d}}}{C_{\mathrm{n}}} \times 100\% \tag{1}$$

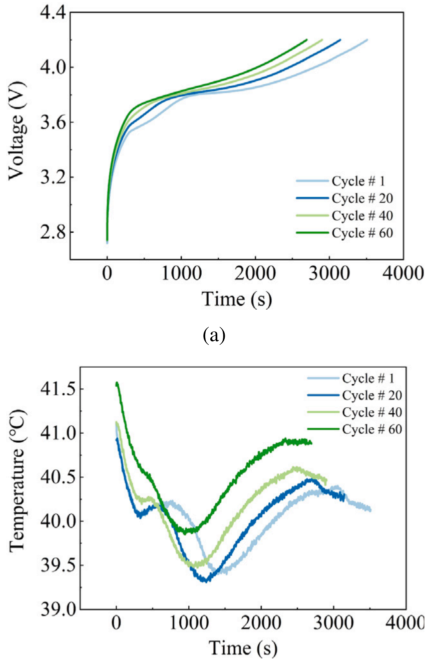


(a)

(b)

(c)

Fig. 1. Examples of battery charging cycle curves from the 124 commercial battery database: (a) voltage curves; (b) current curves; (C) temperature curves.



(a)

(b)

Fig. 2. Examples of battery charging cycle curves from the Oxford database: (a) voltage curves; (b) temperature curves.

where C_n is the nominal capacity, and C_d is the maximum discharging capacity, which can be obtained by the Coulomb counting method. As widely applied, a battery reaches the end of life if its capacity is reduced to 70%~80% of its nominal capacity. Since the nonlinear nature and complex aging mechanisms of batteries, it is challenging to estimate SOH directly and accurately using Eq. (1).

Deep learning methods model the nonlinear mapping relationship between input data and SOH by automatic feature extraction. Such methods do not directly consider the internal reactions and aging mechanisms of the battery. Nevertheless, to estimate SOH accurately, it is necessary to carefully select and analyze measurable data, prioritizing data related to battery aging as the model input. Many characteristics have a direct relationship with battery health, e.g., charging/discharging speed, charging/discharging voltage and current, and temperature. In Fig. 1, we give examples of charging cycle curves for the Li-ion battery. Note that Fig. 1 is drawn by using the actual charging data provided by the 124 commercial battery database [28]. For comparison, examples of charging curves from another database are shown in Fig. 2, using the data from the Oxford Battery Degradation Dataset [29]. Since the charging data could be achieved more stably than the discharging data in practical applications, we only focus on the charging cycle curves. In the experiment part of the paper, we also apply the two databases to evaluate our proposed method.

As indicated in Fig. 1, the voltage and current curves significantly differ in the Constant Current–Constant Voltage (CC–CV) step. As the battery ages, the CC charging time is considerably shortened, and the voltage rises faster. Meanwhile, the average temperature of the battery gradually increases. Consistently, the same changing trends of the battery in the Oxford database are also shown in Fig. 2. These trends are explicit features for neural networks, so the complete or partial charging voltage, current, and temperature data for each cycle are applied as the model input to estimate SOH. Moreover, the charging capacity corresponding to complete or partial charging data is highly correlated with SOH, which can be calculated online by Eq. (2), i.e., the Coulomb counting method, where t_1 and t_2 are the start and end times, and I is the charging current. Therefore, the charging capacity C_c is used as the weak label to assist self-supervised pre-training (detailed in Section 3.2), making the model more suitable for capacity-related downstream tasks.

$$C_c = \int_{t_1}^{t_2} I(t)dt \tag{2}$$

So far, numerous methods have been proposed for SOH estimation, and deep learning techniques have become increasingly popular because of the advantages of no modeling, automatic feature extraction, and better generalization performance. In the next section, a self-supervised-based approach is developed to characterize the relationship between charging parameters and the SOH change and then realize the SOH estimation with a small amount of annotated data.

3. SOH estimation method based on self-supervision

In this section, a new data-driven approach called SSF-WL is presented by integrating the Transformer encoder and self-supervised learning pipeline. As stated in the previous section, we use the complete or partial charging data as the input of the proposed method. Since the collected raw charging data is redundant and noisy and contains limited useful information for SOH estimation, the first is to pre-process the raw data, enhance features, and enrich information. Advanced feature extraction networks can improve the accuracy of SOH estimation, and the Transformer has achieved the best results in various fields due to the ability of global information extraction. Therefore, the second is to build a Transformer encoder for feature extraction. Supervised learning methods require a large amount of annotated data to ensure model training. However, collecting and labeling battery aging data is time-consuming and cumbersome. Hence, the third is to build a self-supervised learning pipeline to obtain competitive estimation results using a small amount of annotated data. In addition, self-supervised learning enables the model to have better generalization performance.

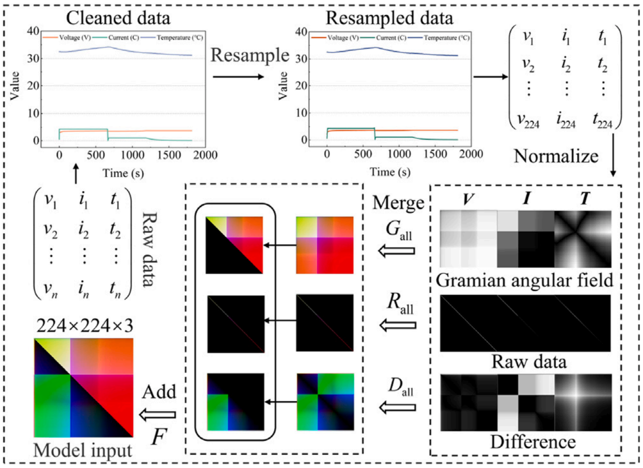


Fig. 3. Data processing deduction process.

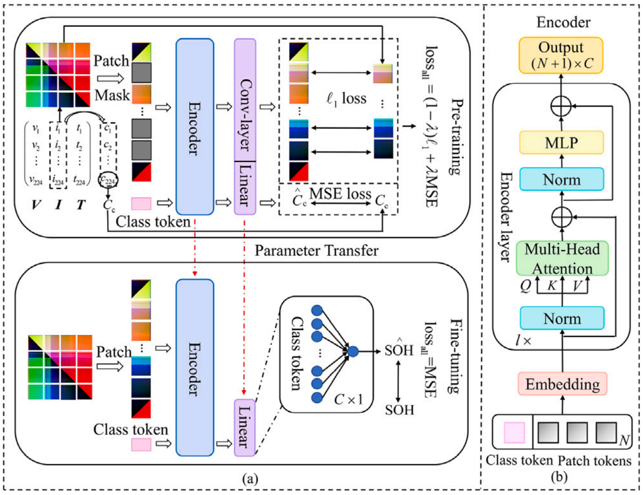


Fig. 4. Overall structure of the SSF-WL and encoder: (a) SSF-WL; (b) encoder, where Norm is the layer normalization, Conv is the convolutional layer, and Linear is the fully connected layer.

3.1. Data pre-processing and feature enhancement

For each cycle, the raw data is an $n \times 3$ matrix consisting of three sequences of voltage (V), current (I), and temperature (T) with n sampling points. After enhancing the features through data processing, the raw data is converted into a 224 × 224 × 3 three-dimensional matrix (image matrix) as the model input. The deduction process of the proposed data processing method is shown in Fig. 3.

