Cross-Domain State-of-Charge Estimation of Li-Ion Batteries Based on Source Domain Selection and Attention Mechanism

Wenliang Zhang, Jingwei Hu, Bing Lin, Dui Liu, Mingfen Wang, Delong Mu, Chenye Lin, and Yu Lu

1College of Physics and Energy, Fujian Normal University, Fuzhou 350117, China
2Department of Software Engineering, Xiamen University, Xiamen 361005, China
3Concord University College, Fujian Normal University, Fuzhou 350117, China

Correspondence should be addressed to Yu Lu; fzluyu@163.com

Received 22 August 2023; Revised 19 October 2023; Accepted 22 November 2023; Published 3 January 2024

Copyright © 2024 Wenliang Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the domain of lithium-ion (Li-ion) battery state-of-charge (SOC) estimation, deep neural network models commonly assume a congruent distribution between training and testing data. Nonetheless, this assumption often proves inadequate in real-world scenarios, due to variations in environmental temperature, aging levels, and operational conditions. To tackle this challenge, this study proposes a novel approach centered around a deep transfer network, incorporating source domain selection and an attention mechanism, for the task of cross-domain SOC estimation. This approach leverages a transfer network to extract bidirectional temporal features and accentuate salient information within sequences. The selection of an appropriate source domain for pretraining is contingent upon establishing domain similarity between the source and target domains. Experimental results demonstrate that the proposed method excels at feature extraction from Libs sequence data, yielding enhanced performance even when confronted with limited data.

1. Introduction

As the consumption of oil and the environmental impact of greenhouse gases continue to grow, countries are increasingly focusing on the development of electric vehicles (EVs) [1]. Over the past few decades, lithium-ion batteries have undergone significant technological advancements and rapid development. They exhibit advantages such as high energy density, long lifespan, and low maintenance costs, making them widely applicable in various fields, including electric vehicles and energy storage solutions [2–4]. A crucial element in the practical use of EVs is the Li-ion battery management system (BMS) [5]. Within the BMS, one of the most important functions is the estimation of the state-of-charge (SOC) of the Libs, as this reflects the Libs performance. Accurate SOC estimation is critical to protect the Libs from overcharging or discharging, extend its lifespan, and enable the application of smart control strategies to a vehicle [6]. Therefore, developing accurate SOC estimation methods is essential for the widespread adoption of EVs.

In recent decades, the estimation of SOC has become a widely researched topic, leading to the development of numerous accurate methods for SOC estimation. These methods can be primarily classified into four categories: the open-circuit voltage methods [7], the Coulomb counting methods [8], the model-based methods [9, 10], and the data-driven methods [11–13]. The open-circuit voltage method estimates the SOC of a Libs by utilizing the relationship between the open-circuit voltage and the SOC. The Coulomb counting method calculates the SOC by integrating the charge and discharge current of the Libs. The model-based method establishes a mathematical model based on the physical and chemical characteristics of the Libs and estimates the Libs state based on the measured data. These methods usually require a deep understanding of the
physical and chemical properties of the Libs, have high model complexity, and suffer from low accuracy.

The data-driven approach uses the Libs current, voltage, and other data as features and learns the nonlinear relationships between these features and the Libs SOC without the need for Libs domain and mathematical modeling knowledge. Deep learning's rapid development in recent years has generated a great deal of interest in its use in SOC estimation [14]. There has been growing interest in using deep learning techniques for SOC estimation, with several methods based on neural networks [15], convolutional neural network (CNN) [16], gated recurrent unit (GRU) [17, 18], and long short-term memory (LSTM) [19, 20] networks being proposed.

While deep learning methods have shown promise for SOC estimation, their generalization ability is limited by the difficulty of obtaining Libs data and the variability of real-world Libs utilization environments. To address these issues, transfer learning has been proposed as a solution. Transfer learning involves leveraging knowledge and experience from one or more related source domains to improve the performance and generalization ability of a target domain. Unlike traditional machine learning approaches that learn the target task from scratch, transfer learning uses existing knowledge to accelerate the learning process and improve accuracy [21].

Several studies have investigated the use of transfer learning for SOC estimation. For example, Tian et al. [22] incorporate a deep neural network (DNN) into a Kalman filter and fine-tune (FT) different layers of the pretrained network. Bhattacharjee et al. [23] proposed a one-dimensional CNN-based SOC estimation algorithm that leverages transfer learning. Wang et al. [24] proposed an improved transfer learning method for SOC estimation based on GRUs that is specifically designed for small target sets. Qin et al. [25] developed a novel SOC estimation method that exploits the temporal dynamics of measurements and transfers consistent estimation ability across different temperatures.

