Robust Transfer Learning for Battery Lifetime Prediction Using Early Cycle Data

Kang, Wenda; Wang, Dianpeng; Jongbloed, Geurt; Hu, Jiawen; Chen, Piao

DOI
10.1109/TII.2025.3545079

Publication date
2025

Document Version
Final published version

Published in
IEEE Transactions on Industrial Informatics

Robust Transfer Learning for Battery Lifetime Prediction Using Early Cycle Data

Wenda Kang, Dianpeng Wang, Geurt Jongbloed, Jiawen Hu, and Piao Chen, *Member, IEEE*

***Abstract*—Battery lifetime prediction is crucial in industrial applications. However, the lack of diversity in training data often poses challenges regarding the robustness and generalization of lifetime predictions for batteries from different batches. Motivated by the early cycle data from lithium-ion batteries, this article proposes a robust transfer learning method by employing a model average framework, where the weights are determined based on the distance between the source domain and the target domain. Kernel regression is used to build the prediction of battery lifetime using early cycle data, and transfer component analysis is utilized to transfer knowledge between different domains. The case study on lithium-ion phosphate/graphite cells demonstrates that the proposed method can mitigate the impact of negative transfer and has superior performance compared to traditional methods.**

***Index Terms*—Battery lifetime, model averaging, negative transfer, prognostic, transfer learning (TL).**

I. Introduction

LARGE battery storage systems have been widely used to stabilize energy systems like the electricity grid and offer various benefits [1]. However, the costliness and uneconomical nature of batteries pose significant challenges, necessitating the efficient utilization of their limited resources [2]. Therefore, extending the battery's lifespan is a crucial objective. To achieve this, accurately predicting battery lifetime through modeling is essential [3].

Battery cell aging is affected by both the duration of use and various usage conditions. The aging process is typically divided into three stages: early, stable, and decline stages [4]. Various approaches in the literature are available for modeling battery aging, including physics-based, semiempirical, and data-driven methods. Generally, physics-based and semiempirical methods require a thorough understanding of degradation behavior based on failure mechanisms, which may limit their applicability [5]. On the other hand, data-driven methods, such as machine learning models, have gained significant attention in recent years due to their ability to operate without explicit knowledge of failure mechanisms [6]. However, the machine learning methods heavily rely on the assumption that the training and test data are sampled from the same distribution, which is not satisfied in many real applications. In battery lifetime prediction, although a bunch of data is available, the data from different batches may not follow the same distribution due to factors such as battery type, operating conditions, and manufacturing variations that can impact battery performance [6].

Transfer learning (TL) can help overcome this issue by transferring knowledge from a source problem (training data) to a target problem (test data) with similar characteristics but different underlying distributions [7]. There are many papers focusing on using TL to predict batteries' lifetime in the literature. For example, Qin et al. [8] considered state-of-charge (SoC) estimation by exploiting the temporal dynamics of the measurements and the ability to transfer consistent estimates across different temperatures. Qin et al. [9] proposed a transferable multistage state-of-health (SoH) estimation model to perform TL across batteries in the same degradation stage. More related references can be found in Section II.

However, these existing methods suffer from two common drawbacks. First, they often rely on data where battery capacity degradation has already occurred, necessitating a certain number of cycles. As emphasized in [4] and [10], accurate predictions based on early-stage data—referred to as early-cycle data in this article—are critical. They provide valuable insights into long-term performance without the need for extended testing, allowing for the rapid detection of early failures, manufacturing defects, or deviations from expected performance. This accelerates battery development and design optimization, lowers production costs, and enables quick assessments of battery quality and performance.

Received 22 October 2024; accepted 16 February 2025. Date of publication 12 March 2025; date of current version 23 May 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 72401253, Grant 12131001, Grant 12171033, and Grant 72171037, in part by the the State Key Laboratory of Biobased Transportation Fuel Technology under Grant 512302-X02301, in part by the Sichuan Science & Technology Program under Grant 2023NSFSC0476, and in part by the start-up grant from the ZJU-UIUC Institute at Zhejiang University under Grant 130200-171207711. Paper no. TII-24-5571. *(Corresponding author: Jiawen Hu.)*

Wenda Kang and Geurt Jongbloed are with the Department of Applied Mathematics, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: w.kang-1@tudelft.nl; g.jongbloed@tudelft.nl).

Dianpeng Wang is with the School of Mathematics and Statistics, and the MIIT Key Laboratory of Mathematical Theory and Computation in Information Security, Beijing Institute of Technology, Beijing 100081, China (e-mail: wdp@bit.edu.cn).

Jiawen Hu is with the School of Aeronautics and Astronautics, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: hdl@uestc.edu.cn).

Piao Chen is with the ZJU-UIUC Institute, Zhejiang University, Haining 314400, China, and also with the Department of Applied Mathematics, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: piaochen@intl.zju.edu.cn).

Digital Object Identifier 10.1109/TII.2025.3545079

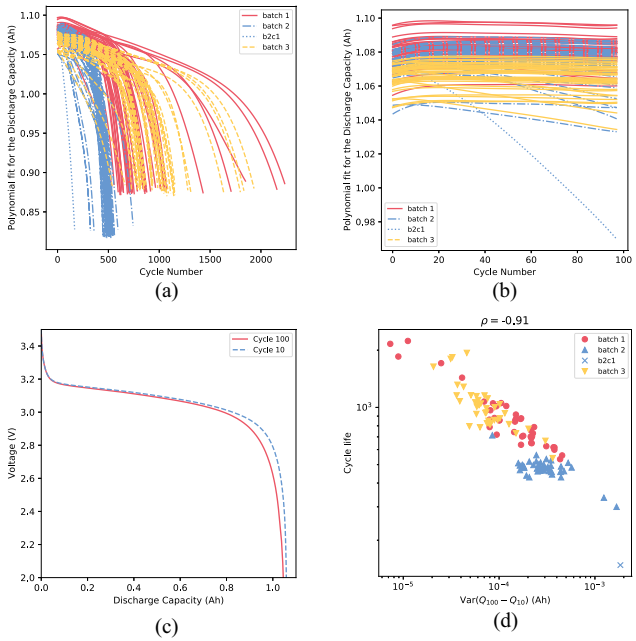

Fig. 1. Overview and examples of features in the dataset. (a) Discharge capacity curves after preprocessing. (b) Discharge capacity curves over the first 100 cycles. (c) Example of the discharge voltage curves for the 100th and 10th cycle. (d) Variance of $\Delta Q_{100-10}(V)$ against battery cycle life for all cells.

Regarding the prediction models utilizing early-cycle data, the two primary categories [4] are models based on long short-term memory (LSTM) [11], [12] or the convolutional neural network (CNN) [13], [14]. Additional references can be found in the review paper [4].

An example of early cycle battery data is depicted in Fig. 1, where Fig. 1(a) and (b) exhibits the capacity curves for the full life cycles and the corresponding curves for early cycles (the first 100 cycles in this case). From Fig. 1(b), it is apparent that there is a negligible degradation trend for capacity curves in the early cycles, underscoring the challenge of accurately predicting battery lifetime using such data. However, as highlighted in [10], certain features extracted from early cycle data demonstrate strong correlations with cycle life. An example of this is shown in Fig. 1(c) and (d), where the later stages of degradation are reflected in the discharge voltage curve and the features extracted from it. More details will be introduced in Section IV-A.

Furthermore, existing work primarily focuses on what part of data should be transferred and how to utilize the TL methods, but without addressing the challenge of reducing the impact of negative transfer. Negative transfer occurs when the knowledge gained in the source domain negatively affects the performance of a model when applied to the target domain, and this phenomenon can be attributed to factors such as domain discrepancies, task misalignment, overfitting to the source domain, etc. To the best of authors' knowledge, there is limited literature exploring the utilization of TL for battery lifetime prediction based on early cycle data and discussing strategies to mitigate the effects of negative transfer.

