An unsupervised domain adaptation framework for cross-conditions state of charge estimation of lithium-ion batteries

Yunpeng Liu, *Member, IEEE,* Moin Ahmed, Jiangtao Feng, *Member, IEEE,* Zhiyu Mao, *Member, IEEE,* and Zhongwei Chen

***Abstract*—With the rapid development of deep learning, battery state of charge (SOC) estimation has made major strides. However, the batteries' inconsistency and changing working conditions lead to the distribution discrepancy across domains, which further affects the prediction accuracy of the pre-trained model. Moreover, collecting sufficient and labeled data is labor-intensive to gain a well-performed SOC estimator. To overcome these drawbacks, this paper proposes a novel SOC estimation framework based on adversarial domain adaptation. Firstly, a distinctive SOC estimator is constructed and trained to capture the mapping relationship between the original input and the battery SOC based on the offline source dataset with a specific working condition. Then, an adversarial network with a reconstruction module and maximum mean discrepancy constraint is designed to extract the domain-invariant features and decrease distribution discrepancy across domains. Thus, the pre-trained model could be transferred to the different working conditions using only the limited and unlabeled target data. Experimental results demonstrate that the best cross-domain RMSE of the proposed transfer framework are 1.33%, 2.57 %, and 1.45 % for fixed ambient temperatures, changing ambient temperatures, and changing battery type, respectively, indicating this framework emerges as a promising solution for the precise battery SOC cross-domain estimation.**

***Index Terms*—Lithium-ion batteries, State of charge estimation, Transfer learning, Unsupervised domain adaptation, Charging conditions.**

NOMENCLATURE

Parameters

α, β	Non-negative trade-off parameters
η_{pt}, η_{tf}	Learning rate during the pre-trained step and the TL process
$\mathcal{D}^s, \mathcal{D}^t$	Source domain,and target domain
$\mathcal{D}^{t,train}, \mathcal{D}^{t,test}$	Labeled and unlabeled target domain
$\mathcal{L}_{\text{mmd}}, \mathcal{L}_{\text{pred}}, \mathcal{L}_{\text{rec}}, \mathcal{L}_{\text{adv}}, \mathcal{L}_{\text{total}}$	MMD loss, SOC estimation loss, reconstruction loss, adversarial training loss, and total loss during the transfer process
$\mathcal{P}(x^t), \mathcal{P}(x^t)$	Marginal probability distributions from $\mathcal{D}^s$ and $\mathcal{D}^t$
$\theta_{f_e^s}, \theta_{f_e^t}$	Parameters of f_e^s and f_e^t
$\theta_{f_{re}}, \theta_{f_{dis}}, \theta_{f_{pred}}$	Parameters of f_{re} and f_{dis}, and f_{pred}
$F^s, F^{t,train}$	Domain-invariant features from $\mathcal{D}^s$ and $\mathcal{D}^t$
f_e^s, f_e^t	Feature encoder for $\mathcal{D}^s$ and $\mathcal{D}^t$
$f_{re}, f_{pred}, f_{dis}$	Reconstructed decoder, SOC predictor, and domain discriminator
x^s, x^t	Input vector from $\mathcal{D}^s$ and $\mathcal{D}^t$
y^s, y^t	SOC corresponding to x^s and x^t

Abbreviations

BiLSTM	Bi-directional LSTM
BMS	Battery management system
CNN	Convolutional neural network
CRNN	Convolutional recurrent neural network
DARM	An adversarial domain adaptation framework with the reconstruction module and the MK-MMD constraint
DNN	Deep learning network
DWS 1D CNN	Depthwise separable 1D CNN
IID	Independent identical distribution
KDE	Kernel density estimation
LIBs	Lithium-ion batteries
LSTM	Long short term memory
MAE	Mean absolute error
MK-MMD	Multiple kernel MMD
MMD	Maximum mean discrepancy
MSE	Mean squared error
Resblock	Residual block
ResNets	Residual network
RKHS	Reproducible kernel Hilbert space
RMSE	Root mean square error
RNN	Recurrent neural network
SOC	State of charge
TL	Transfer learning
UDA	Unsupervised domain adaptation

I. INTRODUCTION

OWING to the advantages of high energy density, low manufacturing cost, and low self-discharge rate, Lithium-ion batteries (LIBs) are widely employed in various

This work was supported in part by the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDB0600400), the National Natural Science Foundation of China (Grant No. 52102112), and the Fundamental Research Funds for the Central Universities, China (Grant No. xzy012022072). (Corresponding authors: Jiangtao Feng, Zhiyu Mao, and Zhongwei Chen)

Yunpeng Liu, Zhiyu Mao, and Zhongwei Chen are with the Power Battery & Systems Research Center, the State Key Laboratory of Catalysis, Dalian Institute of Chemical Physics, Chinese Academy of Sciences, Dalian 116023, China. (e-mail: liuyunpeng1994@dicp.ac.cn; zhymao@dicp.ac.cn; zwchen@dicp.ac.cn).

Moin Ahmed is with the Department of Chemical Engineering, University of Waterloo, 200 University Avenue West, Waterloo, ON N2L 3G1, Canada (e-mail: moin.ahmed@uwaterloo.ca).

Jiangtao Feng is with the Department of Environmental Science & Engineering, School of Energy and Power Engineering, Xi'an Jiaotong University, Xi'an, 710049, China (e-mail: fjtes@xjtu.edu.cn).

applications, such as electric vehicles, electronic equipment, and photovoltaic systems [1, 2]. The battery management system (BMS) can guarantee the safety and high-efficiency operation of batteries [3, 4]. Among many functions, the state of charge (SOC) is a vital metric to reflect the battery's remaining charge at present [5]. However, the batteries' SOC could not be directly gauged, and it is determined using other parameters measured by sensors in most cases [6]. Therefore, developing appropriate SOC estimation methods has attracted much research interest.

The classical SOC estimation approaches can be categorized into two kinds: the open circuit voltage method and the coulomb counting approach [7]. However, these methods not only need precise sensors or inference algorithms to avoid the problem of data drift caused by random noises but also ensure the tested batteries' steady state to interface the over potentials' elimination [8]. These harsh and fussy operation requirements, make them impractical in complex scenarios. To overcome the above drawbacks, numerous model-based and data-driven SOC estimation approaches have been investigated [9]. Model-based approaches can be divided into equivalent circuit models and electrochemical models. The former is relatively easy and common but exhibits terrible performance when the battery is exposed to an extreme operational environment such as low environmental temperature or high current density. Meanwhile, the latter is relatively more accurate, while the modeling parameters must be additionally decided in this process, which is unsuitable for batteries' online monitoring task [10].

Deep learning (DL) approaches for LIBs' SOC estimations have become increasingly attractive because their neural network possesses powerful nonlinear mapping capacity [11]. Generally, without the participation of complicated electrochemical properties of the battery, the SOC values could be inferred based on the online monitoring data, by a series of black-box models, such as the convolutional neural network (CNN) [12], recurrent neural network (RNN) [13], long short term memory (LSTM) [14], gate recurrent unit (GRU) [15], fully connected (FC) network [16], etc. The regular data-driven SOC estimation can construct SOC estimation models based on historical cycle data, which are built on the assumption that the training dataset (source domain) and test dataset (target domain) follow the same distribution [17], namely independent identical distribution (IID). Unfortunately, the source and target domains usually have related but different data distribution in practices. The changing working conditions (such as different ambient temperatures and charging/discharging protocols) or internal properties (such as individual differences in battery manufacturing processes) would inevitably result in discrepant data distributions across domains, namely domain shift [18]. Once the operating conditions are out of the training dataset, the pre-trained SOC estimator could not work well for monitoring the target battery. Theoretically, suppose the SOC estimator's generalization ability would like to be improved for one-type batteries in changing situations. In that case, a vast training dataset should be collected to fully cover the various operating conditions, which is labor-intensive and time-consuming.

Although the pre-trained model based on the source domain could not be directly employed for the target tasks due to domain drift, there is shared knowledge about batteries' SOC across domains. Therefore, it is feasible to enhance the estimator's generalization ability with the help of this shared and valuable knowledge from the source domain. Based on this argument, the transfer learning (TL) concept attracted increasing attention in SOC estimation tasks [19], which has been successfully utilized in computer vision [20], natural language processing [21], and multi-modal tasks [22]. For DL industrial applications, especially when dealing with large-scale and complex systems, the computational burden becomes a signature challenge. TL method as an effective solution, alleviates the computational resources and time required to train one model from scratch by leveraging knowledge transferred from pre-trained models to new tasks. Xie et al. [23] utilized the New England 39-bus system and the South Carolina 500-bus system to validate that the TL process could provide accurate results despite insufficient training data. Liu et al. [24] employed the filed data collected from 50 wind turbines in commercial wind farms to verify their proposed TL methods, effectively decreasing the computational burden. Liu et al. [25] proposed a TL method-based probabilistic wind power forecasting method. The results indicated that the computation burden was satisfied with the TL method for the practical wind farms in China, in which the test datasets included the eight wind farms in Ji Bei, China, with the wind power output data collected every 15 min and covering 18 months. Hence, for the large-scale system, by utilizing the knowledge accumulated from source tasks, the TL method enhances model performance on target tasks and relieves the computational burden, which is particularly important in practical applications such as the BMS. Li et al. [26] developed a deep TL method to boost the training process of the terminal voltage estimation for 53 electric buses. Zheng et al. [27] employed the TL method to construct the state of health estimation model for a large amount of the electric bus operational data. Therefore, the application of the TL method to battery SOC estimation has broad prospects.

As a typical TL method, a parameter transfer-based fine-tuning strategy is relatively simple and popular. For the fine-tuning approach in the batteries' SOC estimation task, the pre-trained model is first gained based on the large-scale source data. This model's parameters are loaded and serve as the initial status of the model for the target battery. Finally, the whole or sectional parameters are retrained and updated based on a few target data, including true SOC values. This strategy could effectively avoid the model training from scratch. Recently, some studies have exhibited the universality of fine-tuning in battery SOC estimation. Vidal et al. [28] pre-trained the LSTM model on one battery type, and then this pre-trained model as a starting point, was retrained and fitted on another battery type. For multi-task learning including SOC estimation, Che et al. [29] combined a CNN-FC model with fine-tuning technology and achieved high accuracy and computational efficiency under different application scenarios by retraining the specific task layers. Tian et al. [30] applied fine-tuning on one layer of the pre-trained deep learning network (DNN) to fit various scenarios, providing more accurate SOC estimation results at lower training costs. However, the SOC estimation accuracy will be greatly compromised when the domain drift is serious, causing the pre-trained model to not be suitable for the target task by fine-tuning strategy. Besides, the quantity of the target data is vital for the SOC estimation, and few-shot target training data might let the fine-tuned model fall into the overfitting. More troublingly, as a typical supervised task, it is essential for SOC estimation based on the fine-tuning strategy to gain precise ground truth of the target data. However, obtaining abundant labeled target domain data is costly and time-consuming in real-world scenarios.

