Battery Health Estimation Based on Multidomain Transfer Learning

Hanmin Sheng, *Member, IEEE*, Biplob Ray, *Senior Member, IEEE*, Shaben Kayamboo, *Student Member, IEEE*, Xintao Xu, *Member, IEEE*, and Shafei Wang

***Abstract*—Machine learning methods are expected to play a significant role in battery state of charge (SOH) estimation, leveraging their strengths in self-learning and nonlinear fitting. One of the key challenges in SOH estimation is the concept drift issue, which refers to changes in the data distribution between the training and test datasets. General machine learning methods assume that the training data shares similar characteristics with the test data. However, in SOH estimation tasks, differences in the environment and the characteristics of the battery itself can cause concept drift, which then impacts the model's effectiveness. As a result, many data-driven models that perform well in laboratory conditions struggle to be applied to other target batteries. This is a common and significant battery diagnosis technology issue, yet it remains unresolved. This article proposes a multidomain transfer Gaussian process regression (MTR-GPR) SOH estimation approach to address this issue. In this model, training data do not directly participate in the model's learning process. Instead, the MTR-GPR model extracts information from different datasets based on the distribution similarity. This method can fully use multisource battery ageing data while reducing the negative impact of distribution differences. Experimental results prove that MTR-GPR can make reliable SOH estimates with only 20% of target battery data. On the other hand, this method can provide the posterior probability distribution of the prediction results.**

***Index Terms*—Gaussian process regression (GPR), learning aided monitoring, state of health (SOH), transfer learning.**

I. Introduction

DUE to energy shortages and environmental concerns, the electric vehicle (EV) industry has rapidly developed in the last ten years. The energy storage system is critical to EV's competitiveness. Lithium-ion batteries have several advantages over other electrochemical energy storage technologies, including high energy density, high power density, low self-discharge rate, long battery life, and excellent transient response characteristics. As a result, lithium battery has been placed in high hopes in the automotive industry. The power battery pack of electric vehicles is composed of a large number of battery cells; therefore, monitoring the health status of the battery is crucial for devising battery balancing and charging/discharging strategies, which ultimately impacts the efficiency and safety of the battery pack. Battery state of health (SOH) is a measurement that reflects the batteries' health status. SOH is frequently defined as the ratio of the available capacity to the rated capacity [1]. Ampere-hour (Ah) capacity testing a widely used SOH estimation approach. But this approach is both time-consuming and energy-consuming [2]. Recently, a considerable amount of literature has emerged on SOH estimation. These methods can be broadly categorized into three main categories: the electrochemical model [3], [4], [5], the equivalent circuit model (ECM) [6], [7], [8], [9], [10], and the data-driven approach. In recent years, data-driven technologies, represented by machine learning, have advanced rapidly. In contrast to electrochemical methods and ECM models, machine learning methods can self-learn during deployment. They possess the excellent nonlinear fitting ability, accommodate multiple features, and dramatically lessen the manual burden of modeling. Moreover, the early identification and exploration of numerous nondestructive testing features [health indexes (HIs)], including the incremental capacity (IC) curve [11], [12], differential voltage [13], ageing cycles [14], sample entropy [15], the interval of equal discharging voltage difference [16], and many others, offer favorable conditions for data-driven modeling.

Artificial neural networks (ANNs), support vector machines (SVMs), and Gaussian process regression (GPR) are popular methods used in SOH research, each providing distinct advantages. ANNs have strong nonlinear approximation ability and flexible structure. Many algorithms have been developed based on neural networks. A recurrent neural network (RNN) approach has been reported in [17], and the long short-term memory (LSTM) technique has been utilized to solve the problem of gradient disappearance in SOH estimation. A convolutional neural network (CNN)-based algorithm has been proposed in [18]. A significant disadvantage of the ANN-based approach is that it is prone to overfit in small sample scenarios. A battery ageing

Manuscript received 7 September 2023; revised 16 November 2023; accepted 16 December 2023. Date of publication 25 December 2023; date of current version 16 February 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62373081 and in part by the China Postdoctoral Science Foundation under Grant 2021M690560. Recommended for publication by Associate Editor Xinke Wu. *(Corresponding author: Biplob Ray.)*

Hanmin Sheng and Xintao Xu are with the School of Automation Engineering, University of Science and Technology, Chengdu 611731, China (e-mail: hmsheng@uestc.edu.cn; 202222060441@std.uestc.edu.cn).

Biplob Ray and Shaben Kayamboo are with the Centre for Machine Learning Networks and Educational Technologies, College of ICT, School of Engineering and Technology, CQUniversity, Melbourne, VIC 3000, Australia (e-mail: b.ray@cqu.edu.au; shaben.kayamboo@cqumail.com).

Shafei Wang is with the Laboratory of Electromagnetic Space Cognition and Intelligent Control, Beijing 065001, China (e-mail: rockingsandstorm@163.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TPEL.2023.3346335.

Digital Object Identifier 10.1109/TPEL.2023.3346335

model based on extreme learning machine has been established in reference [19] as a variant of ANN, it is efficient, but its accuracy and stability have been questioned. Compared to ANNs, SVM performs better in small sample fitting [20]. A least square support vector machine (LSSVM) SOH estimation algorithm has been reported in [21], a kernel function extension form of the least square regression method with high robustness. GPR is a nonparametric method that permits a model expressively calibrated to the data requirement [22]. GPR provides the estimate's posterior probability, which helps to avoid the wrong decisions caused by prediction errors [23].

To summarize, data-driven modeling has been a hot topic in the research field of SOH detection in recent years. However, despite the high expectations for machine learning-based battery testing technology, many issues still need to be solved in applying SOH detection using machine learning in industrial applications. One of the key obstacles is the generalization problem of data-driven models in different external environments, different charging and discharging profiles, and different battery objects. Unlike natural language processing and image recognition, where researchers can quickly obtain massive and diverse training data, data available for battery model training is relatively limited. The valuable but limited ageing experimental data cannot cover complex application scenarios, which means there will always be a difference in distribution between laboratory data and actual battery aging data when applying such models. As a result, it affects the effectiveness of SOH estimation models. Transfer learning is a practical approach to addressing this situation. By transferring knowledge from a source domain with rich data to a target domain with limited data, transfer learning can improve the performance of models in the target domain.

Transfer learning is a novel learning paradigm that leverages knowledge acquired from one domain to enhance the performance of a model in a different domain. In the field of machine learning, in terms of knowledge transfer forms, transfer learning is primarily classified into three research directions: instance-based transfer learning [24], [25], feature-based transfer learning [26], [27], and model-based transfer learning [28], [29], [29], [30]. Research on transfer learning for SOH estimation is still in its infancy, as papers applying transfer learning to SOH have only been published over the past three years. It is common practice to transplant successful transfer learning algorithms from other domains to SOH estimation. Within the machine learning community, deep learning-based transfer learning, which leverages pretraining and fine-tuning, has gained popularity as a knowledge transfer method. This approach has also been extensively documented in the literature on SOH estimation. Initially, the model is trained using source domain data. The trained model is then utilized as the initial model for the target domain, with the model being further refined with a small amount of target domain data. This allows for information transfer through the deep learning architecture, which can be particularly useful when limited target domain data are available [31]. Different deep learning algorithms have been used for model transfer in various literature, including RNNs mentioned in [23], [43], LSTM networks in [32], [33], [34], [35], capsule networks in [36], and CNNs, as discussed in [37]. [38] extended the pretraining and fine-tuning transfer learning paradigm and used a model-agnostic metalearning algorithm to find a parameter set sensitive to task changes. Apart from using the pretraining and fine-tuning strategy, literature [39] incorporated both the classification loss and domain adaptation loss between the source and target domains in their adopted ResNet-50 deep learning network structure. In [40], a transfer learning architecture was proposed that uses two LSTM networks—one to learn shared prediction ability from the source domain and another LSTM trained with a small amount of target data. Two real-time adjusting factors are used to control the influence of source domain information on the model weights of the two submodels. In [41] and [42], Ye used adversarial learning to guide feature generators into providing domain-invariant features. The unsupervised feature alignment metric and maximum mean discrepancy were used to evaluate domain variance. In [43], a siamese network was used to implement few-shot learning, while in [44], a bootstrapped random vector functional link model was used to estimate the importance weights through kernel mean matching. These weights were controlled to minimize the MMD distance. [45] employed a GAN-based network architecture to generate virtual data using an equivalent circuit model. GAN is then used to differentiate between virtual and actual data. Correlation alignment was used in the learning process to align the second-order statistics of the data distribution in the source and target domains.

There are few transfer learning algorithms specifically developed for SOH problems. The standard approach is to apply transfer learning algorithms that have been successful in other fields to battery SOH detection. Many studies have used deep learning architectures for SOH detection. Still, in reality, CNN-based deep learning architectures typically require high-dimensional raw feature data and spatial correlations between features. In contrast, RNN-based architectures hope for weaker randomness between sequential data. However, under complex operating conditions, randomness is unavoidable. In addition, CNN and RNN require massive amounts of data for training, making them less suitable for SOH detection. In our earlier research, we attempted to develop transfer learning algorithms suitable for SOH detection for low-dimensional and small data problems. In [46], we proposed a TCA algorithm and [47] a cross-manifold embedding algorithm, which views the source and target domains as two different manifolds and designs a mapping relationship to transplant information across different domains. In our early paper [48], the Gaussian process transfer algorithm performs better in small sample generalization than deep learning. However, the problem that our earlier research could not break through is the difficulty of solving multidomain transfer problems. Compared with single-source domain transfer, multidomain transfer models can obtain knowledge applicable to the target domain from multiple datasets, greatly expanding the source of information.

Previously, scholars have proposed several transfer learning methods that can be applied to SOH detection. Nevertheless, current research mainly focuses on single-source domain transfer, which encounters challenges when the source and target

domain data exhibit significant differences in feature distribution. The reusable data information that can be transferred will be extremely limited, while the risk of negative transfer increases. The multidomain transfer is highly attractive, enabling knowledge transfer from multiple datasets collected across diverse application scenarios, serving as multiple source domains. This model can autonomously regulate the impact of distinct source domains on the training process, thereby significantly amplifying the extent of accessible source domain information. Furthermore, it can mitigate adverse transfer risks by minimizing the influence of data sources that exhibit notable distribution discrepancies with the target domain.

This article proposes a novel multidomain transfer learning approach based on GPR. First, by including a regularization term in the Gaussian kernel, the regularization term controls the contribution of the source domain data in the learning process, thereby avoiding the negative transfer problem caused by the significant difference in the characteristics of the source domain data and the target domain data. On this basis, knowledge from various data domains is fused using an ensemble framework. This method does not require manual data source selection. Helpful information can be screened from multiple data sources, making reliable SOH estimations based on a small amount of target battery data. The rest of this article is organized as follows. Section II introduces the multidomain transfer Gaussian process regression algorithm (MTR-GPR). Section III verifies the estimation effect with battery ageing data with several similar data-driven algorithms introduced for comparison. Finally, Section IV concludes this article. Relevant certification is provided in the Appendix.

II. Transfer Gaussian Process Regression (Tr-GPR)

In this section, we detailed a brief problem statement in Section II-A followed by a detailed derivation of the Tr-GPR in Section II-B followed by MTR-GPR derivation in Section II-C and deployment process of MTR-GPR in Section II-D.

