Journal of Energy Storage 73 (2023) 109244

Contents lists available at ScienceDirect

Journal of Energy Storage

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

Research Papers

State-space modeling for electrochemical performance of Li-ion batteries with physics-informed deep operator networks

Qiang Zheng [a,b,e], Xiaoguang Yin [c], Dongxiao Zhang [a,c,d,*]

[a] Eastern Institute for Advanced Study, Eastern Institute of Technology, Ningbo, Zhejiang 315200, P.R. China
[b] Department of Mechanical Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong SAR, P. R. China
[c] School of Environmental Science and Engineering, Southern University of Science and Technology, Shenzhen, Guangdong 518055, P.R. China
[d] Department of Mathematics and Theories, Peng Cheng Laboratory, Shenzhen, Guangdong 518000, P.R. China
[e] Department of Automation, University of Science and Technology of China, Hefei, Anhui, 230027, P.R. China

ARTICLE INFO

Keywords:
State-space modeling
Li-ion battery
Operator learning
Physics-informed deep operator networks

ABSTRACT

Online estimation of unobservable internal states is significant for safe operation of Li-ion batteries, and it constitutes one of the main functions of battery management system (BMS). The next-generation BMS expects model-based state estimation, especially with electrochemical models, which are accurate but often costly for solving. Therefore, it is required to build more easily executable state-space representation of electrochemical models for online state estimation. However, the traditional numerical methods for time discretization are relatively complicated, and the discretized system is not very flexible in modifying predictive time intervals. To address such issues, we introduce the concept of physics-informed operator learning for state-space modeling. Specifically, we propose an architecture, termed the physics-informed multiple-input operator network (PI-MIONet), to reformulate the state-space representation of the extended single particle (eSP) model. In this work, the PI-MIONet takes the Li-ion concentration of the whole electrode particle and current densities at the current time as the input functions, and predicts Li-ion concentration at any spatial-temporal location, which means that the forward predictions can be realized with user-defined step size. In addition, due to the capability of taking discretized functions as inputs, the PI-MIONet can be used for estimating states in the form of long vectors, and it can be conducted very efficiently, which makes it highly suitable for online applications in BMS. We verify the predictive performance of PI-MIONet through several synthetic experiments, and successfully apply it to the estimation of Li-ion concentration across the full particle with unscented Kalman filter algorithms.

1. Introduction

As an energy supply system to promote transportation electrification, lithium-ion (Li-ion) batteries have been widely used in electric vehicles, since they possess the major advantages of high energy density, long life cycle, low rate of self-discharging, and no memory effect [1]. However, in realistic and complex working scenarios, Li-ion batteries also face some problems, such as battery degradation and thermal runaway, which bring about certain challenges for their applications in electric vehicles. Therefore, it is imperative to construct an advanced battery management system (BMS), which can be used to monitor the internal electrochemical states and guarantee a safe operation during the life period of Li-ion batteries. Since the internal states, e.g., Li-ion concentration in electrodes and electrolytes, cannot be observed, they need to be estimated from observable quantities, such as current and voltage, which actually play a key role in BMS.

The methods for state estimation are diversified. In terms of most commonly used state-of-charge (SOC), which reflects the remaining battery power, the methods for its estimation mainly comprise Coulomb counting [2,3], voltage translation [4,5], data-driven methods [6,7], and model-based methods [8,9]. Coulomb counting highly depends on initial value and may suffer from the cumulative integration error, and the voltage translation method infers SOC from the predetermined open circuit voltage (OCV)-SOC lookup table using the OCV measurements, which requires a long execution period. As a consequence, the former two methods are not suitable for online applications in BMS. With the

* Corresponding author at: School of Environmental Science and Engineering, Southern University of Science and Technology, Shenzhen, Guangdong 518055, PR China.
E-mail address: zhangdx@sustech.edu.cn (D. Zhang).

https://doi.org/10.1016/j.est.2023.109244
Received 23 May 2023; Received in revised form 5 September 2023; Accepted 7 October 2023
Available online 21 October 2023

rapid development of artificial intelligence, data-driven methods have attracted substantial attention for SOC estimation; however, the predictive robustness will become problematic due to the lack of physical insights. In comparison, model-based methods can provide a relatively more reliable SOC estimation, since they can realize closed-loop control by leveraging state estimation algorithms. The accuracy of model-based methods depends on the choice of battery model, among which the equivalent circuit model (ECM) [10,11] and the electrochemical model (EM) [12,13] are the major ones. Certain studies have pointed out that state estimation (e.g., SOC) based on EMs will become the primary focus of the next-generation BMS [14]. The reasons for this are that, compared to ECM, EM-based state estimation can indicate the characteristics of electrochemical reactions and charging/discharging from the perspective of electrochemical principles, and thus it is beneficial to monitor the degradation process and tendency of state variations during the full life cycle [15].

Model-based state estimation is usually conducted by taking advantage of filter-based methods [16], e.g., Kalman filter and its diversified variants [17], for use in online scenarios. In order to meet the efficiency demands of online applications in BMS, it is required to modify the complex dynamic process with continuous time to a relatively simpler state-space model with discretized time. ECMs are commonly composed of a system of ordinary differential equations (ODEs), which are convenient for time discretization, and are often combined with extended Kalman filter (EKF) to perform state estimation [18,19]. In comparison, EMs are built with a system of partial differential equations (PDEs), which are relatively more difficult to be discretized with time than ODEs, and thus EMs need model reduction and local linearization prior to being combined with EKF [20]. Meanwhile, for the nonlinear PDE system, linearization is certain to bring about truncation errors. As a result, unscented Kalman filter (UKF) and particle filter (PF) could be more suitable, since they do not require local linearization [21–24]. Even though model linearization can be eliminated in UKF or PF, time discretization for the complex PDE system is still needed, and how to obtain a discretized system that is suitable for quantities with infinite dimensions, i.e., at any location, is not only important, but also challenging. Moreover, as a Bayesian method, PF needs to deal with an efficiency problem due to a large number of model executions, especially in online applications. Therefore, state-space modeling essentially realizes the mappings between functions with infinite dimensions, and it needs to not only approximate the original model as accurately as possible, but also execute efficiently.

Deep learning techniques can be adopted to fulfill the above two goals, since they possess strong fitting abilities, especially for nonlinear and high-dimensional problems [25], and they have also been introduced in battery modeling [26,27]. The universal approximation theorem constructs the theoretical basis for deep learning, and it states that neural networks can be used to approximate any continuous function [28]. Furthermore, there is another approximation theorem that has received less attention, which states that a neural network can approximate any continuous nonlinear functional [29] or operator, i.e., a mapping from one function to another function [30]. Based on the universal approximation theorem for operators, Lu et al. [31] designed a novel network architecture, termed the deep operator network (DeepONet), to learn explicit and implicit operators, e.g., in the form of PDEs. In terms of a dynamical system that can be described by ODEs or PDEs, its state-space modeling can also be understood from the perspective of the operator. Therefore, operator learning has great potential to build state and space functions in a data-driven manner, but trial work has not yet been seen in battery modeling.

Even though deep learning has been validated with powerful fitting capabilities, the huge demand for training data and the lack of physical constraints restrict its applications in natural science, where data are often sparse and costly to acquire. To address this issue, the physics-informed neural network (PINN) [32] was developed, and it actually transformed the way that we model the behavior of physical systems, for which we have some available observations and at least a parameterized PDE to provide additional constraints in a semi-supervised type of learning. The successful applications of PINN and its variants in diversified fields, e.g., fluid mechanics [33], biology [34], and material science [35], have contributed to the flourishment of scientific machine learning, which has become a new paradigm for scientific research. In the realm of batteries, great interests and advancements in physics-informed deep learning have occurred, with current emphases on failure diagnosis [36], lifetime prediction [37], and state estimation [38].

Inspired by the concept of PINN, Wang et al. [39] proposed the physics-informed DeepONet (PI-DeepONet) to alleviate the data burden of the original DeepONet. The PI-DeepONet is powerful, and we have adopted it in our previous work to learn a functional mapping from current to voltage [40], but generally, it is still in its infancy for applications in battery modeling. Moreover, in both DeepONet and PI-DeepONet, they can only map one function to another function, and have no ability to take multiple functions as inputs. To solve this problem, Jin et al. [41] proposed a multiple-input operator network (MIONet) with rigorous theoretical validations. In this work, we further expand the MIONet to a physics-informed version, i.e., PI-MIONet, and recast state-space modeling of batteries into an operator learning problem by leveraging PI-MIONet. Then, online state estimation can be conducted with PI-MIONet-based state-space models. The key point of this work is the functional perspective on state-space modeling and adopting the newly proposed PI-MIOMet to reformulate it. The feasibility of PI-MIONet for state-space modeling is validated by synthetic cases, and online estimation of Li-ion concentration in the range of whole electrode particles is also tested with a UKF algorithm. The advantages of the proposed method lie in several aspects. The first one is the elimination of explicit time discretization and local linearization for PDEs, leading to a relatively simple working flow without sacrificing model accuracy with the assistance of rigorous operator learning and physical constraints. The second one is that the trained PI-MIONet allows users to define the step size for forward prediction, and it can be evaluated with high efficiency, which endows it with great applicability and flexibility for online scenarios in BMS, especially in some extreme working environments, e.g., in extremely high/low temperature, where state estimation can be conducted more frequently to better evaluate battery performance [42]. Furthermore, the PI-MIONet takes as input the discretized functions, which makes it feasible to achieve estimation of states in the form of long vectors, i.e., Li-ion concentration across the whole particle.

The remainder of this paper proceeds as follows. In Section 2, we introduce the physical model for simulating electrochemical performance of Li-ion batteries, and the method for estimating internal states. We also place an emphasis on the introduction of PI-MIONet and reformulation of state-space modeling with it. In Section 3, we represent the fitting performance of PI-MIONet and its application in online state estimation with several synthetic cases. The conclusion and suggested directions for future research are given in Section 4.