Motivated by the aforementioned challenges and the early cycle dataset of lithium-ion batteries in [10], this article presents a novel robust TL framework for battery lifetime prediction using early cycle data. The new framework addresses three key issues: determining the relevant information (*what*) for effective transfer, devising an appropriate transfer strategy (*how*) to leverage knowledge from source to target data, and providing a robust TL framework by choosing appropriate weights, although we did not explicitly identify the transferable conditions (*when*).

The main contributions of this article can be summarized as follows.

1) Introduce a novel robust TL framework for predicting battery lifetime using early cycle data, which can mitigate the impact of negative transfer.
2) Present a method for determining the weight in the proposed robust TL framework based on the distributional difference.
3) Validate the performance of the method using a substantial dataset of lithium-ion batteries.

The rest of this article is organized as follows. Section II reviews some related work, including *what*, *how*, *when* to transfer. Section III provides a detailed explanation of the methods employed and details of the proposed robust TL framework. In Section IV, a case study based on a substantial dataset of lithium-ion batteries is conducted to validate the performance of the proposed method. Finally, Section V concludes this article.

II. Related Work

This section reviews the existing literature on TL, focusing on three key questions: what knowledge should be transferred, how to facilitate this transfer, and when it is most beneficial to do so. By addressing these aspects, we aim to clarify the mechanisms and strategies involved in effective TL, highlighting its applications and challenges in various domains.

A. What to Transfer?

What to transfer involves selecting the relevant information or data to transfer. Generally, most of the TL methods for battery lifetime prediction are feature-based. According to the way of extracting features, there are two commonly used methods for transferring information: transferring the physical significance features or data-driven features [15], [16]. In practice, the selection of feature type should be determined based on the specific engineering context and requirements.

1) Physical Significance Features: In the case of physical significance features, the feature extractor remains fixed, wherein the extracted features possess specific physical significance. Subsequently, the transferred data can be integrated into a data-driven predictor.

For example, Jia et al. [17] introduced a TL-based method for predicting target SoH values using degradation data from other batteries by employing transfer component analysis (TCA) with an extreme learning machine algorithm. Li et al. [18] proposed a framework to monitor SoH of batteries by integrating maximum mean discrepancy (MMD), semisupervised TCA, and mutual information, and validates the effectiveness by using a real dataset

containing four batteries operating under different conditions. Jiao et al. [19] proposed a TL-based framework for battery remaining useful life (RUL) prediction, which utilizes features generated from electrochemical theory, capacity-differential voltage curves, and electrochemical impedance spectroscopy curves.

2) Data-Driven Features: Another commonly used method is to transfer the data-driven features generated from an adaptive feature extractor, where the feature extractor itself is also a data-driven model [20]. LSTM and neural networks are the most popular data-driven feature extractors. For example, in the work by [21], a CNN-based feature extractor is utilized to extract data-driven features from the collected data, which include the current, voltage, and capacity characteristics of the battery during the charging/discharging cycle. Pan et al. [22] adopted an LSTM network model to extract data-driven features from the collected capacity curves, and proposes a TL-based battery RUL prediction model by integrating a particle filter model.

B. How to Transfer?

The question *how* to transfer concerns selecting the appropriate TL method and hyperparameters. There are three main approaches to perform TL, namely based on the fine-tuning strategy, the discrepancy metric, or the domain adversarial [15], [16].

1) Fine-Tuning-Based: The strategy to transfer through fine-tuning is widely used in the lifetime prediction of batteries. The basic idea of the fine-tuning strategy is to involve retraining a pretrained data-driven model for better performance on specific tasks, which uses only a small amount of data from the target dataset [15]. Specifically, in battery life prediction, fine-tuning involves first building a pretrained data-driven model using source data and then retraining the model using a small amount of target data, thereby improving accuracy.

For example, Tan and Zhao [23] proposed a TL-based framework for battery SoH prediction, which combines an LSTM network with adjustable fine-tuning-based fully connected layers. Deng et al. [24] merged degradation pattern recognition with a fine-tuning-based LSTM approach to present a TL-based SoH prediction model, which demonstrates superior performance compared to existing methods. Nguyen et al. [25] proposed a TL-based framework for SoH prediction of lithium-ion batteries, which employs a deep neural network architecture that integrates equivalent circuit simulated layers and a fine-tuning network hierarchy. More references can be found in review papers [15], [16].

As mentioned earlier, the fine-tuning strategy typically necessitates a small amount of labeled target data. However, in practical scenarios, such labeled testing data for training is often unavailable, thereby constraining the applicability of the fine-tuning strategy.

2) Discrepancy Metric-Based: The methods based on discrepancy metrics are widely used in TL. The literature applying TCA for transfer generally offers advantages in computational efficiency. Despite these approaches, another common category within discrepancy metric-based methods involves incorporating the discrepancy metric into the loss function during model training. For example, Han et al. [26] presented a novel deep learning framework that addresses this challenge by combining a deep LSTM network to model the nonlinear mapping from monitored data (e.g., terminal voltage and current) to battery capacity, and a domain adaptation layer with MMD to align degradation features between source and target batteries. Ye and Yu [27] combined the MMD with a gated recurrent unit recurrent neural network to reduce the domain discrepancy for battery SoH prediction. Ma et al. [28] integrated the MMD loss with a convolutional neural network to predict the battery SoH.

3) Domain Adversarial-Based: The domain adversarial-based approach is another effective TL method aimed at learning a domain-invariant feature space where the domain classifier cannot distinguish the domain of the input data. Although adversarial adaptation methods are effective, there are only a limited number of references using them as TL strategies at this stage. For example, Shen et al. [29] introduced a temperature-adaptive transfer network that employs adversarial adaptation and MMD to minimize domain divergence for estimating the SoC in batteries. In [6], domain adversarial training and TL methods are integrated into Bayesian deep learning to propose an innovative RUL prediction framework capable of handling diverse machines with limited data.

While domain-adversarial-based TL solutions have demonstrated success, a notable drawback is the resource-intensive nature of the adversarial training process. This computational demand can somewhat restrict their applicability in intricate industrial scenarios. Therefore, further research and development efforts are necessary to align these methods with the practical requirements of the industry [15], [16].

C. When to Transfer?

Combining the methods in *what* (see Section II-A) and *how* (see Section II-B), various TL-based methods can be generated. Despite the effectiveness of these transfer methods, they can encounter challenges in complex and demanding environments, potentially resulting in negative transfer. Hence, there is a need to develop methods for quantifying transferability or assessing negative transfer, which is essentially a question of *when* to transfer.

However, there is limited literature focusing on the *when* to transfer issue in battery health management. Chehade et al. [30] proposed a Gaussian process regression model to forecast the capacity of batteries and uses hyperparameters of the kernel function in the covariance matrix to control the negative transfer. Oyewole et al. [31] introduced the product of MMD and a tuning parameter as the penalty term in their loss function to reduce the likelihood of negative transfer in battery SoC estimation. Nonetheless, these methods either fail to predict battery lifetime or require the capacity to exhibit a degradation trend, which is inconsistent with our task of predicting battery lifetime based on early cycle data shown in Fig. 1. In addition, they require labeled data in the target domain, which is difficult to obtain in practice.

III. Robust TL Framework for Battery Lifetime Prediction

In this section, the proposed robust TL framework is introduced, which is designed to address the three questions outlined in Section II.