A. Dilemma of Model Cross-Target Transplantation

The target battery data may exhibit distributional disparities when applying machine learning techniques for SOH detection. Such differences can have an impact on the reliability of the algorithm. In this section, we illustrate this phenomenon using a straightforward example. Fig. 1 shows the distribution of two different datasets [CX2_36 and CS2_36 provided by the National Aeronautics and Space Administration (NASA)], where the x-axis is the input feature (cycle number), and the y-axis is the reference SOH value. Suppose a data-driven model is trained with dataset CS2_36 (the red samples in Fig. 1); for a valid model, the estimation function should satisfy most red samples. At this time, the function will be invalid for the gray samples. This is a practical problem for the data-driven SOH estimation models, but little attention has been paid to this issue.

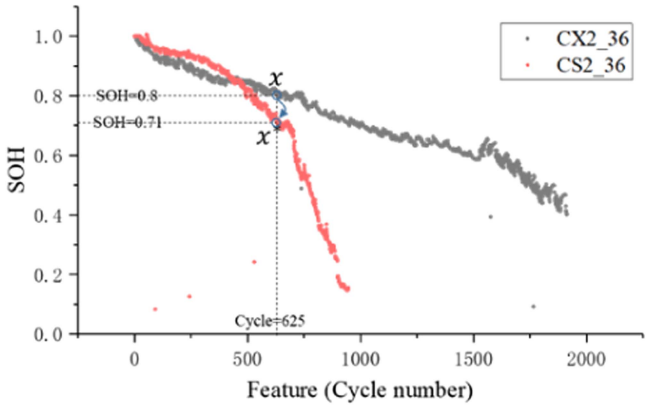

Fig. 1. Differences in data distribution between two datasets.

B. Transfer Gaussian Process Regression

We presented a transfer SOH estimation method based on GPR to address the issue mentioned earlier. TR-GPR introduces a scalar matrix into the covariance matrix of GPR to control the influence of the source domain on the target domain. When there is a significant distribution difference between the source and target domains, the TR-GPR can adjust the parameters in the scalar matrix to reduce the negative transfer problem. The Gaussian is a stochastic process, every finite collection of which has a joint Gaussian distribution. Due to the above-mentioned properties, a Gaussian function can be defined by its mean and variance as

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})] \tag{1}$$

$$k(\mathbf{x}, \mathbf{x}) = \mathbb{E}\left[\left(f(\mathbf{x}) - m(\mathbf{x})\right)\left(f(\mathbf{x}^T) - m(\mathbf{x}^T)\right)\right] \tag{2}$$

where $m(\mathbf{x})$ is the mean function, $f(\mathbf{x})$ denotes the real SOH value, and $\mathbf{x}$ is the input vector. When using the GPR model for SOH estimation, SOH is the target of the model's prediction, and the input matrix $\mathbf{x}$ consists of battery HIs provided by users. GPR aims to make the mean function close to the real process. The validity of HIs directly affects the effectiveness of the GPR model. A Gaussian process model is typically described as

$$f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k\left(\mathbf{x}, \mathbf{x}'\right)\right). \tag{3}$$

When performing SOH prediction, the user inputs the matrix $\mathbf{x}$ to the GPR model, which then provides an estimated SOH in the form of the mean function and the corresponding posterior probability distribution of the estimate in the form of variance. With TR-GPR, a natural idea is to construct a unified covariance matrix to control the correlation between data from different source domains and the target domain, affecting the degree of knowledge transfer from other source domains to the target domain. By constructing a covariance matrix over different source data and target data, it is possible to know which data played a leading role in the training process when performing the posterior probability estimation, which data information was masked, and thereby provide some degree of interpretability. When multiple HIs are used as the input features for SOH estimation, the feature order of the source data and the target

data should be consistent. The estimation $f(\mathbf{x})$ and the real SOH value y have the relationship as derived in the following:

$$y = f(\mathbf{x}) + \varepsilon \tag{4}$$

where $\varepsilon\mathcal{N}(0,\sigma^2)$ is the noise. The following equation denotes the prior relationship of observations:

$$\operatorname{cov}(y_i, y_j) = k(\mathbf{x}_i, \mathbf{x}_j) + \sigma^2\delta_{i,j}. \tag{5}$$

In which δ_{pq} is a Kronecker delta, $\delta_{pq} = 1$ iff $i = j$, elsewise $\delta_{pq}= 0$. $k()$ is the covariance function. This article uses the squared exponential (SE) covariance function, which takes the form

$$k(\mathbf{x}_i, \mathbf{x}_j) = \alpha^2 \exp\left(-\frac{1}{2l^2}(\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)\right). \tag{6}$$

For the SE covariance function, α and l are hyperparameters. $k(\mathbf{x}_i, \mathbf{x}_j)$ is negative related to the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. The general GPR typically only partitions the data into training and testing sets. The training set is employed to train the model, while the testing set constitutes the target data to be evaluated. Unlike the conventional GPR setting, we further segment the training set into source and target domain data in TR-GPR. The target domain data originates from a limited number of battery samples comparable to the test case, often rendering such data insufficient. The source domain data may originate from diverse batteries or distinct scenarios; it is plentiful but may differ from the target scenario. TR-GPR utilizes the target domain data to assess the similarity of the data distribution between the source domain and the target domain, thereby determining the weight of influence of the source domain data in the learning process. Suppose the target data $\mathcal{D}_T = \{\mathbf{X}_T, \mathbf{Y}_T\}(\mathbf{X}_T \in \mathbb{R}^{n\times d}, Y_T \in \mathbb{R}^{m\times 1})$ is obtained from the target battery. Source data $\mathcal{D}_S = \{\mathbf{X}_S, \mathbf{Y}_S\}(\mathbf{X}_S \in \mathbb{R}^{n\times d}, Y_S \in \mathbb{R}^{n\times 1})$. GPR constructs a covariance matrix overall data, which implies a consistency assumption, which can be expressed as

$$\begin{bmatrix}\mathbf{Y}_S\\ \mathbf{Y}_T\end{bmatrix} \mathcal{N}\left(\mathbf{0}, \begin{bmatrix}\mathbf{K}(\mathbf{X}_S, \mathbf{X}_S) + \sigma_S^2\mathbf{I} & \mathbf{K}(\mathbf{X}_S, \mathbf{X}_T)\\ \mathbf{K}(\mathbf{X}_T, \mathbf{X}_S) & \mathbf{K}(\mathbf{X}_T, \mathbf{X}_T)\sigma_T^2\mathbf{I}\end{bmatrix}\right). \tag{7}$$

$K(\mathbf{X}_p, \mathbf{X}_q)$ is the covariance matrix over $\mathcal{D}_p$ and $\mathcal{D}_q$.σ_S^2 and σ_T^2 represent the source domain and target domain variance, respectively. However, in cross-object forecasting, consistency assumptions do not always hold. When there are differences in data distribution, the model should prevent the source domain from adversely affecting the estimation. A transfer covariance function is derived in 8 to achieve this goal

$$\begin{bmatrix}\mathbf{Y}_S\\ \mathbf{Y}_T\end{bmatrix} \mathcal{N}\left(\mathbf{0}, \begin{bmatrix}\boldsymbol{\lambda}\left(\mathbf{K}(\mathbf{X}_S, \mathbf{X}_S) + \sigma_n^2 I\right) & \lambda\mathbf{K}(\mathbf{X}_S, \mathbf{X}_T)\\ \mathbf{K}(\mathbf{X}_T, \mathbf{X}_S)\boldsymbol{\lambda}^T & \mathbf{K}(\mathbf{X}_T, \mathbf{X}_T)\end{bmatrix}\right). \tag{8}$$

In which $\mathbf{X}_S$ is the source feature vector, $\mathbf{X}_T$ is the target feature vector, $\boldsymbol{\lambda}$ is a scalar matrix with elements λ on the diagonal ($\lambda \in (0, 1)$). The regularization item controls the correlation between the source and target domains. When the source domain plays a negative role in modeling, adjusting the transfer factor can reduce the correlation between the source domain and the target domain. When $\lambda = 1$, the source domain will directly participate in the training process; in this case, the transfer GPR model is equivalent to the classical GPR. Based on (7), we have derived the following:

$$p(\mathbf{Y}_T|\mathbf{Y}_S, \mathbf{X}_T, \mathbf{X}_S)\mathcal{N}(\mathbf{m}(\mathbf{Y}_T), \mathbf{var}(\mathbf{Y}_T)) \tag{9}$$

$$\begin{cases}\mathbf{m}(\mathbf{f}_T) = \mathbf{K}(\mathbf{X}_T, \mathbf{X}_S)\lambda^T\left(\boldsymbol{\lambda}\left(\mathbf{K}(\mathbf{X}_S, \mathbf{X}_S) + \sigma_S^2\mathbf{I}\right)\right)^{-1}\mathbf{Y}_S\\ \operatorname{var}(\mathbf{f}_T) = \left(\mathbf{K}(\mathbf{X}_T, \mathbf{X}_T) + \sigma_T^2\mathbf{I}\right) - \mathbf{K}(\mathbf{X}_T, \mathbf{X}_S)\boldsymbol{\lambda}^T\times\\ \qquad\left(\lambda\left(\mathbf{K}(\mathbf{X}_S, \mathbf{X}_S) + \sigma_S^2\mathbf{I}\right)\right)^{-1}\lambda\mathbf{K}(\mathbf{X}_S, \mathbf{X}_T).\end{cases} \tag{10}$$

Under the GPR prior, the marginal likelihood function has the form of

$$\begin{aligned}\log p(\mathbf{Y}_T|\boldsymbol{\theta}) = &-\frac{1}{2}\log|\mathbf{cov}(\mathbf{f}_T)| - \frac{1}{2}(\mathbf{Y}_T - \mathbf{m}(\mathbf{Y}_T))^T\\ &\times \mathbf{var}(\mathbf{f}_T)^{-1}(\mathbf{Y}_T - \mathbf{m}(\mathbf{Y}_T)) - \frac{n}{2}\log(2\pi)\end{aligned} \tag{11}$$

where $\boldsymbol{\theta} = \{l, \lambda, \alpha, \sigma_S, \sigma_T\}$ is the optimal hyperparameters which can be obtained by maximum (11) with gradient descent approach. When predicting a new input $\mathbf{x}_*$ a joint distribution can be constructed over the source domain, target domain and the input data as derived in the following:

$$[\boldsymbol{Y}_S, \boldsymbol{Y}_T, \boldsymbol{f}_*]^T \sim \mathcal{N}(0, \widehat{\boldsymbol{K}}) \tag{12}$$

where (13) shown at the bottom of this page.