Domain adaptation is a branch of transfer learning, which aims to improve the model’s ability to generalize the target domain in the target domain by learning the differences between the source domain and the target domain. Oyewole et al. [26] propose a controllable deep transfer learning (CDTL) network for short- and long-term SOC estimations at early stages of degradation using controllable multiple domain adaptation (MDA). Bian et al. [27] proposed a deep transfer neural network (DTNN) with multiscale distribution adaptation for cross-domain SOC estimation. Shen et al. [28] proposed a temperature adaptive transfer network (TATN), which used a bidirectional long short-term memory and a 2-D convolutional neural network to extract temporal features and used adversarial adaptation and maximum mean discrepancy to reduce domain divergence.

Transfer learning has three main issues: when to transfer, where to transfer, and how to transfer. Current research on SOC estimation based on transfer learning often overlooks the selection of a suitable source domain and the filtering/classification of pretraining data, which can lead to suboptimal pretrained models and longer domain adaptation times during model training.

In response to the aforementioned challenges, this study introduces a novel approach involving both source domain selection and an attention mechanism to enhance the SOC estimation for Libs. The proposed methodology comprises three key components. Firstly, we establish an integrated network architecture consisting of a bidirectional long short-term memory (BiLSTM) and an attention mechanism. This architecture serves to extract bidirectional time-varying features from Libs data recorded under diverse operating conditions. By incorporating the attention mechanism, the model is capable of discerning critical information while filtering out irrelevant data, resulting in an enhanced ability to capture essential Libs behavior. Secondly, we select a source domain exhibiting a high degree of similarity with the target domain to act as the pretraining source domain. This selection process is driven by viewing distinct working conditions as source domains, each characterized by its own probability distribution. By identifying and utilizing the working condition that bears the closest resemblance to the target domain, we capitalize on the inherent similarities and optimize the transfer process. To address distributional disparities across various scales between the source and target domains, we employ the maximum mean discrepancy (MMD) method to calibrate the domain-shared features. The main contributions of this article are summarized as follows.

1. We propose a novel estimation model for Libs SOC by integrating an attention mechanism and BiLSTM. The architecture not only empowers the model to prioritize vital information while disregarding extraneous details but also facilitates concurrent learning from the past and future Libs data.

2. The selection of the pretraining source domain is informed by the identification of the working condition most akin to the target domain. This strategic choice results in improved estimation accuracy and faster network convergence by capitalizing on the intrinsic similarities between source and target domains.

3. We conduct a comprehensive comparison of various transfer learning techniques and propose a deep transfer network-based approach for Libs SOC estimation.

4. Experimental results demonstrate the superiority of source domain selection over random selection or employing all source domains, as it leads to enhanced estimation accuracy and accelerated network convergence.

The rest of this article is organized as follows. Section 2 presents the proposed method. Section 3 explains the dataset description and implementation details. Section 4 gives the experiment results and analysis. Finally, Section 5 concludes this article.

2. Proposed Method

2.1. Overview of the Proposed Method. To improve the learning ability and reduce the possibility of negative transfer in
Libs cross-domain SOC estimation, a transfer network based on source domain selection and attention mechanism is proposed in this article, as shown in Figure 1. The network structure comprises BiLSTM, attention, and fully connected (FC) layers, which are pretrained on selected data from the source domain. During the transfer stage, we compute the distribution difference between the source and target domains and perform domain adaptation to achieve feature alignment and optimize the network parameters of the SOC estimator.

To preprocess the Libs data, the sliding window method was used to convert the original two-dimensional data into three-dimensional sequences. The length of the data sequence is $T$, the length of the time window is $W$, and the data used to train the network is $\{x_1, y_1\} \cdots \{x_N, y_N\}$, where $N = T - W - 1$; and $x_i = \Phi_i, \cdots \Phi_i, \cdots \Phi_i$. $\Phi_i$ represents the input sequence of the $i$th sample, where $\Phi_i = (V_i, I_i)$ denotes the measured Libs voltage and current at time $t$. The corresponding label $y_i$ is the real SOC of the Libs in the last time step. Using the sliding window algorithm helps the network learn from the past and capture the temporal dynamics of the Libs, as shown in Figure 2.