Regarding the *what* to transfer issue, we prioritize the use of features with physical significance due to their interpretability. Specifically, in battery lifetime prediction research involving early cycle data, domain knowledge pertaining to lithium-ion batteries is commonly leveraged. Key features, such as initial discharge capacity, charge time, and cell temperature are widely employed. For further details regarding these features, refer to Section IV and [10].

A. How to Transfer: TCA and the Kernel Regression

As discussed in Section II-B, there are three commonly employed approaches to tackle the *how* to transfer issue. In this article, we employ discrepancy metric strategies for information transfer because they do not require labeled data or substantial computing resources in the target domain, unlike fine-tuning and domain adversarial approaches.

The procedures of discrepancy metric strategies can be further divided into three steps: choosing the discrepancy metric, domain adaptation, and developing the predictor. These steps are discussed in detail as follows.

1) Discrepancy Metric: Discrepancy metrics play a pivotal role in the discrepancy metric-based strategy, and among the various metrics available, MMD is particularly popular.

Consider a labeled source domain dataset $(\boldsymbol{X}_S, \boldsymbol{Y}_S) = \{(X_{S,1}, Y_{S,1}), \ldots, (X_{S,n_S}, Y_{S,n_S})\}$, where $X_{S,i}$, $i = 1, \ldots, n_S$, represents the ith input of the source domain dataset, and $Y_{S,i}$ denotes its corresponding output. In addition, there exists an unlabeled target domain dataset $\boldsymbol{X}_T = \{X_{T,1}, \ldots, X_{T,n_T}\}$, where $X_{T,j}$, $j = 1, \ldots, n_T$, signifies the jth input of the target dataset. The total size of the source and target datasets is represented by $n = n_S + n_T$. Let $\mathcal{P}(\boldsymbol{X}_S)$ and $\mathcal{Q}(\boldsymbol{X}_T)$ (or $\mathcal{P}$ and $\mathcal{Q}$ in short) denote the marginal distributions of inputs from the source and target datasets, respectively. The task is to predict the output of the target domain, $Y_{T,j}, j = 1, \ldots, n_T$. We assume that $\mathcal{P} \neq \mathcal{Q}$ while there exists a mapping function $\phi(\cdot)$ such that the marginal distributions $P(\phi(\boldsymbol{X}_S)) \approx P(\phi(\boldsymbol{X}_T))$ and the conditional distributions $P(\boldsymbol{Y}_S|\phi(\boldsymbol{X}_S)) \approx P(\boldsymbol{Y}_T|\phi(\boldsymbol{X}_T))$.

The key issue in discrepancy metric-based TL is mapping different domain instances onto a common space. Specifically, MMD employs a function $\phi(\cdot)$ to map each instance to the reproduced Hilbert space $\mathcal{H}$ associated with the kernel $k : \chi \times \chi \to \mathbb{R}$, where $k(X_i, X_j) = \phi(X_i)^{\mathrm{T}}\phi(X_j)$, and χ is the feature space of the source and target domains.

The MMD can be defined as the distance between two different projections of the means. By using a kernel trick, The squared MMD distance can be reformulated as

$$\mathrm{MMD}^2(X_S, X_T) = \|\frac{1}{n_S}\sum_{i=1}^{n_S}\phi(X_{S,i}) - \frac{1}{n_T}\sum_{j=1}^{n_T}\phi(X_{T,j})\|_{\mathcal{H}}^2 = \mathrm{Tr}(\mathrm{KL}) \tag{1}$$

where $K = \begin{bmatrix} K_{\boldsymbol{X}_S,\boldsymbol{X}_S} & K_{\boldsymbol{X}_S,\boldsymbol{X}_T} \\ K_{\boldsymbol{X}_T,\boldsymbol{X}_S} & K_{\boldsymbol{X}_T,\boldsymbol{X}_T} \end{bmatrix} \in \mathbb{R}^{n\times n}$ is a composite kernel matrix, with $K_{\boldsymbol{X}_S,\boldsymbol{X}_S}$, $K_{\boldsymbol{X}_S,\boldsymbol{X}_T}$, $K_{\boldsymbol{X}_T,\boldsymbol{X}_S}$, and $K_{\boldsymbol{X}_T,\boldsymbol{X}_T}$ being the kernel matrices defined by k based on the data in the source domain, the target domain, and across domains, respectively. L is a matrix with the (i, j)th entry L_{ij} defined as $L_{ij} = \begin{cases} 1/n_S^2 & X_i, X_j \in \boldsymbol{X}_S \\ 1/n_T^2 & X_i, X_j \in \boldsymbol{X}_T \\ -1/n_S n_T & \text{other.} \end{cases}$

MMD is a kernel-based distance metric that plays a crucial role in identifying underlying patterns in data and recognizing distribution differences. The choice of an appropriate kernel is essential for the effective utilization of the metric. However, the task of determining the most suitable kernel for MMD can be intricate and remains an ongoing research area.

2) Domain Adaption: Once the discrepancy metric is selected, the subsequent step focuses on minimizing the chosen metric using the data from the source and target domains. In this article, we employ a widely used domain adaptation method known as TCA. Using MMD, TCA can be achieved by solving the following kernel learning problem [7]:

$$\begin{aligned} \min_{W} \quad & \mathrm{Tr}(W^\top KLKW) + \lambda \mathrm{Tr}(W^\top W) \\ \text{s.t.} \quad & W^\top KHKW = I_m \end{aligned} \tag{2}$$

where W is a matrix that transforms the kernel map features to an m-dimensional space, $H = I_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^{\mathrm{T}}$ is the centering matrix, with $I_n \in \mathbb{R}^{n\times n}$, $I_m \in \mathbb{R}^{m\times m}$ being identity matrices and $\mathbf{1}_n \in \mathbb{R}^{n\times 1}$ being the column vector with all ones. Subsequently, the matrix W is obtained by identifying the first m smallest eigenvalues of $(I_m + \lambda KLK)^{-1}KHK$, where λ is a hyperparameter determined through cross-validation. More details and explanations of TCA can be found in [7].

3) Predictor: Following the acquisition of the most closely matched transferred data, the subsequent phase involves constructing a predictor. This is achieved by training a model using the transferred source data. In practical applications, the choice of a specific predictor depends on the context. For example, in the case of the motivated battery dataset described in [10], the authors employed a regularization model known as the elastic net, which combines the Lasso and Ridge regression methods for making predictions. The underlying assumption of the elastic net is a linear relationship between the input variables and the corresponding response. However, such a parametric assumption may not be maintained in the transferred data after TCA. Therefore, we adopt the widely used nonparametric approach, the kernel regression [32], as our predictor.

Let $(X_{S,1}, Y_{S,1}), \ldots, (X_{S,n_S}, Y_{S,n_S})$ be the given source data set, the estimated result of the ith sample based on the kernel regression can be expressed as

$$\hat{Y}_{S,i} = \frac{\sum_{j\neq i} Y_{S,j} \times k(X_{S,i}, X_{S,j})}{\sum_{j\neq i} k(X_{S,i}, X_{S,j})}. \tag{3}$$

For the target data set $\{X_{T,1}, \ldots, X_{T,n_T}\}$, the prediction for the ith sample can be derived as

$$\hat{Y}_{T,i} = \frac{\sum_{j=1}^{n_S} Y_{S,j} \times k(X_{T,i}, X_{S,j})}{\sum_{j=1}^{n_S} k(X_{T,i}, X_{S,j})}. \tag{4}$$

More details and explanations of kernel regression can be found in [32].

B. When to Transfer: A Robust TL Method

The problem about *when* to transfer is important because inappropriate TL may lead to negative transfer. It is also difficult to verify performance since there is no labeled data in the target domain [15].

1) Relative Change in MMD Is Not Reliable: A straightforward idea to solve this issue is using the relative change in MMD to assess the success of TL since they are specifically designed to measure the difference between the source and target domains. However, the following simulation study reveals that the reduction in MMD values does not consistently correspond to increased similarities. In addition, the effectiveness of TCA is observed to be contingent on the initial MMD value.