Conditioning the prior distribution on observations, the estimation result of MTR-GPR takes the form

$$\mathbf{f}_* \mid \mathbf{x}_*, \mathbf{X}_T, \mathbf{X}_S\mathcal{N}(\mathbf{m}(\mathbf{f}_*), \mathbf{cov}(\mathbf{f}_*)) \tag{14}$$

$$\begin{cases}\mathbf{m}(\mathbf{f}_*) = \hat{\mathbf{K}}(\mathbf{x}_*, \mathbf{X})[\hat{\mathbf{K}}(\mathbf{X}, \mathbf{X})]^{-1}\mathbf{Y}\\ \operatorname{var}(\mathbf{f}_*) = \hat{\mathbf{K}}(\mathbf{x}_*, \mathbf{x}_*) - \hat{\mathbf{K}}(\mathbf{x}_*, \mathbf{X})[\hat{\mathbf{K}}(\mathbf{X}, \mathbf{X})]^{-1}\hat{K}(\mathbf{X}, \mathbf{x}_*).\end{cases} \tag{15}$$

When the user inputs $\mathbf{x}_*$, which represents the HIs of the target battery, GPR infers the posterior probability distribution of SOH based on the prior data distribution. Finally, the estimated result of SOH output by GPR is stored in the variable $\mathbf{f}*$, in equation where values are derived from (16), (17), (18), and (19) as follows:

$$\hat{\mathbf{K}}(\mathbf{x}_*, \mathbf{X}) = [\lambda\mathbf{K}(\mathbf{x}_*, \mathbf{X}_S), \quad \mathbf{K}(\mathbf{x}_*, \mathbf{X}_T)] \tag{16}$$

$$\hat{\mathbf{K}}(\mathbf{X}, \mathbf{X}) = \begin{bmatrix}\left(\boldsymbol{\lambda}\mathbf{K}(\mathbf{X}_S, \mathbf{X}_S) + \sigma_S^2\mathbf{I}\right) & \boldsymbol{\lambda}\mathbf{K}(\mathbf{X}_S, \mathbf{X}_T)\\ \boldsymbol{\lambda}\boldsymbol{K}(\mathbf{X}_T, \mathbf{X}_S) & \mathbf{K}(\mathbf{X}_T, \mathbf{X}_T) + \sigma_T^2\mathbf{I}\end{bmatrix} \tag{17}$$

$$\hat{\mathbf{K}}(\mathbf{X}, \mathbf{x}_*) = [\lambda\mathbf{K}(\mathbf{x}_S, \mathbf{x}_*), \quad \mathbf{K}(\mathbf{x}_T, \mathbf{x}_*)]^T \tag{18}$$

$$\mathbf{Y} = [\mathbf{Y}_S, \mathbf{Y}_T]^T. \tag{19}$$

C. MTR-GPR Learning

For single-source domain transfer learning, if the distribution of the source domain and target domain are quite different, the information that can be transferred will be extremely limited. Therefore, the transfer GPR is extended to the multidomain

transfer learning in this section. An intuitive way is to incorporate multidomain data into one covariance matrix in (20) shown at the bottom of this page.

If individual weights can be assigned to different covariance matrix blocks, then the influence weights of different source domains on the prediction results can be independently controlled. However, in GPR, the covariance function needs to be semidefinite, which requires $\boldsymbol{\lambda}_1 = \boldsymbol{\lambda}_2 =, \ldots, = \boldsymbol{\lambda}_k$, ($(\boldsymbol{\lambda}_i \in (0,1), i = 1, 2, \ldots, k)$ the proof is in Appendix A). Instead, different subtransfer GPRs are integrated as an ensemble structure. The submodels with a higher confidence level are assigned with higher weights. For the ith submodel, the weight can be expressed as

$$\omega_i = \frac{\left(\mathbf{var}_i\left(\mathbf{f}_*\right)^{-1/2}\right)}{\left(\sum_i \mathbf{var}_i\left(\mathbf{f}_*\right)^{-1/2}\right)}. \tag{21}$$

Suppose the prediction output of the ith model is $\mathbf{f}_* \mid \mathbf{x}_*, \mathbf{X}_T, \mathbf{X}_{S_i}, \mathbf{f}\mathcal{N}(\mathbf{m}_i(\mathbf{f}_*), \mathbf{var}_i(\mathbf{f}_*))$, and the submodels are independent, the output of MTR-GPR model can be expressed in the following:

$$\mathbf{f}_* \mid \mathbf{x}_*, \mathbf{X}_T, \mathbf{X}_{S_i}, \ldots, \mathbf{X}_{S_n} \mathcal{N}\left(\mathbf{m}_{\text{ensemble}}\left(\mathbf{f}_*\right), \mathbf{var}_{\text{ensemble}}\left(\mathbf{f}_*\right)\right) \tag{22}$$

$$\begin{cases} \mathbf{m}_{\text{ensemble}}\left(\mathbf{f}_*\right) = \sum_i \omega_i \mathbf{m}_i\left(\mathbf{f}_*\right) \\ \mathbf{var}_{\text{ensemble}}\left(\mathbf{f}_*\right) = \sum_i \omega_i^2 \mathbf{var}_i\left(\mathbf{f}_*\right). \end{cases} \tag{23}$$

The computational complexity of GPR's model is closely related to the number of elements in the covariance matrix. The computational complexity of the classic GPR model is $\mathcal{O}(\frac{n^3}{6})$ [49], where $\mathbf{n}$ refers to the number of samples that form the covariance matrix. Assume there are one test sample and k source domains, and each has $n_S^{(k)}$ samples, if all the samples are included in a unified covariance matrix, the computational complexity of GPR is $\mathcal{O}(\frac{(n_S^{(1)}+n_S^{(2)}+,\ldots,+n_S^{(k)}+1)^3}{6})$. While the computational complexity of the MTR-GPR is $\mathcal{O}(\frac{\sum_i (n_S^{(i)}+1)^3}{6})$ (with Cholesky decomposition). The flow chart of MTR-GPR is illustrated in Fig. 2.

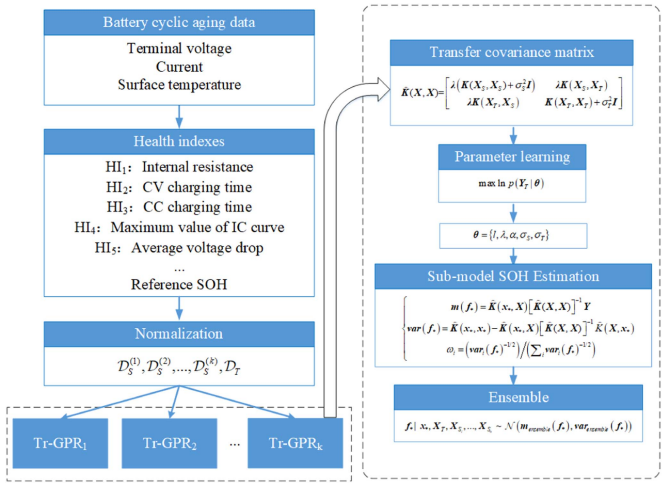

Fig. 2. Flow chart of multidomain transfer GPR.

D. Deployment Process of MTR-GPR

This article proposes a multisource domain transfer learning approach, MTR-GPR, for SOH estimation. The specific deployment process of this algorithm is as follows.

1) Collect charging and discharging data of batteries. These data can come from different batteries and different usage environments. The voltage and current curves during the charging and discharging process should be recorded, and the label value $\mathbf{y}$ can be calculated as a reference value for SOH using the time integral method. In addition, to utilize multidomain transfer learning, a small amount of target battery data should be obtained for knowledge screening.
2) Extract battery health factors from the auxiliary and target battery data, such as internal resistance, maximum IC curve, charging voltage, and charging time for constant voltage (CV) and constant current (CC). Suppose random charging and discharging data are used. In that case, it is necessary to prepare stable features that can still be obtained under alternating charging and discharging conditions as the HIs for the model, such as resistance at a specific SOC and HPPC pulse data. The source domain data and target domain data should use the same HIs.

$$\hat{\mathbf{K}} = \begin{bmatrix} \boldsymbol{\lambda}\left(\mathbf{K}\left(\mathbf{X}_S, \mathbf{X}_S\right) + \sigma_S^2 \mathbf{I}\right) & \boldsymbol{\lambda}\mathbf{K}\left(\mathbf{X}_S, \mathbf{X}_T\right) & \boldsymbol{\lambda}\mathbf{K}\left(\mathbf{X}_S, \mathbf{x}_*\right) \\ \mathbf{K}\left(\mathbf{X}_T, \mathbf{X}_S\right)\boldsymbol{\lambda}^T & \mathbf{K}\left(\mathbf{X}_T, \mathbf{X}_T\right) + \sigma_T^2 \mathbf{I} & \mathbf{K}\left(\mathbf{X}_T, \mathbf{x}_*\right) \\ \mathbf{K}\left(\mathbf{x}_*, \mathbf{X}_S\right)\boldsymbol{\lambda}^T & \mathbf{K}\left(\mathbf{x}_*, \mathbf{X}_T\right) & \mathbf{K}\left(\mathbf{x}_*, \mathbf{x}_*\right) \end{bmatrix}. \tag{13}$$

$$\hat{\mathbf{K}}(\mathbf{X}, \mathbf{X}) = \begin{bmatrix} \lambda_1\left(\mathbf{K}\left(\mathbf{x}_{S_1}, \mathbf{x}_{S_1}\right) + \sigma_S^2 \mathbf{I}\right) & \cdots & \lambda_1\left(\mathbf{K}\left(\mathbf{X}_{S_1}, \mathbf{x}_{S_N}\right) + \sigma_S^2 \mathbf{I}\right) & \lambda_1 \mathbf{K}\left(\mathbf{x}_{S_1}, \mathbf{x}_T\right) \\ \vdots & \ddots & \vdots & \vdots \\ \lambda_k\left(\mathbf{K}\left(\mathbf{x}_{S_n}, \mathbf{x}_{S_1}\right) + \sigma_S^2 \mathbf{I}\right) & \cdots & \lambda_k\left(\mathbf{K}\left(\mathbf{x}_{S_n}, \mathbf{x}_{S_n}\right) + \sigma_S^2 \mathbf{I}\right) & \lambda_k \mathbf{K}\left(\mathbf{x}_{S_n}, \mathbf{x}_T\right) \\ \mathbf{K}\left(\mathbf{x}_{S_1}, \mathbf{x}_T\right)\lambda_1^T & \cdots & \mathbf{K}\left(\mathbf{x}_{S_n}, \mathbf{x}_T\right)\lambda_k^T & \mathbf{K}\left(\mathbf{x}_T, \mathbf{x}_T\right) + \sigma_T^2 \mathbf{I} \end{bmatrix}. \tag{20}$$

TABLE I
DISCHARGE PROFILES OF THE SELECTED NASA-PCOE BATTERY DATASETS

Battery ID	Discharge current	Cutoff voltage	Ambient temperature	Capacity loss
B0005	2 A	2.7 V	24 °C	30%
B0006	2 A	2.5 V	24 °C	30%
B0007	2 A	2.2 V	24 °C	30%
B0018	2 A	2.5 V	24 °C	30%
B0029	4 A	2.0 V	43 °C	20%
B0030	4 A	2.2 V	43 °C	20%
B0034	4 A	2.2 V	24 °C	20%

TABLE II
DISCHARGE PROFILES OF THE SELECTED CALCE DATASETS

Battery ID	Discharge current	Cutoff voltage	Nominal capacity
CS2-33	0.55 A	2.7 V	1100 mAh
CS2-34	0.55 A	2.7 V	1100 mAh
CS2-35	1.1 A	2.7 V	1100 mAh
CS2-36	1.1 A	2.7 V	1100 mAh
CS2-38	1.1 A	2.7 V	1100 mAh
CX2-36	0.55 A	2.7 V	1350 mAh
CX2-38	0.55 A	2.7 V	1350 mAh

3) The extracted features from different datasets are placed in separate matrices, where the features extracted from the auxiliary dataset are noted as $\mathbf{x}_{S_i}$ and the features extracted from the target battery are noted as $\mathbf{x}_T$. Construct the weighted covariance function for the source and target data, noted as $\hat{K}(\mathbf{x}_{S_i}, \mathbf{x}_T)$.
4) Optimize hyperparameters θ to maximize the likelihood function according to (11).
5) Calculate the estimated SOH for each TR-GPR submodel based on (14).
6) Calculate the weight for each submodel based on (21).
7) Calculate the estimated SOH value with the ensemble structure described in (22).