The raw input data contains many outliers, such as zero and negative values. Therefore, data cleaning is first performed to ensure the validity of the data. Then, to remove redundant information and standardize the input dimensions while maintaining the curve characteristics of the raw data, the cubic spline interpolation, which has advantages such as preserving smoothness and effectively capturing local features, is introduced to resample the cleaned data. To maintain compatibility with existing visual models such as ResNet [30] and Vision Transformer (ViT) [31], the number of samples is set to 224. As shown in Fig. 3, the charging curves before and after resampling are basically the same. This indicates that the method is able to meet our task requirements effectively. Further, the min–max normalization is applied to normalize the resampled V, I, and T sequences. For each sequence X, the normalized is calculated by Eq. (3), where $x_{\min}$ and $x_{\max}$ are the minimum and maximum values in the sequence X, x_i is the i-th value of the sequence, and $\tilde{x_i}$ is the normalized value.

$$\tilde{x_i} = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{3}$$

Furthermore, GAF [26] is applied to add time correlations to the normalized data. First, the normalized sequence $\tilde{X}$ is transformed into polar axes, as follows:

$$\begin{cases} \phi_i = \arccos\left(\tilde{x}_i\right), \tilde{x}_i \in \tilde{X} \\ r_i = \frac{t_i}{M}, \qquad t_i \in \mathbb{N} \end{cases} \tag{4}$$

where $i \in \{1, 2, \ldots, 224\}$, $\tilde{x}_i \in [0, 1]$, ϕ_i is the angle, r_i is the radius, t_i is the time stamp, and M is the constant factor. Subsequently, the cosine sum is calculated from the angles to obtain the time correlations among different values as Eq. (5), where $j \in \{1, 2, \ldots, 224\}$. Then, the GAF matrix G is calculated by Eq. (6). The corresponding 224 × 224 matrices G_v, G_i, and G_t can be obtained from the sequences V, I, and T. Finally, the G_v, G_i, and G_t are concatenated into a 224 × 224 × 3 three-dimensional matrix G_all as shown in Fig. 3.

$$g_{i,j} = \cos\left(\phi_i + \phi_j\right) \tag{5}$$

$$G(\tilde{X}) = \begin{pmatrix} g_{1,1} & g_{1,2} & \cdots & g_{1,224} \\ g_{2,1} & g_{2,2} & \cdots & g_{2,224} \\ \vdots & \vdots & \ddots & \vdots \\ g_{224,1} & g_{224,2} & \cdots & g_{224,224} \end{pmatrix} \tag{6}$$

The effectiveness of the feature enhancement method using only GAF has been verified in [32]. However, the matrix G_all contains significant redundant information, as the upper and lower triangles have the same values. Obviously, relying solely on GAF for feature enhancement is not an ideal approach. Therefore, we further improve the data processing method by analyzing differential information and original characteristics. In fact, raw data is the best carrier of original properties, and the difference calculation can highlight the differences between values in the time sequence, effectively describing the changing trends of the sequence. Hence, we mask the lower triangular and diagonal values of the G_all and introduce the difference calculation and raw data. Specifically, the V, I, and T raw normalized sequences constitute the diagonal values of the second three-dimensional matrix R_all. The difference matrix D of a single sequence is calculated by Eqs. (7) and (8). Similarly, we concatenate the matrices D_v, D_i, and D_t to obtain the third three-dimensional matrix D_all and mask the upper triangular and diagonal values. As a result, the matrices G_all, R_all, and D_all are added to acquire the final model input matrix (image) F with the dimension 224 × 224 × 3, which contains time correlations, differential information, and original characteristics.

$$d_{i,j} = |x_i - x_j| \tag{7}$$

$$D(\tilde{X}) = \begin{pmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,224} \\ d_{2,1} & d_{2,2} & \cdots & d_{2,224} \\ \vdots & \vdots & \ddots & \vdots \\ d_{224,1} & d_{224,2} & \cdots & d_{224,224} \end{pmatrix} \tag{8}$$

In summary, we first perform data cleaning, resampling, and normalization operations for the n × 3 raw data matrix consisting of charging V, I, and T sequences. Then, GAF and the difference calculation are performed on the three sequences of the normalized matrix to obtain the GAF and difference matrices G_all and D_all with the dimension 224 × 224 × 3. After masking the redundant data, the above two matrices and the normalized raw data matrix R_all are added to construct the final model input image matrix F with the dimension 224 × 224× 3.

3.2. Self-supervised framework incorporating weak labels

The proposed SSF-WL is shown in Fig. 4(a). It includes an encoder and a self-supervised learning pipeline. The encoder is used for automatic feature extraction, and its performance directly affects the accuracy of SOH estimation. Self-supervised learning aims to construct

Table 1
The default parameters of ViT and self-supervised learning.

Hyperparameter	Value	Hyperparameter	Value
Input dimension	224 × 224 × 3	Mask size	32 × 32
Encoder layers (l)	5	Mask ratio	0.5
Embedding dimension (C)	384	λ	0.8
Number of heads (H)	12	Main loss	ℓ_1 loss
MLP dimension	$C \times 4$	Auxiliary loss	MSE loss
Patch size	16×16	Activation function	GELU

a pretext task on an unannotated database for pre-training to extract representation information. The pre-trained model is fine-tuned on the downstream task with a small amount of annotated data in a supervised manner. Compared with supervised learning, self-supervised learning can achieve competitive SOH estimation results on a small amount of annotated battery aging data and exhibit better generalization performance.

Transformer encoder. The input data of the network is a three-dimensional matrix, which is the inherent representation of the image, and the SOH estimation is similar to the image regression task in the computer vision field. Due to the significant performance of the Transformer [33], it has been emerging since the rise of ViT [31] in 2020. Experimental suggests that its performance is better than the state-of-the-art CNN module [30]. Therefore, ViT is used as the encoder in SSF-WL for feature extraction.

The encoder is shown in Fig. 4(b). First, the input image F is cropped into N patch tokens. Then, the patch tokens are concatenated with an additional learnable class token vector, which is the difference between the Transformer [33], as the input of the embedding layer for embedding and position encoding to obtain an $(N+1) \times C$ output matrix, where C is the embedding dimension. Further, the embedded matrix is forwarded into the encoder layer. The normalized matrix (layer normalization [34]) F_{n} is fed into the multi-head attention mechanism for matrix calculation to acquire the output matrix O as Eqs. (9)~(12), where $F_{\mathrm{n}} \in \mathbb{R}^{(N+1)\times C}$ is the normalized matrix, $W^Q, W^K, W^V \in \mathbb{R}^{C\times C}$ are the transformation matrices, $Q, K, V \in \mathbb{R}^{(N+1)\times C}$ are the Query, Key, and Value matrices in the Transformer, respectively, $W_h^Q, W_h^K, W_h^V \in \mathbb{R}^{C\times C_{\mathrm{h}}}$ are the multi-head transformation matrices, C_{h} is the dimension of each head, H is the number of heads, $C = H \times C_{\mathrm{h}}$, $h \in \{1, 2, \dots, H\}$, $\hat{A}_h \in \mathbb{R}^{(N+1)\times(N+1)}$ and $O_h \in \mathbb{R}^{(N+1)\times C_{\mathrm{h}}}$ are the attention matrix and the output of the h-th head, respectively, $d_k = C$ is the dimension of the K matrix, $W^O \in \mathbb{R}^{HC_{\mathrm{h}}\times C}$ is the output transformation matrix, and $O \in \mathbb{R}^{(N+1)\times C}$ is the output of the multi-head attention mechanism. Then, the output O is added to the embedded matrix, which is the first residual connection. The feature matrix after the second normalization is fed into the MLP layer consisting of two fully connected layers. After the second residual connection, the final output matrix of ViT is obtained. A complete encoder is acquired by stacking l encoder layers, and the activation function in this network is the Gaussian Error Linear Unit (GELU) [35]. Table 1 shows the default parameters of the encoder in this paper.

$$\begin{cases} Q = F_{\mathrm{n}} \times W^Q \\ K = F_{\mathrm{n}} \times W^K \\ V = F_{\mathrm{n}} \times W^V \end{cases} \tag{9}$$

$$\hat{A}_h = \mathrm{Softmax}\left(A_h\right) = \mathrm{Softmax}\left(\frac{QW_h^Q \times \left(KW_h^K\right)^T}{\sqrt{d_k}}\right) \tag{10}$$

$$O_h = \hat{A}_h \times VW_h^V \tag{11}$$

$$O = \mathrm{Concat}\left(O_1, O_2, \dots, O_h, \dots, O_H\right) W^O \tag{12}$$

Self-supervised learning. So far, some self-supervised methods have been proposed, including relatively general generative and contrastive learning [36–38], as well as learning strategies for specific downstream tasks, such as the probabilistic introspection-based self-supervised method for geometry-oriented tasks [39]. In general, the task compatibility of generative and contrastive learning is better than the task-specific self-supervised method [39], and generative learning is simpler and more effective than contrastive learning, so generative learning is used in this paper. The proposed self-supervised learning constructs the pretext task with masked image reconstruction (inspired by SimMIM [36]) and charging capacity estimation. The masked image reconstruction task is used to learn general representation features, while charging capacity estimation aims to guide the general features to be capacity-related, making our self-supervised pretext task more suitable for the SOH estimation task. Fig. 4(a) shows the overall strategy.