### 2.2. The Network Structure.

During the pretraining phase in the source domain, the process is primarily divided into three sequential steps. First, the feature time series of the Libs is fed through the input layer to the BiLSTM layer to extract nonlinear features. Next, the extracted features are passed through the attention layer to enable the network to focus on important information while disregarding irrelevant information in the time series. Finally, the significant features are mapped to the SOC value by the FC layer. The structure of the network is depicted in Figure 3.

#### 2.2.1. BiLSTM Layer.

LSTM [29] architecture is a variant of recurrent neural network (RNN) designed to address the problem of vanishing or exploding gradients commonly encountered when processing long sequential data. By incorporating three types of gating mechanisms, i.e., input gate, forget gate, and output gate, LSTM is able to manage the flow of information and improve its ability to retain and manipulate long sequential data. Bidirectional LSTM (BiLSTM) [30] is an extension of LSTM that combines two unidirectional LSTMs, one from the beginning to the end and the other from the end to the beginning. The outputs of the two LSTM layers are merged at each time step to form the final output. Due to its ability to consider both past and future information, BiLSTM is often more effective than unidirectional LSTM in processing sequential data.

Let the learned parameters $W$ and $b$ be learnable parameters that map the hidden state $h_i$ to the space of attention scores. Secondly, the attention scores are normalized using the softmax function to obtain attention weights:

$$a_i = \text{softmax}(u_i).$$

Finally, the hidden states at all time steps and their corresponding attention weights $a_i$ are weighted summed to obtain the final output of the attention layer:

$$c = \sum_{i=1}^{T} a_i h_i.$$
**Figure 1:** Architecture of the proposed method. The architecture consists of two stages: the pretraining stage and the transfer stage.

**Figure 2:** The sliding window method.

**Figure 3:** Proposed net structure composed of BiLSTM and attention for SOC estimation.
2.2.3. FC Layer. In the end, a Libs data sequence of length $W$ within a time window is processed through BiLSTM and attention layers for feature extraction. The resulting features are then transmitted to a FC layer, which outputs an estimated Libs SOC value of length 1.

For pretraining the network, the input data is a feature matrix of size $(T - W + 1 \times W \times 2)$, and the corresponding output SOC size is a column vector of size $(T - W + 1 \times 1)$. The input-output formula for the network at time step $i$ is represented by Eq. (7), where $SOC_i$ represents the estimated SOC value at time step $i$, $f$ represents the FC layer, $x_i$ denotes the input sequence at time $i$, and $\Phi_{i,1}$ represents the first set of features of the input sequence at time $i$.

$$SOC_i = f(x_i) = (\Phi_{i,1}, \cdots, \Phi_{i,W}).$$  

During the pretraining stage, the network parameters are
initialized randomly. Since the SOC estimation is a regression problem, the mean squared error (MSE) is used as the loss function, as shown in the Eq. (8). $SOC_i$ and $\hat{SOC}_i$ represent the true SOC values and predicted SOC values at time $i$, respectively, and $n$ denotes the number of samples.

$$\text{Loss}_{\text{MSE}} = \frac{1}{n} \sum_{i=1}^{n} (SOC_i - \hat{SOC}_i)^2. \quad (8)$$

2.3. Transfer Learning for SOC Estimation. Transfer learning is aimed at applying the knowledge learned from one task to another task, especially when there is similarity or correlation between the two tasks. The model or dataset that has been trained and optimized is referred to as the source domain, while the new task that requires the application of the model or dataset is referred to as the target domain [21]. Domain adaptation is a form of transfer learning that aims to improve model performance by finding common or similar features between the source and target domains, through transforming the features of the source and target domains to make their distributions as similar as possible in a specific space [35]. The maximum mean discrepancy (MMD) is a statistical measure commonly used in domain adaptation to evaluate the difference between the two probability distributions. Specifically, it is used to minimize the distributional discrepancies between the source and target domains by aligning their feature distributions [36]. The MMD technique is described in Eq. (9).

$$\text{MMD}(X_S, X_T) = \frac{1}{n} \sum_{i=1}^{n} k(x_i) - \frac{1}{m} \sum_{j=1}^{m} \hat{k}(x'_j)^2, \quad (9)$$

where $k(\cdot)$ is the kernel function, typically a Gaussian kernel or a linear kernel; $X_S$ and $X_T$ represent the probability distributions of the source and target domains, respectively; $x_i$ and $x_j$ represent samples from $X_S$ and $X_T$, respectively; and $n$ and $m$ represent the number of samples. MMD maps samples from the source and target domains in the feature space to a reproducing kernel Hilbert space (RKHS) and computes the squared difference of their means in that space.