III. EXPERIMENT

This section detailed experimental data in Section III-A, followed by HIs in Section III-B. In Sections III-C, III-D, and III-E, we have detailed validation parameters, comparative models, and experiment results with analysis, respectively.

A. Introduction of Experimental Data

The datasets used in this article are randomly selected from the database of the Ames Prognostics Center of Excellence (PCoE), NASA [50] and the database of the Center for Advanced Life Cycle Engineering, (CALCE) [51]. Batteries are repeatedly charged and discharged in the ageing test. The PCoE and CALCE datasets record the changes in the battery's external characteristics from the initial state to a deteriorated condition. PCoE dataset uses 18 650-sized lithium-ion batteries. Batteries are first charged with a CC of 1.5 A. When the terminal voltage reaches 4.2 V, switch to the CV charge mode until the current drops to 20 mA. The discharge setting is illustrated in Table I. The CALCE database uses the LiCoO2 battery. The batteries marked as "CS2" have a nominal capacity of 1100 mAh, whereas those marked as "CX2" have a nominal capacity of 1350 mAh. For the CALCE database, batteries are charged with a 0.5 C charge rate; when the voltage reaches 4.2 V, switch to the CV mode until the current drops to 50 mA. The settings of the discharge mode are illustrated in Table II. Fig. 3 shows the ageing curve of the selected datasets. The characteristics and differences between datasets are apparent. To fully demonstrate the effect, three sets of experiments are set up. The training dataset used in the experiments contains multiple source domains and the first 20% of target domain data. Since the first 20% of cycle data cannot completely cover the feature space, it cannot be used alone to support the model training. In these experiments, we will reveal the performance degradation of general machine learning models for cross-battery SOH estimation and show the advantages of MTR-GPR in cross-object knowledge generalization.

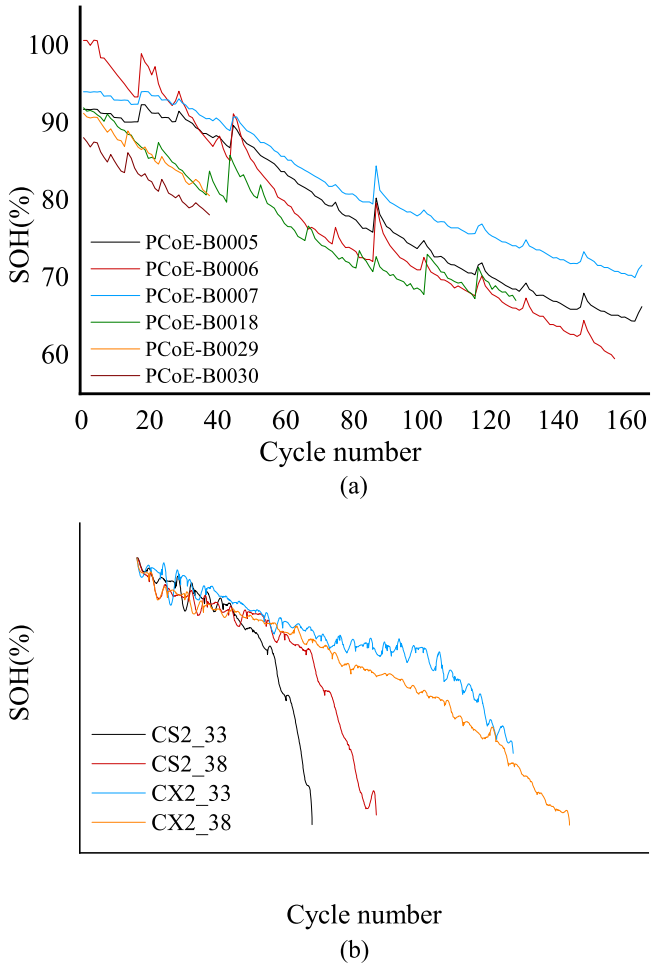

Fig. 3. Battery ageing rate in different experiments. (a) Ageing curve of selected PCoE datasets. (b) Ageing curves of selected CALCE datasets.

B. Health Indexes

Some well-known HIs are used in this study, including CC and CV charging time, the maximum value of the IC curve, and time interval of equal charging voltage [52]. These widely validated HIs are obtained from long-time charge and discharge data,

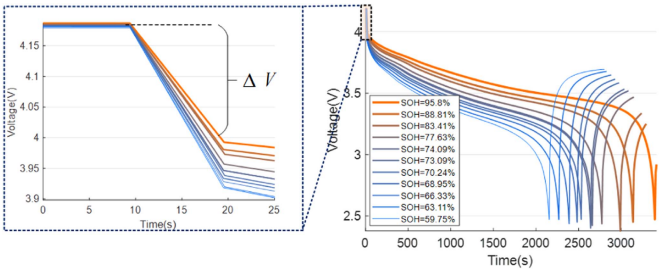

Fig. 4. Difference in the initial voltage drop of batteries with different SOH.

which is not conducive to rapid SOH estimation. An HI called 100% SOC-resistance is proposed, which allows immediate pickup. In the hybrid pulse power characterization (HPPC) test, the battery's Ohmic and polarization resistance can be obtained by dividing the voltage drop by current [53]. However, the HPPC test relies on a specific battery testing device and profiles. To estimate the internal resistance based on the battery management system, accurate SOC control is essential.

In fact, the 100% SOC status can is easily accessible. Once the battery is fully charged, the 100% SOC status is achieved. The Ohmic internal resistance can be calculated by capturing the battery's instantaneous voltage drop, as shown in (3)

$$R_{100\%\ \mathrm{SOC}} = \frac{\Delta V}{I} \tag{24}$$

where Δ Vis the voltage drop, as shown in Fig. 4. It can be clearly seen from the figure that the voltage drops rapidly in the initial discharging stage. As the battery ages, Δ V increases gradually under the same discharge current, which means the 100% SOC Ohmic resistance increases as the battery ages. This HI is named the 100% SOC-Ohmic resistance. It is not encouraged to calculate the 100% SOC-polarization resistance. Obtaining the polarization effect needs to maintain load stability for a long time, which is difficult to achieve with electric vehicles.

C. Evaluation Indicators

To evaluate the effectiveness of the model, mean bias error (MBE), root-mean-square error (RMSE), and The prediction interval normalized average width (PINAW) [54] are used as the evaluation indicators. RMSE and MAE evaluate the variance of the estimates, MBE indicates the bias of the model; MAPE shows the relative error. PINAW and interval validity evaluate the error interval. The mentioned indicators can be obtained by the following (25), (26), (27), and (28):

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_*^{(i)} - y_{\mathrm{reference}}^{(i)}\right)^2} \tag{25}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_*^{(i)} - y_{\mathrm{reference}}^{(i)}\right| \tag{26}$$

$$\mathrm{MBE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_*^{(i)} - y_{\mathrm{reference}}^{(i)}\right) \tag{27}$$

TABLE III
EXPERIMENTAL SETTINGS OF COMPARATIVE MODELS

Model	Parameter Configuration
GPR	Mean function : zero mean function Covariance function : SE kernel Cost function : Gaussian likelihood function Initial Hyper-parameter: $l = 1, \sigma_f = 1$
ANN	Structure: single hidden layer Hidden layer nodes : 2×feature dimension + 5 Cost function : sum of squared errors Training function : scaled conjugate gradient Learning function : gradient descent with momentum Maximum epoch number: 1000
LSSVM	Kernel function : radial basis function Cost function : mean square error Hyper parameters: $\gamma = 1, \sigma = 0.1$ Parameter verification method : 10-fold cross validate Optimization function : simplex

TABLE IV
EXPERIMENTAL DATA SETTING

Roles	Case I	Case II	Case III
Source 1	PCoE-B0005	PCoE-B0005	PCoE-B0005
Source 2	PCoE-B0006	PCoE-B0007	PCoE-B0029
Source 3	PCoE-B0007	PCoE-B0029	CALCE-CS2_38
Source 4	PCoE-B0029	CALCE-CS2_33	CALCE-CS2_33
Source 5	PCoE-B0030	CALCE-CX2_38	CALCE-CX2_38
Target	PCoE-B0018	PCoE-B0018	PCoE-B0018

$$\mathrm{PINAW} = \frac{1}{n}\sum_{i=1}^{n}\left(U\left(f^{(i)}\right) - L\left(f^{(i)}\right)\right) \tag{28}$$

where $U(f^{(i)})$ and $L(f^{(i)})$ are the upper and lower bounds of 95% confidence interval. y_* is the predicted SOH value, $y_{\mathrm{reference}}$ indicates the reference SOH value. n indicates the number of samples in the test dataset. n_c is the number of true values that falls into the 95% confidence error boundary.

D. Comparative Models

This article compares the MTR-GPR model with several widely used data-driven methods, including ANN, LSSVM, and GPR. The model parameters refer to the experiment settings in [55], [56], and [52], as shown in Table III.

E. Experiment Results and Analysis

Three experiments are included in this section. Unlike previous studies, the training and test data in this article are not drawn at random from one ageing experiment; instead, datasets from multiple experiments are used. These datasets are obtained with different cutoff voltages, ambient temperatures, charge–discharge profiles, and battery types. This allows us to better simulate the actual use of the environment. For MTR-GPR, a small amount of target data is required for knowledge screening, only the first 20% of the target data is used in the training process. The experiments show that the concept drift issue can cause large errors. The experiment will validate the performance of MTR-GPR while also verifying the possible problems of general machine learning SOH detection methods in the context of concept drift. The datasets selected for each of the three experiments are shown in Table IV.

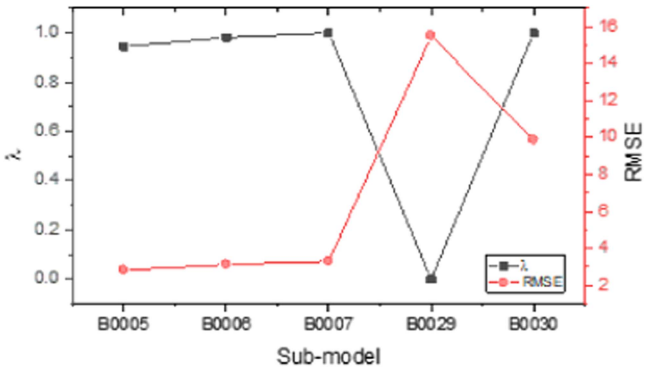

Fig. 5. Transfer factor and RMSE for each submodel.