As described in Fig. 4(a), the input data is cropped into patch tokens and forwarded into the encoder. To construct the generative pretext task, the random masking method is applied to mask the input image with a ratio. Simply, the mask block, a square with a side length of mask size, is set to cover the patch token completely, and the input image is divisible by the mask block. In the pre-training, the masked patch tokens are fed into the encoder to extract the representation features. The output of the encoder is forwarded into the output layer, which consists of a 1×1 convolutional layer with a stride of 1 and a $C \times 1$ linear layer. The convolutional layer reconstructs the $N \times C$ matrix corresponding to the patch tokens to obtain the predicted patches, including masked and unmasked. The linear layer only reduces the dimension of the $1 \times C$ vector corresponding to the class toke to obtain the estimated charging capacity $\hat{C}_{\mathrm{c}}$. During pre-training, only the masked part of the reconstructed patches and the original input are used to calculate the loss. We refer to and follow the loss function setting in SimMIM [36] and use ℓ_1 loss to train the mask reconstruction task. Indeed, the ℓ_1 loss, which is based on absolute differences, tends to preserve the edges and details of the image and is more robust to outliers or extreme pixel values than MSE, which may be more suitable for the mask reconstruction task. Charging capacity estimation is a regression task where accurate estimation of extreme values or sudden changes is crucial. Therefore, we employ MSE loss, which is more sensitive to large errors due to the squared operation, to calculate the loss between the $\hat{C}_{\mathrm{c}}$ and C_{c}. The ℓ_1 loss and MSE loss can be obtained by Eqs. (13) and (14), where n is the number of samples, y_i is the true value, and $\hat{y}_i$ is the predicted value. Finally, the loss of pre-training is shown in Eq. (15), where λ is the weighting factor. It is worth mentioning that the pre-training does not require the labels of SOH. It just learns general representation information from input images.

For the downstream task of SOH estimation, the pre-trained model needs to be fine-tuned. Unlike pre-training, the input of the encoder is the raw patch tokens. Only the encoder and linear layer parameters are transferred, as the patches need not be reconstructed. The output of the linear layer is the estimated SOH value. The loss of fine-tuning can be acquired by Eq. (14). The default parameters of SSF-WL are shown in Table 1.

$$\ell_1 = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{13}$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 \tag{14}$$

$$\mathrm{loss}_{\mathrm{all}} = (1-\lambda)\,\ell_1 + \lambda \mathrm{MSE} \tag{15}$$

4. Results and discussion

4.1. Databases

The 124 commercial Li-ion phosphate (LFP)/graphite batteries are manufactured by the A123 System (APR18650M1A). The nominal capacity and voltage of all batteries are 1.1 Ah and 3.3 V, respectively. The charging of all batteries follows the "C1(Q1)-C2" fast-charging

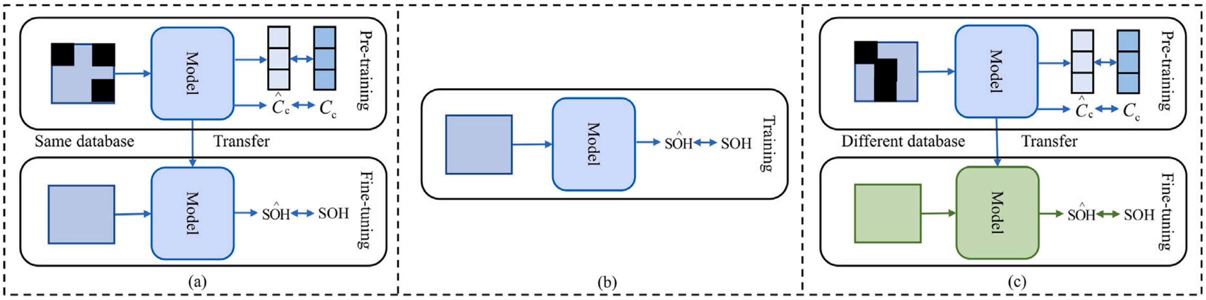


Fig. 5. The details of the three training strategies: (a) SSF-WL; (b) supervised learning; (c) transfer learning.

Table 2
The detailed charging information of the first 8 batteries.

Battery	C1	Q1	C2	Cycles
1	5C	67%	4C	1008
2	5.3C	54%	4C	1062
3	5.6C	19%	4.6C	1266
4	5.6C	36%	4.3C	1114
5	5.6C	19%	4.6C	1047
6	5.6C	36%	4.3C	827
7	3.7C	31%	5.9C	666
8	4.8C	80%	4.8C	1835

policy at constant temperature 30 °C. The C1 and C2 are the first and second CC steps, and the Q1 is the State-of-Charge (SOC). First, the batteries are charged at C1 until the SOC reaches Q1, and then the current switches to C2. When the SOC reaches 80%, the batteries switch to 1 C CC–CV charging. In fact, all batteries are charged under different conditions but discharged with a 4 C CC policy. The upper and lower cutoff voltages are 3.6 V and 2.2 V, respectively, and the cutoff current is C/20. All batteries cycle to 80% of nominal capacity. The first 8 batteries in "batch3" are employed in this work. The detailed charging information is shown in Table 2. Among them, batteries 4 and 8 are used as the test set, and the remaining are the training set. Notably, the charging policy of battery 8 is not included in the training set.

The Oxford database is published by the University of Oxford. It contains 8 small commercial Kokam pouch cells with a nominal capacity of 740 mAh. All cells are constructed with lithium cobalt oxide, lithium nickel cobalt oxide, and graphite. The cells are tested with 2 C CC charging and dynamic discharging under Artemis urban drive cycle at a temperature of 40 °C. After every 100 urban driving cycles, a characterization test with 1 C CC charging and discharging is performed to measure the maximum available capacity. There are only about 70 available cycle data per cell. Moreover, the cutoff voltage of all cells is 2.7 V. In this work, the current is set to 1 C for network input, cells 4 and 8 are extracted as the test set, and others are the training set.

Note that the batteries in both databases contain entirely different materials and charge–discharge policies. This positively contributes to verifying the generalization ability of the model. To distinguish the two databases, the batteries in the 124 commercial battery and Oxford databases are labeled as Batteries $1 \sim 8$ and Cells $1 \sim 8$, respectively.

4.2. Implementation details

All experiments are carried out on a device with GPU: 3 NVIDIA RTX 3090 (24G), memory: 64G, and CPU: Intel (R) Core (TM) i9-10920X. The AdamW [40], warmup, and cosine learning rate [41] are used to optimize the parameters. The Mean Absolute Error (MAE $= \ell_1$), Mean Square Error (MSE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) are introduced to evaluate the proposed method as Eqs. (13), (14), (16), and (17).

Table 3
The default training parameters for the 124 commercial battery and Oxford databases.