By introducing attention mechanisms in domain adaptation, network models can be effectively adapted to the characteristics of the target domain while mitigating disparities between the source and target domains. The attention mechanism aids the network not only in better identifying critical features within Libs time series data during feature extraction but also in learning domain-specific weight adjustments between the source and target domains. This facilitates improved utilization of knowledge from the source domain in the target domain context.

In the field of SOC estimation, different Libs types, charging and discharging strategies, or Libs discharge data under different usage environments are defined as the source and target domains, as shown in

$$X_S = \{x_i, y_i\}, i = 1, \cdots, n, \quad (10)$$

$$X_T = \{x'_i, y'_i\}, i = 1, \cdots, m, \quad (11)$$

where $x_i$ is the feature vector of the Libs, which is the current and voltage in this work; $y_i$ is the corresponding label of $x_i$, that is, the SOC value; and $n$ and $m$ represent the total number of samples in the source and target domains, respectively. Different from the traditional neural network-based SOC estimation, the proposed method adds the distance between the source domain and the target domain to the loss function of the neural network. The loss function of this method is shown in

$$\text{Loss} = \text{Loss}_{\text{MSE}} + \lambda \cdot \text{MMD}(X_S, X_T), \quad (12)$$

where $\lambda$ is the trade-off parameter, which determines the relative weight of the two loss functions in the total loss, and $\text{Loss}_{\text{MSE}}$ is the mean squared error loss function commonly used in neural network regression tasks.

Transferring pretrained network weights from the source domain to the target domain and conducting domain adaptation training on the target domain is a common method in transfer learning. In addition to the MMD method described above, there are also fine-tuning (FT) and correlation alignment (CORAL) methods [37]. FT adjusts the parameters of a pretrained model through supervised learning on the target domain data, without considering the distributional differences between the source and target domains. CORAL achieves domain adaptation by minimizing the difference between the covariance matrices of the source and target domains.

2.4. Source Domain Selection. The question of determining where to transfer knowledge in the context of transfer learning poses a significant research challenge. Although transfer learning has gained considerable attention, a comprehensive discussion about the optimal choice of source domains for transfer has been largely lacking. In this paper, we propose an innovative domain selection approach that leverages the

---

<table>
<thead>
<tr>
<th>Network structures</th>
<th>US06</th>
<th>RMSE</th>
<th>MAE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM [45]</td>
<td>3.69%</td>
<td>5.35%</td>
<td>2.93%</td>
<td>3.40%</td>
</tr>
<tr>
<td>BiLSTM [46]</td>
<td>2.82%</td>
<td>3.74%</td>
<td>2.75%</td>
<td>3.21%</td>
</tr>
<tr>
<td>LSTM-attention [47]</td>
<td>3.13%</td>
<td>4.22%</td>
<td>2.76%</td>
<td>3.25%</td>
</tr>
<tr>
<td>BiLSTM-attention(ours)</td>
<td><strong>2.36%</strong></td>
<td><strong>3.08%</strong></td>
<td><strong>1.82%</strong></td>
<td><strong>2.42%</strong></td>
</tr>
</tbody>
</table>

The best estimation results are indicated in bold.
Figure 5: SOC-estimated values and errors tested at ambient temperature of 25°C using different network structures: (a) US06 and (b) HWFTA.
similarity of current and voltage sequences as a metric between different domains. We utilize a time series-based clustering technique to effectively categorize all source and target domain data into distinct groups. Operational conditions that fall within the same category as the target domain data exhibit higher similarity and more closely aligned distribution characteristics, making them suitable candidates for pretraining the model.

The chosen metric for quantifying similarity is dynamic time warping (DTW) [38], a renowned method for measuring similarity between two time series sequences. The strength of the DTW algorithm lies in its ability to handle time series with varying lengths or temporal offsets. It achieves this by employing dynamic programming to identify the optimal alignment path between the two sequences, minimizing the cumulative distances between the corresponding points along this path. Originally designed for speech recognition, the DTW algorithm has since found widespread application in diverse fields such as time series analysis [39], voice recognition [40], and handwritten character recognition [41].