TABLE V
PERFORMANCE COMPARISON (CASE I)

	LSSVM	ANN	GPR	MTR-GPR
RMSE	2.0837	3.2925	4.6189	1.064
MAE	1.6214	2.4843	3.5935	0.8275
MBE	-1.1765	-1.6845	-2.2913	-0.1813
MAPE	0.02220	0.03260	0.0487	0.0108
PINAW (95% confidence)			14.9677	4.4657

1) CASE I: In case I, both the source datasets and target datasets are from the PCoE database, the ageing experiments are conducted with the same type of battery, so the data distribution is not too dissimilar. Compared with the target dataset, the first three source datasets have differences in cut-off voltage, while source 4 and source 5 have differences in ambient temperature and capacity loss in addition to cutoff voltage.

Fig. 5 shows the weights of different GTR-GPR submodels. As can be seen from the figure, the submodel corresponding to the B0005, B0006, and B0007 has better performance in terms of RMSE. As a result, these three models will be given a higher weight in the ensemble model based on (20). Fig. 6 and Table V depict the estimation result. The data shown in the figure to the left of the dotted line took part in the training. MTR-GPR has higher accuracy in terms of RMSE, MAE, MBE, MAPE, and PINAW, as shown in the graph. ANN and GPR have large errors around cycle 50, the error of LSSVM rises after cycle 70. In general, the distribution difference between the training and test sets is small, and all of the algorithms can track the battery's deterioration trend in this case. Fig. 6 depicts the estimation result. MTR-GPR has superior accuracy in terms of RMSE, MAE, MBE, MAPE, and PINAW. When the number of cycles is around 50, ANN and GPR have high errors, while LSSVM does not track the reference SOH properly when the number of cycles is greater than 70. In general, because the data features of the training set and the test set are similar in this group of trials, the comparison algorithm can also track the ageing trend of the measured item, but the accuracy is not as good as the suggested MTR-GPR algorithm.

Compared with the conventional data-driven models, MTR-GPR has a more stable and accurate estimate. The diversity of the source data ensures the model's generalization ability and the

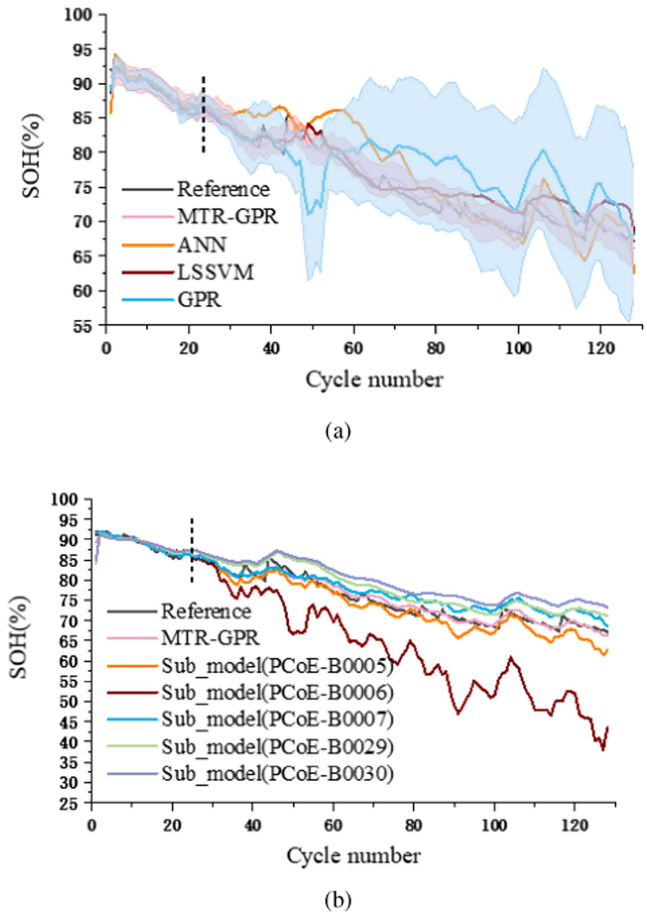

Fig. 6. SOH estimation result (case I). (a) comparison of different SOH estimation methods. (b) demonstration of MTR-GPR's submodels.

ensemble structure brings the advantage of predictive stability. In the figure, the red and blue error intervals represent the 95% confidence intervals for MTR-GPR and GPR, respectively. PINAW determines the average width of error intervals. Under the same confidence range, a narrower confidence interval indicates that the model has a higher prediction accuracy. As seen in the figure, MTR-GPR has a distinct advantage in terms of error interval accuracy.

2) Case II: The datasets used in case II are randomly selected from the PCoE and CALCE datasets. In this case, the experimental environment and battery types are different. In this scenario, the distribution of the source and target datasets differs more significantly than in CASE I. As a result, when the model is trained on these data, its validity is severely questioned. Fig. 7(a) and Table VI illustrate the prediction performance of various comparative models as well as the proposed MTR-GPR model.

Besides the source data, only first 20% of the target data is used for training, the GPR, ANN, and LSSVM models perform well within this range, and when the number of cycles exceeds that, the models' performance degrades considerably. The LSSVM exhibits significant overfitting in this set of trials. When the number of cycles exceeds 40, it is unable to adequately track

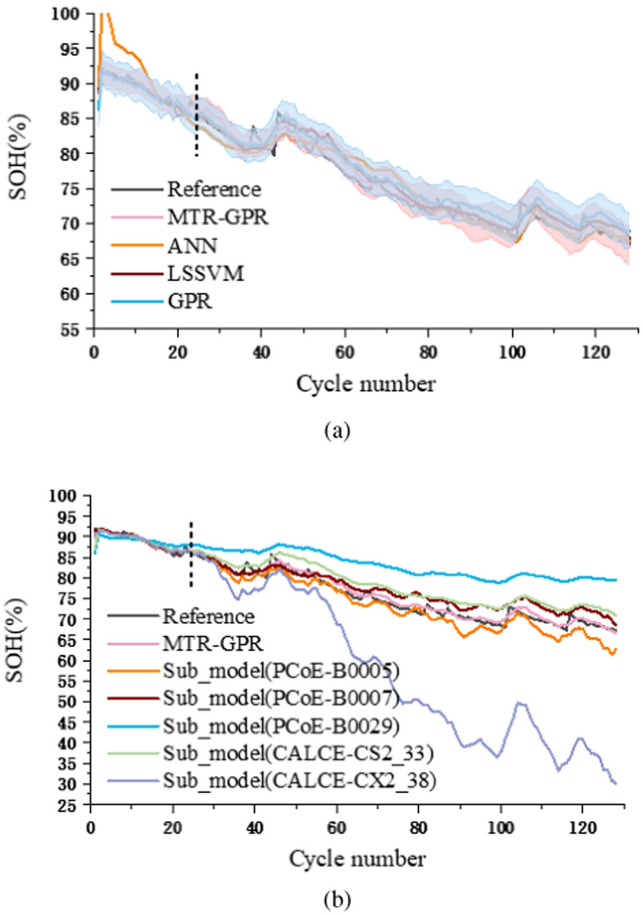

Fig. 7. SOH estimation result (case II). (a) comparison of different SOH estimation methods. (b) demonstration of MTR-GPR's submodels.

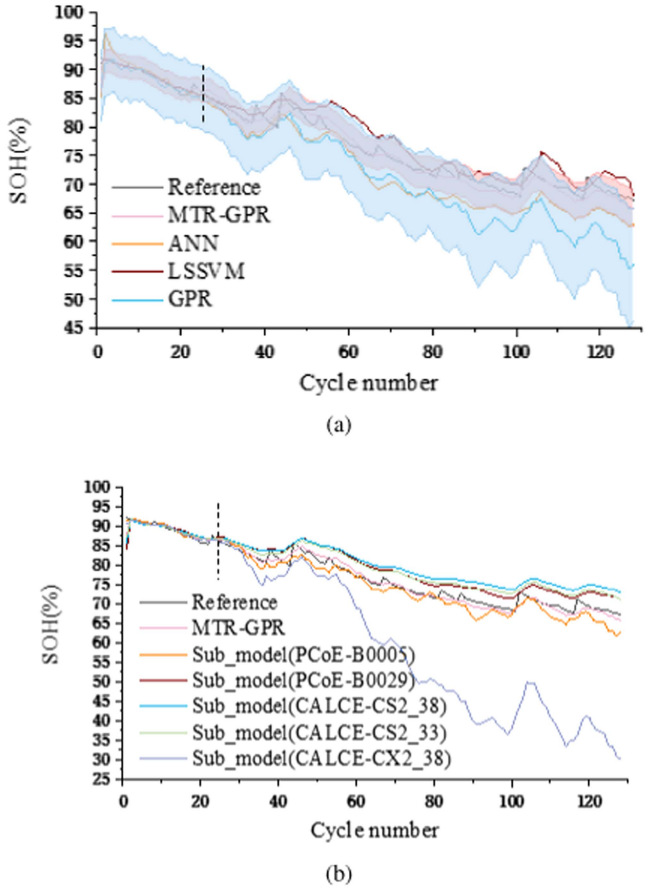

Fig. 8. SOH estimation result (case III). (a) Comparison of different SOH estimation methods. (b) Demonstration of MTR-GPR's submodels.

TABLE VI
PERFORMANCE COMPARISON (CASE II)

	LSSVM	ANN	GPR	MTR-GPR
RMSE	1.3105	2.3652	1.6618	1.0467
MAE	1.0352	1.6212	1.3909	0.8422
MBE	−0.5260	−0.4881	−0.9530	−0.3845
MAPE	0.01360	0.0201	0.0186	0.0110
PINAW (95% confidence)			4.8581	4.0988

TABLE VII
PERFORMANCE COMPARISON (CASE III)

	LSSVM	ANN	GPR	MTR-GPR
RMSE	2.3576	3.5113	4.9979	1.1411
MAE	1.7986	3.0047	4.0344	0.8593
MBE	−1.5429	2.6804	3.9719	0.1894
MAPE	0.0241	0.0404	0.0555	0.0112
PINAW (95% confidence)			14.0102	4.9325

the battery's deteriorating trend. Although the battery SOH calculated by GPR and ANN is comparable with the battery's actual SOH trend, the estimation accuracy is much lower than that of MTR-GPR. LSSVM embodies its advantages in generalization performance and controls MAE at 1.0352. Under this condition, MTR-GPR has limits the MAE at 0.8422 and MBE at –0.3845. A reliable prediction result is obtained.

Fig. 7(b) shows the prediction effect of submodels of MTR-GPR. The results show that submodels 1 and 2 outperform the others. According to Table IV, the variations between training and test sets for these submodels are minor. In the ensemble framework, MTR-GPR assign these models with a high weight, thereby avoiding the occurrence of negative transfer problems. At the same time, the experimental results show that when the distribution of the source domain and the target domain is very different, knowledge transfer via a single source domain cannot guarantee the stability and reliability of the prediction results.