Hyperparameter	124 battery database		Oxford database	
	Pre-training	Fine-tuning	Pre-training	Fine-tuning
Epochs	100	500	200	500
Warmup epochs	5	2	10	2
Learning rate	2e−4	1.25e−4	2e−4	1.25e−3

To highlight the performance of the proposed SSF-WL, traditional supervised learning and transfer learning are introduced. Fig. 5 shows the details of the three training strategies. As described in the previous section, SSF-WL is first pre-trained on the unannotated training data according to the pretext task, and then the pre-trained model is fine-tuned using the annotated training data from the same database. Supervised learning, the most commonly employed training strategy among existing methods, solely utilizes annotated training data for training without any pre-training process. To assess the generalization ability of the proposed method, we fine-tune the pre-trained model of SSF-WL using a database distinct from that used for pre-training and refer to this training strategy as transfer learning in this paper. The model parameters for all training processes remain consistent, following the default settings in Table 1. The batch size is 128. Some main training parameters of SSF-WL are different on the two databases, as shown in Table 3. The training parameters of supervised learning for each database follow the fine-tuning column outlined in Table 3. For transfer learning, we determine the training parameters based on the database used for the specific task. For example, when the 124 commercial battery database is employed for pre-training, the pre-training parameters correspond to the pre-training column for the 124 commercial battery database in Table 3. The fine-tuning parameters correspond to the Oxford fine-tuning column.

The training sets of the 124 commercial battery and Oxford databases contain 5876 and 396 available cycle data, respectively, and the test sets have 2949 and 123 available cycle data, respectively. For each database, we first sort the available training data according to battery ID and cycle number. Then, three different amounts of data: the top 30%, 60%, and 100% of the sorted training data are chosen for relevant experimental analyses. To ensure the robustness and generalization ability of the model, these data are shuffled before being fed into the model. In addition, the complete and partial charging data are used in Sections 4.3 and 4.4, respectively, to verify the effectiveness and practicality of the proposed method for SOH estimation. The remaining experimental analyses are based on the complete charging data to analyze the influence of various factors on the performance of the model.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2} \tag{16}$$

$$\mathrm{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{17}$$

Table 4
Experimental results using 100% of the training data on the 124 commercial battery database.

Method	Battery	MAE (%)	MSE (%)	RMSE (%)	MAPE (%)
SSF-WL	Battery 4	0.4925	0.0048	0.6204	0.5376
	Battery 8	0.5431	0.0060	0.6419	0.5966
	All	**0.5241**	0.0055	0.6831	0.5743
Supervised	Battery 4	0.3465	0.0017	0.3902	0.3729
	Battery 8	0.6370	0.0055	0.6798	0.6917
	All	0.5273	**0.0041**	**0.5816**	**0.5713**
Transfer	Battery 4	0.2353	0.0011	0.2789	0.2570
	Battery 8	0.7144	0.0103	0.7543	0.7950
	All	0.5334	0.0068	0.6299	0.5918

Table 5
Experimental results using 60% of the training data on the 124 commercial battery database.

Method	Battery	MAE (%)	MSE (%)	RMSE (%)	MAPE (%)
SSF-WL	Battery 4	0.4205	0.0034	0.5233	0.4589
	Battery 8	0.6013	0.0072	0.7140	0.6596
	All	**0.5330**	**0.0058**	**0.6758**	**0.5838**
Supervised	Battery 4	0.4583	0.0031	0.5153	0.4977
	Battery 8	1.0927	0.0481	1.3071	1.2395
	All	0.8531	0.0311	1.0387	0.9593
Transfer	Battery 4	0.3232	0.0022	0.3908	0.3540
	Battery 8	1.3344	0.0555	1.4849	1.5023
	All	0.9524	0.0354	1.1158	1.0685

4.3. Estimation results of complete charging data

We conduct experimental analyses on SSF-WL, supervised learning, and transfer learning using different amounts of annotated training data to evaluate the performance of the proposed method on SOH estimation. The different amounts of annotated training data are only available for fine-tuning and supervised learning, and all pre-training processes use 100% unannotated training data.

4.3.1. 124 Commercial battery database

As shown in Tables 4–6, SSF-WL achieves exciting results on the 124 commercial battery database test set. As the training data decreases, the overall errors of SSF-WL on the test set only have minor changes. When using 30% of the training data, the MAE and RMSE are 0.5207% and 0.6585%, respectively. Compared with using 100% of the training data, they are reduced by 0.0034% and 0.0246%, respectively. In fact, SSF-WL utilizes the pretext task to learn general representation features, and the pre-trained model can be used for SOH estimation only by fine-tuning on a small amount of annotated data, thereby reducing the data dependency of the model. In contrast, the overall errors of supervised learning increase gradually. The MAE and RMSE reach 1.0426% and 1.2670%, respectively, when using 30% of the training data. Compared with using the full training data, they are increased by 0.5153% and 0.6854%, respectively. This is because the insufficient annotated data prevents the supervised training model from extracting rich and effective features, leading to a decline in model performance. Further, SSF-WL utilizes additional unannotated data for pre-training to enhance feature representation, which is lacking in supervised learning. This results in the superior performance of SSF-WL over supervised learning when a sufficient 100% annotated database is available. Transfer learning uses the 124 commercial battery database to fine-tune the pre-trained model on the Oxford database. It can be seen that transfer learning achieves similar errors compared to supervised learning. This may be attributed to the fact that the Oxford database has only 376 training data. As a result, the pre-trained model cannot sufficiently learn general representation information.

Fig. 6 illustrates the estimation results and errors of a single battery in the 124 commercial battery database. With the reduction of data, the estimation results of SSF-WL on both batteries are consistent, and the error fluctuations are slight. When using 30% of the data, both supervised learning and transfer learning exhibit significant estimation errors in the fitting curves on Battery 4, with even more pronounced errors exceeding 10% on Battery 8. In fact, the charging policy of Battery 8 is not included in the training set, so the above results provide a positive perspective that SSF-WL can still achieve superior estimation results for batteries with unknown charge–discharge conditions relying on a small amount of annotated training data.

Table 6
Experimental results using 30% of the training data on the 124 commercial battery database.

Method	Battery	MAE (%)	MSE (%)	RMSE (%)	MAPE (%)
SSF-WL	Battery 4	0.4627	0.0041	0.5625	0.5066
	Battery 8	0.5559	0.0071	0.6682	0.6153
	All	**0.5207**	**0.0060**	**0.6585**	**0.5742**
Supervised	Battery 4	0.5282	0.0078	0.6357	0.5906
	Battery 8	1.3548	0.0877	1.5784	1.5611
	All	1.0426	0.0575	1.2670	1.1945
Transfer	Battery 4	0.4884	0.0069	0.6388	0.5407
	Battery 8	1.5799	0.1006	1.7367	1.8147
	All	1.1676	0.0652	1.3813	1.3335

Table 7
Experimental results using 100% of the training data on the Oxford database.

Method	Cell	MAE (%)	MSE (%)	RMSE (%)	MAPE (%)
SSF-WL	Cell 4	0.6186	0.0057	0.7522	0.7071
	Cell 8	0.6019	0.0061	0.7781	0.7235
	All	**0.6083**	**0.0059**	**0.7631**	**0.7172**
Supervised	Cell 4	0.5902	0.0054	0.7333	0.6791
	Cell 8	0.6779	0.0081	0.8989	0.8131
	All	0.6444	0.0070	0.8349	0.7619
Transfer	Cell 4	0.5885	0.0066	0.8125	0.6828
	Cell 8	0.7011	0.0093	0.9668	0.8480
	All	0.6581	0.0083	0.9017	0.7849

Table 8
Experimental results using 60% of the training data on the Oxford database.