2. Methods

2.1. Electrochemical modeling with the extended single particle model

It is widely known that the well-established pseudo-two-dimensional (P2D) model is the full-order battery model to characterize the electrochemical mechanism inside of batteries, including the diffusion process of Li-ion in solid and liquid phases, Ohm's law in solid and liquid phases, charge conservation, and Butler-Volmer kinetics [43–46]. As shown in Fig. 1a, the structure of the P2D model for Li-ion battery mainly contains three components denoted by a thickness: negative electrode (L_n); positive electrode (L_p); and separator (L_{sep}). The electrolyte is distributed among all of the three components for the transport of Li-ion. Even though the P2D model is the physical model with high fidelity, a number of parameters to be calibrated and a costly numerical-

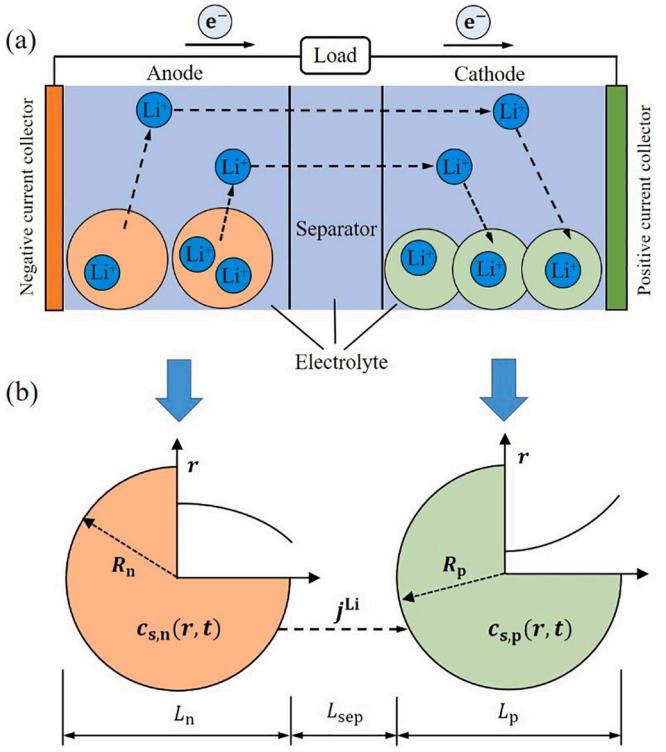

Fig. 1. Schematic diagram of the discharging process of a Li-ion battery, which is mainly composed of four parts, i.e., anode, separator, cathode, and electrolyte. (a) represents the full-order pseudo-two-dimensional (P2D) model, and (b) denotes the simplified single particle (SP) model. The transport direction of Li-ion and electrons in the charging process is opposite to that in the discharging process.

solving scheme hinder its applications in online scenarios, leading to several simplified versions. Among them, the single particle (SP) model is a classical one, in which all particles in the electrodes are assumed to behave in a similar manner, and thus each electrode is modeled as a single particle [47], as shown in Fig. 1b. Although the SP model is fast to execute, the neglect of potential gradient inside of the electrolyte makes it impossible to adequately describe battery dynamics at high C-rates [48]. To address this issue, the extended single particle (eSP) model was proposed by incorporating stress-enhanced diffusion and electrolyte concentration distribution into a modified SP model [49,50]. The eSP model has been validated to give voltage predictions with less than 0.630 ± 0.331 % error than that predicted by P2D for C-rates up to 3C, while offering a significantly higher computational efficiency that is approximately three times faster than the P2D model [49]. In this study, we utilize the eSP model as the underlying physical model for a Li-ion battery due to its straightforward structure and acceptable level of accuracy. This model is then utilized to generate data for the purpose of training operator networks.

The eSP model suggests that the movement of Li-ion in solid particles is influenced by both concentration gradient and mechanical stress field. The diffusion equation of Li-ion coupled with the stress-diffusion effect is defined as follows:

$$\frac{\partial c_{s,m}}{\partial t}+\frac{1}{r^2}\frac{\partial}{\partial t}\left[-r^2\left(1+\Theta_{s,m}c_{s,m}\right)D_{s,m}\frac{\partial}{\partial r}c_{s,m}\right]=0, m=n,p, \tag{1}$$

with boundary condition as:

$$\left.\frac{\partial c_{s,m}}{\partial r}\right|_{r=0}=0,\ -\left(1+\Theta_{s,m}c_{s,m}\right)D_{s,m}\left.\frac{\partial c_{s,m}}{\partial r}\right|_{r=R_m}=j_m^{Li}, \tag{2}$$

where $c_{s,m}$ represents the concentration of Li-ion in the solid phase; $D_{s,m}$ is the solid diffusion coefficient, i.e., $c_{s,n}, D_{s,n}$ for anode and $c_{s,p}, D_{s,p}$ for cathode; and $\Theta_{s,m}=\frac{2\Omega_{s,m}^2 E_{s,m}}{9RT\left(1-\nu_{s,m}\right)}$ is a lumped parameter that takes the stress-effect into account, in which $\Omega_{s,m}$ denotes the partial molar volume; $E_{s,m}$ is Young's modulus; R is the universal gas constant; T is temperature; and $\nu_{s,m}$ represents Poisson's ratio. It is assumed that the electrochemical reaction is uniform in each electrode, and therefore, the pore wall flux j_m^{Li} at the solid electrolyte interface can be given by:

$$j_n^{Li}=\frac{-i_{app}R_n}{3F(1-\varepsilon_n)L_n}, j_p^{Li}=\frac{i_{app}R_p}{3F(1-\varepsilon_p)L_p}, \tag{3}$$

where i_{app} represents the applied current density to the battery; F is Faraday constant; ε_n and ε_p denote the porosity of anode and cathode, respectively; and R_m and L_m $(m=n,p)$ are solid particle radius and electrode length, respectively.

Compared to the SP model, the eSP model, in addition to considering the impact of the diffusion-stress coupling effect, also accounts for the role of electrolyte physics, since the potential drop in the electrolyte phase can play a significant role in affecting Li-ion transport. As shown in Fig. 1a, the electrolyte is distributed in anode, separator, and cathode, and the continuous electrolyte concentration profiles can be approximated, in general, by polynomial functions as follows [49]:

$$c_{e,n}(x,t)=a_1\left(1-e^{-b_1 t}\right)x^2+a_2\left(1-e^{-b_1 t}\right)+c_{e,0}, \tag{4}$$

$$c_{e,p}(x,t)=a_3\left(1-e^{-b_2 t}\right)(L-x)^2+a_4\left(1-e^{-b_2 t}\right)+c_{e,0}, \tag{5}$$

$$\begin{aligned}c_{e,sep}(x,t)=&\frac{\left(a_3L_p^2+a_4\right)\left(1-e^{-b_2 t}\right)-\left(a_1L_n^2+a_2\right)\left(1-e^{-b_1 t}\right)}{L_{sep}}\\&\times\left(x-L_n-L_{sep}\right)+\left(a_3L_p^2+a_4\right)\left(1-e^{-b_2 t}\right)+c_{e,0},\end{aligned} \tag{6}$$

where coefficients a_k, b_k $(k=1,2,3,4)$ are further defined in Appendix A; and $c_{e,0}$ denotes the initial concentration of Li-ion in electrolyte. Then, the analytical equations for electrolyte potential can be derived based on the electrolyte charge conservation equations, and expressed as:

For the anode region $(0\le x\le L_n)$:

$$\phi_{e,n}(x,t)=\phi_{e,n}(0,t)+(1-t_+)\frac{2RT}{F}ln\frac{c_{e,n}(x,t)}{c_{e,n}(0,t)}-\frac{i_{app}}{2L_nk_{e,n}^{eff}}x^2. \tag{7}$$

For the separator region $(L_n\le x\le L_n+L_{sep})$:

$$\phi_{e,sep}(x,t)=\phi_{e,n}(0,t)+(1-t_+)\frac{2RT}{F}ln\frac{c_{e,sep}(x,t)}{c_{e,n}(0,t)}-\frac{i_{app}}{k_{e,sep}^{eff}}(x-L_n)-\frac{i_{app}L_n}{2k_{e,n}^{eff}}. \tag{8}$$

For the cathode region $(L_n+L_{sep}\le x\le L_n+L_{sep}+L_p)$:

$$\begin{aligned}\phi_{e,p}(x,t)=&\phi_{e,n}(0,t)+(1-t_+)\frac{2RT}{F}ln\frac{c_{e,p}(x,t)}{c_{e,n}(0,t)}+\frac{i_{app}}{2L_pk_{e,p}^{eff}}(L-x)^2\\&-\frac{i_{app}}{2}\left(\frac{L_n}{k_{e,n}^{eff}}+\frac{2L_s}{k_{e,sep}^{eff}}+\frac{L_p}{k_{e,p}^{eff}}\right),\end{aligned} \tag{9}$$

where $k_{e,m}^{eff}=k_m*\varepsilon_m^{brug}$ $(m=n,p,sep)$; k_m denotes the electrolyte conductivity in the three regions; *brug* represents the Bruggeman coefficient, which is 1.5; and t_+ is the Li-ion transference number. The electrolyte potential difference is obtained by the difference between the potentials taken from the electrode-current collector interface (i.e., $x=0$ and $x=L$), and calculated by:

$$\phi_{e,p}(L,t)-\phi_{e,n}(0,t)=(1-t_+)\frac{2RT}{F}ln\frac{c_{e,p}(L,t)}{c_{e,n}(0,t)}-\frac{i_{app}}{2}\left(\frac{L_n}{k_{e,n}^{eff}}+\frac{2L_s}{k_{e,sep}^{eff}}+\frac{L_p}{k_{e,p}^{eff}}\right). \tag{10}$$

Finally, the Li-ion battery terminal voltage can be calculated based on the solid phase potential difference between both ends of the cell, and can be expressed as:

$$\begin{aligned} V_{\mathrm{t}}(t) &= \phi_{\mathrm{s,p}}(t)\big|_{x=L} - \phi_{\mathrm{s,n}}(t)\big|_{x=0} \\ &= \left[U_{\mathrm{p}} + \phi_{\mathrm{e,p}}(L,t) + \eta_{\mathrm{p}}\right] - \left[U_{\mathrm{n}} + \phi_{\mathrm{e,n}}(0,t) + \eta_{\mathrm{n}}\right] \\ &= \left[U_{\mathrm{p}}\left(\frac{c_{\mathrm{s,p,surf}}(t)}{c_{\mathrm{s,p},max}}\right) + \phi_{\mathrm{e,p}}(L,t) + \eta_{\mathrm{p}}\right] - \left[U_{\mathrm{n}}\left(\frac{c_{\mathrm{s,n,surf}}(t)}{c_{\mathrm{s,n},max}}\right) + \phi_{\mathrm{e,n}}(0,t) + \eta_{\mathrm{n}}\right] \end{aligned} \tag{11}$$

where U_{m} $(\mathrm{m} = \mathrm{n}, \mathrm{p})$ denotes open circuit voltage, and it is generally a function of the normalized surface concentration, i.e., $c_{\mathrm{s,m,surf}}(t)/c_{\mathrm{s,m},max}$; and η represents the surface over-potential needed to drive the electrochemical reactions at the solid-electrolyte interface, and it can be defined by the Butler-Volmer equation:

$$\mathbf{i} \bullet \mathbf{n} = i_0\left[exp\left(\frac{\alpha_{\mathrm{n}} F\eta}{RT}\right) - exp\left(\frac{-\alpha_{\mathrm{p}} F\eta}{RT}\right)\right], \tag{12}$$

where $\mathbf{i}$ is Faradaic current density at the solid-electrolyte interface; $\mathbf{n}$ is the unit normal vector; i_0 represents the exchange current density; and α_{n} and α_{p} are transfer coefficients of electrochemical reactions for anode and cathode, respectively.