Note that the transferred knowledge in this article refers to the data from the source and target domains that have been transformed using the transformation matrix W. Successful transfer indicates that the distributions of the transferred source and target domain data are significantly closer to each other compared to the distributions of the original source and target domains. To illustrate the behavior of traditional TCA, we present three simulated examples in Fig. 2. The first scenario involves two samples with a small initial MMD value, the second involves two samples with an initial MMD within an appropriate range, and the third involves two samples with a large initial MMD value. We utilize the widely used t-SNE [33] for visualizing the distributional changes in the high-dimensional feature space resulting from TCA, providing an intuitive representation essential for assessing domain adaptation success and understanding the alignment between source and target domains.

Specifically, in Fig. 2(a), we observe that when the initial MMD value is relatively small, applying TCA, while successful in reducing MMD (from 0.1022 to 0.0576), may introduce alterations to the distribution shape. In such cases, the prediction without TL, denoted as NoTL, could be preferred, as the distribution shapes of the raw source and target domains are more similar than those after TCA. Turning to Fig. 2(b), we focus on scenarios where the initial MMD falls within a suitable range. Here, TCA significantly enhances the similarity between the source and target distributions, demonstrating its effectiveness (reducing the MMD value from 0.5177 to 0.0002). In Fig. 2(c), we explore the case with a large initial MMD value. Despite TCA's ability to reduce the MMD metric (from 1.2345 to 0.2711), the transferred data still exhibits considerable dissimilarity, emphasizing the limitations of TCA when faced with large MMD values. Simultaneously, the smaller MMD values may still offer valuable insights, given the distinct dissimilarities in distribution shapes observed in both the raw source and target domains, as well as the transferred source and target domains. Fig. 2(a) and (c) illustrates that although the MMD values decreased after applying TCA, indicating improved global alignment, the increased divergence observed in the t-SNE visualization suggests that local differences may have become larger. This highlights that, in some cases, the relative change in MMD may be less reliable.

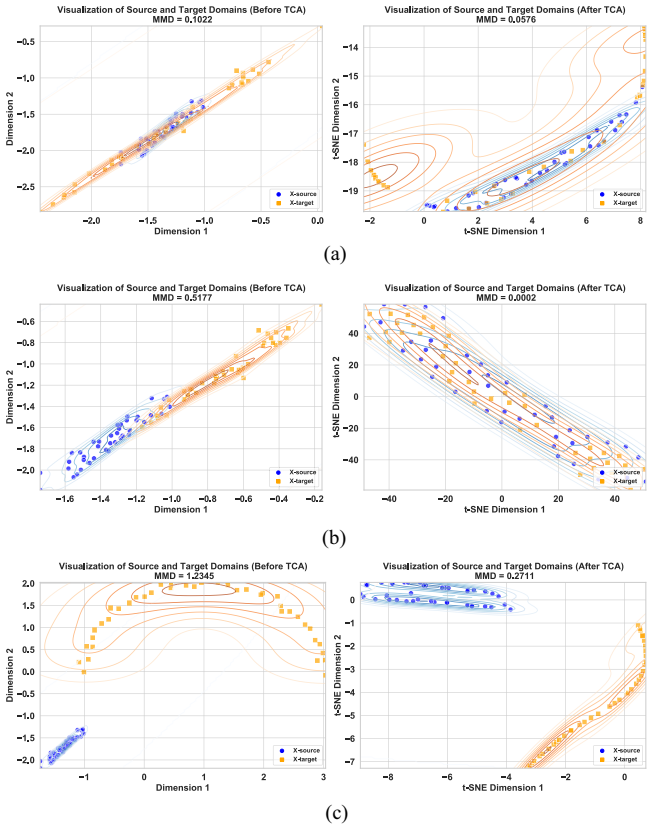

Fig. 2. Illustrative examples of TCA and distribution preservation with varying initial MMD values. (a) Example 1: The impact of a small initial MMD value on distribution shape. (b) Example 2: Improving distribution similarity with an appropriate initial MMD value. (c) Example 3: Non-negligible differences in distribution shape after transfer still exist for a larger initial MMD value.

The insights from Fig. 2 suggest that the initial value of MMD, indicating the similarity between the original source and target domain, highly influences the performance of TCA. Thus, a robust TL framework is essential to strike a balance between NoTL and those with TL.

2) Robust TL Method Based on Model Averaging: To mitigate the impact of negative TL, this article adopts the idea of model averaging [34] to propose a more robust TL method.

Let $\hat{\boldsymbol{Y}}_{T,1}$ be the predicted result using NoTL, and $\hat{\boldsymbol{Y}}_{T,2}$ be the predicted result using TL. Then, the final result $\hat{\boldsymbol{Y}}_T = (1-w) \times \hat{\boldsymbol{Y}}_{T,1} + w \times \hat{\boldsymbol{Y}}_{T,2}$, where $w \in [0, 1]$ is the weight to balance NoTL and TL. Thus, the issue turns to finding a suitable form of w.

Considering the similarity between the raw source data $\mathcal{P}$ and target data $\mathcal{Q}$, if $\mathcal{P}$ and $\mathcal{Q}$ yield a small initial MMD value, indicating their similarity in this case, then w should be set to 0. This concern is affirmed by the results shown in Fig. 2(a), where TCA increased the dissimilarity of the distribution shapes

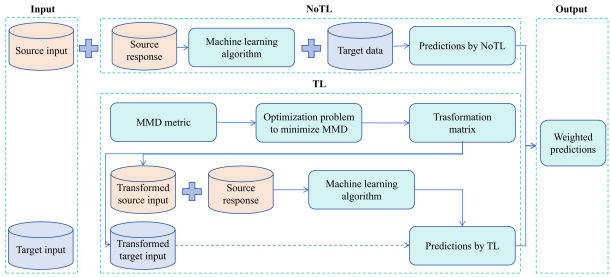

Fig. 3. Basic structure of the proposed robust TL approach.

when $\mathcal{P}$ and $\mathcal{Q}$ yielded a small initial MMD value. In addition, w should also be smaller when there is no similarity between the transferred source data $\mathcal{P}_{\text{new}}$ and target data $\mathcal{Q}_{\text{new}}$ as this case indicates that after TL, the source and target domain are still very different, as shown in Fig. 2(c). As significant distributional differences persist after transfer in such cases, the effectiveness of TCA becomes uncertain. Choosing traditional machine learning algorithms is preferable in these instances, as they do not rely on the assumptions associated with TL.

To quantify the degree of similarity, a threshold is necessary. In this article, we utilize the empirical distribution of the MMD metric to determine such a threshold, which can be obtained through a permutation approach [35]. Specifically, the empirical distribution function of the MMD is derived by repeatedly shuffling the datasets and recalculating the MMD value. Subsequently, the $1-\alpha$ quantile of the derived empirical distribution is employed as the threshold, denoted as T. If the original MMD value, denoted as MMD_0, exceeds T, we conclude that the two distributions are different; otherwise, we state that the two distributions are similar enough. Naturally, the probability of obtaining results at least as extreme as the result actually observed, denoted as p, can be chosen as a weight. Thus, the final form of the weight term w can be expressed as

$$w = p \times \mathbb{I}_{\{\text{MMD}_0 > T\}} \tag{5}$$

where $\mathbb{I}_{\{\text{MMD}_0>T\}}$ is the indicator function, whose value is equal to 1 when the MMD_0 calculated by $\mathcal{P}$ and $\mathcal{Q}$ exceeds T, and 0 otherwise.