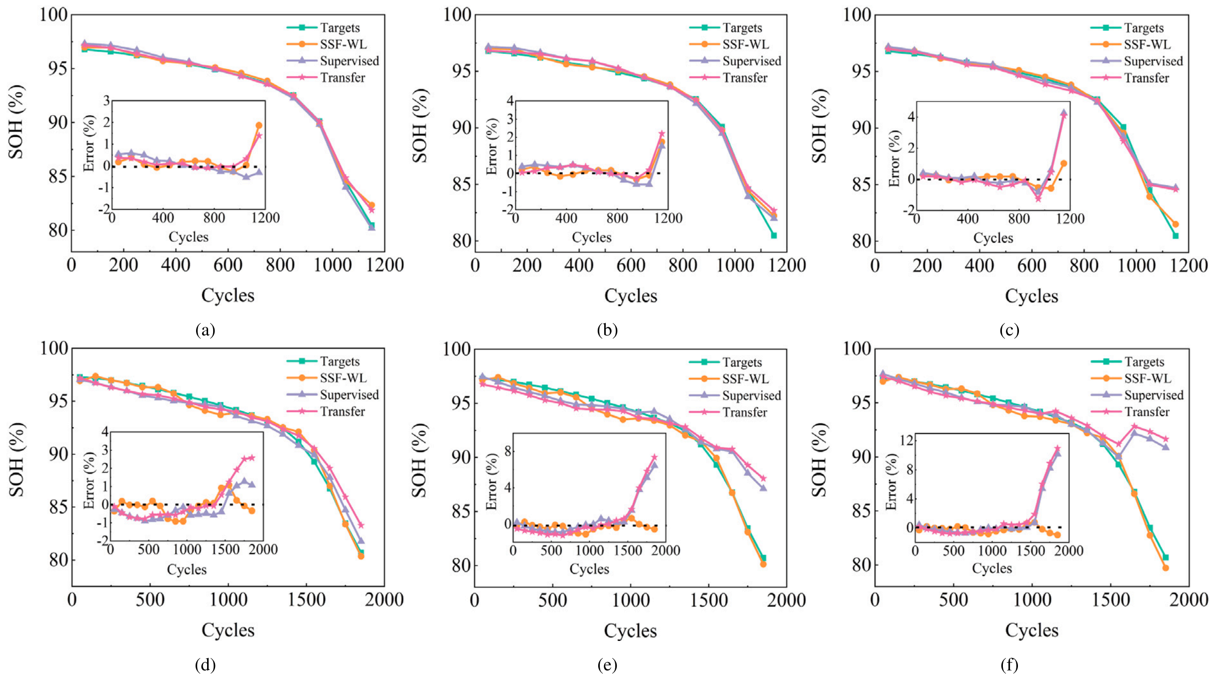
Method	Cell	MAE (%)	MSE (%)	RMSE (%)	MAPE (%)
SSF-WL	Cell 4	0.7687	0.0084	0.9176	0.8834
	Cell 8	0.6087	0.0065	0.8068	0.7305
	All	**0.6698**	**0.0072**	**0.8505**	**0.7889**
Supervised	Cell 4	1.1033	0.0206	1.4341	1.2856
	Cell 8	0.8474	0.0116	1.0761	1.0016
	All	0.9452	0.0150	1.2248	1.1101
Transfer	Cell 4	0.9901	0.0179	1.3386	1.1459
	Cell 8	0.8736	0.0127	1.1281	1.0261
	All	0.9181	0.0147	1.2078	1.0719

Table 9
Experimental results using 30% of the training data on the Oxford database.

Method	Cell	MAE (%)	MSE (%)	RMSE (%)	MAPE (%)
SSF-WL	Cell 4	1.0992	0.0167	1.2931	1.2817
	Cell 8	1.0967	0.0200	1.4141	1.3082
	All	1.0976	0.0187	1.3685	1.2981
Supervised	Cell 4	1.3093	0.0280	1.6748	1.5135
	Cell 8	1.7689	0.0425	2.0620	2.0734
	All	1.5933	0.0370	1.9227	1.8595
Transfer	Cell 4	1.0998	0.0214	1.4639	1.2583
	Cell 8	1.0236	0.0148	1.2163	1.2117
	All	**1.0527**	**0.0173**	**1.3153**	**1.2295**

4.3.2. Oxford database

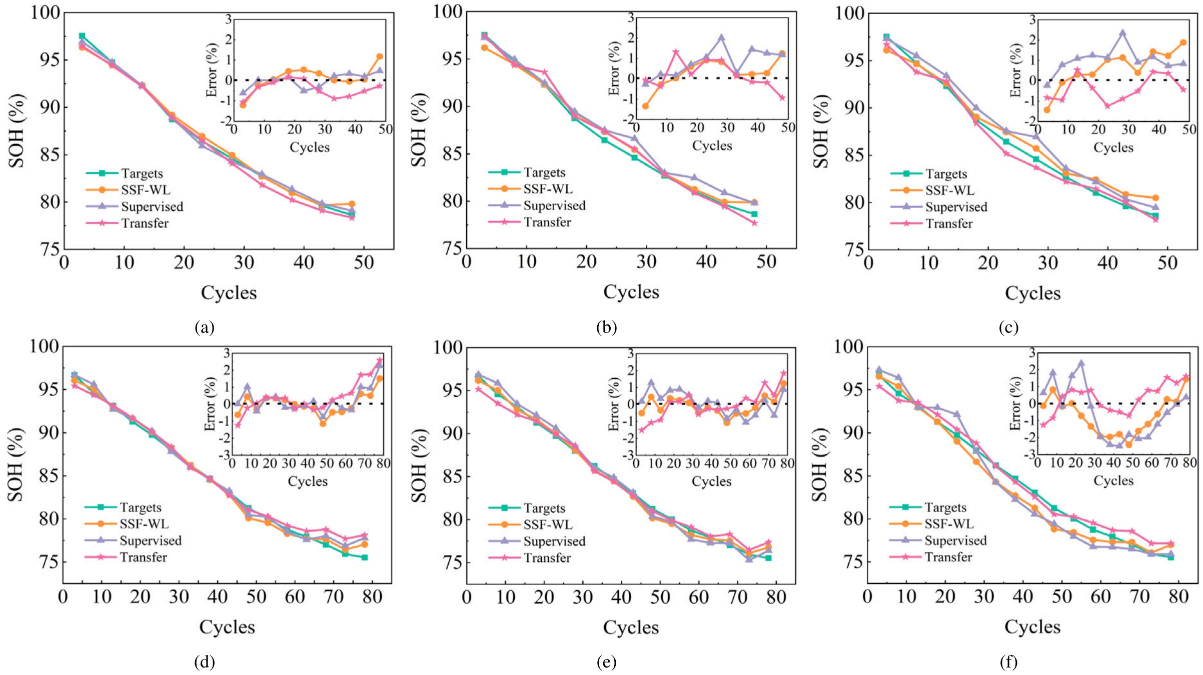
The error results on the Oxford database are shown in Tables 7–9. Due to the lack of training data, all the overall errors of the three training strategies increase with data reduction. However, SSF-WL still



(a) (b) (c)

(d) (e) (f)

Fig. 6. SOH estimation results and errors of Battery 4 and Battery 8 in the 124 commercial battery database: (a), (b), and (c) are Battery 4; (d), (e), and (f) are Battery 8. The first, second, and third columns are the estimated results using 100%, 60%, and 30% of the training data, respectively.



(a) (b) (c)

(d) (e) (f)

Fig. 7. SOH estimation results and errors of Cell 4 and Cell 8 in the Oxford database: (a), (b), and (c) are Cell 4; (d), (e), and (f) are Cell 8. The first, second, and third columns are the estimated results using 100%, 60%, and 30% of the training data, respectively.

maintains the lowest overall errors on the test set compared to supervised learning. When the training data is 30%, the MAE and RMSE are obtained by SSF-WL and supervised learning of 1.0976%/1.3685% and 1.5933%/1.9227%, respectively. Compared to 100% training data, they are increased by 0.4893%/0.6054% and 0.9489%/1.0878%, respectively. This further validates the effectiveness of SSF-WL in reducing data dependencies. The effect of transfer learning is demonstrated in the Oxford database. It achieves the optimal error results when using 30% of the training data with MAE and RMSE of 1.0527% and 1.3153%, respectively. This is because the pre-training is performed on a larger 124 commercial battery database containing 5876 training data. Obviously, pre-training on a large amount of data enables the model to learn better representation features, even if the data is unannotated.