The unsupervised domain adaptation (UDA) method is commonly used in computer vision tasks, which has provided new perspectives and solutions for addressing the scarcity of labeled training data. Long et al. [31] constructed separate image classification networks for the source and target domains, assuming that the differences between the networks of the source and target domains can be represented by a residual function. They also used maximum mean discrepancy (MMD) to align the output features of multiple network layers. Saito et al. [32] proposed a domain adaptation method for object detection based on strong local alignment and weak global alignment. This method applies domain adaptation at two levels: strong alignment at the lower levels of features such as texture and color, and weak alignment at the higher levels of features involving semantic information. Luo et al. [33] combined adversarial learning and collaborative training to propose a category-level adversarial network. They jointly trained two semantic segmentation networks using data from both the source and target domains to check if the semantic segmentation categories across different domains are aligned. Therefore, the UDA method is a potential and effective strategy to handle the problems of fine-tuning approach in the field of battery state estimation [34, 35], which could fundamentally minimize the marginal and conditional distribution discrepancy and structure across domains and finally find the domain-invariant feature [36, 37]. Shen et al. [38] proposed a novel TL framework with Minimum Estimation Discrepancy to tackle the challenges of distribution discrepancy and data limitation in cross-domain SOC estimation under different ambient temperatures. Bian et al. [39] presented a deep transfer neural network with multi-scale MMD for cross-domain SOC estimation, which could adaptively transfer the high-level features with small-scale data from the target domain. Shen et al. [40] constructed an adversarial domain adaptation framework by adding a discriminator and prompting it to fail to judge which domain the features come from via an adversarial mechanism. The appending MMD constraint also decreases the discrepancy of feature distribution across domains. Meng et al. [41] developed an adversarial TL-based training framework to extract the domain invariants across domains, guided by minimizing Wasserstein distance, and results displayed satisfactory SOC estimation performance and adaptation to varying operating conditions and battery health degradation. However, the transfer performances of an adversarial domain adaptation with a distance metric constraint, remain to be enhanced. This framework combined with other kinds of constraints, such as the reconstructed constraint, is worthwhile to be explored for the domain adaption study. Besides, the neural network structure would affect the extraction efficiency of domain-invariant features, and then decide the prediction accuracy of the battery SOC. Nevertheless, the mainstream neural networks are still the variant or combination of shallow CNN or LSTM. Generally, the deep neural network structure allows a higher level of abstraction than a shallow one, hence the novel deep neural network structure has great potential to be employed in the adversarial domain adaption. In a word, this work proposes a novel TL framework, namely an adversarial domain adaptation framework with the reconstruction module and the multiple kernel MMD (MK-MMD) constraint, donated as DARM, for predicting battery SOC under the low requirement of unlabeled target data with changing conditions. The main contributions of this work are summarised as follows:

1) A novel TL-based SOC estimation framework is proposed to extract domain-invariant features and decrease distribution discrepancy across domains. Based on this TL framework, the source estimator trained at a specific working condition can be generalized to different conditions even though the target domain is limited and unlabeled.
2) A new SOC estimator consisting of depthwise separable (DWS) 1D CNN, residual connection, Bi-directional LSTM (BiLSTM), and attention mechanism, is presented to extract latent features and map the measured data and SOC value more effectively.
3) By introducing the reconstructed decoder to form the U-Net structure with the feature encoder, the raw input could be reconstructed to force the feature encoder to extract more valuable information. Meanwhile, an MK-MMD module is added as an auxiliary module to relieve the distribution discrepancy between domains owing to dynamic working conditions.
4) The effectiveness, robustness, and generalization ability of the proposed framework are verified by conducting comprehensive comparisons to other popular methods i.e., pre-trained model prediction and fine-tuning.

The rest of this article is structured as follows. Section II gives the problem statement, methodology overview, and detailed TL-based SOC estimation strategy. Section III presents the comprehensive results of comparative studies and discussions. Finally, the conclusions are drawn in Section IV.

II. Methodology

In this section, the problem formation is first introduced for TL-based SOC estimation study in Section II-A. Secondly, an overview of the main procedure is illustrated in Section II-B, and the whole process could be divided into two stages: offline training and online monitoring. Then, Section II-C illustrates the network structure of the proposed DARM framework with feature extraction, input reconstruction, SOC estimation, adversarial learning, and feature alignment metric in detail. Finally, the pre-trained and transferring total optimizations are elaborated in Section II-D and Section II-E, respectively.

This article has been accepted for publication in IEEE Transactions on Transportation Electrification. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TTE.2024.3483973

A. Problem formation

In this work, the TL-based SOC estimation is studied in the domain discrepancy scenario. The source domain $\mathcal{D}^s$ and target domain $\mathcal{D}^t$ are first defined, and they contain large-scale labeled data, and the limited-scale unlabeled data under different working conditions, respectively. The source domain is defined as $\mathcal{D}^s = \{x_i^s, y_i^s\}_{i=1}^{m_s}$ with m_s samples, where each x^s is a vector containing voltage, current, and temperature, i.e., $x^s = [I, V, T]$, and y^s is the corresponding true SOC value. The $\mathcal{D}^t$ is denoted as $\mathcal{D}^t = \{\mathcal{D}^{t,train}, \mathcal{D}^{t,test}\}$, where $\mathcal{D}^{t,train}$ represents the monitoring unlabeled target samples for the UDA process, that is $\mathcal{D}^{t,train} = \{x_i^{t,train}\}_{i=1}^{m_{t,train}}$ with $m_{t,train}$ samples, and $\mathcal{D}^{t,test}$ represents the monitoring labeled target samples for evaluating SOC prediction performance, that is, $\mathcal{D}^{t,test} = \{x_i^{t,test}\}_{i=1}^{m_{t,test}}$ with $m_{t,test}$ samples. However, the number of $m_{t,train}$ samples is much fewer than m_s samples, and thus it is hard to gain a satisfactorily well-trained model for $\mathcal{D}^{t,test}$, if the model is trained from scratch based on $\mathcal{D}^{t,train}$. Meanwhile, a domain $\mathcal{D}$ usually consists of a feature space X and a probability distribution $\mathcal{P}(x)$, so $\mathcal{D}^s$ and $\mathcal{D}^t$ are separately sampled from two marginal probability distributions $\mathcal{P}(x^s)$, and $\mathcal{P}(x^t)$. Traditional DL models assume that the $\mathcal{D}^s$ and $\mathcal{D}^t$ satisfy the IID in advances. However, due to the diverse working conditions and battery type, this assumption lowers the generalization of the related DL models. In practice, there are inevitable discrepancies in data distribution across domains, i.e., $\mathcal{P}(x^s) \neq \mathcal{P}(x^t)$, and thus the pre-trained SOC estimator based on $\mathcal{D}^s$ cannot be generalized to $\mathcal{D}^t$.

The proposed framework is designed to build a SOC estimation model for $\mathcal{D}^{t,test}$ by taking full advantage of the transferable knowledge from $\mathcal{D}^s$ and the few-shot unlabeled data from $\mathcal{D}^{t,train}$. Namely, this work aims to learn a function h which approximates the online monitoring battery's SOC directly from the raw data, which can be expressed as,

$$y^t \approx h\left(x_i^t\right) \tag{1}$$

where x_i^t and y_i^t donate the raw data and ground truth for i-th time step in target domain $\mathcal{D}^t$.

B. Methodology overview

An overview of the main procedure of the proposed method is displayed in Fig. 1, divided into two main stages, i.e., offline training and online monitoring. Firstly, the raw data such as voltage, current, and temperature, are collected and pre-processed before being used as the model input. Then, for the offline training stage, the SOC estimator is trained by $\mathcal{D}^s$ to gain the pre-trained SOC estimator. After the relative parameters are initialized based on this pre-trained estimator, the DARM framework is simultaneously trained by $\mathcal{D}^s$ and $\mathcal{D}^{t,train}$ to realize the UDA for changing working conditions. Finally, for an online monitoring stage, this offline estimator is deployed in an online monitoring system, and online collected $\mathcal{D}^{t,test}$ is pre-processed and then inputted into the adapted estimator to infer the real-time SOC value.

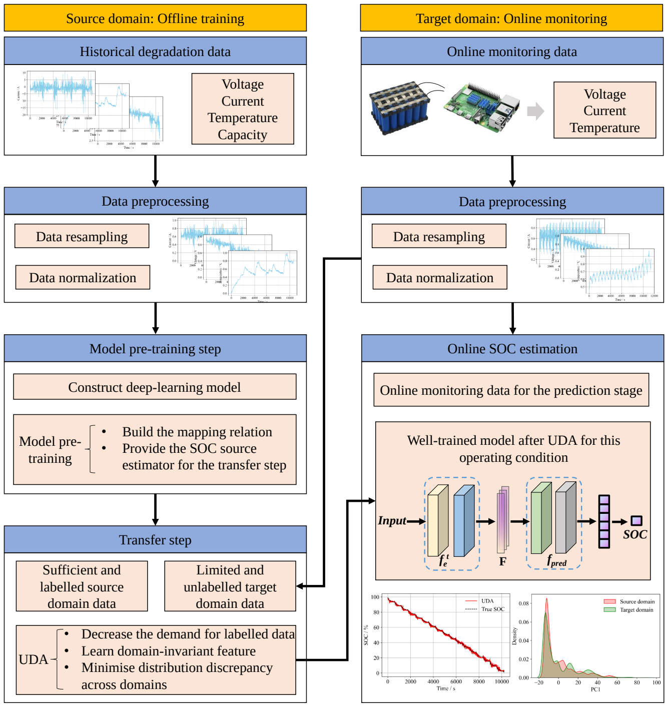
Source domain: Offline training
Historical degradation data
Voltage
Current
Temperature
Capacity
Data preprocessing
Data resampling
Data normalization
Model pre-training step
Construct deep-learning model
Model pre-training
Build the mapping relation
Provide the SOC source estimator for the transfer step
Transfer step
Sufficient and labelled source domain data
Limited and unlabelled target domain data
UDA
Decrease the demand for labelled data
Learn domain-invariant feature
Minimise distribution discrepancy across domains
Target domain: Online monitoring
Online monitoring data
Voltage
Current
Temperature
Data preprocessing
Data resampling
Data normalization
Online SOC estimation
Online monitoring data for the prediction stage
Well-trained model after UDA for this operating condition
Input
f_e^t
F
f_{pred}
SOC
UDA
True SOC
Source domain
Target domain
Time / s
PC1
Density

Fig. 1. The framework of the proposed methodology.

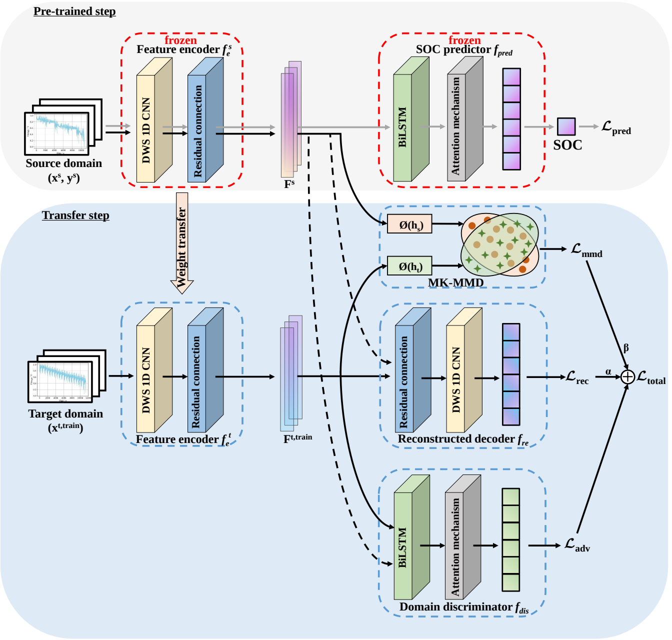

Fig. 2. The framework of the proposed DARM.