The weight form in (5) satisfies the aforementioned properties we desired. When the raw source and target distributions $\mathcal{P}$ and $\mathcal{Q}$ are similar enough, the calculated MMD_0 should be located within the threshold T, resulting in the value of w being equal to 0. Otherwise, TL should be performed, and the corresponding value of w is equal to p, which is derived from the transferred distributions $\mathcal{P}_{\text{new}}$ and $\mathcal{Q}_{\text{new}}$.

C. Robust TL Framework

The proposed robust TL framework is presented in Algorithm 1, and the fundamental structure of the approach is illustrated in Fig. 3. Following data preprocessing, we extract selected features from both the source and target domains. Subsequently, the weight term w in (5) can be determined following the procedure introduced in Section III-B. The weight term w is then utilized to balance the predictions $\hat{\boldsymbol{Y}}_{T,1}$ and $\hat{\boldsymbol{Y}}_{T,2}$.

Algorithm 1: The Proposed Robust Transfer Learning Framework.

Input: Source domain $\boldsymbol{X}_S$ and corresponding response $\boldsymbol{Y}_S$, target domain $\boldsymbol{X}_T$, kernel function type, λ;
Output: Predicted response for target domain $\hat{\boldsymbol{Y}}_T$.

1: Data preprocessing, encompassing techniques such as smoothing and outlier handling;
2: Generate features according to the preprocessed data;
3: Calculate the weight term w following the procedure introduced in Section III-B;
4: Derive the prediction $\hat{\boldsymbol{Y}}_{T,1}$ using NoTL method;
5: If $w \neq 0$, derive the prediction $\hat{\boldsymbol{Y}}_{T,2}$ using TL method, otherwise, set $\hat{\boldsymbol{Y}}_{T,2} = 0$;
6: Derive the final prediction $\hat{\boldsymbol{Y}}_T = (1-w) \times \hat{\boldsymbol{Y}}_{T,1} + w \times \hat{\boldsymbol{Y}}_{T,2}$.

IV. Case Study

In this section, a case study on the lifetime data of 124 lithium-ion batteries is presented to demonstrate the implementation of the proposed framework.

A. Data Overview

The dataset used for the case study is taken from [10], which comprises 3 batches consisting of a total of 124 lithium-ion phosphate/graphite batteries (41/43/40 in batch 1/2/3, respectively). The batches can be considered as separate experiments, with batch 3 conducted nearly a year after batches 1 and 2. For each battery, the following data are attached.

1) *Cycle life*: The number of cycles until the battery's capacity has decreased below 80% (and for batch 2, 75%) of its nominal capacity.
2) *Charge policy*: All batteries are charged according to different fast-charging conditions in a temperature-controlled environment, see [10] for more details.
3) *Summary data*: The summary data contains information for each cycle, including the cycle number, discharge capacity, charge capacity, internal resistance, maximum temperature, average temperature, minimum temperature, and charging time.
4) *Cycle data*: The information contained within a cycle includes the time, charge capacity, current, voltage, temperature, and discharge capacity. In addition, calculated variables such as the discharge rate, the discharge capacity interpolated linearly, and temperature interpolated linearly are also included.

Note that the dataset is the largest publicly available for nominally identical commercial lithium-ion batteries cycled under controlled conditions.

B. Prediction Models and Analytical Scenarios

For the prediction model, the original paper [10] uses a feature-based method. After preprocessing the curves and fitting a polynomial to the discharge capacity as a function of voltage in the 10th and 100th cycle $\Delta Q_{100-10}(V)$, several features can be

TABLE I
DIFFERENT SCENARIOS FOR SOURCE AND TARGET DATA

Scenario	Source data	Target data
1	Batch 1	Batch 2
2	Batch 1	Batch 3
3	Batch 2	Batch 3
4	Batch 1&2	Batch 3
5	Original train set (from [10])	Original test set (from [10])
6	Original train set (from [10])	Batch 3
7	Original test set (from [10])	Batch 3

extracted from these values, such as mean, variance, minimum, skewness, and kurtosis. The logarithmic values of the summary statistics exhibit a strong correlation with the logarithmic cycle life, while the discharge capacities of only the first 100 cycles are weakly correlated with the cycle life. The capacity curves and an example of the discharge voltage curves are shown in Fig. 1. The battery cell "b2c1," which has a relatively low lifetime, will be excluded from the dataset to prevent its negative impact on the results.

As detailed in [10], three prediction models incorporating different features are considered: the *variance*, *discharge*, and *full* models. The *variance* model operates as a univariate model, utilizing $\log(\mathrm{Var}(|\Delta Q_{100-10}(V)|))$ to predict log cycle lives. In contrast, the *discharge* model encompasses summary statistics of $\Delta Q_{100-10}(V)$, including minimum, mean, variance, skewness, and kurtosis, alongside other candidate features derived during discharge. The *full* model extends the features of the variance and discharge models by incorporating additional data streams, such as temperature and internal resistance. Further elucidation on the configuration of these three models is available in [10].

To fully utilize the dataset, we consider seven different scenarios for both the source and target data, allowing for comprehensive comparisons. A summary of these scenarios is presented in Table I, providing a clear overview of the variations studied. Both the root mean squared error (RMSE) and the mean absolute percentage error (MAPE) are used as prediction performance metrics.

C. Prediction Performance

The performance of kernel regression and TL is affected by the hyperparameter λ in (2) and the choice of the kernel type. In this article, we adopt four widely used kernel types: linear, polynomial, radial basis function (Rbf), and Laplace kernel. To ensure fair comparisons across different scenarios, the hyperparameter λ is determined through cross-validation, and the hyperparameters of the kernel functions remain consistent across all scenarios. Taking scenario 2 as an example, RMSEs for different models across various kernel types are displayed in Fig. 4, where the red, green, and blue boxes represent the results derived from NoTL, TL, and the proposed robust TL methods, respectively. Fig. 4 indicates that the proposed robust TL framework does indeed possess the desired ability to balance NoTL and TL methods for different models. Similar results for MAPEs are observed but not displayed to save space.

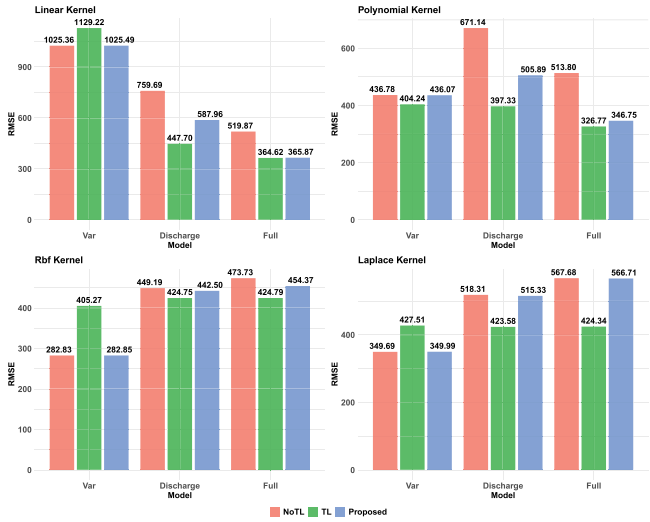

Fig. 4. Example of RMSEs for different methods via kernel types.

Taking the linear kernel as an example, in the variance model case, the NoTL approach outperforms TL, indicating that TL introduces a negative effect. In this scenario, the proposed method favors NoTL, resulting in predictions that align closely with NoTL, thereby minimizing the negative impact caused by TL. Conversely, for the discharge and full models, TL significantly outperforms NoTL. In these cases, the proposed approach prioritizes TL, with final predictions derived from a weighted combination of TL and NoTL using weight w, thereby leveraging the benefits of TL.