The estimation results and errors of a single cell in the Oxford database are indicated in Fig. 7. The data distributions of the cells in the test and training sets are consistent, resulting in a similar estimation trend of the three methods on both cells. For Cell 4, the estimation error in the fitting curve of supervised learning is more significant as the data decreases. In contrast, SSF-WL consistently exhibits a perfect

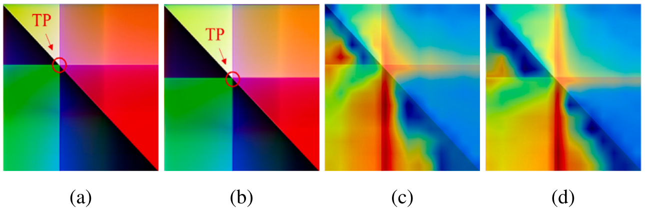


Fig. 8. Feature heatmaps of Battery 8: (a) and (b) are the model input images of the 1st and 1000th cycles, where TP is the transition point; (c) and (d) are the feature heatmaps corresponding to (a) and (b).

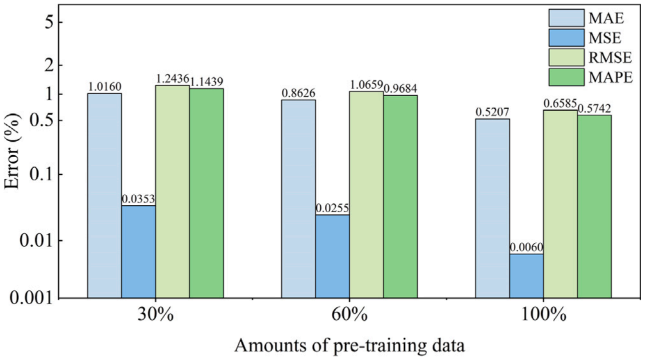


Fig. 9. Experimental results of different amounts of pre-training data.

fitting curve and smaller estimation error. For Cell 8, the fitting curves of the three methods are not significantly different when the training data is more than 60%, but the estimation error curves of SSF-WL are closer to the zero line. Moreover, transfer learning has the best fitting curve when the training data drops to 30%.

The results on both databases show that the performance of SSF-WL is more outstanding when the unannotated data for pre-training is sufficient. Compared with supervised learning, SSF-WL can achieve satisfactory results despite the lack of annotated training data. Transfer learning further proves that the SSF-WL pre-trained model on a large unannotated database can be transferred to batteries with different charging conditions or different materials for SOH estimation with a small amount of annotated data and achieve on-par or better results than supervised learning.

4.4. Estimation results of partial charging data

To verify the performance of SSF-WL on more practical partial charging data, we intercept voltage, current, and temperature data within the 60%~90% SOC range from each charging cycle as the model input. Due to the reduction in charging time as the SOH decreases, the charging capacity corresponding to the partial charging data with the same SOC interval in different cycles still strongly correlates with the SOH. The C_c calculated by Eq. (2) is the charging capacity corresponding to the partial charging data. On the 124 commercial battery database, 100% unannotated training data is used for pre-training, and different proportions of annotated training data are applied for fine-tuning and supervised learning. As shown in Table 10, when the training data is 100%, SSF-WL and supervised learning obtain nearly identical results. However, the performance of supervised learning drops significantly as the training data decreases. When using 30% of the training data, the MAE and RMSE obtained by supervised learning are 1.1751% and 1.4232%, respectively. Under the premise of ensuring that the information contained in partial charging data is strongly related to SOH, SSF-WL does not directly depend on whether the input data is complete or partial, so SSF-WL is still effective on the partial charging data, which has practical significance.

Table 10
Experimental results of partial charging data.

Amount of data	Method	MAE (%)	MSE (%)	RMSE (%)	MAPE (%)
100%	SSF-WL	0.4496	0.0071	0.6028	0.5081
	Supervised	0.4427	0.0095	0.6556	0.5055
60%	SSF-WL	0.5042	0.0087	0.6764	0.5684
	Supervised	0.7086	0.0152	0.9013	0.8006
30%	SSF-WL	0.7151	0.0111	0.8737	0.7897
	Supervised	1.1751	0.0295	1.4232	1.3016

4.5. Interpretability of the model

To provide some interpretation of the features extracted by the encoder, we extract the output of the last encoder layer of SSF-WL, which is pre-trained and fine-tuned using 100% unannotated and annotated training data, respectively. Fig. 8 presents the feature heatmaps of Battery 8 at the 1st and 1000th cycles. As described in Fig. 1, the CC–CV charging time gradually decreases as the battery ages. This key feature is manifested in the input image as a shift of the charging strategy transition point (from C1(Q1)-C2 charging to CC–CV charging). As shown in Fig. 8, the features extracted by the model consistently focus on the vicinity of the data row and column corresponding to the transition point as the battery ages. Obviously, these features conform to our cognition. Further, the model pays more attention to the differential information, possibly because such information directly reflects the state changes at different time points during the charging process, enabling the model to capture these key features more sensitively. In summary, the model can effectively extract features to accurately estimate SOH and has certain interpretability.

4.6. Effect of pre-training data amount

In the previous section, we discovered that the amount of unannotated pre-training data significantly affects the performance of the model. Hence, 30%, 60%, and 100% unannotated training data are selected to analyze the relationship between the amount of pre-training data and model performance in SSF-WL. All experiments are fine-tuned on 30% of the annotated data to highlight performance variations. The results on the 124 commercial battery database are shown in Fig. 9. The model performance is significantly improved by increasing the pre-training data. When only 30% of the pre-training data is used, the MAE and RMSE are 1.0160% and 1.2436%, respectively. They are increased by 0.4953% and 0.5851% compared to using 100%. In addition, the performance difference between 60%~100% is higher than 30%~60%. This may be because a larger amount of unannotated data can contribute to the model extracting more prosperous and diverse representation features, leading to significant performance gains.

4.7. Effect of the main parameters

We carry out several experiments on the 124 commercial battery database to verify the influence of the main parameters on SSF-WL, including the encoder layer, embedding dimension, mask ratio, mask size, and λ. All experiments are pre-trained and fine-tuned on 100% unannotated and annotated training data, respectively.

The errors of different encoder layers on the test set are indicated in Table 11. As the number of layers increases, the four metrics have the same change trend. When the layer is 5, SSF-WL reaches the lowest errors, and the parameter number is 9.29 M. Fig. 10 shows the results of different embedding dimensions. It can be seen that the model with an embedding dimension of 384 has excellent performance. When the embedding dimensions are 96 and 768, the model obtains similar higher errors. Moreover, the embedding dimension of 768 significantly raises the parameter number to 36.23 M. In fact, the number of encoder layers and the embedding dimension have always been two crucial

Table 11
Experimental results of different encoder layers.

Encoder layers	MAE (%)	MSE (%)	RMSE (%)	MAPE (%)	Params (M)
1	0.9780	0.0155	1.0760	1.0568	2.16
3	0.7940	0.0150	1.0138	0.8755	5.72
5	0.5241	0.0055	0.6831	0.5743	9.29
7	1.4247	0.0354	1.5929	1.5370	12.86
9	1.4682	0.0398	1.7168	1.5781	16.43

Table 12
Experimental results of different mask sizes and ratios.

Mask size	Mask ratio	MAE (%)	MSE (%)	RMSE (%)	MAPE (%)
16	0.25	0.6044	0.0074	0.7572	0.6648
	0.50	0.7474	0.0108	0.8771	0.8235
	0.75	0.8264	0.0122	1.0245	0.8968
32	0.25	0.6087	0.0063	0.7355	0.6631
	0.50	0.5241	0.0055	0.6831	0.5743
	0.75	0.7559	0.0096	0.9031	0.8217

parameters in deep learning, determining the parameter number in the network. Too many parameters may cause over-fitting, while too few parameters may cause under-fitting. Therefore, the five-layer ViT structure with the embedding dimension of 384 is most suitable for the SOH estimation in this paper.