C. Proposed framework

To achieve a superior domain-invariant representation for SOC estimation for LIBs, this work proposes the DARM framework as shown in Fig. 2, which consists of five modules: two feature encoder (source domain: f_e^s; target domain: f_e^t), a reconstructed decoder f_{re}, a SOC predictor f_{pred}, a domain discriminator f_{dis}, and an MK-MMD module. This framework consists of two steps, i.e., the pre-trained (grey arrowed line) and transfer steps (black arrowed line). For the pre-trained step, the SOC estimator consists of a feature encoder for source domain f_e^s and a SOC predictor f_{pred}, trained from scratch to gain the SOC source estimator by reducing SOC estimation loss $\mathcal{L}_{\text{pred}}$. For the transfer step, the weights of f_e^s are used to initialize the feature encoder for the target domain f_e^t, and its weights and those of SOC predictor f_{pred} are frozen. An adversarial network would be formed by combining two feature encoders (f_e^s and f_e^t) and a domain discriminator f_{dis}, in which the domain discriminator distinguishes whether the extracted features are from the source domain or the target domain. Via the constant adversarial training, the domain-invariant feature (source domain: F^s; target domain: $F^{t,train}$) can be obtained. Meanwhile, a domain distribution discrepancy metric, i.e., MK-MMD, is adopted to minimize the distance between the extracted features across domains and realize the feature alignment during adversarial learning. Besides, to ensure the extracted feature contains a more meaningful representation of the input data, a reconstructed decoder f_{re} is added to be expected to reconstruct the input data based on the extracted feature as closely as possible. For the transfer learning step, the total loss $\mathcal{L}_{\text{total}}$ is a weighted sum of the MMD loss $\mathcal{L}_{\text{mmd}}$, reconstruction loss $\mathcal{L}_{\text{rec}}$, and adversarial training loss $\mathcal{L}_{\text{adv}}$, which can be expressed as,

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{adv}} + \alpha \cdot \mathcal{L}_{\text{rec}} + \beta \cdot \mathcal{L}_{\text{mmd}} \tag{2}$$

where α and β are non-negative trade-off parameters. Firstly, Section II-C1 introduces the structures of two feature encoder (f_e^s and f_e^t), reconstructed decoder f_{re}, and reconstruction loss $\mathcal{L}_{\text{rec}}$. Secondly, the detail of SOC predictor f_{pred} and SOC estimation loss $\mathcal{L}_{\text{pred}}$ are illustrated in Section II-C2. Then, Section II-C3 introduces the domain discriminator f_{dis} and the adversarial training loss $\mathcal{L}_{\text{adv}}$. Finally, the feature alignment metric MMD and MMD loss $\mathcal{L}_{\text{mmd}}$ are elaborated in Section II-C4

As the basement of the DARM, the SOC estimator consists of the feature extractor f_e and the SOC predictor f_{pred}, and its design inspiration comes from the structure of the convolutional recurrent neural network (CRNN) [42] because the multi-dimensional time-series input data x^s possesses a similar sequence feature representation as the scene text recognition task. Hence, the feature extractor f_e is constructed by DWS 1D CNN and residual connection, which is employed to extract a sequential feature representation F from a multi-dimensional sequence, and its detailed structure will be discussed in Section II-C1. Then a sequence of feature vectors extracted from the feature maps serves as the input for the subsequent SOC predictor. Importantly, each feature vector of a feature sequence is generated from left to right on the feature maps by column. This means the i-th feature vector is the concatenation of the i-th columns of all the maps, thus its global temporal information is captured effectively. Besides, the tensor shape of the extracted features is also suitable for the input requirement of RNN by the map-to-sequence operation. hence, based on the design concept of CRNN structure, the SOC predictor f_{pred} consists of the BiLSTM layers and the attention mechanism, and its detailed structure will be discussed in Section II-C2. Besides, the feature extractor f_e and SOC predictor f_{pred} are mutually independent, which benefits the subsequent transfer stage as shown in Fig. 2.

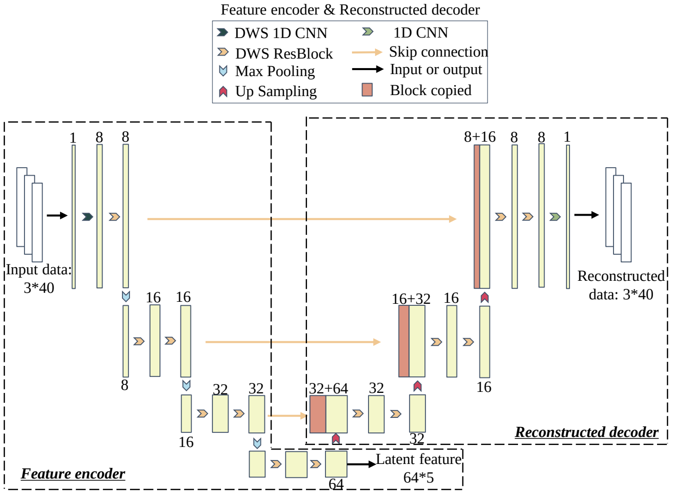

Fig. 3. The network architecture of the U-Net structure, consisted of a feature encoder and reconstructed decoder.

1) Feature extraction and input reconstruction: To guide the feature encoder to learn valuable information and avoid overfitting to few-shot unlabeled target data, a reconstruction constraint is introduced into this framework [43–45]. A reconstructed decoder decodes the features extracted by the feature encoder, so a U-Net structure is employed to restore the input data. In 2015, Ronneberger et al. [46] proposed a novel CNN with U-shaped architecture, which is particularly well-suited for biomedical image analysis, such as the segmentation of cells or organs in medical images. As shown in Fig. 3, the U-shaped architecture is built by a feature encoder f_e (dotted box on the left) as the gathering of a contracting path and a bottleneck, and a reconstructed decoder f_{re} as an expansive path (dotted box on the right). Each layer in the feature encoder is typically connected to the corresponding layer in the reconstructed decoder, forming cross-level skip connections (light yellow arrow), which could enable the network to transmit richer information. Therefore, forming a U-shaped configuration in this work ensures the feature encoder can extract a more meaningful representation from input data through multi-level feature extraction and reconstruction. The feature extraction and data reconstruction results can be calculated as follows.

$$\begin{cases} F & = f_e\left(x\right) \\ \hat{x} & = f_{re}(x) \end{cases} \tag{3}$$

where $\hat{x}$ and F denote the reconstructed data and the latent feature corresponding to input samples x, respectively.

The feature encoder captures multi-scale features by numerous convolution operations and reduces the spatial dimensions of the input data by repeated max pooling down-sampling. Finally, the latent feature F is outputted in the last convolution operation of the feature encoder, which serves as the remaining modules' input including the reconstructed decoder. Table I displays the detailed configuration of the feature encoder, and

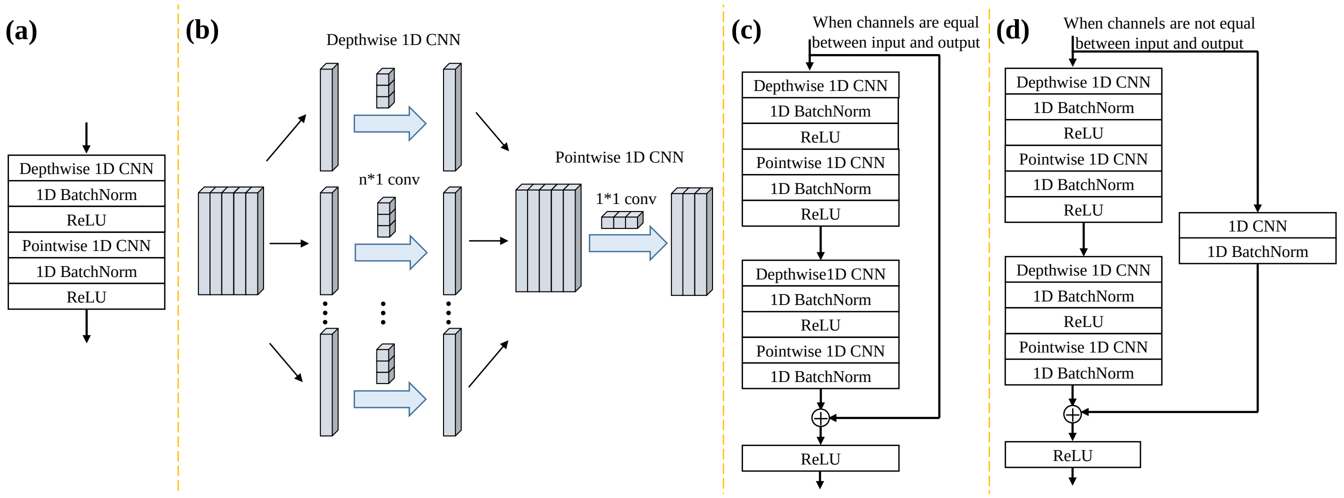

Fig. 4. The detail operation of depthwise 1D CNN and pointwise 1D CNN (a); The detail structure of the DWS 1D CNN (b), Resblock with identity shortcut (c), and Resblock with projection shortcut (d).

TABLE I
DETAILS OF THE DARM.

	Basic block	Input size	Output size
Feature encoder	DWS 1D CNN	(3, 40)	(8, 40)
	DWS ResBlcok	(8, 40)	(8, 40)
	Max Pooling 1D	(8, 40)	(8, 20)
	DWS ResBlcok	(8, 20)	(16, 20)
	DWS ResBlcok	(16, 20)	(16, 20)
	Max Pooling 1D	(16, 20)	(16, 10)
	DWS ResBlcok	(16, 10)	(32, 10)
	DWS ResBlcok	(32, 10)	(32, 10)
	Max Pooling 1D	(32, 10)	(32, 5)
	DWS ResBlcok	(32, 5)	(64, 5)
	DWS ResBlcok	(64, 5)	(64, 5)
Reconstructed decoder	Up sampling	(64, 5)	(64, 10)
	DWS ResBlcok	(96, 10)	(32, 10)
	DWS ResBlcok	(32, 10)	(32, 10)
	Up sampling	(32, 10)	(32, 20)
	DWS ResBlcok	(48, 20)	(16, 20)
	DWS ResBlcok	(16, 20)	(16, 20)
	Up sampling	(16, 20)	(16, 40)
	DWS ResBlcok	(24, 40)	(8, 40)
	DWS ResBlcok	(8, 40)	(8, 40)
	1D CNN	(8, 40)	(3, 40)
SOC predictor	Reshape	(64, 5)	(5, 64)
	BiLSTM	(5, 64)	(5, 128)
	Attention layer	(5, 128)	128
	Linear	128	1
Domain discriminator	Reshape	(64, 5)	(5, 64)
	BiLSTM	(5, 64)	(5, 128)
	Attention layer	(5, 128)	128
	Linear	128	2

it mainly consists of DWS 1D CNN, and DWS residual block (Resblock) in this work.

The DWS 1D CNN is an extension of DWS CNN for processing 1D time-series data, which could maintain effectiveness while reducing the parameter amount and computational complexity [47]. As shown in Fig. 4a, the main idea of DWS 1D CNN is that the convolution is split into depthwise convolution and pointwise convolution. The former applies convolution separately to each input channel using the corresponding kernel, and the latter uses a 1x1 convolutional kernel to perform convolution on the output of each channel, integrating the channel information. The detailed structure of one DWS 1D CNN is shown in Fig. 4b, the activation function ReLU and batch normalization 1D BatchNorm should be added to introduce nonlinear transformations and enhance network stability and training speed. A Resblock is a fundamental building unit in deep neural networks, used to construct residual networks (ResNets) [48]. The design of the Resblock aims to mitigate issues such as vanishing and exploding gradients by introducing residual connections, making the network more trainable. In this work, according to whether the dimensions of the input and output are the same, the residual connections of Resblock could be divided into two types, i.e., identity shortcut (Fig. 4c) and projection shortcut (Fig. 4d). For the main path of Resblock, there are two DWS 1D CNN blocks to extract features and learn representations.

In this work, the reconstructed decoder receives the low-resolution features F from the feature encoder, and gradually restores them to the original resolution of the input data through up-sampling operations. As shown in Fig. 3 and Table I, the U-Net exhibits a symmetric structure, with the feature encoder and the reconstructed decoder having similar hierarchical structures but opposite directions. Note that the last output layer is a pointwise convolution, which is used to adjust the channel dimensions to meet the requirements of the network output. The reconstruction loss $\mathcal{L}_{\text{rec}}$ is calculated from the standard mean squared error (MSE) between $\hat{x}$ and x [45],

$$\mathcal{L}_{\text{rec}} = \frac{1}{m}\sum_{i=1}^{m}(\hat{x}_i - x_i)^2 \tag{4}$$

where x_i and $\hat{x}_i$ respectively correspond to the i-th sample and its corresponding reconstruction result for total m samples.