3) Case III: Based on case II, we continued to use PCoE-B0018 as the target set in this case, removing a dataset from PCoE and adding a dataset from CALCE. It exacerbates the problem of concept drift. As shown in Fig. 8, after the number of cycles exceeds 15 (20% of total 128 cycles), the comparison methods cannot reliably track the actual SOH. 20% of the target data is included in the training dataset. Once the models encounter the unseen input, the model's overfitting problem appears. According to Table VII, the MTR-GPR has a more reliable performance concerning RMSE, MAE, MBE, and MAPE.

When making predictions, machine learning methods often assume that the test data they encounter has the same characteristics as the training data. However, in practice, this is not always the case. Machine learning-based techniques are prone to substantial errors when there is a distribution disparity between the training and test data. The results of previous tests support this. This is a challenge in the laboratory and a difficulty that diagnostic technology frequently encounters in industrial applications. This article proposes a method called MTR-GPR to address this issue. Unlike typical single-source domain knowledge transfer methods, this method does not require manually selecting a dataset with similar characteristics to the target dataset as the source domain data. Instead, it can incorporate multiple datasets into the model, and the model can automatically select the data sources that are most helpful for the target battery. With the help of these source domain data, the model can obtain reliable prediction results with limited target data support.

In the practical application of MTR-GPR, there are several factors that can affect the method's practicality. First, the similarity between the source domain and target domain data. Although the addition of multiple source domains allows for a more abundant data information, if the target data type is significantly different from the multiple source domain data distributions provided by the user, there will still be limited information available for transfer, resulting in limited benefits. Second, this method requires that the source and target domain data have the same modality, meaning that the data features of the source domain and target domain should be the same. Otherwise, the GPR model cannot be solved. Third, the correlation between features will directly affect the model's effectiveness. Therefore, readers should select features based on their practical application needs when using this model, while also adopting some of the more efficient features proposed in recent years. Finally, more source domains mean greater computational complexity, so it is recommended to delete some source domain datasets with low correlation to the target domain in practical applications.

IV. Conclusion

In this article, a multidomain transfer learning approach is proposed for SOH estimation. By constructing the Gaussian transfer covariance function, the GPR model can control the adverse effects of the data distribution difference on the regression to a certain extent. When the data distribution difference is large, it may not be easy to obtain satisfactory results for the single-domain transfer learning model. The ensemble structure makes up for this deficiency. A model with a lower confidence level occupies a relatively small weight. It can be concluded from the experimental results that in the cross-object estimation scenarios, the general machine learning method has an overfitting risk. The MTR-GPR can effectively generalize the information in the source domain to the target battery. It should be noted that we cannot acquire reliable prediction results with MTR-GPR merely by relying on source domain data; a modest amount of target domain data is required. In the following research, we will focus on further reducing the reliance of intelligent detection algorithms on target domain data.

Appendix

The article's Appendix demonstrates the sufficient and necessary conditions for the solvability of TR-GPR. It explains why it is impossible to construct a covariance matrix that encompasses all data from the source domain and the target domain to achieve MTR-GPR. It also provides theoretical support for the ensemble structured MTR-GPR approach. GPR requires the covariance matrix to be symmetric and semipositive definite, therefore, to guarantee the transfer covariance function $\hat{\mathbf{K}}(\mathbf{X},\mathbf{X})$ still valid, $\hat{\mathbf{K}}(\mathbf{X},\mathbf{X})$ should be positive semidefinite (PSD).

Proposition: For valid covariance matrix $\mathbf{K}(\mathbf{X}_S,\mathbf{X}_S)$, $\mathbf{K}(\mathbf{X}_S,\mathbf{X}_T)$, $\mathbf{K}(\mathbf{X}_T,\mathbf{X}_S)$, and $\mathbf{K}(\mathbf{X}_T,\mathbf{X}_T)$, $\hat{\mathbf{K}}(\mathbf{X},\mathbf{X}) \succcurlyeq 0 \Leftrightarrow \lambda_1 = \lambda_2 =, \ldots, = \lambda_k$, and $\lambda \in (0,1)$. Define shown at the bottom of this page For notation simplicity, denote

$$\mathbf{K}(\mathbf{X},\mathbf{X}) = \begin{bmatrix} \mathbf{K}(\mathbf{X}_S,\mathbf{X}_S) + \sigma_S^2\mathbf{I} & \mathbf{K}(\mathbf{X}_S,\mathbf{X}_T) \\ \mathbf{K}(\mathbf{X}_T,\mathbf{X}_S) & \mathbf{K}(\mathbf{X}_T,\mathbf{X}_T)\,\sigma_T^2\mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix} \tag{32}$$

$\hat{K}$ can be denoted as follows.

Lemma 1: A Gram matrix $K \in \mathbb{R}^{n\times n}$, with respect to $\{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$, where $K_{ij} = k(x_i,x_j)$, must be PSD [57].

Lemma 2: The product of two symmetric PSD matrices is PSD [58].

Lemma 3 (Schur complement theorem [59]): Given $\mathbf{A} = \mathbf{A}^T$ and $\mathbf{C} = \mathbf{C}^T$

$$\mathbf{K} \succcurlyeq 0 \Leftrightarrow \begin{cases} \mathbf{A} \succcurlyeq 0 \\ \left(\mathbf{I} - \mathbf{A}\mathbf{A}^{\dagger}\right)\mathbf{B} = 0 \\ \mathbf{C} - \mathbf{B}^T\mathbf{A}^{\dagger}\mathbf{B} \succcurlyeq 0. \end{cases} \tag{33}$$

Proof of necessity:

For a valid GPR model, the covariance matrix $\mathbf{K}$, $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are PSD (Lemma 1). According to Lemma 3, we have

$$\mathbf{A} \succcurlyeq 0 \tag{34}$$

$$\left(\mathbf{I} - \mathbf{A}\mathbf{A}^{\dagger}\right)\mathbf{B} = 0 \Leftrightarrow \begin{bmatrix} \mathbf{B}_{11} & \cdots & \mathbf{B}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{B}_{m1} & \cdots & \mathbf{B}_{mn} \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{A}_{11} & \cdots & \mathbf{A}_{1m} \\ \vdots & \ddots & \vdots \\ \mathbf{A}_{m1} & \cdots & \mathbf{A}_{mm} \end{bmatrix} \begin{bmatrix} \mathbf{A}_{11}^{\dagger} & \cdots & \mathbf{A}_{1m}^{\dagger} \\ \vdots & \ddots & \vdots \\ \mathbf{A}_{m1}^{\dagger} & \cdots & \mathbf{A}_{mm}^{\dagger} \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{B}_{11} & \cdots & \mathbf{B}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{B}_{m1} & \cdots & \mathbf{B}_{mn} \end{bmatrix} = 0. \tag{35}$$

Equation (33) can be rewritten as

$$\begin{cases} [\mathbf{B}_{11},\ldots,\mathbf{B}_{1n}]-\left[\sum_{j=1}^{m}\sum_{i=1}^{m}\mathbf{A}_{1i}\mathbf{A}_{ij}^{\dagger}\mathbf{B}_{j,1},\ldots,\right. \\ \left.\sum_{j=1}^{m}\sum_{i=1}^{m}\mathbf{A}_{1i}\mathbf{A}_{ij}^{\dagger}\mathbf{B}_{j,n}\right]=0 \\ [\mathbf{B}_{m1},\ldots,\mathbf{B}_{mn}]-\left[\sum_{j=1}^{m}\sum_{i=1}^{m}\mathbf{A}_{mi}\mathbf{A}_{ij}^{\dagger}\mathbf{B}_{j,1},\ldots,\right. \\ \left.\sum_{j=1}^{m}\sum_{i=1}^{m}\mathbf{A}_{mi}\mathbf{A}_{ij}^{\dagger}\mathbf{B}_{j,n}\right]=0. \end{cases} \tag{36}$$

Equation (31) also tells us

$$\mathbf{C}-\mathbf{B}^{T}\mathbf{A}^{\dagger}\mathbf{B}\succcurlyeq 0 \tag{37}$$

$$\hat{\mathbf{K}}\geqslant 0\Leftrightarrow\begin{cases}\boldsymbol{\lambda}\mathbf{A}\geqslant 0\\ \left(\mathbf{I}-\lambda\mathbf{A}\mathbf{A}^{\dagger}\lambda^{\dagger}\right)\lambda\mathbf{B}=0\\ \mathbf{C}-\mathbf{B}^{T}\boldsymbol{\lambda}^{T}\mathbf{A}^{\dagger}\boldsymbol{\lambda}^{\dagger}\boldsymbol{\lambda}\mathbf{B}\geqslant 0\end{cases} \tag{38}$$

where $K^{\dagger}$ denotes the generalized inverse of K

$$\hat{\mathbf{K}}=\begin{bmatrix}\lambda_1\mathbf{A}_{11}&\cdots&\lambda_1\mathbf{A}_{1m}&\lambda_1\mathbf{B}_{11}&\cdots&\lambda_1\mathbf{B}_{1n}\\ \vdots&\ddots&\vdots&\vdots&\ddots&\vdots\\ \lambda_m\mathbf{A}_{m1}&\cdots&\lambda_m\mathbf{A}_{mm}&\lambda_m\mathbf{B}_{m1}&\cdots&\lambda_m\mathbf{B}_{mn}\\ \lambda_1\mathbf{B}_{11}&\cdots&\lambda_m\mathbf{B}_{1m}&\mathbf{C}_{11}&\cdots&\mathbf{C}_{1n}\\ \vdots&\ddots&\vdots&\vdots&\ddots&\vdots\\ \lambda_1\mathbf{B}_{n1}&\cdots&\lambda_m\mathbf{B}_{nm}&\mathbf{C}_{n1}&\cdots&\mathbf{C}_{nn}\end{bmatrix} \tag{39}$$

$$\left(\mathbf{I}-\boldsymbol{\lambda}\mathbf{A}\mathbf{A}^{\dagger}\boldsymbol{\lambda}^{\dagger}\right)\boldsymbol{\lambda}\mathbf{B}=0\Leftrightarrow\begin{bmatrix}\lambda_1\mathbf{B}_{11}&\cdots&\lambda_1\mathbf{B}_{1n}\\ \vdots&\ddots&\vdots\\ \lambda_m\mathbf{B}_{m1}&\cdots&\lambda_m\mathbf{B}_{mn}\end{bmatrix}-\begin{bmatrix}\lambda_1\mathbf{A}_{11}&\cdots&\lambda_1\mathbf{A}_{1m}\\ \vdots&\ddots&\vdots\\ \lambda_m\mathbf{A}_{m1}&\cdots&\lambda_m\mathbf{A}_{mm}\end{bmatrix}\begin{bmatrix}\lambda_1^{-1}\mathbf{A}_{11}^{\dagger}&\cdots&\lambda_m^{-1}\mathbf{A}_{1m}^{\dagger}\\ \vdots&\ddots&\vdots\\ \lambda_1^{-1}\mathbf{A}_{m1}^{\dagger}&\cdots&\lambda_m^{-1}\mathbf{A}_{mm}^{\dagger}\end{bmatrix}\begin{bmatrix}\lambda_1\mathbf{B}_{11}&\cdots&\lambda_1\mathbf{B}_{1n}\\ \vdots&\ddots&\vdots\\ \lambda_m\mathbf{B}_{m1}&\cdots&\lambda_m\mathbf{B}_{mn}\end{bmatrix}=0. \tag{40}$$