Regarding the RMSEs of lifetime predictions under all scenarios, kernel types, and models, the results are reported in Table II. Cases where the proposed robust TL framework successfully mitigates the impact of negative TL in direct TL are highlighted in bold for enhanced visibility.

In terms of computational cost, the TCA-based approach took 0.02 s for data transformation. The experiments were conducted on the following system specifications.

1) Operating System: Windows 10 (Version 10.0.22621).
2) CPU: Intel 16-core processor (Intel64 Family 6 Model 141).
3) Memory: 32 GB.

The results presented in Fig. 4 and Table II demonstrate that the proposed robust TL framework effectively balances NoTL and TL in general. Specifically, the proposed method performs well in extracting positive outcomes from TL when it significantly improves prediction results. Conversely, when TL yields unfavorable results, the proposed method favors the NoTL approach, aligning the results with NoTL for comparability. These observations are particularly noticeable in scenarios 2, 3, 4, 6, and 7.

However, exceptions exist, particularly in scenarios 1 and 5. The performance in scenario 1 exhibits instability. Upon investigating the root cause of this instability, we identify a potential factor: a violation of the fundamental assumption of TL, namely, the sharing of the same distribution between the

TABLE II
RMSEs VIA DIFFERENT SCENARIOS, KERNEL TYPES, AND MODELS

	Scenarios		1	2	3	4	5	6	7
Variance	Linear	NoTL	7525	1025	1037	1018	1050	1026	1019
		TL	17247	1129	1030	1015	1512	1017	1014
		Weight	0	0.0034	0	0	0.3368	0	0
		Proposal	**7525**	**1025**	1037	1018	1171	1026	1019
	Polynomial	NoTL	270	437	895	459	484	641	568
		TL	250	404	779	458	483	527	462
		Weight	0	0.0209	0	0	0.3997	0	0
		Proposal	270	436	895	459	483	641	568
	Rbf	NoTL	138	283	814	333	439	491	399
		TL	238	405	703	502	434	565	537
		Weight	0	0.0003	0	0	0.7194	0	0
		Proposal	**138**	**283**	814	**333**	435	491	**399**
	Laplace	NoTL	139	350	895	362	459	551	476
		TL	256	428	694	534	436	586	560
		Weight	0.1067	0.0041	0.0018	0.0004	0.8797	0.0005	0.0004
		Proposal	**139**	**350**	894	**362**	439	551	**476**
Discharge	Linear	NoTL	545	760	918	725	560	792	759
		TL	924	448	809	517	586	606	524
		Weight	0.9623	0.5315	0.2098	0.4951	0.9880	0.3529	0.4458
		Proposal	905	588	899	630	585	736	666
	Polynomial	NoTL	372	671	845	598	427	711	691
		TL	428	397	662	507	494	548	517
		Weight	0.8974	0.6159	0.5358	0.4432	0.9854	0.2546	0.4704
		Proposal	414	506	761	559	493	675	616
	Rbf	NoTL	197	449	721	458	427	561	533
		TL	254	425	687	531	429	581	557
		Weight	0.6902	0.2069	0.1471	0.5403	0.9272	0.0964	0.4891
		Proposal	219	442	716	498	429	**563**	544
	Laplace	NoTL	187	518	771	483	436	605	571
		TL	249	424	689	530	428	580	556
		Weight	0.0514	0.0318	0.0583	0.0205	0.9965	0.1143	0.0658
		Proposal	**186**	515	767	**484**	428	602	570
Full	Linear	NoTL	1144	520	846	497	496	583	580
		TL	945	365	660	431	484	486	461
		Weight	0.9892	0.9847	0.9806	0.6028	1.0000	0.9781	0.9635
		Proposal	947	366	665	455	484	488	466
	Polynomial	NoTL	650	514	815	526	428	634	617
		TL	467	327	651	491	466	509	489
		Weight	0.7882	0.8852	0.4078	0.5665	0.9958	0.6382	0.8614
		Proposal	494	347	757	506	466	557	508
	Rbf	NoTL	190	474	741	516	417	596	572
		TL	256	425	688	533	429	582	557
		Weight	0.1624	0.3549	0.0310	0.5318	0.9234	0.1696	0.2266
		Proposal	**195**	454	740	524	428	594	569
	Laplace	NoTL	149	568	804	565	419	649	631
		TL	251	424	690	531	426	582	557
		Weight	0.0190	0.0073	0	0	0.9993	0	0
		Proposal	**149**	567	804	565	426	649	631

TABLE III
P-VALUES FOR HYPOTHESIS TEST ON RESPONSES DISTRIBUTION HOMOGENEITY ACROSS SCENARIOS AND KERNEL TYPES

Scenario	Linear	Polynomial	Rbf	Laplace
1	0.0001	0.0001	0.0001	0.0001
2	0.1751	0.8388	0.1192	0.0967
3	0.0001	0.0001	0.0001	0.0001
4	0.0001	0.0308	0.0094	0.0028
5	0.5480	0.5828	0.1637	0.2371
6	0.0001	0.0153	0.0158	0.0067
7	0.0002	0.1548	0.0229	0.0107

responses of the source and target domains. As the performance of the proposed framework is influenced by kernel types, a kernel two-sample test introduced in [35] is employed. The p-values resulting from hypothesis tests across scenarios and kernel types, assessing whether the responses of the source and target domains share the same distribution, are presented in Table III. Taking scenario 2 as an example, the p-values across different kernel types are all larger than 0.01, indicating that we cannot reject the null hypothesis. This suggests that, in scenario 2, the response distribution in scenario 2 is not statistically different and likely shares the same distribution. Notably, for various kernel types, scenario 1 consistently rejects the null hypothesis that the responses of source and target domains share the same distribution.

For scenario 5, where the initial MMD value of the raw source and target data is already smaller enough, the results should align with the NoTL method, as $w = 0$ in this case. The results presented in Table II reflect the performance without considering this setting, confirming the necessity of the indicator function in (5). Moreover, results for scenario 5 indicate that when the raw source and target data derive a small initial MMD value, TL may provide unstable results, potentially worsening overall predictions.