2) SOC estimation: As shown in Fig. 2, because only source data possess the ground truth during the pre-trained step, the SOC predictor f_{pred} is trained to map the source latent features F^s to the battery's SOC. The latent features serve as the SOC predictor's inputs to gain the SOC estimation value $\hat{y}^s$, which can be expressed as,

$$\hat{y}^s = f_{pred}(F^s) \tag{5}$$

The minimization SOC estimation loss function in pre-

This article has been accepted for publication in IEEE Transactions on Transportation Electrification. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TTE.2024.3483973

trained stage $\mathcal{L}_{\text{pred}}$ is formulated as the MSE of the supervised SOC estimation of the source domain.

$$\mathcal{L}_{\text{pred}} = \frac{1}{m_s}\sum_{i=1}^{m_s}(\hat{y}_i^s - y_i^s)^2 \tag{6}$$

where y_i^s and $\hat{y}_i^s$ respectively correspond to the i-th sample and its corresponding estimation result for total m_s sample. The detailed structure of the SOC predictor f_{pred} will be introduced as follows.

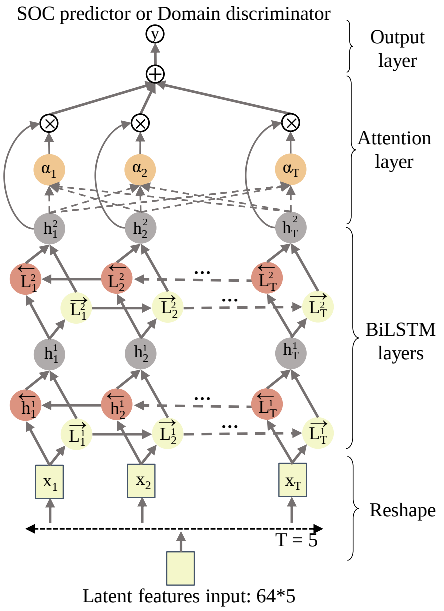

Fig. 5. The network architecture of the SOC predictor.

The BiLSTM is a variant of RNN designed to better capture and understand long-term dependencies in sequential data by processing input sequences in both forward and backward directions simultaneously [49]. Specifically, BiLSTM consists of two directions of LSTM, one processing the input sequence in the forward direction and the other in the backward direction. This allows the network to consider past and future information at each time step, aiding in capturing patterns and dependencies within the sequence. Meanwhile, to comprehensively capture contextual information and focus on different parts of the input sequence, the output of BiLSTM is then connected to the attention mechanism as shown in Fig. 5. The introduction of the attention mechanism could dynamically assign different weights to different parts of the input sequence, allowing the model to focus more intently on important information [50]. Specifically, the attention mechanism calculates weights for each input position, which are then used for the weighted summation of the BiLSTM outputs, and the final sequence representation could be gained. Note that there is a data shape processing operation before the latent feature is fed into the BiLSTM layer, which could convert feature maps obtained from the feature encoder into a sequence format for input into subsequent BiLSTM layers.

3) Adversarial learning: In the as-proposed framework, adversarial learning aims to minimize the distributional discrepancy between the source domain and the target domain by introducing adversarial loss $\mathcal{L}_{\text{adv}}$. This process could achieve feature alignment across domains and thus facilitate better model generalization ability on the target domain [51, 52]. Specifically, an adversarial network consists of two components: feature encoder f_e and domain discriminator f_{dis}. Feature encoder is responsible for offering the common features for domain discriminator. In contrast, the domain discriminator attempts to classify whether the extracted features originate from the source domain or the target domain. A game-like scenario is established between the feature encoder and the domain discriminator in the adversarial training process. The feature encoder's objective is to extract indistinguishable latent features for the domain discriminator, and the domain discriminator's goal is to accurately distinguish whether the input features originate from the source domain or the target domain. Thus, the optimization objective of the adversarial training between the domain discriminator and the feature encoder can be calculated as follows [52]:

$$\begin{aligned}\min_{f_e^t}\max_{f_{dis}}\mathcal{L}_{\text{adv}} = -\mathbb{E}_{x^s\in\mathcal{D}^s}\log\left[f_{dis}\left(f_e^s\left(x^s\right)\right)\right] - \\ \mathbb{E}_{x^t\in\mathcal{D}^{t,train}}\log\left[1-f_{dis}\left(f_e^t\left(x^t\right)\right)\right]\end{aligned} \tag{7}$$

As shown in Table I, there is nearly no difference between the network structure of the SOC predictor and domain discriminator, except the last output layer, because the output of the domain discriminator is a binary classification problem.

4) Feature alignment metric: There are some defects in adversarial learning. Typically, even though the domain discriminator is successfully confused, the learned feature representation might be domain-invariant [38]. To deal with this problem, as a typical unsupervised feature alignment metric, an MK-MMD module is introduced into this framework to quantify the mean discrepancy between the extracted features from different domains. By minimizing the MK-MMD loss $\mathcal{L}_{\text{mmd}}$, the model is compelled to find a feature representation that is insensitive to domain differences [53]. Thus, this module encourages the model to learn domain-invariant features and helps improve the model's generalization performance on the target domain. Specifically, $\mathcal{L}_{\text{mmd}}$ is the squared distance of F^s and $F^{t,train}$ embedded in the reproducible kernel Hilbert space (RKHS), which can be expressed as [54],

$$\begin{aligned}\mathcal{L}_{\text{mmd}} &= \|\frac{1}{m_s}\sum_{i=1}^{m_s}\Phi(x_i^s) - \frac{1}{m_{t,train}}\sum_{j=1}^{m_{t,train}}\Phi(x_j^{t,train})\|_{\mathcal{H}}^2 \\ &= \frac{1}{m_s^2}\sum_{i=1}^{m_s}\sum_{j=1}^{m_s}\mathcal{K}\left(f_i^s, f_j^s\right) + \\ &\frac{1}{m_t^2}\sum_{i=1}^{m_{t,train}}\sum_{j=1}^{m_{t,train}}\mathcal{K}\left(f_i^{t,train}, f_j^{t,train}\right) - \\ &\frac{2}{m_s m_{t,train}}\sum_{i=1}^{m_s}\sum_{j=1}^{m_t}\mathcal{K}\left(f_i^s, f_j^{t,train}\right)\end{aligned} \tag{8}$$

where $\|\|_{\mathcal{H}}^2$ is the two-norm operation in RKHS, $\Phi(\cdot)$ is the nonlinear mapping function from raw data space to RKHS. f_i^s and $f_i^{t,train}$ are the subspace's feature learned from the x_i^s

Algorithm 1 Pseudo Code of the Pre-trained Process

```
Input: $\mathcal{D}^s = \{x_i^s, y_i^s\}_{i=1}^{m_s}$: Labeled source training data; $epoch_{thresh}^{pre}$: Pre-trained epochs
Output: $f_e^s$: Source feature encoder, $f_{pred}$: SOC predictor
for $epoch \leftarrow 0$ to $epoch_{thresh}^{pre}$ do
    Estimate SOC $\hat{y}^s$ using $f_e^s$ and $f_{pred}$ modules (Eq.3 and Eq.5);
    Compute the estimation loss $\mathcal{L}_{\text{pred}}$ (Eq.6);
    Update $f_e^s$ and $f_{pred}$ (Eq.11);
    $epoch \leftarrow epoch + 1$;
end for
```

and $x_i^{t,train}$, respectively. $\mathcal{K}$ donates the combination of the Gaussian radial basis function (RBF) kernels as follows [55]:

$$\mathcal{K}(x_i, y_j) = \sum_{n=1}^{N_k} k_n(x_i, y_j) \tag{9}$$

where k_n denotes the n-th RBF kernel and then it can be expressed as,

$$k_n(x_i, y_j) = exp(-\|x_i - y_j\|^2 / 2\gamma_n^2) \tag{10}$$

where γ_n^2 donates bandwidth parameter.

D. Pre-trained loss optimization

The learning object of the pre-trained step is to gain a well-trained SOC source estimator. Hence, by minimizing $\mathcal{L}_{\text{pred}}$, the weight parameters of the SOC source estimator can be updated, and the detailed model parameter update process is displayed as follows:

$$\begin{cases} \theta_{f_e^s} = \theta_{f_e^s} - \eta_{pt} \cdot \dfrac{\partial \mathcal{L}_{\text{pred}}}{\partial \theta_{f_e^s}} \\ \\ \theta_{f_{pred}} = \theta_{f_{pred}} - \eta_{pt} \cdot \dfrac{\partial \mathcal{L}_{\text{pred}}}{\partial \theta_{f_{pred}}} \end{cases} \tag{11}$$

where $\theta_{f_e^s}$, and $\theta_{f_{pred}}$ represent the parameters of the feature encoder for source domain, and SOC predictor, respectively; η_{pt} represents the learning rate during the pre-trained step. When the pre-trained step is finished, the feature encoder for source domain f_e^s is utilized in the next transfer step, and the f_{pred} is transferred to the SOC online estimation process. The pseudo code of the pre-trained process is shown in Algorithm 1.

E. Transferring total loss optimization

The optimization objective $\mathcal{L}_{\text{total}}$ of the DARM framework consists of the adversarial loss $\mathcal{L}_{\text{adv}}$, the reconstruction loss $\mathcal{L}_{\text{rec}}$, and the MK-MMD loss $\mathcal{L}_{\text{mmd}}$. The overall loss function is formulated as,

Algorithm 2 Pseudo Code of the Transfer Process

```
Input: $\mathcal{D}^s = \{x_i^s, y_i^s\}_{i=1}^{m_s}$: Labeled source training data; $\mathcal{D}^{t,train} = \{x_i^{t,train}\}_{i=1}^{m_{t,train}}$: Unlabeled target training data; $epoch_{thresh}^{tf}$: Transfer epochs; $f_e^s$: Source feature encoder from the pre-trained stage; $f_{pred}$: SOC predictor from the pre-trained stage
Output: $f_T$: Transferred target model
Initialize target feature extractor $f_e^t$ with pre-trained source feature extractor $f_e^s$
for $epoch \leftarrow 0$ to $epoch_{thresh}^{tf}$ do
    Extract domain-invariant feature ($F^s$ and $F^{t,train}$) from $\mathcal{D}^s$ and $\mathcal{D}^{t,train}$ (Eq.3);
    Compute the MMD loss $\mathcal{L}_{\text{mmd}}$ (Eq.8);
    Compute the adversarial training loss $\mathcal{L}_{\text{adv}}$ (Eq.7);
    Compute the reconstruction loss $\mathcal{L}_{\text{rec}}$ (Eq.4);
    Compute the total loss $\mathcal{L}_{\text{rec}}$ (Eq.2);
    Update $f_e^t$, $f_{re}$, and $f_{dis}$ (Eq.12);
    $epoch \leftarrow epoch + 1$;
end for
Combine target feature extractor $f_e^t$ with pre-trained SOC predictor $f_{pred}$ to gain transferred target model $f_T$
```

$$\begin{cases} \theta_{f_e^t} = \theta_{f_e^t} - \eta_{tf} \cdot \left(\dfrac{\partial \mathcal{L}_{\text{adv}}}{\partial \theta_{f_e^t}} + \alpha \cdot \dfrac{\partial \mathcal{L}_{\text{rec}}}{\partial \theta_{f_e^t}} + \beta \cdot \dfrac{\partial \mathcal{L}_{\text{mmd}}}{\partial \theta_{f_e^t}} \right) \\ \\ \theta_{f_{re}} = \theta_{f_{re}} - \eta_{tf} \cdot \dfrac{\partial \mathcal{L}_{\text{rec}}}{\partial \theta_{f_{re}}} \\ \\ \theta_{f_{dis}} = \theta_{f_{dis}} - \eta_{tf} \cdot \dfrac{\partial \mathcal{L}_{\text{adv}}}{\partial \theta_{f_{dis}}} \end{cases} \tag{12}$$

where $\theta_{f_e^t}$, $\theta_{f_{re}}$, and $\theta_{f_{dis}}$ represent the parameters of the feature encoder for the target domain, reconstructed decoder, and domain discriminator, respectively; η_{tf} denotes the learning rate during the TL process. When the TL step is finished, the feature encoder for target domain f_e^t and the well-trained SOC predictor in pre-trained step f_{pred} are combined for the online battery's SOC estimation. The pseudo code of the pre-trained process is shown in Algorithm 2.