Even though the eSP model is sufficiently simplified compared to the P2D model while keeping its accuracy, it cannot be directly used for estimating internal states, e.g., solid phase Li-ion concentration. Instead, in a common way, the eSP model needs to be reformulated as a state-space representation, in order to guarantee the requirements of execution efficiency for online applications in BMS.

2.2. State estimation with unscented Kalman filter

In a dynamical system, the physical model built with continuous time needs to be reformulated as a state-space model with discrete time, in order to conduct optimal control and state estimation more efficiently. For a non-linear dynamical system, the discrete-time state-space model can be generally written as:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \tag{13}$$

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{u}_k), \tag{14}$$

where $\mathbf{x}_k$ represents the internal states to be estimated at k time-step; $\mathbf{u}_k$ means the input control; $\mathbf{z}_k$ is the measurable quantity, which can be used to infer $\mathbf{x}_k$ with state estimation methods; and $f(\bullet)$ and $h(\bullet)$ are termed state function and space function, respectively.

The unscented Kalman filter (UKF) is a powerful and efficient algorithm used for estimating the state of a non-linear system in real-time [51]. Unlike the traditional Kalman filter, which uses linear approximations to represent the system, the UKF applies a deterministic sampling approach, known as unscented transform, to accurately capture the mean and covariance of the system state. This approach provides a more robust estimation in nonlinear environments, and it also reduces the computational burden compared to other nonlinear filtering methods, e.g., particle filter [16].

The UKF algorithm consists of two main steps, i.e., the prediction step and the analysis step. In the prediction step, several sigma points $\mathscr{X}$ and their corresponding weights W^{m} and W^{c} are generated based on the current states $\mathbf{x}_k$ and prior covariance $\mathbf{P}_k$. Then, the sigma points need to be fed into the nonlinear process model, i.e., Eq. (13), to form a new predictive prior, which is a set of sigma points termed $\mathscr{Y}$, for the subsequent step. The mean and covariance of the prior can be computed with the unscented transform:

$$\widehat{\mathbf{x}} = \sum_{i=1}^{N} W_i^{\mathrm{m}}\, \mathscr{Y}_i, \tag{15}$$

$$\widehat{\mathbf{P}} = \sum_{i=1}^{N} W_i^{\mathrm{c}}(\mathscr{Y}_i - \widehat{\mathbf{x}})(\mathscr{Y}_i - \widehat{\mathbf{x}})^{\mathrm{T}} + \mathbf{Q}, \tag{16}$$

where $\mathbf{Q}$ quantifies the covariance of process noise; and N is the number of sigma points, and it is often determined as $2n+1$, where n refers to the length of $\mathbf{x}_k$.

In the analysis step, the UKF updates the predicted state with the latest measurement data. Firstly, the sigma points of the predictive prior need to be converted into measurements with the space function Eq. (14), i.e., $\mathscr{Z} = h(\mathscr{Y})$. Then, the mean ($\boldsymbol{\mu}_z$) and covariance ($\mathbf{P}_z$) of these points and the cross-covariance ($\mathbf{P}_{xz}$) between states and measurements can also be computed by unscented transform, which can be expressed as:

$$\boldsymbol{\mu}_z = \sum_{i=1}^{N} W_i^{\mathrm{m}}\, \mathscr{Z}_i, \tag{17}$$

$$\mathbf{P}_z = \sum_{i=1}^{N} W_i^{\mathrm{c}}\left(\mathscr{Z}_i - \boldsymbol{\mu}_z\right)\left(\mathscr{Z}_i - \boldsymbol{\mu}_z\right)^{\mathrm{T}} + \mathbf{R}, \tag{18}$$

$$\mathbf{P}_{xz} = \sum_{i=1}^{N} W_i^{\mathrm{c}}(\mathscr{Y}_i - \widehat{\mathbf{x}})\left(\mathscr{Z}_i - \boldsymbol{\mu}_z\right)^{\mathrm{T}}, \tag{19}$$

where $\mathbf{R}$ is the covariance of measurement noise. Finally, the internal states can be updated with the measurement z in the following form:

$$\mathbf{x} = \widehat{\mathbf{x}} + \mathbf{P}_{xz}\mathbf{P}_z^{-1}\left(z - \boldsymbol{\mu}_z\right), \tag{20}$$

and the updated covariance is given by:

$$\mathbf{P} = \widehat{\mathbf{P}} + \left(\mathbf{P}_{xz}\mathbf{P}_z^{-1}\right)\mathbf{P}_z\left(\mathbf{P}_{xz}\mathbf{P}_z^{-1}\right)^{\mathrm{T}}. \tag{21}$$

Actually, the prediction step and analysis step form the core of the UKF algorithm, and they are repeated at each time-step to provide a dynamic and accurate estimate of the system state. For additional details about UKF, one can refer to [51].

When it comes to state estimation for batteries with UKF in BMS, we take solid phase Li-ion concentration (c_{s}) as the internal state to be estimated from measured voltage in this work, since the commonly used state-of-charge (SOC) can be directly computed from c_{s} [52,53]. Then, the key step is to reformulate the eSP model into a state-space representation in the form of Eqs. (13) and (14). In the eSP model, the measurable voltage is analytically related to other quantities, e.g., Li-ion concentration and electrolyte potential, which means that the space function $h(\bullet)$ in Eq. (14) can be directly provided by the eSP model with analytical equations, and they can be executed efficiently. Meanwhile, the state function $f(\bullet)$ that describes the transition rule of c_{s} should be derived based on Eq. (1) with time discretization. Additionally, the numerical-solving scheme for time-discretized $f(\bullet)$, e.g., using COMSOL Multiphysics, is neither cheap nor efficient, leading to a certain difficulty for its online applications. In this work, we leverage deep operator learning to reformulate state-space modeling, in order to avoid complex time discretization and numerical solving, and meanwhile provide flexibility for prediction intervals and accelerate model execution efficiency.

2.3. State-space modeling with operator networks

2.3.1. Physics-informed multiple-input operator networks (PI-MIONet)

As a complement to approximate function, neural networks have been validated to be able to approximate any continuous operator, i.e., a mapping from one function to another function with infinite dimensions, based on the universal approximation theorem for operators [30]. Inspired by this theorem, Lu et al. [31] designed a specific network architecture, termed the deep operator network (DeepONet), to learn explicit operators, e.g., integral and fractional Laplacians, as well as

implicit operators, e.g., deterministic and stochastic differential equations. The DeepONet consists of two sub-networks: a branch net, which receives the functions in the form of discretized points as input; and a trunk net, which encodes the evaluation locations for the output functions. Even though DeepONet is powerful, it is only designed to learn operators defined on a single Banach space, i.e., the input of the operator is a single function. To expand the existing neural operators, Jin et al. [41] proposed the multiple-input operator network (MIONet) to receive multiple functions as inputs based on rigorous theoretical validations. Different from DeepONet, MIONet has several branch nets for encoding diversified input functions, while the trunk net still remains the same as DeepONet, using one net to encode the domain of the output function.

As shown in Fig. 2a, the MIONet aims to learn an operator $G:(u_1,\ldots,u_n)\rightarrow G(u_1,\ldots,u_n)$. For the ith function u_i, it needs to be

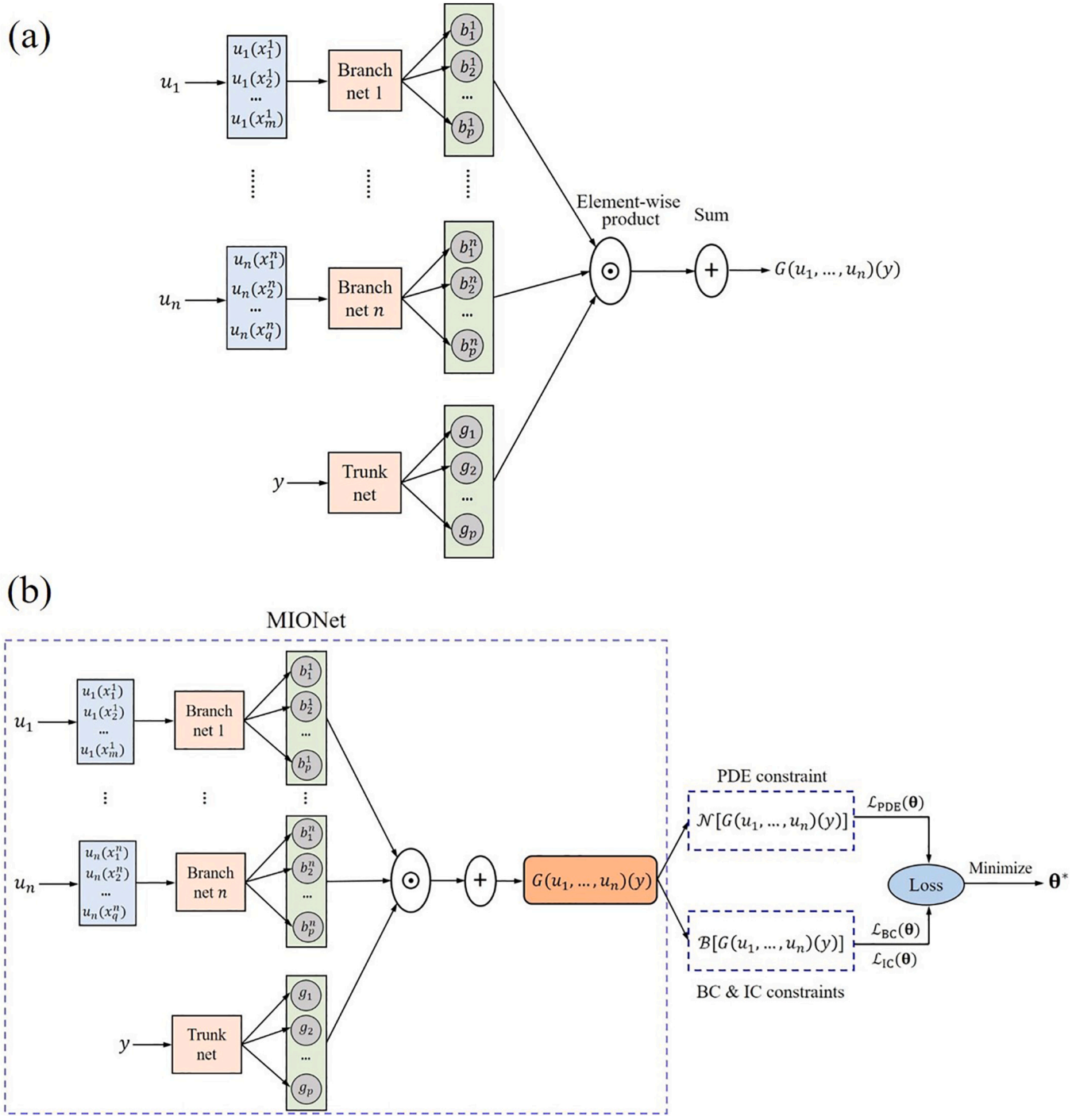

Fig. 2. The architecture of (a) MIONet and (b) physics-informed MIONet (PI-MIONet). In MIONet, suppose that there are n input functions (i.e., $u_1,\ldots,u_n$), each of which has its own discretization points (e.g., m locations $x_1^i, x_2^i,\ldots, x_m^i$ for evaluating ith function), and then the discretized values are fed into their corresponding branch nets. The trunk net takes coordinates y as input, where the output function $G(u_1,\ldots,u_n)$ can be evaluated. The outputs of the trunk net and different branch nets should keep the same dimension, so that they can conduct the element-wise production and sum operation to obtain the final output $G(u_1,\ldots,u_n)(y)$. Based on the outputs of MIONet, some additional regularization terms can be formulated via an automatic differentiation technique, as shown in subfigure (b), according to the information from partial differential equation (PDE), boundary condition (BC), and initial condition (IC). These physical constraints will guide the training of PI-MIONet, and bias its outputs to satisfy a given system of PDEs.