For two domain-specific battery feature sequences, denoted as $x = (x_0, \cdots, x_{n-1})$ and $y = (y_0, \cdots, y_{m-1})$, the DTW distance between them is considered as the measure of similarity between the two domains. The DTW distance is defined as follows:

$$\text{DTW}(x, y) = \min_{\pi} \sqrt{\sum_{(i,j) \in \pi} d(x_i, y_j)^2},$$

where $\pi$ is a path that can be seen as a temporal alignment of time series such that Euclidean distance between aligned time series is minimal, where $d(\cdot, \cdot)$ is the Euclidean distance.

To apply the K-means clustering method to time series data, we have refined the classical algorithm to better accommodate the grouping of time series exhibiting similar patterns or trends. This adaptation involves the division of time series into $K$ distinct clusters, where the data within each cluster exhibits maximal similarity while the data between different clusters is deliberately dissimilar. The specific procedural steps are outlined below:

1. Initialization of centroids: $K$ initial centroids are randomly selected, each representing a time series for a specific cluster
2. Distance calculation: the distance between each time series and each centroid is calculated, utilizing DTW as the distance metric
3. Assignment to clusters: each time series is assigned to the cluster whose centroid is closest in terms of DTW distance
4. Centroid recalculation: the centroids for each cluster are recalculated based on the time series contained within that cluster. Typically, this recalibration employs the mean or median as an aggregation function

where $\pi$ is a path that can be seen as a temporal alignment of time series such that Euclidean distance between aligned time series is minimal, where $d(\cdot, \cdot)$ is the Euclidean distance.

To apply the K-means clustering method to time series data, we have refined the classical algorithm to better accommodate the grouping of time series exhibiting similar patterns or trends. This adaptation involves the division of time series into $K$ distinct clusters, where the data within each cluster exhibits maximal similarity while the data between different clusters is deliberately dissimilar. The specific procedural steps are outlined below:

1. Initialization of centroids: $K$ initial centroids are randomly selected, each representing a time series for a specific cluster
2. Distance calculation: the distance between each time series and each centroid is calculated, utilizing DTW as the distance metric
3. Assignment to clusters: each time series is assigned to the cluster whose centroid is closest in terms of DTW distance
4. Centroid recalculation: the centroids for each cluster are recalculated based on the time series contained within that cluster. Typically, this recalibration employs the mean or median as an aggregation function

In our specific case, we found that two clusters were sufficient to distinguish source domains with significant similarities to the target domain and those that were notably different. The choice of two clusters allowed us to effectively highlight the key separation in the data. It provided a straightforward and interpretable way to convey the concept of source domain selection for the purpose of transfer learning.

This innovative approach guarantees that only pertinent and analogous operational conditions are incorporated into the pretraining phase of the model. This strategic selection mitigates the risk of overfitting arising from excessive data or detrimental negative transfer due to pronounced disparities between the source and target domains. As a result, our proposed method offers a robust solution for addressing the complex challenges associated with domain adaptation in transfer learning scenarios.

3. Experiment

In this section, the dataset and experiment implementation details will be discussed.

3.1. Dataset Description. In order to evaluate the performance of the proposed method in cross-domain SOC estimation, two publicly available Libs datasets were used for experimentation, namely, the Panasonic 18650 PF battery dataset and the Dynamic Test Profile from the CALCE battery team.

The Panasonic 18650 PF battery dataset consists of nickel-cobalt-aluminum lithium oxide (LiNiCoAlO2 or NCA) chemistry [42]. The collected data were obtained from
the tests conducted at different ambient temperatures ranging from -20°C to 25°C. In each test, the battery cell was first fully charged and then subjected to a power drive cycle. These drive cycles correspond to those of a Ford F150 truck and were scaled to a single 18650PF cell, from which energy was extracted during the discharge process until the cell voltage reached a cutoff value of 2.5 V. The four standard drive cycles used for testing the battery are as follows: the Urban Dynamometer Driving Schedule (UDDS), the Highway Fuel Economy Driving Schedule (HWFET), the Los Angeles 92 (LA92), and the Supplemental Federal Test Procedure Driving Schedule (US06). Some additional drive cycles, i.e., cycle 1, cycle 2, cycle 3, cycle 4, and neural network (NN), were manually created to obtain additional power. These cycles, which are composed of random combinations of UDDS, HWFET, US06, and LA92, are shown in Figure 4(a) for the US06 test at 25°C ambient temperature, which presents the measured current and voltage [43].