III. Experimental results

A. Dataset description

To evaluate the battery SOC estimation performance of the DARM framework at real-world working conditions and compare this approach with others, the Panasonic 18650PF dataset collected by the University of Wisconsin-Madison [56], is utilized to train and validate the proposed model architecture. In this dataset, a brand-new Panasonic 18650PF cell is tested in the thermal chamber with a Digatron firing circuits universal battery tester channel, and other detailed description is displayed in Table II. The Panasonic 18650PF dataset is a benchmark dataset in the community. In this

TABLE II
THE SPECIFICATIONS OF THE PANASONIC 18650PF DATASET.

Characteristics	Values
Cathode material	$LiNiCoAlO_2$
Anode material	Graphite
Nominal voltage	3.6 V
Minimum/typical capacity	2.75 Ah / 2.9 Ah
Min/max voltage	2.5 V / 4.2 V
Mass/energy storage	48 g / 9.9 Wh
Charge temperature	10 °C to 45 °C
Discharge temperature	-20 °C to 60 °C

TABLE III
EXPERIMENTAL HYPERPARAMETERS.

Parameters	Value	Parameters	Value
α	10	Optimizer	Adam
β	1	Learning rate for pre-trained step	0.001
Pre-trained epochs	500	Learning rate for transfer step	0.0001
Transfer epochs	100	Batch size	256
Early stop patience	20		

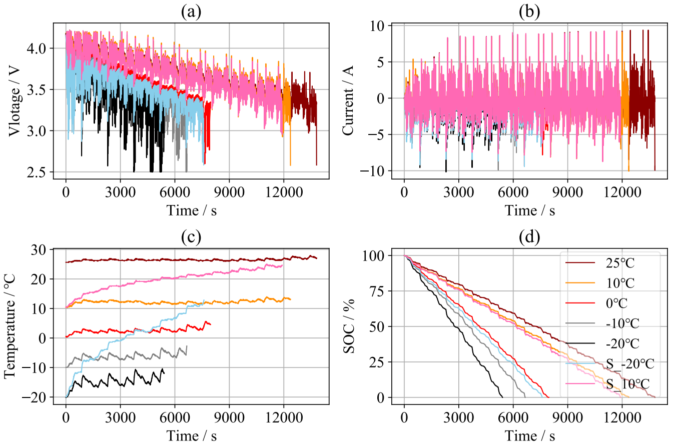

Fig. 6. Voltage (a), current (b), temperature (c), and SOC value (d) of LA92 drive cycle in raw measured at five discrete ambient temperatures and two increasing ambient temperatures.

dataset, a series of power profiles (Cycle1, Cycle2, Cycle3, Cycle4, US06, HWFET, UDDS, LA92, and neural network (NN)) were tested repeatedly at multiple pre-set fixed ambient temperatures (25 °C, 10 °C, 0 °C, -10 °C, -20 °C), and increasing ambient temperatures (starting at -20 °C and 10 °C), which is donated as S_-20 °C and S_10 °C, respectively. Cycles 1-4 are a random mix of US06, HWFET, UDDS, LA92, and NN drive cycles. NN drive cycle consists of portions of the US06 and LA92 drive cycles. The drive cycle power profile is calculated for an electric Ford F-150 truck with a 35-kWh battery pack scaled for the single cell.

The dataset mainly includes timestamp, voltage, current, ampere-hours, and cell temperature, recorded at a 10 Hz sampling rate. Based on the previous investigation [57, 58], the collected data were resampled at the sampling rate (1 Hz) commonly employed by BMS. Fig. 6 displays the collected parameters and SOC values of the LA92 driving cycle tested at five discrete temperatures.

B. Implementation detail

All approaches are trained and tested for subsequent experiments with datasets resampled at 1 Hz. Voltage, current, and cell temperature input are provided within a window of 40 time steps. To facilitate the model's training, the Min-Max normalization transforms these measurements to the range [0, 1]. The proposed SOC estimator is compared with the commonly used CNN, LSTM, and CNN-LSTM models, and their parameter numbers are set to be close to that of the proposed model for fairness. Adam optimizer is employed in the training and TL process. Via the grid-search method, the trade-off parameter in Eq.12 can be quickly obtained, and other hyperparameters values are shown in Table III. All approaches are trained and tested using PyTorch 1.11.0 with NVIDIA RTX 3080 GPU. We demonstrate the quantitative validation of the proposed model using the following two well-known metrics: root mean square error (RMSE) and mean absolute error (MAE), which can be expressed as,

$$\begin{cases} RMSE = \sqrt{\dfrac{1}{n}\displaystyle\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2} \\ \\ MAE = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n}\left|\hat{y}_i - y_i\right| \end{cases} \tag{13}$$

where n refers to the number of input samples; y_i and $\hat{y}_i$ are the real value and estimated SOC value, respectively.

C. SOC estimation performance of the source estimator

To evaluate the SOC estimation performance of the pre-trained estimator at specific ambient temperature, the measured data (including voltage, current, temperature, and capacity) on cycles 1-4 at 25 °C serve as the training dataset for the pre-trained step, and those of the remaining drive cycle profiles at 25 °C serve as the test dataset. Fig. 7 and Table IV display the SOC estimation results (LA92, UDSS, US06, HWFET, and NN at 25 °C) based on the proposed source estimator and the other commonly used models. Compared with the other models, the prediction value from the proposed estimator is closer to the ground truth at the various dynamic drive cycles, and the proposed SOC estimator exhibits better SOC estimation performance at the same ambient temperature. Therefore, the proposed SOC estimator could extract the latent features of the source data effectively, which has a potential generalization ability for the online battery at different ambient temperatures.

D. Transferring SOC estimator to different working conditions via the DARM framework

The ambient temperature has a significant influence on the SOC estimation. An increase in ambient temperature promotes the chemical reaction rate on the electrodes, while lower temperatures slow down the reaction rate. This leads to faster charging and discharging of the battery at higher temperatures and slower reactions at lower temperatures. Hence there is domain drift across various temperatures for SOC estimation,

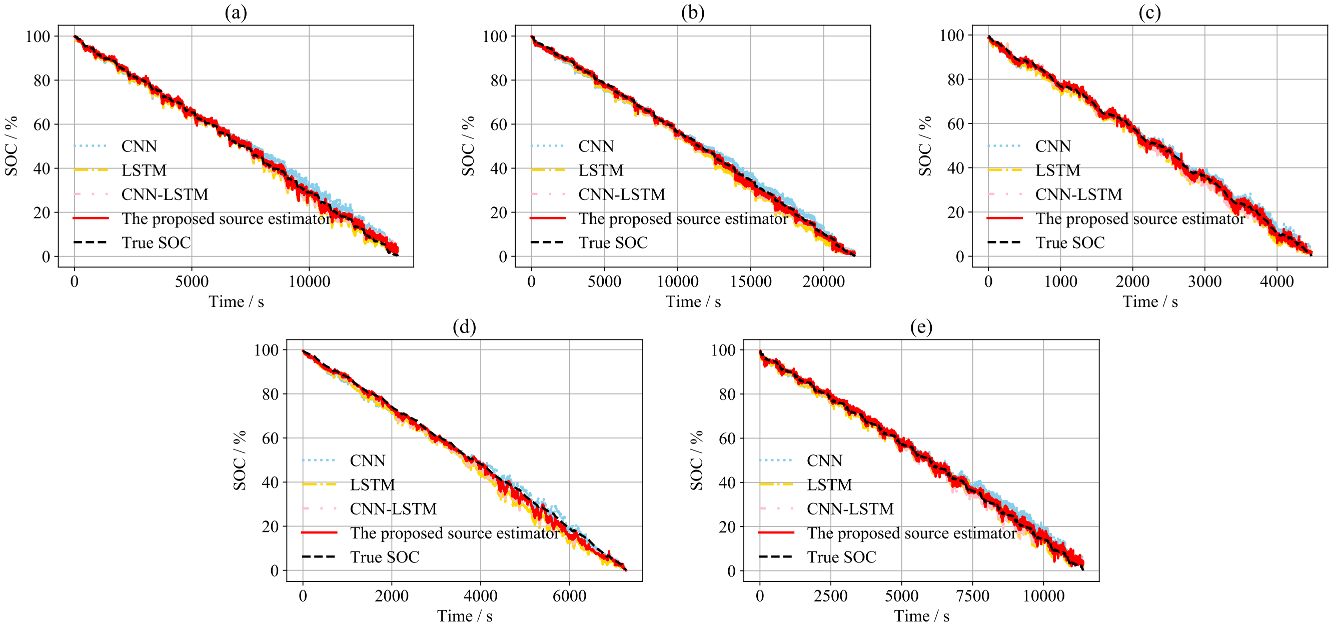

Fig. 7. SOC estimation results from various models at 25 °C. (a) LA92, (b) UDSS, (c) US06, (d) HWFET, and (e) NN.

TABLE IV
SOC ESTIMATION RESULTS FROM THE VARIOUS MODELS AT 25 °C.

Models	RMSE / %					MAE / %				
	LA92	UDSS	US06	HWFET	NN	LA92	UDSS	US06	HWFET	NN
CNN	2.55	1.76	1.72	2.02	2.43	2.16	1.52	1.40	1.71	1.85
LSTM	1.64	2.30	2.06	3.81	1.34	1.28	2.00	1.75	3.34	1.07
CNN-LSTM	1.56	1.57	1.84	2.61	1.49	1.14	1.24	1.46	2.08	1.11
The proposed source estimator	1.33	1.00	1.24	1.85	1.22	1.02	0.74	0.97	1.37	0.94

and the TL strategy needs to be carried out to eliminate the distributional difference across domains. To evaluate the TL performance of the DARM framework, the pre-trained SOC estimator at 25 °C is transferred to the other data at different temperatures (10 °C, 0 °C, -10 °C, and -20 °C). During the TL process, the data of Cycle 1 (including voltage, current, and temperature, except capacity) at target ambient temperature serve as the $\mathcal{D}^{t,train}$, and the data of five driving cycles (including LA92, UDSS, US06, HWFET, and NN) at target ambient temperature serve as the $\mathcal{D}^{t,test}$.

To better display the UDA performance of the proposed DARM framework, the fine-tuning approach as a typical supervised TL strategy is utilized to be a comparison, in which the transferred pre-trained model is the SOC source estimator at 25 °C and all the modules and the hyperparameters are similar to the proposed framework for fairness. Taking the LA92 power profile as an example, Fig. 8 displays SOC transferring results at different target temperatures, and the metrics of all power profiles are shown in Table V. Meanwhile, the direct prediction performance of the source estimator is added to reflect the improvement of SOC estimation accuracy with the help of the TL process.