evaluated at m points to obtain $[u_i(x_1^i), u_i(x_2^i), \ldots, u_i(x_m^i)]$, which then serve as the input of the ith branch net. It is worth noting that there are no constraints for the number and location of these evaluation points. The trunk net takes the coordinates y as input where the output function can be evaluated. The output of each branch net $[b_1^i, b_2^i, \ldots, b_p^i]$ and that of the trunk net $[g_1, g_2, \ldots, g_p]$ should share the same length, so that they can conduct element-wise product and sum operation to obtain the final outputs, i.e.:

$$G(u_1, \ldots, u_n)(y) = \sum_{k=1}^{p} \left[\prod_{i=1}^{n} b_k^i \right] g_k. \tag{22}$$

In terms of training MIONet, the data should be organized in the form of structured tuples, i.e., $(u_1, \ldots, u_n, y, G(u_1, \ldots, u_n)(y))$, with the former $n+1$ terms serving as inputs and the last one for outputs. Since MIONet can receive input functions from different spaces, function samples should be drawn from their corresponding spaces. Suppose that we draw N_i separate function samples $u_i \in \mathscr{U}_i$, i.e., $\left\{u_i^{(j)}\right\}_{j=1}^{N_i}$, from function space $\mathscr{U}_i$, and thus we have $\prod_{i=1}^{n} N_i$ function samples in total. For each realization of the input function composite, we collect N_y points, i.e., $\left\{y^{(j)}\right\}_{j=1}^{N_y}$, to evaluate the output function. Therefore, the total number of training samples should be $\prod_{i=1}^{n} N_i \times N_y$, and the networks can be trained by the commonly used mean square error (MSE) as follows:

$$L_{\mathrm{MSE}} = \frac{1}{\prod_{i=1}^{n} N_i \times N_y} \sum_{i_1=1}^{N_1} \cdots \sum_{i_n=1}^{N_n} \times \sum_{j=1}^{N_y} \left[G\left(u_1^{(i_1)}, \ldots, u_n^{(i_n)}\right)\left(y^{(j)}\right) - \widehat{G}\left(u_1^{(i_1)}, \ldots, u_n^{(i_n)}\right)\left(y^{(j)}\right) \right]^2, \tag{23}$$

where $\widehat{G}\left(u_1^{(i_1)}, \ldots, u_n^{(i_n)}\right)\left(y^{(j)}\right)$ is the predicted value; and $G\left(u_1^{(i_1)}, \ldots, u_n^{(i_n)}\right)\left(y^{(j)}\right)$ is the corresponding label.

Since MIONet is used to learn a mapping between infinite-dimensional spaces, the amount of training samples should be as large as possible, in order to explore the function space more sufficiently. Consequently, the training dataset with size $\prod_{i=1}^{n} N_i \times N_y$ will be too massive, which will markedly aggravate the burden of data collection and memory requirements for training. In addition, compared to DeepONet, MIONet requires more training samples since it explores several input function spaces simultaneously. To alleviate the data demands, we follow the concept of the physics-informed DeepONet [39] and propose the physics-informed MIONet (PI-MIONet) as shown in Fig. 2b, and leverage the physical loss induced from the underlying governing equations to compensate for the huge cost of collecting data from real experiments.

It can be seen from Fig. 2b, in the architecture of PI-MIONet, that the additional physical loss coming from the constraints of partial differential equation (PDE), initial condition (IC), and boundary condition (BC) can be obtained through automatic differentiation techniques, owing to the differentiable structure of networks and the spatial-temporal coordinates taken by the trunk net as inputs. In terms of training data to make up the physical loss, we follow the above settings for MIONet and prepare different datasets to enforce IC, BC, and PDE constraints, for the purpose of exploring function spaces as efficiently as possible. The datasets $\left[\left\{u_i^{(j)}\right\}_{j=1}^{N_{i,\mathrm{IC}}}\right]_{i=1}^{n}$ and $\left\{y^{(j)}\right\}_{j=1}^{N_{y,\mathrm{IC}}}$ are prepared for encoding IC loss, and likewise, the datasets $\left[\left\{u_i^{(j)}\right\}_{j=1}^{N_{i,\mathrm{BC}}}\right]_{i=1}^{n}$ and $\left\{y^{(j)}\right\}_{j=1}^{N_{y,\mathrm{BC}}}$ are for BC loss, and the datasets $\left[\left\{u_i^{(j)}\right\}_{j=1}^{N_{i,\mathrm{PDE}}}\right]_{i=1}^{n}$ and $\left\{y^{(j)}\right\}_{j=1}^{N_{y,\mathrm{PDE}}}$ are for PDE loss. Consider a generic parametric PDE and its BC taking the following forms:

$$\mathscr{N}(u_1, \ldots, u_n, G(u_1, \ldots, u_n)) = 0, \tag{24}$$

$$\mathscr{B}(u_1, \ldots, u_n, G(u_1, \ldots, u_n)) = 0, \tag{25}$$

where $u_1, \ldots, u_n$ represents the parameters (i.e., input functions); $G(u_1, \ldots, u_n)$ denotes the corresponding solutions of PDE; and $\mathscr{N}$ and $\mathscr{B}$ are linear or nonlinear differential operators for representing PDE and BC, respectively. Finally, the training loss from different sources can be computed as follows:

$$L_{\mathrm{data}} = \frac{\sum_{i_1=1}^{N_1} \cdots \sum_{i_n=1}^{N_n} \sum_{j=1}^{N_y} \left[G\left(u_1^{(i_1)}, \ldots, u_n^{(i_n)}\right)\left(y^{(j)}\right) - \widehat{G}\left(u_1^{(i_1)}, \ldots, u_n^{(i_n)}\right)\left(y^{(j)}\right) \right]^2}{\prod_{i=1}^{n} N_i \times N_y}, \tag{26}$$

$$L_{\mathrm{IC}} = \frac{\sum_{i_1=1}^{N_{1,\mathrm{IC}}} \cdots \sum_{i_n=1}^{N_{n,\mathrm{IC}}} \sum_{j=1}^{N_{y,\mathrm{IC}}} \left[G\left(u_1^{(i_1)}, \ldots, u_n^{(i_n)}\right)\left(y^{(j)}\right) - \widehat{G}\left(u_1^{(i_1)}, \ldots, u_n^{(i_n)}\right)\left(y^{(j)}\right) \right]^2}{\prod_{i=1}^{n} N_{i,\mathrm{IC}} \times N_{y,\mathrm{IC}}}, \tag{27}$$

$$L_{\mathrm{BC}} = \frac{\sum_{i_1=1}^{N_{1,\mathrm{BC}}} \cdots \sum_{i_n=1}^{N_{n,\mathrm{BC}}} \sum_{j=1}^{N_{y,\mathrm{BC}}} \left[\mathscr{B}\left(u_1^{(i_1)}, \ldots, u_n^{(i_n)}, \widehat{G}\left(u_1^{(i_1)}, \ldots, u_n^{(i_n)}\right)\left(y^{(j)}\right)\right) \right]^2}{\prod_{i=1}^{n} N_{i,\mathrm{BC}} \times N_{y,\mathrm{BC}}}, \tag{28}$$

$$L_{\mathrm{PDE}} = \frac{\sum_{i_1=1}^{N_{1,\mathrm{PDE}}} \cdots \sum_{i_n=1}^{N_{n,\mathrm{PDE}}} \sum_{j=1}^{N_{y,\mathrm{PDE}}} \left[\mathscr{N}\left(u_1^{(i_1)}, \ldots, u_n^{(i_n)}, \widehat{G}\left(u_1^{(i_1)}, \ldots, u_n^{(i_n)}\right)\left(y^{(j)}\right)\right) \right]^2}{\prod_{i=1}^{n} N_{i,\mathrm{PDE}} \times N_{y,\mathrm{PDE}}}, \tag{29}$$

$$L_{\mathrm{total}} = \lambda_{\mathrm{data}} L_{\mathrm{data}} + \lambda_{\mathrm{IC}} L_{\mathrm{IC}} + \lambda_{\mathrm{BC}} L_{\mathrm{BC}} + \lambda_{\mathrm{PDE}} L_{\mathrm{PDE}}, \tag{30}$$

where the coefficients λ_{data}, λ_{IC}, λ_{BC}, and λ_{PDE} are weights used to adjust the relative importance of each item. Specifically, the amount of labelled data can even be reduced to zero under the guidance of physical constraints, so that the label-free PI-MIONet is built where L_{data} is no longer needed. In this work, we adopt the label-free PI-MIONet to reformulate state-space representation of the eSP model, with an aim to improve execution efficiency without sacrificing physical accuracy.