TABLE IV
RMSEs VIA DIFFERENT SCENARIOS USING DIFFERENT APPROACHES UNDER THE FULL MODEL

	Scenarios	1	2	3	4	5	6	7
Linear	NoTL-KR	1144	520	846	497	496	583	580
	TL-KR	945	365	660	431	484	486	461
	Weight-KR	0.9892	0.9847	0.9806	0.6028	1.0000	0.9781	0.9635
	Proposal-KR	**947**	**366**	**665**	**455**	**484**	**488**	**466**
	NoTL-LSTM	331	202	646	216	124	227	236
	TL-LSTM	276	197	574	187	103	365	210
	Weight-LSTM	0.7174	0.8332	0.8297	0.3681	1.0000	0.0776	0.4477
	Proposal-LSTM	**290**	195	**587**	**185**	**103**	**234**	**213**
	NoTL-CNN	415	366	675	342	396	532	426
	TL-CNN	517	329	609	440	399	443	498
	Weight-CNN	0.3667	0.9824	0.9816	0.0754	0.9999	0.9757	0.1655
	Proposal-CNN	**439**	**328**	**610**	**342**	399	**445**	**431**
Polynomial	NoTL-KR	650	514	815	526	428	634	617
	TL-KR	467	327	651	491	466	509	489
	Weight-KR	0.7882	0.8852	0.4078	0.5665	0.9958	0.6382	0.8614
	Proposal-KR	**494**	**347**	**757**	**506**	466	**557**	**508**
	NoTL-LSTM	331	202	646	216	124	227	236
	TL-LSTM	459	256	634	388	99	333	336
	Weight-LSTM	0.1945	0.0062	0.2361	0.0035	0.6787	0.0279	0.0042
	Proposal-LSTM	**344**	**202**	643	**217**	**98**	**229**	**237**
	NoTL-CNN	415	366	675	342	396	532	426
	TL-CNN	321	404	639	468	399	506	546
	Weight-CNN	0.4718	0.0395	0.4130	0.0138	0.5025	0.6614	0.0193
	Proposal-CNN	**362**	**367**	660	**342**	402	**514**	**427**
Rbf	NoTL-KR	190	474	741	516	417	596	572
	TL-KR	256	425	688	533	429	582	557
	Weight-KR	0.1624	0.3549	0.0310	0.5318	0.9234	0.1696	0.2266
	Proposal-KR	**195**	**454**	740	524	428	594	569
	NoTL-LSTM	331	202	646	216	124	227	236
	TL-LSTM	402	374	617	502	191	394	566
	Weight-LSTM	0.0242	0.0220	0.0335	0	0.9205	0.0022	0
	Proposal-LSTM	**331**	**203**	645	**216**	185	**227**	**237**
	NoTL-CNN	415	366	675	342	396	532	426
	TL-CNN	304	411	623	477	394	577	533
	Weight-CNN	0.1584	0.3509	0.0076	0.0949	0.7795	0.0142	0.0746
	Proposal-CNN	**393**	379	674	**344**	393	**532**	**429**
Laplace	NoTL-KR	149	568	804	565	419	649	631
	TL-KR	251	424	690	531	426	582	557
	Weight-KR	0.0190	0.0073	0	0	0.9993	0	0
	Proposal-KR	**149**	567	804	565	426	649	631
	NoTL-LSTM	331	202	646	216	124	227	236
	TL-LSTM	477	400	643	573	193	547	499
	Weight-LSTM	0.0172	0.0085	0	0	0.9993	0	0
	Proposal-LSTM	331	**203**	646	**216**	193	**227**	**237**
	NoTL-CNN	415	366	675	342	396	532	426
	TL-CNN	308	405	649	514	398	516	530
	Weight-CNN	0.0168	0.0091	0	0	0.9938	0	0.0002
	Proposal-CNN	412	**367**	675	**342**	398	532	**426**

To further demonstrate the robustness of the proposed approach, we conducted additional experiments based on the *full* model, integrating the framework with LSTM and CNN-based techniques. The results, presented in Table IV, further highlight the robustness of the proposed framework. Specifically, "-KR," "-LSTM," and "-CNN" represent the corresponding results based on kernel regression, LSTM, and CNN, respectively. Regardless of the method used, the results in Table IV indicate that when TL is significantly impacted by negative transfer, our method tends to favor the NoTL approach, effectively minimizing the negative effects.

In terms of computational cost, the approach based on kernel regression took approximately 0.89 s per scenario, while the LSTM-based approach took around 5.59 s and the CNN-based approach around 4.05 s. This level of computational efficiency suggests that the proposed method is feasible for use in real-world battery management systems.

D. Sensitivity Analysis

According to the results, both the hyperparameter λ in (2) and the choice of kernel type significantly impact the performance.

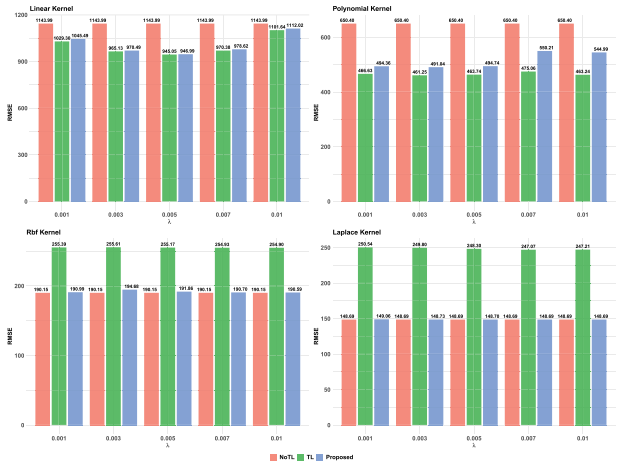

Fig. 5. Example of RMSEs for different methods via the tuning parameter λ and kernel types.

Given that selecting suitable kernel function types remains an ongoing area of research, we specifically focus on exploring the sensitivity associated with the hyperparameter λ. This uncertainty can influence the performance of both the TL and the proposed robust TL framework.

Taking the full model in scenario 1 as an example, we find that the optimal λs for four commonly used kernel types in the proposed robust TL framework fall within the range $[0.001, 0.01]$. Consequently, we present the RMSEs for different methods across various kernel types, with λ values set to 0.001, 0.003, 0.005, 0.007, and 0.01, in Fig. 5.

The results presented in Fig. 5 demonstrate that the proposed robust TL framework consistently achieves a balance between the NoTL and TL methods, irrespective of the kernel types. Notably, this balancing function remains unaffected by specific values of λ.

V. Conclusion

This study introduces a robust TL framework that balances the outcomes of NoTL and TL through a weight form associated with the MMD metric. The strength of the proposed framework lies in its flexibility, enabling the integration of various traditional machine learning algorithms within the TL process. The case analysis results, conducted on early cycle battery data for predicting lifespan, demonstrate that the proposed method tends to favor TL results when TL performs well and leans toward NoTL results when TL performance is suboptimal. This robustness is crucial in practical applications as TL can not guarantee consistently better performance.

This article proposes a novel analytical framework where the specific weight form and TL algorithm can be adapted based on the nature of the problem at hand. Thus, as for future research, one potential direction involves exploring alternative forms of weights and TL algorithms. Another possible research direction entails further investigation into the choice of kernel function types, aiming to provide more practical and tailored selection methods. In addition, potential optimizations, such as parallelizing computations, could further enhance the computational efficiency of the proposed method. Furthermore, improving model interpretability using techniques like Shapley additive explanations or local interpretable model-agnostic explanations could offer valuable insights for engineers. Exploring ensemble learning methods may also enhance accuracy by combining the strengths of multiple models.

Acknowledgment

The authors would like to thank the editor, the associate editor, and five anonymous referees for their valuable comments and constructive suggestions, which have significantly improved the quality and presentation of this work. The authors would also like to extend our heartfelt thanks to Femke Schürmann, a master's student from the Department of Applied Mathematics, Delft University of Technology, for her inspiring work. This research was partly influenced by her master's thesis.

References

[1] B. Dunn, H. Kamath, and J.-M. Tarascon, "Electrical energy storage for the grid: A battery of choices," *Science*, vol. 334, no. 6058, pp. 928–935, 2011.

[2] Y. Jiang, Y. Ke, F. Yang, J. Ji, and W. Peng, "State of health estimation for second-life lithium-ion batteries in energy storage system with partial charging-discharging workloads," *IEEE Trans. Ind. Electron.*, vol. 71, no. 10, pp. 13178–13188, Oct. 2024.

[3] Y. Ke, Y. Jiang, R. Zhu, W. Peng, and X. Tan, "Early prediction of knee point and knee capacity for fast-charging lithium-ion battery with uncertainty quantification and calibration," *IEEE Trans. Transport. Electrific.*, vol. 10, no. 2, pp. 2873–2885, Jun. 2024.

[4] M. Yang, X. Sun, R. Liu, L. Wang, F. Zhao, and X. Mei, "Predict the lifetime of lithium-ion batteries using early cycles: A review," *Appl. Energy*, vol. 376, 2024, Art. no. 124171.

[5] Q. Zhang, C.-G. Huang, H. Li, G. Feng, and W. Peng, "Electrochemical impedance spectroscopy based state-of-health estimation for lithium-ion battery considering temperature and state-of-charge effect," *IEEE Trans. Transport. Electrific.*, vol. 8, no. 4, pp. 4633–4645, Dec. 2022.