Table 12 illustrates the influence of mask size and ratio. Since the mask block needs to cover the image patch, and the input image is divisible by the mask block, we only choose mask sizes 16 and 32 for analyses. Overall, the errors at the smaller mask size of 16 are not as lower as that at a larger mask size of 32, which is also proved in [36]. Nevertheless, further increasing the mask radio is observed with increased errors when the mask size is 16, probably because the upper limit of the prediction is already reached at the mask ratio of 0.25. When the mask size is 32, the optimal results are obtained with a mask ratio of 0.5. Mask size and rate directly influence the difficulty of the pretext task. If the mask size is too small or the mask rate is too low, the pretext task may become relatively simple. Conversely, the pretext task may become excessively complex, making it challenging for the model to learn and solve the task, thereby impacting overall performance. In this paper, the mask size and rate of 32 and 0.5 potentially achieve a balance between task difficulty and learning effectiveness.

Different λ values are selected to analyze and evaluate the weight allocation in the pretext tasks. As shown in Fig. 11, when the λ is 0, i.e., no weak labels are introduced, the errors obtained by the model are the largest. Increasing the λ from 0 to 0.8 is observed with significantly decreased errors. The optimal results are obtained with the λ of 0.8. When pre-training with only weak labels, i.e., the mask policy is not introduced, the errors increase again. Weak labels play a guiding role in pre-training, making general representation features correlated with capacity. Further, the charging capacity prediction task is closely related to the SOH estimation, so favoring the model to address this pretext task is advantageous for downstream SOH estimation. Therefore, λ of 0.8 may be the optimal choice.

4.8. Effect of data processing

We conduct experimental analyses of the data processing method to evaluate its effectiveness. When the data processing method is not used, the input is the normalized matrix with the dimension 224 × 3 (multiple sequences), as shown in Fig. 3, which may not be suitable for ViT. Therefore, the Transformer [33], specially designed for sequence tasks, is introduced in SSF-WL to replace ViT, and other parameters are unchanged. The final output of the Transformer is used to reconstruct the masked sequences. A global average pooling layer and a linear layer are combined to obtain estimated labels. This substitution is fair because ViT maintains the fundamental structure of the Transformer, and the proposed data processing method can be regarded as an innovative strategy for transforming the task from sequence to image. The model is pre-trained and fine-tuned on 100% unannotated and annotated training data, and the results are shown in Table 13. On the 124 commercial battery database, the MAE and RMSE obtained by the Transformer are 0.9758% and 1.0702% when the input data are raw sequences. Compared to the proposed method, they are increased by 0.4517% and 0.3871%. On the Oxford database, the Transformer even achieves the MAE of 4.2135%. Indeed, the proposed data processing method ensures that the input data incorporates time correlations, differential information, and original features, which is crucial in enhancing model performance.

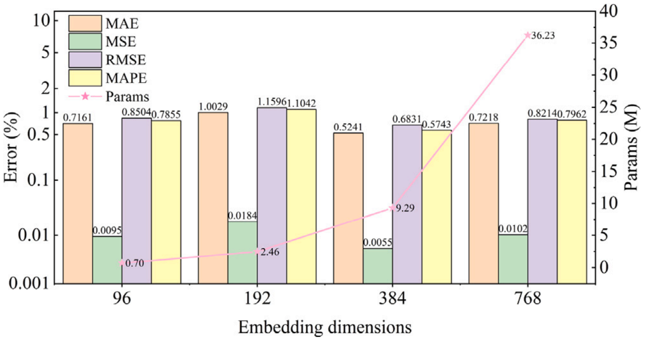


Fig. 10. Experimental results of different embedding dimensions.

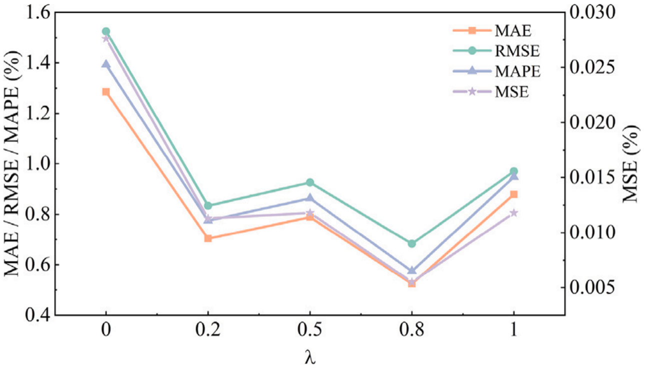


Fig. 11. Experimental results of different λ.

4.9. Comparison with other encoders

To verify the performance of the constructed five-layer ViT encoder, several classical models are used for comparison. All models follow the training strategy of SSF-WL, only replacing the "Encoder". The output of the last convolutional layer of the CNN models is used for masked image reconstruction, while the final output of the Long Short-Term Memory (LSTM) [42] is used to reconstruct the masked sequences. For all models, estimated labels are obtained by a global average pooling layer and a linear layer. The experimental results of pre-training and fine-tuning on 100% unannotated and annotated training data are shown in Table 13. The parameter number and Floating Point Operations (FLOPs) of the five-layer encoder constructed in this paper are only 9.29 M and 1.36 G, which are much lower than CNN-based models. Further, its performance is better than ResNet18 [30] and VGG16 [43] on the two databases. On the 124 commercial battery database, VGG16 achieves sub-optimal estimation errors, and the MAE is increased by 0.1447% compared to ours. On the Oxford database, the errors of VGG16 increase significantly, probably due to the simplicity of the model and the lack of training data. In addition, LSTM has the

Table 13
Experimental results of the data processing method and different encoders.

Method	Type	Params (M)	FLOPs (G)	124 commercial battery database				Oxford database			
				MAE (%)	MSE (%)	RMSE (%)	MAPE (%)	MAE (%)	MSE (%)	RMSE (%)	MAPE (%)
Transformer [33]	Sequence	8.96	1.32	0.9758	0.0355	1.0702	1.1102	4.2135	0.3104	5.3827	4.9859
LSTM [42]	Sequence	5.33	1.20	2.7639	0.1601	2.7796	3.0817	5.1909	0.3648	5.9458	5.9443
VGG16 [43]	Image	14.72	15.35	0.6688	0.0060	0.7340	0.7165	5.5831	0.4251	6.5192	6.4645
ResNet18 [30]	Image	11.18	6.93	0.7115	0.0071	0.7851	0.7629	0.6558	0.0067	0.8132	0.7687
Ours	Image	9.29	1.36	**0.5241**	**0.0055**	**0.6831**	**0.5743**	**0.6083**	**0.0059**	**0.7631**	**0.7172**

largest errors on both databases. The encoder is the feature extraction component within the entire framework, and its performance directly affects the estimation result of SOH. The strength of ViT is that it can effectively extract global information, which is lacking in CNN or LSTM models. Generally, the experimental results show that the constructed five-layer ViT can effectively improve the performance of the proposed method.

5. Conclusion

This paper proposed a novel method for Li-ion battery SOH estimation. The main contributions of this paper include (1) proposing an effective data processing method to enhance features and enrich information content, (2) constructing a five-layer encoder structure to improve algorithm performance fundamentally, (3) developing the SSF-WL self-supervised framework to reduce the model's data dependency, and (4) ensuring the model's generalization ability across different types of batteries. Experiments suggest that the proposed data processing method can effectively improve the estimation performance, and the encoder outperforms classical models. Further, the estimation errors obtained by SSF-WL are significantly lower than that of supervised learning on only 30% of the annotated training data. With sufficient unannotated training data, SSF-WL can achieve performance transfer between different types of batteries based on a small amount of annotated data. These prove the feasibility of the proposed method in reducing the data dependency and improving the generalization ability of the model. Furthermore, SSF-WL consistently achieves satisfactory fitting curves and lower error fluctuations as the available annotated data decreases, demonstrating the accuracy of SSF-WL on SOH estimation. Moreover, competitive experimental results based on partial charging data exhibit that SSF-WL is not reliant on whether the input data is partial or complete, indicating its strong applicability and practicality.

In fact, the performance of SSF-WL is influenced to some extent by the amount of unannotated data, i.e., larger data amount leads to better performance improvements. This phenomenon is widely observed in existing self-supervised methods. In addition, the collection of the annotated database is still cumbersome and error-prone, even though SSF-WL requires less annotated data. Therefore, in the future, we will focus on reducing the amount of unannotated data while maintaining model performance and consider combining weakly supervised or unsupervised methods to achieve effective and accurate Li-ion battery SOH estimation.