2.3.2. PI-MIONet based state-space modeling for Li-ion batteries

In this work, we take the eSP model to characterize the electrochemical performance of Li-ion battery, and the eSP model needs to be reformulated as state-space representations, i.e., Eqs. (13, 14), to serve the subsequent state estimation or optimal control in BMS. Here, we take the commonly used state-of-charge (SOC) as the state to be estimated. Since SOC is dimensionless, and it can be determined physically based on solid phase Li-ion concentration (i.e., c_s), we step back to take c_s as the internal state of interest that needs to be estimated from voltage measurements. Therefore, the state-space representation with respect to the eSP model can be tentatively written as:

$$\boldsymbol{c}_{s,k+1} = f\left(\boldsymbol{c}_{s,k}, i_{\mathrm{app},k}\right), \tag{31}$$

$$V_k = h\left(\boldsymbol{c}_{s,k}, i_{\mathrm{app},k}\right), \tag{32}$$

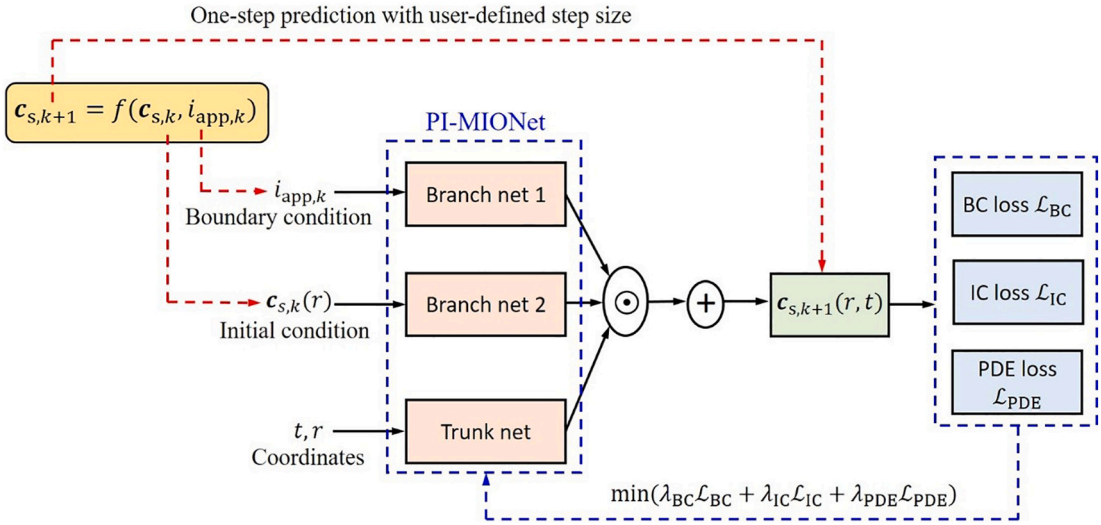

Fig. 3. Reformulation of state function with the label-free PI-MIONet. In order to approximate the state function with operator regression, the input vectors are treated as discretized values of functions and serve as the input of branch nets. The inputs for trunk net are spatial-temporal coordinates, which endow the trained model to be able to give predictions with user-defined step size.

where subscript k means time-step; applied current i_{app} plays the role of input control; and V is the terminal voltage. In the eSP model, only c_s needs to be solved from a PDE with numerical methods, while the other quantities, e.g., liquid phase Li-ion concentration (c_e) and potential (ϕ_e), can be computed with analytical solutions. This means that the state function $f(\cdot)$ needs to be reformulated based on the PDE, i.e., Eq. (1), while the space function $h(\cdot)$ can be directly derived from analytical equations, i.e., Eq. (4) – Eq. (12), which are very efficient to execute. Therefore, the main focus of state-space modeling in this study is how to acquire $f(\cdot)$ with both accuracy and efficiency.

Since the state c_s is actually a function of time and space, and the input signal i_{app} is also a function of time, the transition relationship of c_s can be learned from the aspect of operator regression, for which the proposed PI-MIONet can be adopted. From a different perspective, numerically solving Eq. (1) for c_s at time-step k+1 (i.e., $c_{s,k+1}$) is able to be approximated by operator learning, with c_s at previous time-step k serving as the initial condition (i.e., $c_{s,k}$), and the corresponding i_{app} playing the role of boundary condition (i.e., $i_{app,k}$). Therefore, regardless of the perspective of operator regression or approximating numerical scheme via operator learning, PI-MIONet is highly suitable for state-space modeling, specifically reformulating state function $f(\cdot)$ in this work.

As shown in Fig. 3, we adopt a label-free PI-MIONet with two branch nets to complete this task, with the first branch net taking $i_{app,k}$ as input, and the second one accepting $c_{s,k}$ as input. The trunk net takes time t and radial location r as inputs, and finally output c_s at any spatial-temporal location. It is worth noting that the step size for forward prediction now depends on the user-defined t, instead of being restricted to the original step interval embedded in the training data, which could come from PDE solutions via numerical solving. Upon completion of training, the model can give predictions within a moment, and the enforced physical constraints during training can guarantee the accuracy of predictions. Therefore, compared to the traditional methods, e.g., numerical discretization, for deriving state function, the proposed PI-MIONet-based method is both efficient and accurate, and can also allow users to arbitrarily modify the step size for forward prediction.

3. Results

3.1. PI-MIONet for predicting Li-ion concentration in the solid phase

3.1.1. Generation of training data for PI-MIONet

In this section, we build a one-step predictive model for Li-ion concentration in solid electrode particles with PI-MIONet, in which the step size is allowed to be defined by users. Since solid phase Li-ion concentration cannot be measured directly and PI-MIONet is actually used to approximate eSP model in a discrete manner, we utilize the eSP model working in different applied currents (i.e., i_{app}) to generate synthetic training and testing datasets, to verify the accuracy of PI-MIONet. In this study, we consider the scenarios of galvanostatic discharge under different C-rates, i.e., {1C, 2C, 3C, 4C, 5C}, and 1C rate is taken as 27 A/m^2 for lasting 3600 s. The eSP model is constructed using COMSOL Multiphysics, which includes the specialized module for Li-ion battery modeling, and the default parameter values with respect to different materials are also embedded. For this work, the anode and cathode materials are chosen as graphite and $LiCoO_2$, respectively. The requisite parameter values for creating the model, as outlined in Table 1, are determined based on previously published research [54,55]. Finally, the model can output solid phase Li-ion concentration at 21 equally distributed positions along the radial direction at each time-step. The number of time-steps is always set as 901 under different discharge rates, and the time-step size depends on the discharge rate N_{rate}, i.e., $\Delta t = 4/N_{rate}$ s. The solid phase Li-ion concentrations at discretized spatial-temporal grids under five discharge rates are collected for training PI-MIONet below. In addition, if the material parameters in Table 1 are changed, the trained PI-MIONet can be easily fine-tuned via transfer learning, since it is enabled to learn the general trend of output variables through one parameter setting.

3.1.2. Inference on new currents and initial Li-ion concentrations

In terms of model construction, the PI-MIONet with architecture shown in Fig. 3 is adopted in this work, and it is constructed with two branch nets and one trunk net. The applied current i_{app} serves as the input of the first branch net, whose input dimension is set as one due to the scalar value of i_{app}. The second branch net takes Li-ion concentration c_s of a whole electrode particle as input, which is discretized at 21 points, leading to the input dimension being set as 21. Apart from the input layer, the settings of the other layers for the two branch nets are the same, with four hidden layers and 200 neurons for each layer. The

Table 1
Parameters used in the eSP model.

Parameters	Symbol	Value		Unit
General				
Universal gas constant	R	8.314		J/mol/K
Temperature	T	298		K
Initial Li-ion concentration in electrolyte	$c_{e,0}$	2000		m^3/mol
Anodic symmetry factor	α_a	0.5		–
Cathodic symmetry factor	α_c	0.5		–
Separator				
Porosity	ε_{sep}	0.42		–
Length	L_{sep}	20		μm
Effective diffusion coefficient of electrolyte	$D^{eff}_{e,sep}$	$3\times10^{-10}\times\varepsilon_{sep}^{1.5}$		m^2/s
Effective electrical conductivity of electrolyte	$k^{eff}_{e,sep}$	$3\times\varepsilon_{sep}^{2.3}$		S/m
Electrodes		*Anode (graphite)*	*Cathode ($LiCoO_2$)*	
Porosity	ε_m	0.35	0.35	–
Length	L_m	70	70	μm
Radius of solid particle	$R_{s,m}$	12	12	μm
Maximum Li-ion concentration in particle	$c^{max}_{s,m}$	31,250	49,500	m^3/mol
Initial Li-ion concentration in particle	$c^{ini}_{s,m}$	25,000	25,800	m^3/mol
Exchange current density	$i_{0,m}$	36	26	A/m^2
Diffusion coefficient of solid phase	$D_{s,m}$	4×10^{-14}	1×10^{-13}	m^2/s
Partial molar volume	Ω_m	3.1×10^{-6}	-7.28×10^{-7}	m^3/mol
Young's modulus	E_m	15	375	GPa
Poisson's ratio	ν_m	0.3	0.2	–
Effective diffusion coefficient of electrolyte	$D^{eff}_{e,m}$	$3\times10^{-10}\times\varepsilon_n^{1.5}$	$3\times10^{-10}\times\varepsilon_p^{1.5}$	m^2/s
Effective electrical conductivity of electrolyte	$k^{eff}_{e,m}$	$3\times\varepsilon_n^{4.1}$	$3\times\varepsilon_p^{4.1}$	S/m
Effective electrical conductivity of solid phase	$k^{eff}_{s,m}$	$100\times\varepsilon_n^{1.5}$	$10\times\varepsilon_p^{1.5}$	S/m

Note: The symbol subscript "m" changes to "n" when representing anode, and "p" when representing cathode.

trunk net takes coordinates r and t as inputs, and thus the input layer only needs two neurons, while the remaining layer settings keep consistent with those of branch nets. Softplus function [56] with $\beta = 1$ is chosen as the activation function for both branch and trunk nets.

Based on the simulation results of the eSP model, the training data can be collected and rearranged into a suitable format, as suggested in Section 2.3. Specifically, the data under scenarios $i_{app} = \{1C, 3C, 5C\}$ are used for training, i.e., $N_{1,IC} = N_{1,BC} = N_{1,PDE} = 3$, and the remaining datasets in $i_{app} = \{2C, 4C\}$ are for testing. To enable PI-MIONet to give accurate predictions based on any initial condition, we adopt $\boldsymbol{c}_s$ of full particle at 10 time-steps, i.e., {1, 5, 10, 20, 30, 50, 200, 400, 600, 800}th steps across the original 901 steps, as initial concentrations for training. Therefore, the number of function $\boldsymbol{c}_s$ for the second branch net is 10, i.e., $N_{2,IC} = N_{2,BC} = N_{2,PDE} = 10$. The trunk net, in this experiment, receives different inputs when encoding constraints from different sources. For enforcing IC constraints, t is set as zero, and r is uniformly and randomly sampled in $[0, R_s]$ with size $N_{y,IC} = 5,000$, and the model outputs are the corresponding initial concentrations, which can be sampled from the input of the second branch net with necessary spatial interpolations. For enforcing BC constraints, r is set as zero or R_s, and t is randomly sampled in $[0, 3600/N_{rate}]$ with size $N_{y,BC} = 4,000$, and the model outputs should satisfy Eq. (2). For enforcing PDE constraints, r and t are both randomly sampled from their ranges with size $N_{y,PDE} = 10,000$, and the model outputs need to satisfy Eq. (1). It is worth noting that the training data only borrow the simulation results from the eSP model at 10 time-steps, and they only serve as initial constraints, which are often known in modeling scenarios. No labelled data for prediction are used in this work, and thus we can call the model as the label-free PI-MIONet, in which L_{data} in Eq. (30) is eliminated. The coefficients to weigh different losses in Eq. (30) are set as $\lambda_{IC} = 800, \lambda_{BC} = 10$, and $\lambda_{PDE} = 1e-7$. The Adam [57] optimizer with a learning rate of 5e-4 is employed to train PI-MIONet with MSE loss, which takes approximately 6 h on one Tesla T4 GPU for 15,000 epochs.