Compared with the SOC estimation results of the TL strategies, those of the source estimator without TL are further away from the ground truth at each target temperature, indicating the source estimator is unsuitable to be applied directly to the SOC estimation at different temperatures due to domain drift. Meanwhile, this shift becomes even more violent when ambient temperature gradually decreases, implying the gradually increasing distribution differences between the source domain and the target domain; Although the ground truth of the Cycle 1 data is known for the fine-tuning strategy, its SOC estimation performance is inferior than the DARM framework. On the one hand, there might be an overfitting problem in that the model too overly focuses on the target domain's specific features to decrease its generalization ability. On the other hand, there are still unmitigable domain drift problems due to the large distribution discrepancy across domains; Although the utilized Cycle 1 data is without SOC true values, the estimation results of the proposed framework are closer to the ground truth. Almost the RMSE and MAE are controlled under 2.0% and 1.5% for fixed temperatures, respectively, and the best experimental result (RMSE: 1.33%, bold in Table V) could be gained when transferring from 25 °C to 10 °C. Meanwhile, the errors have slight increases for the increasing ambient temperature, the best experimental result (RMSE: 2.57%, bold in Table V) could also be found when transferring from 25 °C to S_-20 °C. This result implies the proposed DARM framework could transfer effectively source SOC estimator to the unlabeled and limited target domain by extracting domain-invariant features and decreasing the distances between the different features.

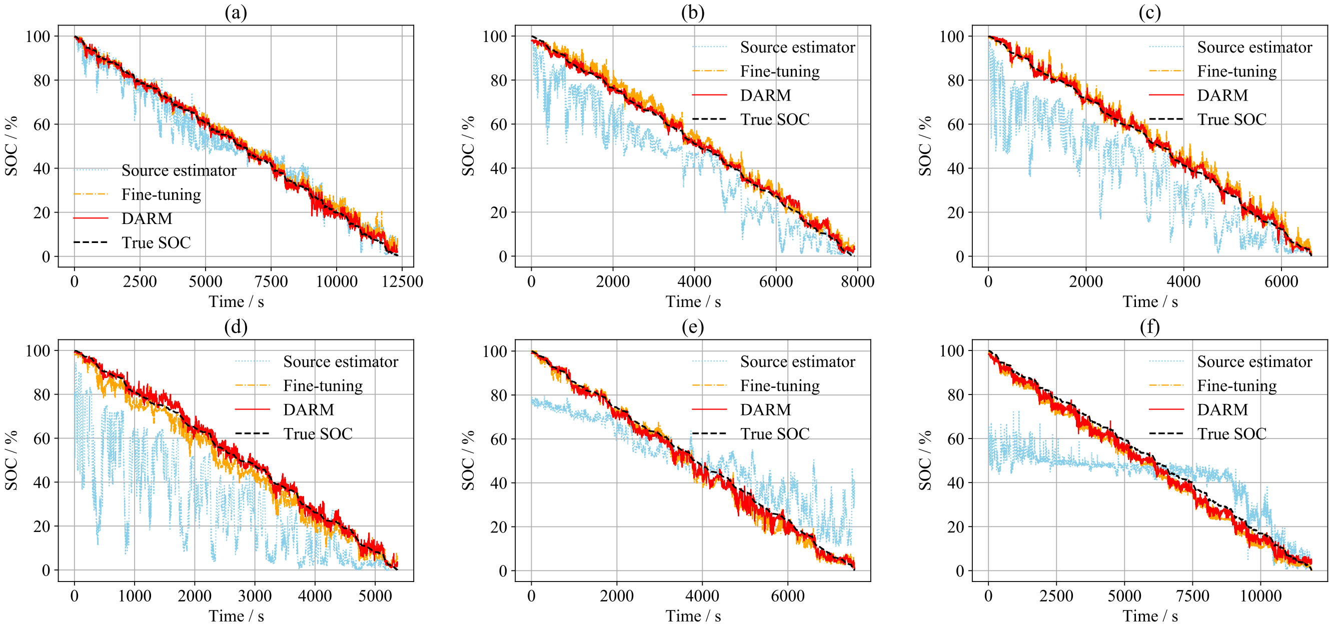

Fig. 8. Taking the LA92 power profile as an example, the SOC estimation results of the source estimator, the fine-tuning strategy, and the DARM framework at different ambient temperatures. (a) 10 °C, (b) 0 °C, (c) -10 °C, (d) -20 °C, (e) S_-20 °C, and (e) S_10 °C.

TABLE V
SOC ESTIMATION RESULTS OF THE SOURCE ESTIMATOR, THE FINE-TUNING STRATEGY, AND THE DARM FRAMEWORK AT DIFFERENT TARGET TEMPERATURES.

Target temperatures	Methods	RMSE / %					Average	MAE/ %					Average
		LA92	UDSS	US06	HWFET	NN	RMSE / %	LA92	UDSS	US06	HWFET	NN	MAE / %
10 °C	Source estimator	5.20	6.26	3.61	8.95	3.77	5.56	3.97	4.65	2.89	7.84	2.94	4.46
	Fine-tuning	2.25	2.61	1.26	1.86	1.76	1.95	1.70	1.96	0.98	1.44	1.34	1.48
	DARM	1.42	1.34	1.27	1.51	1.12	**1.33**	1.05	0.97	0.99	1.12	0.81	0.99
0 °C	Source estimator	12.21	9.13	15.53	14.15	11.95	12.59	10.30	7.08	14.48	12.98	10.27	11.02
	Fine-tuning	2.99	3.18	3.36	2.59	3.79	3.18	2.50	2.85	2.80	2.20	3.32	2.73
	DARM	1.68	1.29	1.46	1.59	1.33	1.47	1.28	1.03	1.15	1.18	1.03	1.13
-10 °C	Source estimator	18.75	15.25	28.54	22.77	18.95	20.85	15.66	12.08	25.63	20.38	15.56	17.86
	Fine-tuning	3.23	3.38	3.53	3.08	3.77	3.40	2.52	2.73	2.82	2.60	3.12	2.76
	DARM	1.78	1.49	1.42	1.66	1.73	1.62	1.41	1.21	1.14	1.33	1.42	1.30
-20 °C	Source estimator	28.57	23.82	38.78	35.99	27.64	30.96	23.68	19.79	33.13	32.17	21.83	26.12
	Fine-tuning	4.66	4.38	3.02	4.95	3.26	4.05	4.00	3.83	2.52	4.22	2.63	3.44
	DARM	2.04	1.59	1.60	1.79	1.78	1.76	1.58	1.27	1.25	1.34	1.36	1.36
S_-20 °C	Source estimator	12.43	18.14	23.16	20.23	19.44	18.68	10.47	16.91	18.81	16.06	14.99	15.45
	Fine-tuning	3.77	3.82	3.53	3.30	4.58	3.80	2.84	3.04	2.66	2.66	3.38	2.92
	DARM	2.82	2.17	2.23	2.48	3.15	**2.57**	2.12	1.79	1.63	1.63	2.17	1.93
S_10 °C	Source estimator	20.43	19.61	25.27	17.08	-[a]	20.60	16.02	15.28	20.77	10.11	-	15.54
	Fine-tuning	4.54	4.62	3.96	3.91	-	4.26	4.14	4.17	3.61	3.39	-	3.83
	DARM	3.29	3.11	3.28	2.87	-	3.14	2.98	2.85	3.02	2.32	-	2.79

[a] the relative raw data is not found.

E. Ablation experiments

To study each module's effect on the DARM framework in terms of transferring performance, the ablation experiments are carried out by gradually appending each module's loss function into the total loss function. As shown in Table III-E, the average RMSE gradually decreases with the addition of modules into the framework, especially when all modules are added to the proposed framework. This result indicates that every module effectively enhances the SOC estimation accuracy during the TL process. The model gradually finds the optimal solution during the TL process through the constraints of multiple loss functions.

F. Visualization of extracted feature distributions

The source estimator automatically extracts features containing meaningful information about battery SOC from the input data (including voltage, current, and temperature), and maps them to the battery SOC. However, using the extracted feature from the feature encoder directly to explain the estimation mechanism is hard because of the high-dimensional feature space. A visual analysis of the original data and extracted features is conducted to investigate the domain adaptation effect of the DARM framework on feature extraction. The high-level feature extracted feature from the feature encoder is gained by principal component analysis (PCA), where the first three principal components (PCs 1 to 3) are employed. The kernel density estimation (KDE) [59] is used to calculate

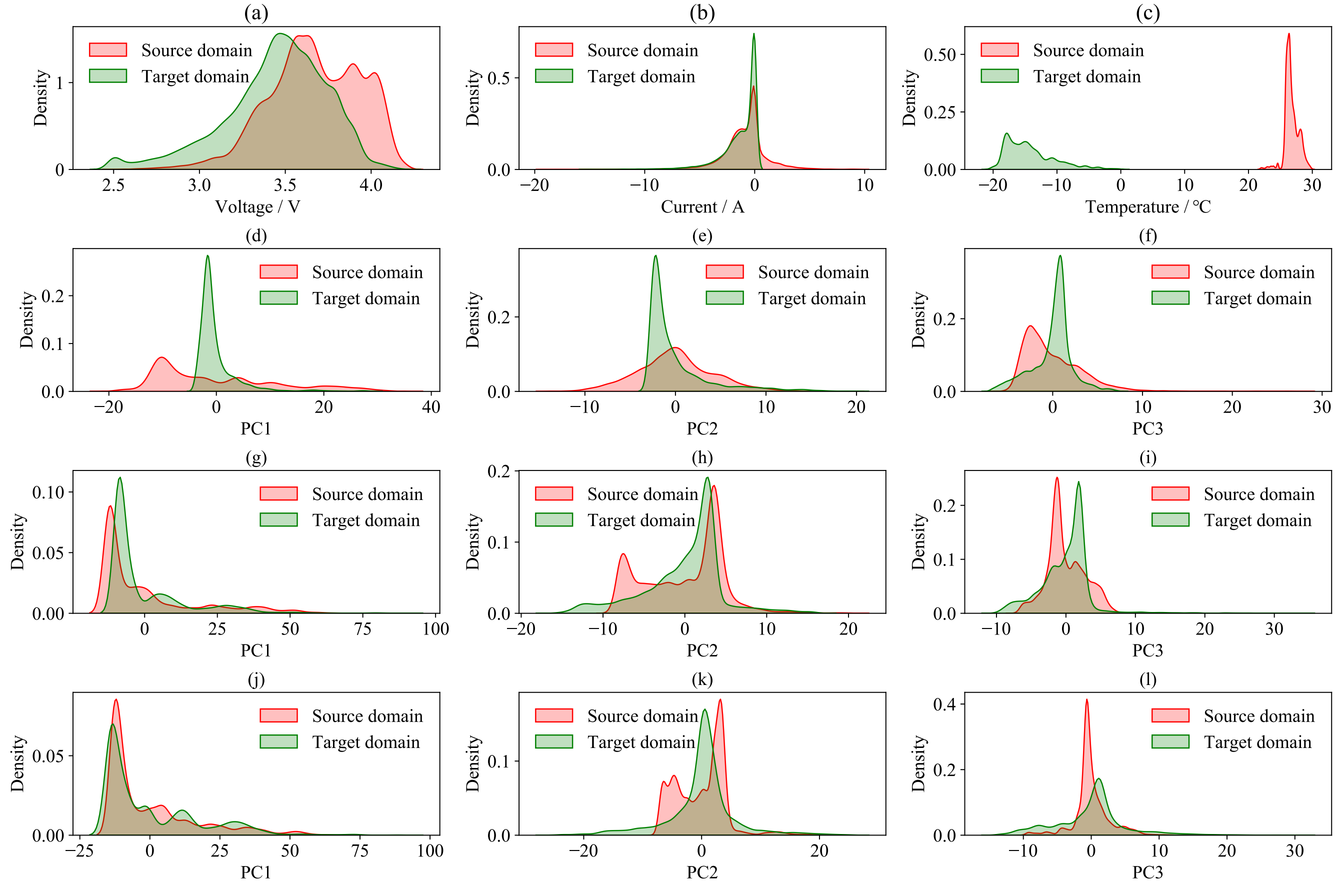

Fig. 9. Visualizations of the selected two domains ($\mathcal{D}^s$, 25 °C; $\mathcal{D}^t$, -20 °C) and extracted feature distributions using PCA: Data distributions across domain (a, voltage; b, current; c, temperature); Extracted feature distributions via the source estimator (d, PC1; e, PC2; f, PC3), the fine-tuning (g, PC1; h, PC2; i, PC3), and the DARM framework (j, PC1; k, PC2; l, PC3), respectively.