CRediT authorship contribution statement

Tianyu Wang: Writing – original draft, Software, Methodology, Conceptualization. **Zhongjing Ma:** Writing – review & editing, Supervision, Project administration, Funding acquisition. **Suli Zou:** Writing – review & editing, Funding acquisition, Formal analysis. **Zhan Chen:** Methodology, Investigation. **Peng Wang:** Visualization, Data curation.

Declaration of competing interest

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

Data availability

The data used in the paper is publicly available, and we have shared the link.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62003037 and Grant U22A2048; in part by the Science and Technology Foundation of Key Laboratory of Space Intelligent Control under Grant HTKJ2020KL502009.

References

[1] Pradhan SK, Chakraborty B. Battery management strategies: An essential review for battery state of health monitoring techniques. J Energy Storage 2022;51:104427.
[2] Jiang S, Song Z. A review on the state of health estimation methods of lead-acid batteries. J Power Sources 2022;517:230710.
[3] Sui X, He S, Vilsen SB, Meng J, Teodorescu R, Stroe D-I. A review of non-probabilistic machine learning-based state of health estimation techniques for Lithium-ion battery. Appl Energy 2021;300:117346.
[4] Lee J, Won J. Enhanced Coulomb counting method for SOC and SOH estimation based on Coulombic efficiency. IEEE Access 2023;11:15449–59.
[5] Pillai P, Sundaresan S, Kumar P, Pattipati KR, Balasingam B. Open-circuit voltage models for battery management systems: A review. Energies 2022;15(18):6803.
[6] Li Y, Xiong B, Vilathgamuwa DM, Wei Z, Xie C, Zou C. Constrained ensemble Kalman filter for distributed electrochemical state estimation of lithium-ion batteries. IEEE Trans Ind Inf 2020;17(1):240–50.
[7] Amir S, Gulzar M, Tarar MO, Naqvi IH, Zaffar NA, Pecht MG. Dynamic equivalent circuit model to estimate state-of-health of lithium-ion batteries. IEEE Access 2022;10:18279–88.
[8] Xiong R, Li L, Li Z, Yu Q, Mu H. An electrochemical model based degradation state identification method of Lithium-ion battery for all-climate electric vehicles application. Appl Energy 2018;219:264–75.
[9] Li J, Adewuyi K, Lotfi N, Landers RG, Park J. A single particle model with chemical/mechanical degradation physics for lithium ion battery State of Health (SOH) estimation. Appl Energy 2018;212:1178–90.
[10] Hosseininasab S, Lin C, Pischinger S, Stapelbroek M, Vagnoni G. State-of-health estimation of lithium-ion batteries for electrified vehicles using a reduced-order electrochemical model. J Energy Storage 2022;52:104684.
[11] Zheng Y, Qin C, Lai X, Han X, Xie Y. A novel capacity estimation method for lithium-ion batteries using fusion estimation of charging curve sections and discrete Arrhenius aging model. Appl Energy 2019;251:113327.
[12] Li X, Yuan C, Li X, Wang Z. State of health estimation for Li-Ion battery using incremental capacity analysis and Gaussian process regression. Energy 2020;190:116467.
[13] Gong D, Gao Y, Kou Y, Wang Y. State of health estimation for lithium-ion battery based on energy features. Energy 2022;257:124812.
[14] Feng X, Weng C, He X, Han X, Lu L, Ren D, Ouyang M. Online state-of-health estimation for Li-ion battery using partial charging segment based on support vector machine. IEEE Trans Veh Technol 2019;68(9):8583–92.
[15] Li Q, Li D, Zhao K, Wang L, Wang K. State of health estimation of lithium-ion battery based on improved ant lion optimization and support vector regression. J Energy Storage 2022;50:104215.
[16] Driscoll L, de la Torre S, Gomez-Ruiz JA. Feature-based lithium-ion battery state of health estimation with artificial neural networks. J Energy Storage 2022;50:104584.
[17] Fan Y, Xiao F, Li C, Yang G, Tang X. A novel deep learning framework for state of health estimation of lithium-ion battery. J Energy Storage 2020;32:101741.
[18] Bao Z, Jiang J, Zhu C, Gao M. A new hybrid neural network method for state-of-health estimation of lithium-ion battery. Energies 2022;15(12):4399.
[19] Gong Q, Wang P, Cheng Z. An encoder-decoder model based on deep learning for state of health estimation of lithium-ion battery. J Energy Storage 2022;46:103804.

[20] Bockrath S, Lorentz V, Pruckner M. State of health estimation of lithium-ion batteries with a temporal convolutional neural network using partial load profiles. Appl Energy 2023;329:120307.

[21] Zhang H, Gao J, Kang L, Zhang Y, Wang L, Wang K. State of health estimation of lithium-ion batteries based on modified flower pollination algorithm-temporal convolutional network. Energy 2023;283:128742.

[22] Xiong R, Sun Y, Wang C, Tian J, Chen X, Li H, Zhang Q. A data-driven method for extracting aging features to accurately predict the battery health. Energy Storage Mater 2023;57:460–70.

[23] 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.

[24] Ma G, Xu S, Yang T, Du Z, Zhu L, Ding H, Yuan Y. A transfer learning-based method for personalized state of health estimation of lithium-ion batteries. IEEE Trans Neural Netw Learn Syst 2022;1–11.

[25] Lu J, Xiong R, Tian J, Wang C, Sun F. Deep learning to estimate lithium-ion battery state of health without additional degradation experiments. Nature Commun 2023;14(1):2760.

[26] Wang Z, Oates T. Imaging time-series to improve classification and imputation. 2015, arXiv preprint arXiv:1506.00327.

[27] Chang C, Wang Q, Jiang J, Wu T. Lithium-ion battery state of health estimation using the incremental capacity and wavelet neural networks with genetic algorithm. J Energy Storage 2021;38:102570.

[28] Severson KA, Attia PM, Jin N, Perkins N, Jiang B, Yang Z, Chen MH, Aykol M, Herring PK, Fraggedakis D, et al. Data-driven prediction of battery cycle life before capacity degradation. Nat Energy 2019;4(5):383–91.

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

[30] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, p. 770–8.

[31] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, et al. An image is worth 16x16 words: Transformers for image recognition at scale. 2020, arXiv preprint arXiv:2010.11929.

[32] Lin M, Wu J, Meng J, Wang W, Wu J. Screening of retired batteries with Gramian angular difference fields and ConvNeXt. Eng Appl Artif Intell 2023;123:106397.

[33] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. Adv Neural Inf Process Syst 2017;30.

[34] Ba JL, Kiros JR, Hinton GE. Layer normalization. 2016, arXiv preprint arXiv:1607.06450.

[35] Hendrycks D, Gimpel K. Gaussian Error Linear Units (GELUs). 2023, arXiv preprint arXiv:1606.08415.

[36] Xie Z, Zhang Z, Cao Y, Lin Y, Bao J, Yao Z, Dai Q, Hu H. Simmim: A simple framework for masked image modeling. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022, p. 9653–63.

[37] He K, Fan H, Wu Y, Xie S, Girshick R. Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020, p. 9729–38.

[38] Chen T, Kornblith S, Norouzi M, Hinton G. A simple framework for contrastive learning of visual representations. In: International conference on machine learning. PMLR; 2020, p. 1597–607.

[39] Novotny D, Albanie S, Larlus D, Vedaldi A. Self-supervised learning of geometrically stable features through probabilistic introspection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, p. 3637–45.

[40] Loshchilov I, Hutter F. Decoupled weight decay regularization. 2019, arXiv preprint arXiv:1711.05101.

[41] Loshchilov I, Hutter F. Sgdr: Stochastic gradient descent with warm restarts. 2017, arXiv preprint arXiv:1608.03983.

[42] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9(8):1735–80.

[43] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 2015, arXiv preprint arXiv:1506.00327.