The Dynamic Test Profile dataset from the CALCE battery team consists of LiNiMnCo/graphite cell chemistry for the INR 18650-20R battery and includes various dynamic current profiles such as Dynamic Stress Test (DST), Federal Urban Driving Schedule (FUDS), US06 Highway Driving Schedule, and Beijing Dynamic Stress Test (BJDST) [44]. All tests were conducted under the conditions of 0°C, 25°C, and 45°C, at 80% and 50% Libs SOC, as shown in Figure 4(b) for the DST test at 25°C ambient temperature, which presents the measured current and voltage [7].

3.2. Implementation Details. Experimental evaluations were conducted on three aspects to assess the effectiveness of the BiLSTM framework, transfer learning method, and source domain selection method for cross-domain SOC estimation.

(1) The Panasonic Libs dataset was selected as the training and testing set for the model, where the US06 and HWFET were used as the test set and the remaining operational cycle data was used as the training set. The network structure consisted of a BiLSTM layer with a layer size of 1, 50 hidden units, and a fully connected layer with 50 neurons. The proposed BiLSTM-attention structure was compared with LSTM, BiLSTM, and LSTM-attention models.

(2) All datasets of the Panasonic Libs were used as the source domain, including nine driving cycles, for the pretraining of the network. The INR Libs dataset was used as the target domain, where the DST and FUDS were used to fine-tune the model or adapt it to the domain. Network parameters were adjusted to reduce the distribution differences between the source and target domains. The effectiveness of transfer learning was validated by comparing fine-tuning, MMD, and CORAL methods using US06 and BJDST.

(3) Source domain selection was performed by using DTW to measure the distance between different driving cycles, and time series clustering was used to separate the source domain data into two clusters. The SOC estimation results were compared using all data, cluster 1, and cluster 2 as pretraining data. MMD was used as the transfer method, and the distribution of training and test sets for the target domain was consistent with experiment 2.

For the pretraining phase of the source domain, the epoch was set to 300, and the learning rate was set to $10^{-2}$. For the training phase of the target domain, the epoch was set to 200, and the learning rate was set to $10^{-3}$. A lower learning rate was used in this phase to preserve the bottom-level features of the pretrained model and only adjust the top-level features. Additionally, $\lambda$, a hyperparameter that balances the two loss functions, was tested using a trial-and-error approach, searching for the optimal value of $\lambda$ from 0.1, 0.5, to 1, 5. When $\lambda$ was set to 0.5, the transfer learning method demonstrated the best performance. Both phases used Adam optimizer based on gradient descent to update their weights, and backpropagation was used to perform weight updates.

Each experiment was repeated five times to address the randomness of deep learning algorithms, and the average of these test results was used as the evaluation metric. The PyTorch framework was used to implement the models, and the server operated on the Linux operating system, allowing for easy deployment of the models onto BMS devices. These models were trained using two NVIDIA RTX 6000 GPUs, providing efficient ray tracing and AI training performance.

Two evaluation metrics, including root mean square error (RMSE) in Eq. (15) and mean absolute error (MAE) in Eq. (14), were used to assess the performance of the proposed method in cross-domain SOC estimation.

\[
\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |\text{SOC}_{i} - \hat{\text{SOC}}_{i}|, \quad (14)
\]

\[
\text{RMSE} = \left( \frac{1}{N} \sum_{i=1}^{N} (\text{SOC}_{i} - \hat{\text{SOC}}_{i})^2 \right)^{1/2}, \quad (15)
\]

Table 2: Estimation results with different transfer learning methods.

<table>
<thead>
<tr>
<th>Transfer method</th>
<th>MAE</th>
<th>RMSE</th>
<th>MAE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>FT</td>
<td>1.80%</td>
<td>2.24%</td>
<td>1.76%</td>
<td>2.07%</td>
</tr>
<tr>
<td>MMD</td>
<td>1.35%</td>
<td>1.91%</td>
<td>1.28%</td>
<td>1.58%</td>
</tr>
<tr>
<td>CORAL</td>
<td>2.73%</td>
<td>3.48%</td>
<td>2.84%</td>
<td>3.46%</td>
</tr>
</tbody>
</table>

The best estimation results are indicated in bold.
Figure 7: SOC-estimated values and errors tested at ambient temperature of 25°C using different transfer learning methods: (a) US06 and (b) BJDST.
where \( N \) is the number of test samples, \( \text{SOC}_i \) is the true SOC value of the \( i \)th sample, and \( \hat{\text{SOC}}_i \) is the predicted SOC value of the \( i \)th sample.