TABLE VI
RESULTS OF ABLATION EXPERIMENTS. GRADUALLY APPENDING EACH LOSS FUNCTION TO THE DARM FRAMEWORK TO TESTIFY EACH MODULE'S EFFECT.

Transferring direction	Average RMSE of different module combination / %			
	D_1[a]	D_2[b]	D_3[c]	D_4[d]
25 °C to 10 °C	1.90	1.57	1.63	1.33
25 °C to 0 °C	2.15	2.06	1.84	1.47
25 °C to -10 °C	2.23	1.92	1.89	1.62
25 °C to -20 °C	2.70	2.40	2.09	1.76

[a] the DARM framework without reconstruction and MK-MMD modules.
[b] the DARM framework without MK-MMD modules.
[c] the DARM framework without reconstruction modules.
[d] the DARM framework.

the probability density distribution of the original input and the extracted features based on different methods (source estimator, fine-tuning, and DARM framework), as shown in Fig. 9.

Figs. 9a to c display a significant difference in the voltage, current, and temperature data across domains because the different ambient temperatures lead to distinct electrochemical reaction rates and inner resistances in a cell. The domain discrepancy would cause a huge challenge for directly applying source domain knowledge to the target domain. As shown in Figs. 9d to f, there is a remarkable domain discrepancy of extracted features via source estimator, so the TL strategy is necessary. As shown in Figs. 9g to l, all the feature distribution discrepancies via the fine-tuning and the DARM framework across domains of PCs 1 to 3 have been improved. Importantly, the proposed framework exhibits the minimum domain discrepancy, implying this framework could ensure the feature encoder extracts the domain-invariant features more effectively. Therefore, the superior SOC estimation performance for different target temperatures could be gained via the proposed framework, which is the same as Section III-D results.

G. Generalization validation

TABLE VII
TRANSFER RESULTS FROM THE WISCONSIN-MADISON DATASETS TO THE MCMASTER DATASETS.

Target temperatures / °C	Average RMSE / %	Average MAE / %
40	1.66	1.38
25	**1.45**	1.07
10	1.60	1.22
0	1.62	1.33
-10	1.74	1.37
-20	1.96	1.47

TABLE VIII
COMPARISON WITH OTHER METHODS OF SOC DOMAIN ADAPTATION RESULTS.

Methods	TL strategies	Validation datasets	Transfer settings	Best results		Computation complexities during transfer process		
				RMSE / %	MAE / %	Transfer time	Estimation time	Parameters
DCNN-TL[61]	Supervised	NASA datasets[a]	Cross battery	1.36	-[b]	12.29 s	5 ms	0.03 M
DTNN[39]	Supervised	Wisconsin-Madison, A123 and INR datasets[c]	Cross battery	1.54	1.21	411.54 s	2 ms	-
SFTTN[38]	Unsupervised	Wisconsin-Madison, A123, and McMaster datasets	Cross fixed temperature Cross changing temperature Cross battery	3.38 6.46 2.06	2.67 4.36 1.98	38.12 s/epoch	2.35 ms	0.95 M
WD-TL[41]	Unsupervised	Wisconsin-Madison datasets	Cross fixed temperature	1.46	1.18	603.68 s	5.63 ms	0.06 M
TATN[40]	Unsupervised	Wisconsin-Madison datasets	Cross fixed temperature	1.50	1.15	-	-	-
The proposed framework	Unsupervised	Wisconsin-Madison and McMaster datasets	Cross fixed temperature Cross changing temperature Cross battery	1.33 2.57 1.45	0.99 1.87 1.07	390.51 s	11.08 ms	0.19 M

[a] NASA datasets obtains from the NASA.
[b] the symbol ”-” denotes ”unavailable”.
[c] A123 and INR datasets obtain from the Center for Advanced Life Cycle Engineering at the University of Maryland.

Because the differences in internal properties between the different types of batteries also would lead to the domain shift when applying the SOC source estimator, the experimental LIBs datasets coming from McMaster University [60] are used in this study. A brand-new 3Ah LG 18650HG2 battery in which cathodic and anode materials are $LiNiMnCoO_2$ and graphite respectively, is tested in a Digatron battery tester, and the ambient temperature is controlled by a thermal chamber. These datasets contain a series of power profiles (US06, HWFET, UDDS, LA92, and Mixed 1-8) were tested repeatedly at multiple preset temperatures (40 °C, 25 °C, 10 °C, 0 °C, -10 °C, and -20 °C), in which Mixed 1 to 8 drive cycles are randomly generated from the standard drive cycles. To display the adaptability and effectiveness of the framework, the SOC source estimator from the Wisconsin-Madison datasets is transferred to the McMaster datasets, in which the source and target domains during the transfer stage are Cycle 1-4 of the Wisconsin-Madison datasets at 25 °C, and Mixed 1-8 of the McMaster datasets at 6 types of ambient temperatures, respectively. The cross-battery transfer result is displayed in Table VII, it is clear that the DARM framework exhibits excellent performance for all the transferred tasks with an average RMSE error of no more than 2%, and the best experimental result (RMSE: 1.45%, bold in Table VII) could be gained when transferring between the 25 °C of the two datasets. This result indicates that the proposed UDA framework possesses outstanding generalization ability, and it would decrease the data demand of the source domain and thus avoid the training process of the source estimator from scratch.

H. Comparison with other methods

Table VIII summarizes the comparison between the transfer results and computation complexities obtained with the proposed method and other transfer methods. In the field of temperature and battery-type domains, the proposed framework could realize outstanding transfer results compared with other state-of-the-art supervised and unsupervised methods. Meanwhile, the training parameter amounts during the transfer process are relatively small, and the transfer and estimation time are within a receptible range by contrast. Therefore, the proposed framework has a scalability advantage on the premise of ensuring accuracy. As shown in Table IX, the proposed method also exhibits excellent SOC estimation performances compared with the other common methods for SOC estimation in electric vehicle (EV) applications. Therefore, this method can potentially be applied to EV SOC estimation.

TABLE IX
COMPARISON WITH OTHER METHODS OF SOC ESTIMATION IN EV APPLICATIONS.

Methods	Average results	
	RMSE / %	MAE / %
Gaussian process regression[62]	-[a]	3.8
DNN[63]	3.68	9.31
Unscented Kalman filtering-based method[64]	≤4	-
The proposed method	≤3.14	≤2.79

[a] the symbol ”-” denotes ”unavailable”.

IV. CONCLUSION

This work proposes a novel TL-based DARM framework for LIBs SOC estimation to handle the domain shift problem caused by changing conditions. A new SOC estimator consisting of DWS 1D CNN, residual connection, BiLSTM, and attention mechanism, is constructed and exhibits the superior SOC estimation performance for offline source dataset compared with commonly used models. An adversarial framework introduces a domain discriminator to extract the domain-invariant features. Moreover, a reconstruction constraint is developed to ensure the feature encoder extracts more valuable information, and the reconstructed decoder and the feature encoder also form a U-shaped structure by skip connection. An MK-MMD module, a typical unsupervised feature alignment metric, is also introduced to decrease the distribution discrepancy of extracted features across domains. Experimental results demonstrate that the best cross-domain RMSE of the proposed transfer framework are 1.33%, 2.57 %, and 1.45 % for fixed ambient temperatures, changing ambient temperatures, and changing battery type, respectively, indicating that the DARM framework can achieve outstanding cross-domain estimation performances compared with the other state-of-the-art transfer methods. Meanwhile, the ablation experiment implies that all the constraints of this framework have a positive effect on enhancing the transferring performance of the source estimator. The KDE result further testifies to this framework's effective domain-invariant feature extraction ability. Therefore, this framework exhibits a potential application for the SOC estimation of online LIBs under changing conditions.

However, there are still some limitations of this work or challenges for using in the real world as follows: First, in more cases, the batteries provide power in the form of battery packs, such as electric vehicles and energy storage power stations. That means the collected data will be dirtier and more complex than the used data in this study; Second, the collected temperature of one pack is the temperature in a certain region, not the temperature of the specific cell, thus the temperature domain is difficult to classify. Besides, the deficiency of temperature parameters might have a negative influence on the efficiencies and scalability of the transfer framework; Third, when the multi-variable coupled condition (such as working condition, aging state, ambient temperature, etc.) changes simultaneously and dynamically in practice, the domain adaptation processes proceed all the time. It will challenge the computational ability and data collection. Therefore, for these limitations and challenges, we will continue to study the efficiency and scalability of this transfer framework for the practical battery pack. Meanwhile, the domain shift caused by multi-variable coupled conditions will be studied for the transferring ability of the proposed framework.

In the future, the efficiency and scalability of the proposed transfer framework for the practical battery packs will be studied. Meanwhile, the domain shift caused by multi-variable coupled conditions will also be investigated for the transferring ability of the proposed framework.

REFERENCES

[1] Y. Liu, B. Hou, M. Ahmed, Z. Mao, J. Feng, and Z. Chen, "A hybrid deep learning approach for remaining useful life prediction of lithium-ion batteries based on discharging fragments," *Appl. Energy*, vol. 358, p. 122555, 2024.

[2] J. He and L. Wu, "Cross-conditions capacity estimation of lithium-ion battery with constrained adversarial domain adaptation," *Energy*, vol. 277, p. 127559, 2023.

[3] G. Ma, Y. Zhang, C. Cheng, B. Zhou, P. Hu, and Y. Yuan, "Remaining useful life prediction of lithium-ion batteries based on false nearest neighbors and a hybrid neural network," *Appl. Energy*, vol. 253, p. 113626, 2019.

[4] Y. Che, S. B. Vilsen, J. Meng, X. Sui, and R. Teodorescu, "Battery health prognostic with sensor-free differential temperature voltammetry reconstruction and capacity estimation based on multi-domain adaptation," *Etransportation*, vol. 17, p. 100245, 2023.

[5] J. Tian, R. Xiong, W. Shen, and J. Lu, "State-of-charge estimation of LiFePO$_4$ batteries in electric vehicles: A deep-learning enabled approach," *Appl. Energy*, vol. 291, p. 116812, 2021.

[6] J. Tian, R. Xiong, J. Lu, C. Chen, and W. Shen, "Battery state-of-charge estimation amid dynamic usage with physics-informed deep learning," *Energy Stor. Mater.*, vol. 50, pp. 718–729, 2022.

[7] Y. Zheng, M. Ouyang, X. Han, L. Lu, and J. Li, "Investigating the error sources of the online state of charge estimation methods for lithium-ion batteries in electric vehicles," *J. Power Sources*, vol. 377, pp. 161–188, 2018.

[8] K. Movassagh, S. A. Raihan, and B. Balasingam, "Performance analysis of coulomb counting approach for state of charge estimation," in *2019 IEEE Electr. Power Energy Conf. (EPEC)*. IEEE, 2019, pp. 1–6.

[9] J. Channegowda, V. Maiya, and C. Lingaraj, "A deep learning approach towards generating high-fidelity diverse synthetic battery datasets," *IEEE Trans. Ind. Appl.*, vol. 59, no. 4, pp. 4620–4627, 2023.