After training, the model can be tested on new currents and initial Li-ion concentrations. The prediction of concentration with trained PI-MIONet only takes about 0.05 s, while the numerical solving with COMSOL Multiphysics needs approximately 6–9 s, and thus the trained model is more appropriate for online scenarios. To further evaluate the predictive performance of trained model on anode, we set $i_{app} = 2C$ for the input of the first branch net, and select $\boldsymbol{c}_{s,a}$ of whole anodic particle at two time-stamps, i.e., 30 s and 600 s, from the original numerical results to serve as the initial concentration and simultaneously the inputs of the second branch net. We compare the predictions of $\boldsymbol{c}_{s,a}$ in the remaining period, i.e., until 1800 s, versus the reference values provided by numerical simulations. As shown in the upper two rows of Fig. 4 (i.e., from Fig. 4a1 to Fig. 4b2), the relative errors across the whole spatial-temporal domain are all less than 1 %. Furthermore, the predictions at six randomly selected time-steps, which are marked by dashed lines in subfigures of the left column, clearly match well with the reference values in the whole particle, as demonstrated in the subfigures of the right column.

Likewise, we also use $i_{app} = 4C$ as new current, and $\boldsymbol{c}_{s,a}$ at 15 s and 300 s as initial concentrations to test model performance. As shown in the lower two rows of Fig. 4 (i.e., from Fig. 4c1 to Fig. 4d2), the maximum relative errors are relatively larger than those of the results with $i_{app} = 2C$. However, the regions with relatively larger values are very tiny, almost all distributed near the particle surface (i.e., $r = R_s$) and at the end of simulation period, and this phenomenon exists in both scenarios with $i_{app} = 2C$ and 4C. This is because, when approaching the end of the discharging process, the anodic surface Li-ion concentration is very small and even approaches zero, leading to a larger relative error. Moreover, the higher is the discharge rate, the lower is the surface Li-ion concentration, and consequently the results with respect to $i_{app} = 4C$ exhibit larger deviations. However, the general fitting performance is still excellent, which is further confirmed by 1D matching in subfigures of the right column.

Similar to what we conducted for anode, we continue to validate the predictive performance of PI-MIONet on cathode. We also set two new current scenarios, i.e., $i_{app} = 2C$ and 4C, whose results are demonstrated in the upper and lower two rows of Fig. 5, respectively. Each row takes

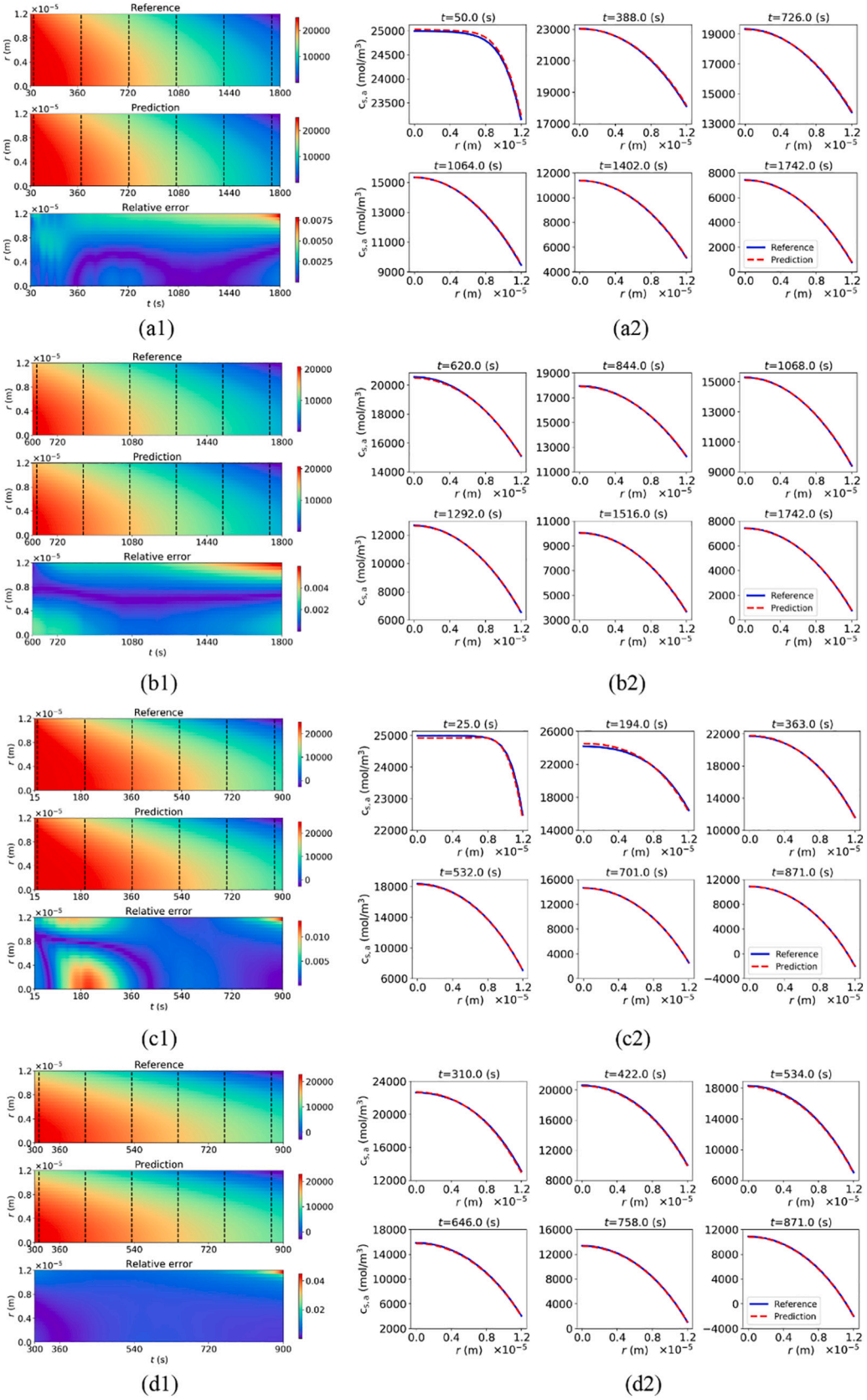

Fig. 4. The predictive performance of Li-ion concentration in anode ($c_{s,a}$) using PI-MIONet with new currents and new initial concentrations as model inputs. In terms of new currents, the upper two rows represent the performance at 2C discharge rate, while the lower two rows are for 4C discharge rate. Each row takes $c_{s,a}$ at different times (i.e., 30 s in (a1), 600 s in (b1), 15 s in (c1), and 300 s in (d1)) as initial conditions to perform forward predictions. In each row, the left column denotes the distribution of $c_{s,a}$ and its relative errors between references and predictions along temporal and spatial directions, and the right column demonstrates the fitting performance of $c_{s,a}$ at different times, which are also marked by dashed lines in the figures of the left column.

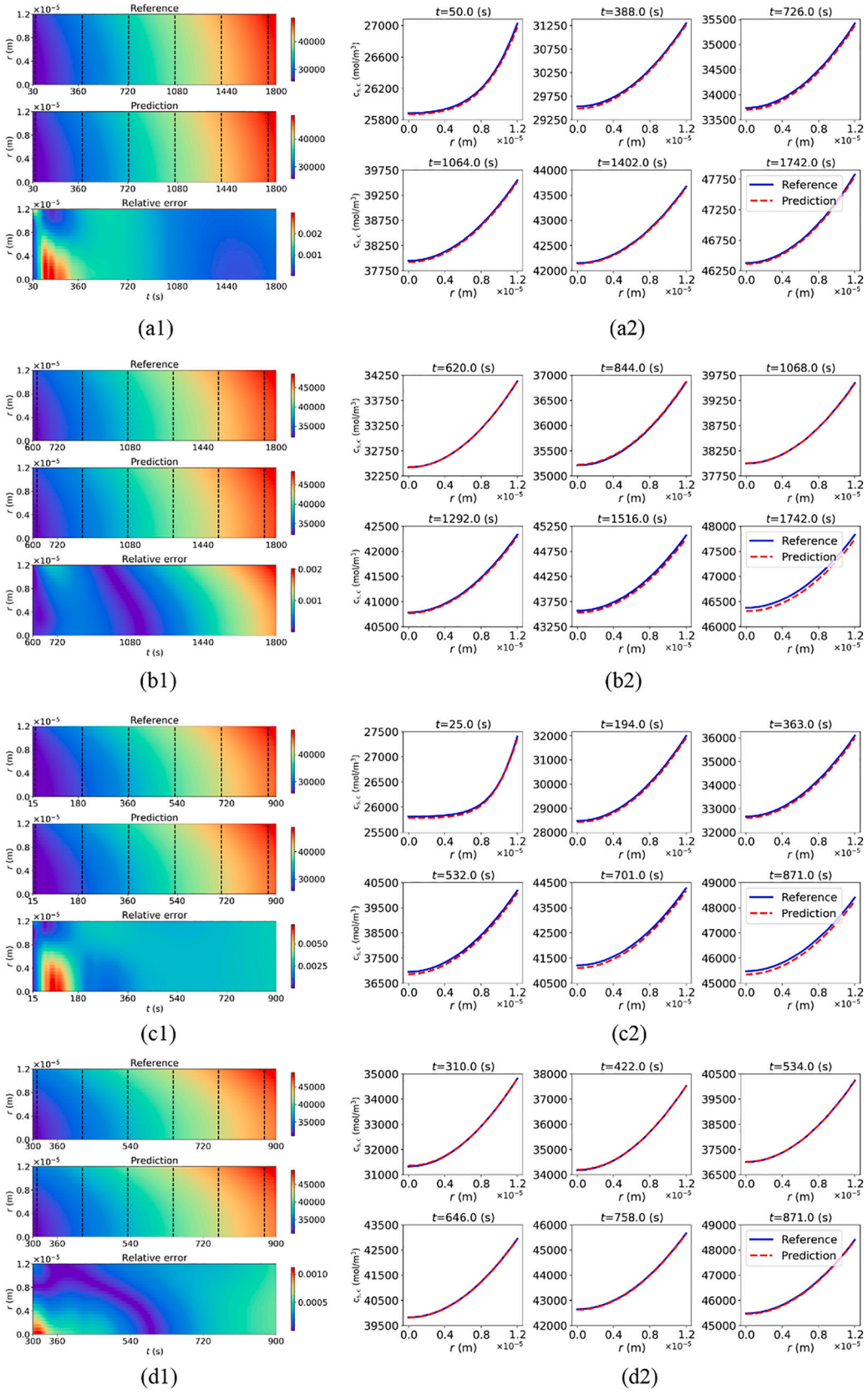

Fig. 5. The predictive performance of Li-ion concentration in cathode ($c_{s,c}$) using PI-MIONet with new currents and new initial concentrations as model inputs. The layout of subfigures is the same as that of Fig. 4.