Rewritten (39), we have

$$\begin{cases}\lambda_1\left[\mathbf{B}_{11},\ldots,\mathbf{B}_{1n}\right]-\left[\sum_{j=1}^{m}\sum_{i=1}^{m}\lambda_j\mathbf{A}_{1i}\mathbf{A}_{ij}^{\dagger}\mathbf{B}_{j,1},\ldots,\right.\\ \left.\sum_{j=1}^{m}\sum_{i=1}^{m}\lambda_j\mathbf{A}_{1i}\mathbf{A}_{ij}^{\dagger}\mathbf{B}_{j,n}\right]=0\\ \lambda_m\left[\mathbf{B}_{m1},\ldots,\mathbf{B}_{mn}\right]-\left[\sum_{j=1}^{m}\sum_{i=1}^{m}\lambda_j\mathbf{A}_{1i}\mathbf{A}_{ij}^{\dagger}\mathbf{B}_{j,1},\ldots,\right.\\ \left.\sum_{j=1}^{m}\sum_{i=1}^{m}\lambda_j\mathbf{A}_{1i}\mathbf{A}_{ij}^{\dagger}\mathbf{B}_{j,n}\right]=0.\end{cases} \tag{41}$$

It can be deducted from (35) and (40) that

$$\begin{cases}\sum_{j=1}^{m}\sum_{i=1}^{m}\left(\lambda_j-\lambda_1\right)\mathbf{A}_{1i}\mathbf{A}_{ij}^{\dagger}B_{j,1},\ldots,\\ \sum_{j=1}^{m}\sum_{i=1}^{m}\left(\lambda_j-\lambda_1\right)\mathbf{A}_{1i}\mathbf{A}_{ij}^{\dagger}\mathbf{B}_{j,n}=0\\ \sum_{j=1}^{m}\sum_{i=1}^{m}\left(\lambda_j-\lambda_m\right)\mathbf{A}_{1i}\mathbf{A}_{ij}^{\dagger}\mathbf{B}_{j,1},\ldots,\\ \sum_{j=1}^{m}\sum_{i=1}^{m}\left(\lambda_j-\lambda_m\right)\mathbf{A}_{1i}\mathbf{A}_{ij}^{\dagger}\mathbf{B}_{j,n}=0.\end{cases} \tag{42}$$

To satisfy (41), the following equation holds:

$$\lambda_1=\lambda_2=\ldots\lambda_m. \tag{43}$$

Since (42) holds, $\boldsymbol{\lambda}$ is a scalar matrix, equation $\mathbf{C}-\mathbf{B}^{T}\boldsymbol{\lambda}^{T}\mathbf{A}^{\dagger}\boldsymbol{\lambda}^{\dagger}\lambda\mathbf{B}\succcurlyeq 0$ equals to

$$\mathbf{C}-\boldsymbol{\lambda}\mathbf{B}^{T}\mathbf{A}^{\dagger}\mathbf{B}\succcurlyeq 0. \tag{44}$$

If $\mathbf{C}-(\lambda\mathbf{B})^{T}(\lambda\mathbf{A})^{\dagger}(\lambda\mathbf{B})\succcurlyeq 0$ holds, for any valid covariance matrix $\mathbf{A},\mathbf{B},\mathbf{C}$, one must have $|\boldsymbol{\lambda}|\leq 1$. Moreover, $\lambda K\succcurlyeq 0$ holds for any valid covariance matrix $\mathbf{K}$, we have $\lambda_i\geq 0$. In summary, if matrix K is PSD, one have $\lambda_1=\lambda_2=\ldots\lambda_m$, and $0\leq\lambda_i\leq 1$.

Proof of sufficiency:

Suppose $\lambda_1=\lambda_2=\ldots\lambda_m$, then $\boldsymbol{\lambda}$ is a scalar matrix. $\forall\mathbf{A}\succcurlyeq 0$, we have

$$\lambda\mathbf{A}\succcurlyeq 0. \tag{45}$$

According to Lemma 1, $(\mathbf{I}-\mathbf{A}\mathbf{A}^{\dagger})\mathbf{B}=0$, i.e.,

$$\left(\mathbf{I}-\mathbf{A}\mathbf{A}^{\dagger}\right)\mathbf{B}=0\Leftrightarrow\begin{bmatrix}\mathbf{B}_{11}&\cdots&\mathbf{B}_{1n}\\ \vdots&\ddots&\vdots\\ \mathbf{B}_{m1}&\cdots&\mathbf{B}_{mn}\end{bmatrix}$$

$$\lambda=\begin{bmatrix}\lambda_1&&\\ &\ddots&\\ &&\lambda_1\end{bmatrix} \tag{29}$$

$$\mathbf{K}=\begin{bmatrix}\mathbf{K}\left(\mathbf{X}_S,\mathbf{X}_S\right)+\sigma_S^2\mathbf{I}&\mathbf{K}\left(\mathbf{X}_S,\mathbf{X}_T\right)\\ \mathbf{K}\left(\mathbf{X}_T,\mathbf{X}_S\right)&\mathbf{K}\left(\mathbf{X}_T,\mathbf{X}_T\right)\sigma_T^2\mathbf{I}\end{bmatrix} \tag{30}$$

$$\hat{\mathbf{K}}(\mathbf{X},\mathbf{X})=\begin{bmatrix}\lambda_1\left(\mathbf{K}\left(\mathbf{X}_{S_1},\right.\right.&\cdots&\lambda_1\left(\mathbf{K}\left(\mathbf{X}_{S_1},\right.\right.&\lambda_1\mathbf{K}\left(\mathbf{X}_{S_1},\right.\\ \left.\left.\mathbf{X}_{S_1}\right)+\sigma_S^2\mathbf{I}\right)&&\left.\left.\mathbf{X}_{S_N}\right)+\sigma_S^2\mathbf{I}\right)&\left.\mathbf{X}_T\right)\\ \vdots&\ddots&\vdots&\vdots\\ \lambda_k\left(\mathbf{K}\left(\mathbf{X}_{S_n},\right.\right.&\cdots&\lambda_k\left(\mathbf{K}\left(\mathbf{X}_{S_n},\right.\right.&\lambda_k\mathbf{K}\left(\mathbf{X}_{S_n},\right.\\ \left.\left.\mathbf{X}_{S_1}\right)+\sigma_S^2\mathbf{I}\right)&&\left.\left.\mathbf{X}_{S_n}\right)+\sigma_S^2\mathbf{I}\right)&\left.\mathbf{X}_T\right)\\ \mathbf{K}\left(\mathbf{X}_{S_1},\mathbf{X}_T\right)\lambda_1^T&\cdots&\mathbf{K}\left(\mathbf{x}_{S_n},\mathbf{X}_T\right)\lambda_k^T&\mathbf{K}\left(\mathbf{X}_T,\mathbf{X}_T\right)\\ &&&+\sigma_T^2\mathbf{I}\end{bmatrix}. \tag{31}$$

$$-\begin{bmatrix}\mathbf{A}_{11} & \cdots & \mathbf{A}_{1m}\\ \vdots & \ddots & \vdots\\ \mathbf{A}_{m1} & \cdots & \mathbf{A}_{mm}\end{bmatrix}\begin{bmatrix}\mathbf{A}_{11}^{\dagger} & \cdots & \mathbf{A}_{1m}^{\dagger}\\ \vdots & \ddots & \vdots\\ \mathbf{A}_{m1}^{\dagger} & \cdots & \mathbf{A}_{mm}^{\dagger}\end{bmatrix}\begin{bmatrix}\mathbf{B}_{11} & \cdots & \mathbf{B}_{1n}\\ \vdots & \ddots & \vdots\\ \mathbf{B}_{m1} & \cdots & \mathbf{B}_{mn}\end{bmatrix}=0. \tag{46}$$

Since $\boldsymbol{\lambda}$ is a scalar matrix, based on (45), we have

$$\left(\mathbf{I}-\lambda\mathbf{A}\left(\lambda\mathbf{A}\right)^{\dagger}\right)\lambda\mathbf{B}=0\Leftrightarrow\left(\mathbf{I}-\mathbf{A}\mathbf{A}^{\dagger}\right)\lambda\mathbf{B}=0. \tag{47}$$

Also

$$\mathbf{C}-\left(\lambda\mathbf{B}\right)^{T}\left(\lambda\mathbf{A}\right)^{\dagger}\left(\lambda\mathbf{B}\right)=\mathbf{C}-\lambda\mathbf{B}^{T}\mathbf{A}^{\dagger}\mathbf{B}. \tag{48}$$

Since

$$\mathbf{C}-\mathbf{B}^{T}\mathbf{A}^{\dagger}\mathbf{B}\succcurlyeq 0. \tag{49}$$

If $\forall i, 0<\lambda_i<1$, the following equation holds:

$$\mathbf{C}-\lambda\mathbf{B}^{T}\mathbf{A}^{\dagger}\mathbf{B}\succcurlyeq 0. \tag{50}$$

Juggle the (31), (46), and (49), we have

$$\begin{cases}\lambda\mathbf{A}\succcurlyeq 0\\ \left(\mathbf{I}-\lambda\mathbf{A}(\lambda\mathbf{A})^{\dagger}\right)\lambda\mathbf{B}=0\\ \mathbf{C}-(\lambda\mathbf{B})^{T}(\lambda\mathbf{A})^{\dagger}(\lambda\mathbf{B})\succcurlyeq 0.\end{cases} \tag{51}$$

Therefore, if transfer factor $\lambda_1=\lambda_2=\ldots\lambda_m$, and $\forall i, 0<\lambda_i<1, K$ is PSD. If a joint covariance matrix is constructed for multiple source domains and the target domain, all transfer factors must be equal, which means that it is impossible to adjust each source domain's contribution individually. On the other hand, if multiple TR-GPR submodels are used for knowledge transfer separately, each submodel can adopt an independent knowledge transfer parameter during the transfer process. This allows for precise control over the contributions of different source domains and enables each submodel to easily satisfy the semipositive definiteness condition of the covariance matrix. And in the calculation process, it can effectively reduce the computational complexity.

REFERENCES

[1] H. Meng and Y.-F. Li, "A review on prognostics and health management (PHM) methods of lithium-ion batteries," *Renewable Sustain. Energy Rev.*, vol. 116, 2019, Art. no. 109405.

[2] A. Paryani, S. I. Kohn, B. Boggs, A. D. Baglino, and C. B. Carlson, "Battery capacity estimating method and apparatus," U.S. Patent 8,004,243. U.S. Patent and Trademark Office, Washington, DC, USA, Aug. 2011.

[3] J. Li, K. Adewuyi, N. Lotfi, R. G. Landers, and J. Park, "A single particle model with chemical/mechanical degradation physics for lithium ion battery state of health (SOH) estimation," *Appl. Energy*, vol. 212, pp. 1178–1190, 2018.

[4] S. Atalay, M. Sheikh, A. Mariani, Y. Merla, E. Bower, and W. D. Widanage, "Theory of battery ageing in a lithium-ion battery: Capacity fade, nonlinear ageing and lifetime prediction," *J. Power Sources*, vol. 478, 2020, Art. no. 229026.

[5] Y. Zhang, X. Du, and M. Salman, "Battery state estimation with a self-evolving electrochemical ageing model," *Int. J. Elect. Power Energy Syst.*, vol. 85, pp. 178–189, 2017.