[6] R. Zhu, W. Peng, D. Wang, and C.-G. Huang, "Bayesian transfer learning with active querying for intelligent cross-machine fault prognosis under limited data," *Mech. Syst. Signal Process.*, vol. 183, 2023, Art. no. 109628.

[7] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

[8] Y. Qin, S. Adams, and C. Yuen, "Transfer learning-based state of charge estimation for lithium-ion battery at varying ambient temperatures," *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7304–7315, Nov. 2021.

[9] Y. Qin, C. Yuen, X. Yin, and B. Huang, "A transferable multistage model with cycling discrepancy learning for lithium-ion battery state of health estimation," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 1933–1946, Feb. 2023.

[10] K. A. Severson et al., "Data-driven prediction of battery cycle life before capacity degradation," *Nature Energy*, vol. 4, no. 5, pp. 383–391, 2019.

[11] Z. Tong, J. Miao, S. Tong, and Y. Lu, "Early prediction of remaining useful life for lithium-ion batteries based on a hybrid machine learning method," *J. Cleaner Prod.*, vol. 317, 2021, Art. no. 128265.

[12] X. Pang et al., "A novel hybrid model for lithium-ion batteries lifespan prediction with high accuracy and interpretability," *J. Energy Storage*, vol. 61, 2023, Art. no. 106728.

[13] Y. Yang, "A machine-learning prediction method of lithium-ion battery life based on charge process for different applications," *Appl. Energy*, vol. 292, 2021, Art. no. 116897.

[14] S. Kim, H. Jung, M. Lee, Y. Y. Choi, and J.-I. Choi, "Model-free reconstruction of capacity degradation trajectory of lithium-ion batteries using early cycle data," *ETransportation*, vol. 17, 2023, Art. no. 100243.

[15] J. Chen, R. Huang, Z. Chen, W. Mao, and W. Li, "Transfer learning algorithms for bearing remaining useful life prediction: A comprehensive review from an industrial application perspective," *Mech. Syst. Signal Process.*, vol. 193, 2023, Art. no. 110239.

[16] 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. Transport. Electrific.*, vol. 10, no. 1, pp. 1465–1481, Mar. 2024.
[17] B. Jia, Y. Guan, and L. Wu, "A state of health estimation framework for lithium-ion batteries using transfer components analysis," *Energies*, vol. 12, no. 13, 2019, Art. no. 2524.
[18] Y. Li, H. Sheng, Y. Cheng, D.-I. Stroe, and R. Teodorescu, "State-of-health estimation of lithium-ion batteries based on semi-supervised transfer component analysis," *Appl. Energy*, vol. 277, 2020, Art. no. 115504.
[19] Z. Jiao et al., "LightGBM-based framework for lithium-ion battery remaining useful life prediction under driving conditions," *IEEE Trans. Ind. Informat.*, vol. 19, no. 11, pp. 11353–11362, Nov. 2023.
[20] Y. Jiang, Y. Chen, F. Yang, and W. Peng, "State of health estimation of lithium-ion battery with automatic feature extraction and self-attention learning mechanism," *J. Power Sources*, vol. 556, 2023, Art. no. 232466.
[21] S. Su, W. Li, J. Mou, A. Garg, L. Gao, and J. Liu, "A hybrid battery equivalent circuit model, deep learning, and transfer learning for battery state monitoring," *IEEE Trans. Transport. Electrific.*, vol. 9, no. 1, pp. 1113–1127, Mar. 2023.
[22] D. Pan, H. Li, and S. Wang, "Transfer learning-based hybrid remaining useful life prediction for lithium-ion batteries under different stresses," *IEEE Trans. Instrum. Meas.*, vol. 71, 2022, Art. no. 3501810.
[23] Y. Tan and G. Zhao, "Transfer learning with long short-term memory network for state-of-health prediction of lithium-ion batteries," *IEEE Trans. Ind. Electron.*, vol. 67, no. 10, pp. 8723–8731, Oct. 2020.
[24] Z. Deng, X. Lin, J. Cai, and X. Hu, "Battery health estimation with degradation pattern recognition and transfer learning," *J. Power Sources*, vol. 525, 2022, Art. no. 231027.
[25] C. Dai Nguyen and S. J. Bae, "Equivalent circuit simulated deep network architecture and transfer learning for remaining useful life prediction of lithium-ion batteries," *J. Energy Storage*, vol. 71, 2023, Art. no. 108042.
[26] 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, 2022, Art. no. 230823.
[27] 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, Mar. 2022.
[28] G. Ma et al., "A transfer learning-based method for personalized state of health estimation of lithium-ion batteries," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 1, pp. 759–769, Jan. 2024.
[29] 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, Mar. 2023.
[30] A. A. Chehade and A. A. Hussein, "A collaborative Gaussian process regression model for transfer learning of capacity trends between li-ion battery cells," *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 9542–9552, Sep. 2020.
[31] I. Oyewole, A. Chehade, and Y. Kim, "A controllable deep transfer learning network with multiple domain adaptation for battery state-of-charge estimation," *Appl. Energy*, vol. 312, 2022, Art. no. 118726.
[32] Y. Ding, M. Jia, Q. Miao, and P. Huang, "Remaining useful life estimation using deep metric transfer learning for kernel regression," *Rel. Eng. Syst. Saf.*, vol. 212, 2021, Art. no. 107583.
[33] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, 2008.
[34] T. Ando and K.-C. Li, "A model-averaging approach for high-dimensional regression," *J. Amer. Stat. Assoc.*, vol. 109, no. 505, pp. 254–265, 2014.
[35] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, no. 25, pp. 723–773, 2012.

Wenda Kang received the bachelor's degree in mathematics and applied mathematics from the Ocean University of China, Qingdao, China, in 2018. He is currently working toward the Ph.D. degree in statistics with the Delft Institute of Applied Mathematics, Delft University of Technology, Delft, The Netherlands.

His research interests include, reliability engineering, transfer learning, and applied statistics.

Dianpeng Wang received the bachelor's, master's, and the Ph.D. degrees in statistics from the Beijing Institute of Technology, Beijing, China, in 2007, 2010, and 2016, respectively.

He is currently an Associate Professor with the School of Mathematics and Statistics, Beijing Institute of Technology. His research interests include computer experiment design, Bayesian calculation, uncertainty quantification, and industrial big data analysis.

Geurt Jongbloed received the master's degree in applied mathematics and the Ph.D. degree in statistics from the Delft University of Technology, Delft, The Netherlands, in 1991 and 1995, respectively.

He is currently a Full Professor with the Delft Institute of Applied Mathematics, Delft University of Technology. His research interests include inverse problems, statistical inference, nonparametric estimation, and computational statistics.

Jiawen Hu received the bachelor's degree in mechanical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2009, the master's degree in mechanical engineering from the Chinese Academy of Science, Beijing, China, in 2012, and the Ph.D. degree in industrial engineering from Shanghai Jiao Tong University in 2017.

He is currently an Associate Professor with the School of Aeronautics and Astronautics, University of Electronic Science and Technology of China, Chengdu, China. His research interests include reliability engineering and maintenance planning.

Piao Chen (Member, IEEE) received the bachelor's degree in industrial engineering from Shanghai Jiao Tong University, Shanghai, China, in 2013 and the Ph.D. degree in industrial and systems engineering management from the National University of Singapore, Singapore, in 2017.

He is currently an Associate Professor with the ZJU-UIUC Institute, Zhejiang University, Hangzhou, China. During 2020–2023, he was an Assistant Professor with the Delft Institute of Applied Mathematics, Delft University of Technology, Delft, The Netherlands. His research interests include reliability engineering, statistical learning, and operations research.