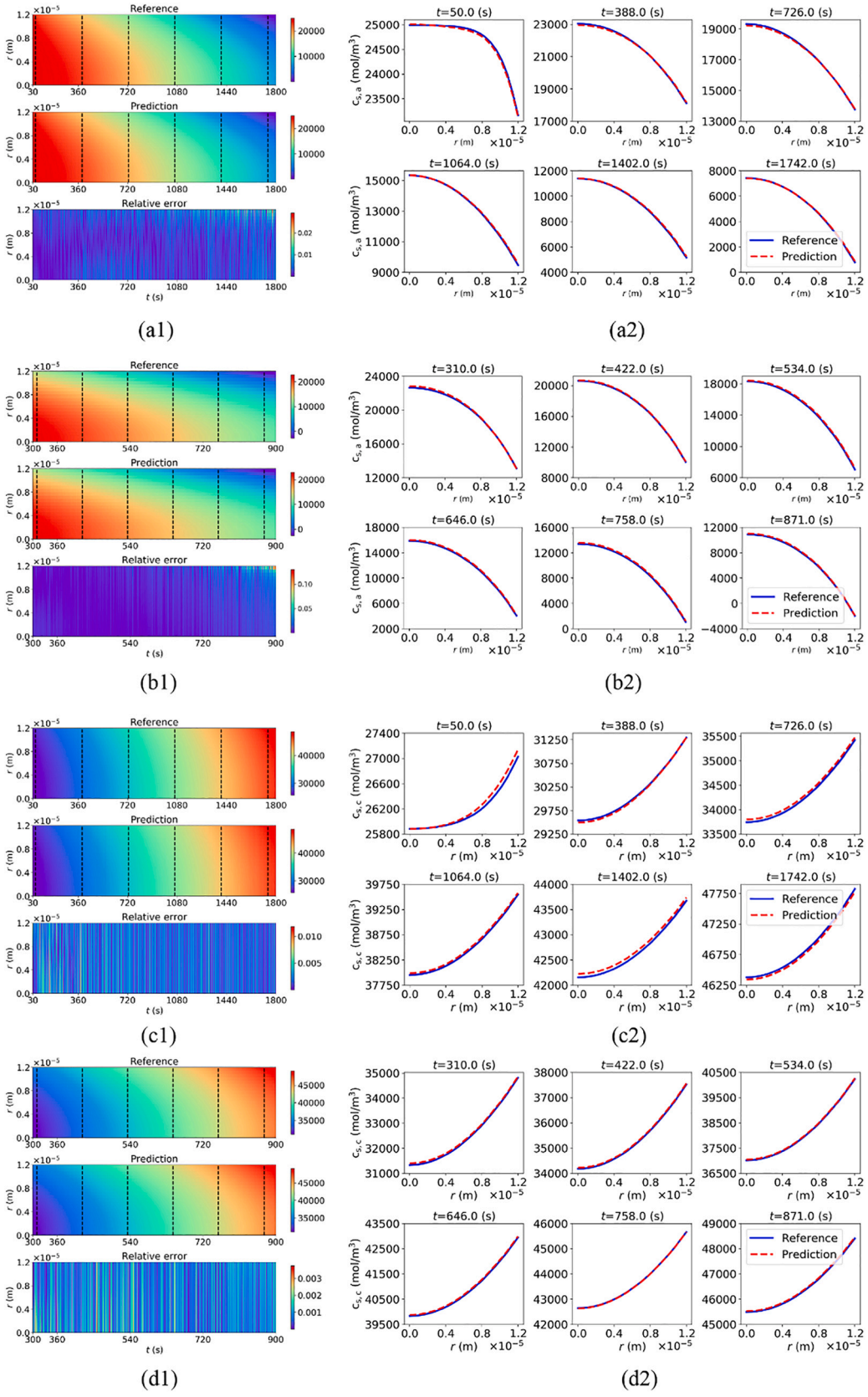

Fig. 6. The predictive performance of Li-ion concentration in two electrodes using PI-MIONet, which is trained with noisy initial conditions. The upper two rows represent the results of anode, while the lower two rows are for cathode. The first and third rows take 2C discharge rate and c_s at 30 s as the initial condition for model inputs, while the second and fourth rows take 4C discharge rate and c_s at 300 s as the initial condition for model inputs. The subfigures in the left column are 2D distributions of c_s, while those in the right column denote 1D fitting performance at some arbitrarily selected moments.

different initial $\boldsymbol{c}_{s,c}$ as model inputs, which are drawn from original numerical results at different times as we chose for anode previously. As shown in Fig. 5, the matching performance between predictions and references based on different currents and initial concentrations is quite excellent, with maximum relative errors less than 0.5 % across the whole spatial-temporal domain. Through the experiments on two electrodes, we conclude that PI-MIONet has effectively incorporated the physics of stress-enhanced diffusion of Li-ion in solid particles, and it can be utilized to reformulate state function for describing the transition of solid phase Li-ion concentration. More interestingly, the step interval for prediction, i.e., Δt between $\boldsymbol{c}_{s,k}$ and $\boldsymbol{c}_{s,k+1}$, can be determined by users, since $\boldsymbol{c}_{s,k}$ is understood as the initial condition, and consequently the input t naturally serves as Δt. It is worth noting that the flexibility for users to modify predictive step size cannot be easily realized through the traditional numerical discretization methods, and it definitely constitutes the advantage of data-driven methods.

3.1.3. Training with noisy initial condition

In this work, the state for estimation is Li-ion concentration $\boldsymbol{c}_s$, which is a discretized function, i.e., a long vector, in practice. When it comes to real online applications, the state-space functions certainly will be executed sequentially combined with state estimation methods, such as Kalman filter, which means that the estimated state at the previous time-step will serve as the initial condition for the next-step prediction. This may lead to a potential question of whether, if the estimated vector of $\boldsymbol{c}_s$ is not smooth, PI-MIONet can give reliable predictions based on the non-smooth initial conditions. To address this question, we repeat the training task in the last section by adding some noise to the input initial condition, i.e., the discretized evaluations of $\boldsymbol{c}_s$ for feeding the second branch net. Specifically, based on the normalized input $\boldsymbol{c}_s$ vector, we impose Gaussian noise, i.e., $\mathcal{N}(0, 0.01^2)$, with the same length of the input vector, and all of the other settings remain the same as previously.

After training, we evaluate the model performance by adding noise with the same magnitude as that used in training on initial conditions. The results of anode are shown in the upper two rows of Fig. 6, while the lower two rows represent those of cathode. In the first and third row, the discharge scenario with $i_{app} = 2C$ is considered, and $\boldsymbol{c}_s$ at 30 s of the original numerical results is selected as the initial condition. Meanwhile, in the second and fourth row, we consider the scenario with $i_{app} = 4C$ and take $\boldsymbol{c}_s$ at 300 s as the initial condition. Compared to the results in Fig. 4 and Fig. 5, the predictions with noisy model inputs generally have larger relative errors, which means that the added noise to the input function will deteriorate the performance of operator learning to a certain degree. Another interesting phenomenon is that the 2D distribution maps exhibit obvious stripes along the r-axis, especially for the results of cathode. This means that the forecasts of $\boldsymbol{c}_s$ show different degrees of overall deviation against their references at different times, which can be seen from some 1D matching plots in right-column subfigures, e.g., Fig. 6c2. In general, the predictive accuracy of PI-MIONet with noisy function as input is still acceptable, and it can be used for the subsequent experiments of state estimation. Furthermore, how much noise in input functions will make the PI-MIONet fail remains a valuable and open question, but it goes beyond the scope of this work.

In this study, we admit that only scenarios with constant current can be adopted to test the performance of PI-MIONet. This is because the input for branch net 1 (i.e., $i_{app,k}$) is essentially a scalar value, and it can represent current density function only in constant current scenarios. Consequently, we can interpolate on current density functions and make predictions under new conditions. If we consider scenarios with dynamic current, the proposed framework in Fig. 3 should be expanded to add a new branch, with an aim to receive the current density function itself as input, as did in our previous work [40], so that the interpolation and prediction on new dynamic current curves can be realized. Since the architecture of PI-MIONet needs to be modified, we leave it as a future work based on the current one.

3.2. Online estimation of Li-ion concentration with UKF

Based on the successful practice of state-space modeling with PI-MIONet, we continue to apply it to efficient online state estimation, which is actually the motivation for developing PI-MIONet. Herein, we take the discretized function of Li-ion concentration in solid particles, i.