[10] A. Tang, Y. Huang, S. Liu, Q. Yu, W. Shen, and R. Xiong, "A novel lithium-ion battery state of charge estimation method based on the fusion of neural network and equivalent circuit models," *Appl. Energy*, vol. 348, p. 121578, 2023.

[11] C. Hu, L. Ma, S. Guo, G. Guo, and Z. Han, "Deep learning enabled state-of-charge estimation of LiFePO$_4$ batteries: A systematic validation on state-of-the-art charging protocols," *Energy*, vol. 246, p. 123404, 2022.

[12] M. A. Hannan, D. N. How, M. H. Lipu, P. J. Ker, Z. Y. Dong, M. Mansur, and F. Blaabjerg, "Soc estimation of li-ion batteries with learning rate-optimized deep fully convolutional network," *IEEE Trans. Power Electron.*, vol. 36, no. 7, pp. 7349–7353, 2020.

[13] J. Chen, Y. Zhang, J. Wu, W. Cheng, and Q. Zhu, "Soc estimation for lithium-ion battery using the lstm-rnn with extended input and constrained output," *Energy*, vol. 262, p. 125375, 2023.

[14] M. Liu, J. Xu, Y. Jiang, and X. Mei, "Multi-dimensional features based data-driven state of charge estimation method for LiFePO$_4$ batteries," *Energy*, vol. 274, p. 127407, 2023.

[15] M. Jiao, D. Wang, and J. Qiu, "A gru-rnn based momentum optimized algorithm for soc estimation," *J. Power Sources*, vol. 459, p. 228051, 2020.

[16] J. Tian, C. Chen, W. Shen, F. Sun, and R. Xiong, "Deep learning framework for lithium-ion battery state of charge estimation: recent advances and future perspectives," *Energy Stor. Mater.*, vol. 61, p. 102883, 2023.

[17] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[18] L. Shen, J. Li, L. Meng, L. Zhu, and H. T. Shen, "Transfer learning-based state of charge and state of health estimation for li-ion batteries: A review," *IEEE Trans. Transp. Electrification*, 2023.

[19] Z. Ni, B. Li, and Y. Yang, "Deep domain adaptation network for transfer learning of state of charge estimation among batteries," *J. Energy Storage*, vol. 61, p. 106812, 2023.

[20] Y. Lu, L. Chen, A. Saidi, E. Dellandrea, and Y. Wang, "Discriminative transfer learning using similarities and dissimilarities," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 3097–3110, 2017.

[21] J. Yi, J. Tao, Z. Wen, and Y. Bai, "Language-adversarial transfer learning for low-resource speech recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 3, pp. 621–630, 2018.

[22] Y. Kang, H. Park, B. Smit, and J. Kim, "A multi-modal pre-training transformer for universal transfer learning in metal–organic frameworks," *Nat. Mach. Intell.*, vol. 5, no. 3, pp. 309–318, 2023.

[23] J. Xie and W. Sun, "A transfer and deep learning-based method for online frequency stability assessment and control," *IEEE Access*, vol. 9, pp. 75 712–75 721, 2021.

[24] X. Liu, Z. Cao, and Z. Zhang, "Short-term predictions of multiple wind turbine power outputs based on deep neural networks with transfer learning," *Energy*, vol. 217, p. 119356, 2021.

[25] Y. Liu and J. Wang, "Transfer learning based multi-layer extreme learning machine for probabilistic wind power forecasting," *Appl. Energy*, vol. 312, p. 118729, 2022.

[26] S. Li, H. He, P. Zhao, and S. Cheng, "Health-conscious vehicle battery state estimation based on deep transfer learning," *Appl. Energy*, vol. 316, p. 119120, 2022.

[27] W. Zheng, X. Zhou, C. Bai, D. Zhou, and P. Fu, "Adaptation of deep network in transfer learning for estimating state of health in electric vehicles during operation," *Batteries*, vol. 9, no. 11, p. 547, 2023.

[28] C. Vidal, P. Kollmeyer, E. Chemali, and A. Emadi, "Li-ion battery state of charge estimation using long short-term memory recurrent neural network with transfer learning," in *2019 IEEE Transp. Electrif. Conf. Expo (ITEC)*. IEEE, 2019, pp. 1–6.

[29] Y. Che, Y. Zheng, Y. Wu, X. Lin, J. Li, X. Hu, and R. Teodorescu, "Battery states monitoring for electric vehicles based on transferred multi-task learning," *IEEE Trans. Veh. Technol.*, vol. 72, no. 8, pp. 10 037–10 047, 2023.

[30] J. Tian, R. Xiong, W. Shen, and J. Lu, "State-of-charge estimation of LiFePO$_4$ batteries in electric vehicles: A deep-learning enabled approach," *Appl. Energy*, vol. 291, p. 116812, 2021.

[31] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," *Adv. Neural Inf. Process. Syst.*, vol. 29, 2016.

[32] K. Saito, Y. Ushiku, T. Harada, and K. Saenko, "Strong-weak distribution alignment for adaptive object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 6956–6965.

[33] Y. Luo, L. Zheng, T. Guan, J. Yu, and Y. Yang, "Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 2507–2516.

[34] A. Tonioni, M. Poggi, S. Mattoccia, and L. Di Stefano, "Unsupervised domain adaptation for depth prediction from images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2396–2409, 2019.

[35] J. Lu, R. Xiong, J. Tian, C. Wang, and F. Sun, "Deep learning to estimate lithium-ion battery state of health without additional degradation experiments," *Nat. Commun.*, vol. 14, no. 1, p. 2760, 2023.

[36] T. Han, Z. Wang, and H. Meng, "End-to-end capacity estimation of lithium-ion batteries with an enhanced long short-term memory network considering domain adaptation," *J. Power Sources*, vol. 520, p. 230823, 2022.

[37] Z. Ye and J. Yu, "State-of-health estimation for lithium-ion batteries using domain adversarial transfer learning," *IEEE Trans. Power Electron.*, vol. 37, no. 3, pp. 3528–3543, 2021.

[38] L. Shen, J. Li, L. Zuo, L. Zhu, and H. T. Shen, "Source-free cross-domain state of charge estimation of lithium-ion batteries at different ambient temperatures," *IEEE Trans. Power Electron.*, vol. 38, no. 6, pp. 6851–6862, 2023.

[39] C. Bian, S. Yang, and Q. Miao, "Cross-domain state-of-charge estimation of li-

ion batteries based on deep transfer neural network with multiscale distribution adaptation," *IEEE Trans. Transp. Electrification*, vol. 7, no. 3, pp. 1260–1270, 2020.

[40] L. Shen, J. Li, J. Liu, L. Zhu, and H. T. Shen, "Temperature adaptive transfer network for cross-domain state-of-charge estimation of li-ion batteries," *IEEE Trans. Power Electron.*, vol. 38, no. 3, pp. 3857–3869, 2022.

[41] Z. Meng, K. A. Agyeman, and X. Wang, "Lithium-ion battery state of charge estimation with adaptability to changing conditions," *IEEE Trans. Energy Convers.*, vol. 38, no. 4, pp. 2860–2870, 2023.

[42] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2298–2304, 2016.

[43] F. Wang, Z. Zhao, Z. Zhai, Y. Guo, H. Xi, S. Wang, and X. Chen, "Feature disentanglement and tendency retainment with domain adaptation for lithium-ion battery capacity estimation," *Reliab. Eng. Syst. Saf.*, vol. 230, p. 108897, 2023.

[44] R. Xiong, J. Tian, W. Shen, J. Lu, and F. Sun, "Semi-supervised estimation of capacity degradation for lithium ion batteries with electrochemical impedance spectroscopy," *J. Energy Chem.*, vol. 76, pp. 404–413, 2023.

[45] L. Ma and T. Zhang, "Deep learning-based battery state of charge estimation: Enhancing estimation performance with unlabelled training samples," *J. Energy Chem.*, vol. 80, pp. 48–57, 2023.

[46] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[47] N. R. Koluguri, T. Park, and B. Ginsburg, "Titanet: Neural model for speaker representation with 1d depth-wise separable convolutions and global context," in *ICASSP 2022-2022 IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*. IEEE, 2022, pp. 8102–8106.

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE conf. comput. vis. pattern recognit.*, 2016, pp. 770–778.

[49] H. Mirzaee and S. Kamrava, "Estimation of internal states in a li-ion battery using bilstm with bayesian hyperparameter optimization," *J. Energy Storage*, vol. 74, p. 109522, 2023.

[50] S. Ji, J. Zhu, Z. Lyu, H. You, Y. Zhou, L. Gu, J. Qu, Z. Xia, Z. Zhang, and H. Dai, "Deep learning enhanced lithium-ion battery nonlinear fading prognosis," *J. Energy Chem.*, vol. 78, pp. 565–573, 2023.

[51] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. March, and V. Lempitsky, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 59, pp. 1–35, 2016.

[52] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7167–7176.

[53] X. Chen, Y. Qin, W. Zhao, Q. Yang, N. Cai, and K. Wu, "A self-attention knowledge domain adaptation network for commercial lithium-ion batteries state-of-health estimation under shallow cycles," *arXiv preprint arXiv:2304.05084*, 2023.

[54] M. Long, Y. Cao, Z. Cao, J. Wang, and M. I. Jordan, "Transferable representation learning with deep adaptation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 12, pp. 3071–3085, 2018.

[55] M. Wang, R. Liu, N. Hajime, A. Narishige, H. Uchida, and T. Matsunami, "Improved knowledge distillation for training fast low resolution face recognition model," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 0–0.

[56] P. Kollmeyer, "Panasonic 18650PF Li-ion battery data," *Mendeley Data*, vol. 1, 2018, doi:10.17632/wykht8y7tg.1.

[57] K.-H. Kim, K.-H. Oh, H.-S. Ahn, and H. Choi, "Time-frequency domain deep convolutional neural network for li-ion battery soc estimation," *IEEE Trans. Power Electron.*, vol. 39, no. 1, pp. 125–134, 2023.

[58] H. Yu, L. Zhang, W. Wang, S. Li, S. Chen, S. Yang, J. Li, and X. Liu, "State of charge estimation method by using a simplified electrochemical model in deep learning framework for lithium-ion batteries," *Energy*, vol. 278, p. 127846, 2023.

[59] T. Han, Z. Wang, and H. Meng, "End-to-end capacity estimation of lithium-ion batteries with an enhanced long short-term memory network considering domain adaptation," *J. Power Sources*, vol. 520, p. 230823, 2022.

[60] P. Kollmeyer, C. Vidal, M. Naguib, and M. Skells, "LG 18650HG2 Li-ion battery data and example deep neural network xEV SOC estimator script," *Mendeley Data*, vol. 3, 2020, doi:10.17632/cp3473x7xv.3.

[61] S. Shen, M. Sadoughi, M. Li, Z. Wang, and C. Hu, "Deep convolutional neural networks with ensemble learning and transfer learning for capacity estimation of lithium-ion batteries," *Appl. Energy*, vol. 260, p. 114296, 2020.

[62] I. Babaeiyazdi, A. Rezaei-Zare, and S. Shokrzadeh, "State of charge prediction of ev li-ion batteries using eis: A machine learning approach," *Energy*, vol. 223, p. 120116, 2021.

[63] D. N. T. How, M. A. Hannan, M. S. H. Lipu, K. S. M. Sahari, P. J. Ker, and K. M. Muttaqi, "State-of-charge estimation of li-ion battery in electric vehicles: A deep neural network approach," *IEEE Trans. Ind. Appl.*, vol. 56, no. 5, pp. 5565–5574, 2020.

[64] W. He, N. Williard, C. Chen, and M. Pecht, "State of charge estimation for electric vehicle batteries using unscented kalman filtering," *Microelectron. Reliab.*, vol. 53, no. 6, pp. 840–847, 2013.