The effectiveness of the proposed method was determined based on the RMSE and MAE values obtained from the experiments. Lower RMSE and MAE values indicate better performance in cross-domain SOC estimation.

### 4. Results and Discussion

#### 4.1. Estimation Results with Different Network Structures

To validate the effectiveness of the BiLSTM-attention model in SOC estimation, the Panasonic Libs dataset was selected as the training and testing set. The dataset includes nine driving cycles, with US06 and HWFET used as test sets and the remaining operational cycle data used for training. LSTM was used as the baseline model since it generally performs better than RNN and GRU networks. The results of the comparison between the BiLSTM-attention model and the other network structures are presented in Table 1, which shows the MAE and RMSE values for each network structure across all test sets. The BiLSTM-attention model outperforms other network structures, with the lowest average MAE and RMSE values of 2.36% and 3.08%, respectively, for the US06 driving cycle and 1.82% and 2.42%, respectively, for the HWFET driving cycle.

Compared with the baseline LSTM-based SOC estimator that only considers unidirectional temporal features, the proposed BiLSTM-attention network can capture the temporal dependency of Libs data sequences from both directions and extract important information in the time series. By considering both past and future information in the sequence, BiLSTM-attention is able to better model the complex dynamic behaviors of the Libs and provide more accurate SOC estimations. In addition, the attention mechanism in the network allows for dynamic weighting of the input sequence, highlighting the most relevant features for SOC estimation, which further improves the accuracy of the predictions. BiLSTM and LSTM-attention, which only consider bidirectional temporal features or important information, respectively, show significantly improved performance compared to LSTM.

Figure 5 shows the estimation results and error results of US06 and HWFTA at 25°C. Compared with the other three network structures, the SOC value estimated by BiLSTM-attention is closer to the true value, and the error is significantly smaller.

The loss reduction trends during the pretraining process of BiLSTM-attention and other network structures are shown in Figure 6. All transfer losses rapidly decreased in the early training stage and then slowly decreased. Compared with other network structures, BiLSTM-attention showed faster convergence and lower loss. This analysis indicates that BiLSTM-attention can accelerate network convergence and improve the accuracy of SOC estimation.

#### 4.2. Estimation Results with Different Transfer Learning Methods

The pretraining of all datasets of Panasonic Libs as the source domain and INR Libs dataset as the target domain was conducted. The DST and FUDS driving cycle data were optimized for network parameters through domain adaptation. To evaluate different transfer learning methods, the FT, MMD, and CORAL methods were applied to the US06 and BJDST datasets. Table 2 shows the estimation error results. MMD achieved the best transfer learning effect, with an average MAE and RMSE of 1.32% and 1.75%, respectively.

The FT method did not consider the distribution difference between the source and target domains, while CORAL could only align the covariance matrix between the two domains and was more suitable for linear problems. Compared with these two methods, MMD can better handle nonlinear problems by using a kernel function to map data to high-dimensional space for nonlinear transformation. Figure 7 shows the estimation and error diagrams of US06 and BJDST at 25°C.

#### 4.3. Estimation Results with Source Domain Selection

To validate the effectiveness of the source domain selection method, clustering and selection were performed on the source domain data. The distance between the source and target domains was measured, and the source domain clustering results were divided into CL0 and CL1 based on their similarity to the target domain Libs SOC. The DTW
Figure 8: SOC-estimated values and errors tested with source domain selection: (a) 0°C and (b) 25°C.
We conducted transfer experiments across different domains to address the negative transfer issue caused by significant distribution differences under different operating conditions. The data used to support the findings of this study have been deposited in the Mendeley Data (doi:10.17632/wykh8y7g.1 and doi:10.17632/cp3473x7xv.3).
Conflicts of 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.

Acknowledgments

The authors acknowledge the financial support from the University-Industry Cooperation of Fujian Province under Grant Nos. 2021H6026 and 2022H6024 and the Intelligent Computing and Application Research Team of Concord University College (2020-TD-001).

References