[6] Q. Yang, J. Xu, X. Li, D. Xu, and B. Cao, "State-of-health estimation of lithium-ion battery based on fractional impedance model and interval capacity," *Int. J. Elect. Power Energy Syst.*, vol. 119, 2020, Art. no. 105883.

[7] D. Yang, Y. Wang, R. Pan, R. Chen, and Z. Chen, "State-of-health estimation for the lithium-ion battery based on support vector regression," *Appl. Energy*, vol. 227, pp. 273–283, 2018.

[8] M. Khalid, S. S. Sheikh, A. K. Janjua, and H. A. Khalid, "Performance validation of electric vehicle's battery management system under state of charge estimation for lithium-ion battery," in *Proc. Int. Conf. Comput., Electron. Elect. Eng.*, 2018, pp. 1–5.

[9] K.-J. Pai, "A reformatory model incorporating PNGV battery and three-terminal-switch models to design and implement feedback compensations of LiFePO4 battery chargers," *Electronics*, vol. 8, no. 2, 2019, Art. no. 126.

[10] C. Cheng, X. Rui, Y. Ruixin, and L. Hailong, "A novel data-driven method for mining battery open-circuit voltage characterization," *Green Energy Intell. Transp.*, vol. 1, no. 1 2022, Art. no. 100001, doi: 10.1016/j.geits.2022.100001.

[11] R. R. Richardson, C. R. Birkl, M. A. Osborne, and D. A. Howey, "Gaussian process regression for in situ capacity estimation of lithium-ion batteries," *IEEE Trans. Ind. Informat.*, vol. 15, no. 1, pp. 127–138, Jan. 2019.

[12] E. Riviere, P. Venet, A. Sari, F. Meniere, and Y. Bultel, "LiFePO4 battery state of health online estimation using electric vehicle embedded incremental capacity analysis," in *Proc. IEEE Veh. Power Propulsion Conf.*, 2015, pp. 1–6.

[13] I. Bloom et al., "Differential voltage analyses of high-power, lithium-ion cells: 1. technique and application," *J. Power Sources*, vol. 139, no. 1/2, pp. 295–303, 2005.

[14] W. He, N. Williard, M. Osterman, and M. Pecht, "Prognostics of lithium-ion batteries based on Dempster-Shafer theory and the Bayesian monte carlo method," *J. Power Sources*, vol. 196, no. 23, pp. 10314–10321, 2011.

[15] X. Hu, J. Jiang, D. Cao, and B. Egardt, "Battery health prognosis for electric vehicles using sample entropy and sparse Bayesian predictive modeling," *IEEE Trans. Ind. Electron.*, vol. 63, no. 4, pp. 2645–2656, Apr. 2016.

[16] D. Liu, J. Zhou, H. Liao, Y. Peng, and X. Peng, "A health indicator extraction and optimization framework for lithium-ion battery degradation modeling and prognostics," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 45, no. 6, pp. 915–928, Jun. 2015.

[17] G.-W. You, S. Park, and D. Oh, "Diagnosis of electric vehicle batteries using recurrent neural networks," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 4885–4893, Jun. 2017.

[18] D. Zhou, Z. Li, J. Zhu, H. Zhang, and L. Hou, "State of health monitoring and remaining useful life prediction of lithium-ion batteries based on temporal convolutional network," *IEEE Access*, vol. 8, pp. 53307–53320, 2020.

[19] R. Razavi-Far, S. Chakrabarti, M. Saif, and E. Zio, "An integrated imputation-prediction scheme for prognostics of battery data with missing observations," *Expert Syst. Appl.*, vol. 115, pp. 709–723, 2019.

[20] X. Tan, D. Zhan, P. Lyu, J. Rao, and Y. Fan, "Online state-of-health estimation of lithium-ion battery based on dynamic parameter identification at multi timescale and support vector regression," *J. Power Sources*, vol. 484, 2021, Art. no. 229233.

[21] W. Pan, Q. Chen, M. Zhu, J. Tang, and J. Wang, "A data-driven fuzzy information granulation approach for battery state of health forecasting," *J. Power Sources*, vol. 475, 2020, Art. no. 228716.

[22] X. Li, C. Yuan, X. Li, and Z. Wang, "State of health estimation for li-ion battery using incremental capacity analysis and Gaussian process regression," *Energy*, vol. 190, 2020, Art. no. 116467.

[23] D. Yang, X. Zhang, R. Pan, Y. Wang, and Z. Chen, "A novel Gaussian process regression model for state-of-health estimation of lithium-ion battery using charging curve," *J. Power Sources*, vol. 384, pp. 387–395, 2018.

[24] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 193–200, doi: 10.1145/1273496.1273521.

[25] W. Ding et al., "Subkilometer crater discovery with boosting and transfer learning," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 4, pp. 1–22, 2011.

[26] G. Blanchard, A. A. Deshmukh, A. Dogan, G. Lee, and C. Scott, "Domain generalization by marginal transfer learning," *J. Mach. Learn. Res.*, vol. 22, no. 1, 2021, Art. no. 2.

[27] Z. Ding and Y. Fu, "Deep transfer low-rank coding for cross-domain learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 6, pp. 1768–1779, Jun. 2019. [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/30371396

[28] X. Shan, Y. Lu, Q. Li, and Y. Wen, "Model-based transfer learning and sparse coding for partial face recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 11, pp. 4347–4356, Nov. 2021.

[29] A. A. Ardakani, A. R. Kanafi, U. R. Acharya, N. Khadem, and A. Mohammadi, "Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: Results of 10 convolutional neural networks," *Comput. Biol. Med.*, vol. 121, 2020, Art. no. 103795.

[30] W.-W. Lin, M.-W. Mak, and J.-T. Chien, "Multisource I-vectors domain adaptation using maximum mean discrepancy based autoencoders," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 12, pp. 2412–2422, Dec. 2018.

[31] S. Maleki, A. Mahmoudi, and A. Yazdani, "Knowledge transfer-oriented deep neural network framework for estimation and forecasting the state of health of the lithium-ion batteries," *J. Energy Storage*, vol. 53, 2022, Art. no. 105183.

[32] Z. Chen, W. Shen, L. Chen, and S. Wang, "Adaptive online capacity prediction based on transfer learning for fast charging lithium-ion batteries," *Energy*, vol. 248, 2022, Art. no. 123537.

[33] 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.

[34] 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.

[35] X. Shu, J. Shen, G. Li, Y. Zhang, Z. Chen, and Y. Liu, "A flexible state-of-health prediction scheme for lithium-ion battery packs with long short-term memory network and transfer learning," *IEEE Trans. Transport. Electrific.*, vol. 7, no. 4, pp. 2238–2248, Dec. 2021.

[36] J. Couture and X. Lin, "Novel image-based rapid RUL prediction for li-ion batteries using a capsule network and transfer learning," *IEEE Trans. Transport. Electrific.*, vol. 9, no. 1, pp. 958–967, Mar. 2023.

[37] J. Yao and T. Han, "Data-driven lithium-ion batteries capacity estimation based on deep transfer learning using partial segment of charging/discharging data," *Energy*, vol. 271, 2023, Art. no. 127033.

[38] J. Schmitt, I. Horstkötter, and B. Bäker, "Effective estimation of battery state-of-health by virtual experiments via transfer- and meta-learning," *J. Energy Storage*, vol. 63, 2023, Art. no. 106969.

[39] X. Lu, J. Qiu, G. Lei, and J. Zhu, "Degradation mode knowledge transfer method for LFP batteries," *IEEE Trans. Transport. Electrific.*, vol. 9, no. 1, pp. 1142–1152, Mar. 2023.

[40] 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.

[41] Z. Ye, J. Yu, and L. Mao, "Multisource domain adaption for health degradation monitoring of lithium-ion batteries," *IEEE Trans. Transport. Electrific.*, vol. 7, no. 4, pp. 2279–2292, Dec. 2021.

[42] 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.

[43] K. Wu, J. Tan, and C. Liu, "Cross-domain few-shot learning approach for lithium-ion battery surface defects classification using an improved siamese network," *IEEE Sensors J.*, vol. 22, no. 12, pp. 11847–11856, Jun. 15, 2022.

[44] S. B. Vilsen and D.-I. Stroe, "Transfer learning for adapting battery state-of-health estimation from laboratory to field operation," *IEEE Access*, vol. 10, pp. 26514–26528, 2022.

[45] 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.

[46] 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.

[47] H. Sheng, Y. Zhou, L. Bai, and L. Shi, "Transfer state of health estimation based on cross-manifold embedding," *J. Energy Storage*, vol. 47, 2022, Art. no. 103555.

[48] H. Sheng, X. Liu, L. Bai, H. Dong, and Y. Cheng, "Small sample state of health estimation based on weighted Gaussian process regression," *J. Energy Storage*, vol. 41, 2021, Art. no. 102816.

[49] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning.*, London, England: MIT Press, 2006.

[50] B. Sahaand and K. Goebel, *Lithium-ion Battery Ageing Dataset*. NASA Ames Pognostics Data Rpository, 2007. [Online]. Available: https://ti.arc.nasa.gov/tech/dash/pcoe/prognostic-data-repository/#battery

[51] Y. Xing, E. W. Ma, K.-L. Tsui, and M. Pecht, "An ensemble model for predicting the remaining useful performance of lithium-ion batteries," *Microelectron. Rel.*, vol. 53, no. 6, pp. 811–820, 2013.

[52] H. Sheng, J. Xiao, Y. Cheng, Q. Ni, and S. Wang, "Short-term solar power forecasting based on weighted Gaussian process regression," *IEEE Trans. Ind. Electron.*, vol. 65, no. 1, pp. 300–308, Jan. 2018.

[53] X. Li, Q. Wang, Y. Yang, and J. Kang, "Correlation between capacity loss and measurable parameters of lithium-ion batteries," *Int. J. Elect. Power Energy Syst.*, vol. 110, pp. 819–826, 2019.

[54] D. Yang, J. Kleissl, C. A. Gueymard, H. T. Pedro, and C. F. Coimbra, "History and trends in solar irradiance and PV power forecasting: A preliminary assessment and review using text mining," *Sol. Energy*, vol. 168, pp. 60–101, 2018.

[55] S. Zhang, B. Zhai, X. Guo, K. Wang, N. Peng, and X. Zhang, "Synchronous estimation of state of health and remaining useful lifetime for lithium-ion battery using the incremental capacity and artificial neural networks," *J. Energy Storage*, vol. 26, 2019, Art. no. 100951.

[56] C. Weng, Y. Cui, J. Sun, and H. Peng, "On-board state of health monitoring of lithium-ion batteries using incremental capacity analysis with support vector regression," *J. Power Sources*, vol. 235, pp. 36–44, 2013.

[57] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning.*, Cambridge, MA, USA: MIT Press, 2012.

[58] A. Meenakshi and C. Rajian, "On a product of positive semidefinite matrices," *Linear Algebra Appl.*, vol. 295, no. 1–3, pp. 3–6, 1999.

[59] S. Boyd, S. P. Boyd, and L. Van den Berghe, *Convex Optimization.*, Cambridge, U.K.: Cambridge Univ. Press, 2004.