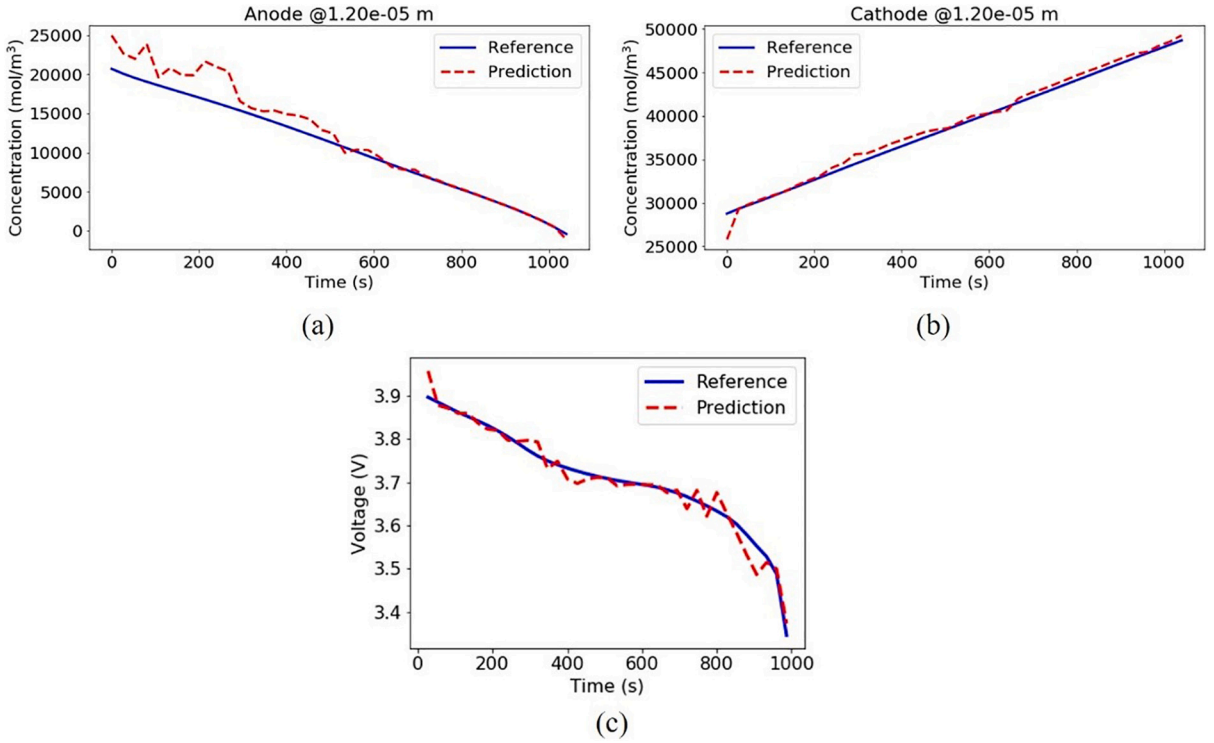

Fig. 7. Tracking of the surface Li-ion concentration of (a) anode and (b) cathode. The matching performance of terminal voltage is illustrated in (c).

e., $\boldsymbol{c}_{\mathrm{s}}(r)$, as the internal states to be estimated from observable voltages using the UKF algorithm introduced in Section 2.2. In this work, we utilize the voltage given by the eSP model working in $i_{\mathrm{app}} = 3\mathrm{C}$, whose stress-enhanced diffusion process is solved using COMSOL Multiphysics, as the synthetic measurements, which are also polluted by a Gaussian noise, i.e., $\mathcal{N}(0, \sigma^2)$ with $\sigma = 5$ mV. The numerically calculated Li-ion concentration serves as the reference to evaluate the estimation performance.

In order to validate the robustness of UKF, we set 20 % deviations to the referential initial concentration, and take it as the initial value for the subsequent state estimation. The estimated $\boldsymbol{c}_{\mathrm{s}}$ vector shares the same length (i.e., 21) as the input for the second branch net of PI-MIONet when we train it, and the anodic and cathodic $\boldsymbol{c}_{\mathrm{s}}$ vectors are concatenated, i.e., $\mathbf{x} = \left[\boldsymbol{c}_{\mathrm{s,a}}, \boldsymbol{c}_{\mathrm{s,c}}\right]^{\mathrm{T}}$, for estimation simultaneously. The sequential estimation continues until the voltage is lower than 3 V, and the step interval for that is taken as 25 s. In each estimation step, we record the updated values of concentration vector and the predictions of voltage given by the space function, i.e., Eq. (14). Since Li-ion concentration at the particle surface (i.e., $r = R_{\mathrm{s}}$) is critically important for reflecting SOC, we track its estimation trajectory to evaluate the performance. As shown in Fig. 7a and b, even though the initial values have a certain deviation, the estimated results can track their references with incorporating information from the measurements. The anodic concentration undergoes a relatively longer period to pursue its reference; however, it does not affect the overall estimation performance, which can also be validated by the matching of voltages, as shown in Fig. 7c.

It is widely known that the definition for SOC is non-unique, e.g., surface SOC or bulk SOC [14], but they both need to be computed from solid phase Li-ion concentration, with an emphasis on different components, e.g., surface or mean values. Actually, the estimation for Li-ion concentration across the whole particle is the special function of our proposed method compared to previous studies, in which only one or two components are estimated [52,53]. Moreover, with the knowledge of Li-ion concentration distribution, optimal control functions, such as fast charging, can be implemented to guarantee safe operation of batteries without experiencing over-charging or over-discharging problems [58].

To better demonstrate the matching performance of Li-ion concentration in the full particle, we select three times, i.e., 400 s, 640 s, and 980 s, to show the predictions versus their references and the relative errors between them. As shown in Fig. 8, in the early stage, e.g., at 400 s, the estimated concentrations across two electrodes have obvious deviations away from their references, while in the middle stage, e.g., at 640 s, the deviations decrease remarkably with certain fluctuations. In

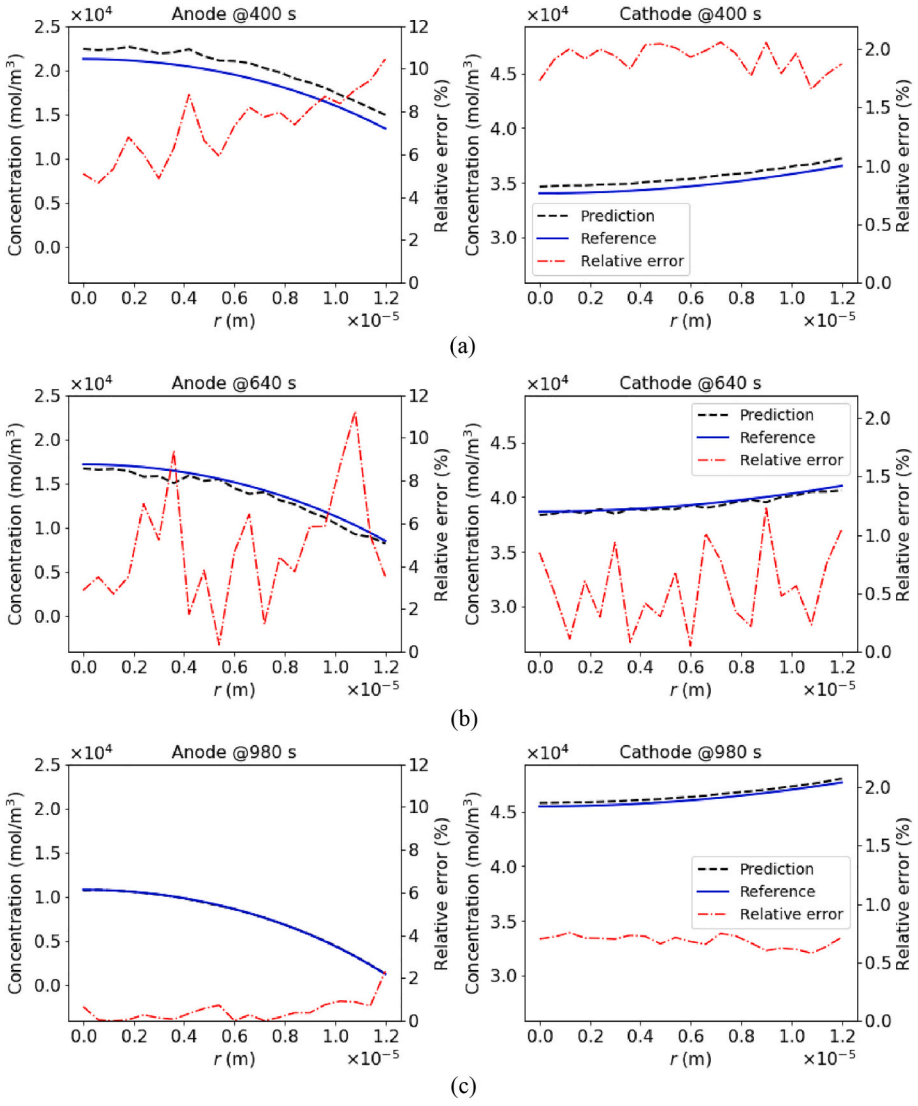

Fig. 8. The estimation performance of Li-ion concentration in the full particles of two electrodes at three time-moments, i.e., (a) 400 s, (b) 640 s, and (c) 980 s. The relative errors are also calculated between the references and predictions.

the late stage, e.g., at 980 s, the predictions of concentration match well with the references in anode, or show up very tiny and global deviations against the referential values in cathode. The pattern can also be quantified through relative errors, which overall undergo a fluctuating decreasing process until the terminal stage with not only small, but also smooth, values. Through the experiments of online Li-ion concentration estimation, we can further validate that the trained PI-MIONet is accurate, and can effectively and efficiently serve as the state function for being incorporated by state estimation methods.

4. Conclusion

Motivated by the demands for efficient online estimation of internal states of Li-ion batteries, we reformulated the state-space modeling from the perspective of physics-informed operator learning, and proposed a novel framework, termed the physics-informed multiple-input operator network (PI-MIONet), to achieve this goal. Through several experiments, PI-MIONet was validated to give reliable predictions of solid phase Li-ion concentration based on its arbitrary initial value under new current densities. Furthermore, the proposed PI-MIONet allowed users to arbitrarily modify the step interval for forward prediction, which cannot be easily realized by traditional methods with numerical discretization. Based on the highly efficient and physically accurate PI-MIONet, the state-space representation of the eSP model can be easily obtained. Subsequently, the estimation for states in the form of long vector, i.e., Li-ion concentration across the whole electrode particle, was successfully realized with the UKF algorithm, and the estimated distribution of Li-ion concentration in solid particle can be used for calculating SOC with distinct definitions, and it also could be meaningful for safe operation of batteries.

Even though the concept of physics-informed operator learning is validated to have the potential to reformulate PDE-based state-space modeling, much room for improvement remains. In this work, we only adopted the eSP model as the physical model to describe electrochemical performance, and proposed PI-MIONet to reformulate its state function, while leaving the space function directly provided by analytical expressions in the eSP model. Therefore, the ceiling accuracy of new state-space modeling depends on how accurate the eSP model is while characterizing internal electrochemical performance. In the near future, we plan to apply the architecture of physics-informed operator learning to more comprehensive physical models, such as the P2D model of high fidelity, and rebuild both state and space functions with P2D-informed operator learning. Furthermore, the electrochemical model also needs to be calibrated through real experimental data or public datasets before state-space modeling with PI-MIONet. Alternatively, the parameters to be calibrated can also be incorporated as the input of PI-MIONet, and consequently, the model calibration and state estimation can be realized simultaneously. This practice also endows the trained PI-MIONet with potential to be applicable to broader scenarios, where internal material parameters vary as the battery ages. Moreover, it is also recommended to construct state-space representation for electro-thermal models [59], which integrates electrochemical models with thermal effects, in order to simulate the dynamical system of real battery packs.

CRediT authorship contribution statement

Zheng, Yin, & Zhang: Conceptualization, Methodology, Writing-Original draft preparation; **Zheng**: Software, Data curation; Visualization, Investigation. **Zhang**: Supervision, Writing- Reviewing and Editing.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgments

This work is partially funded by the Shenzhen Key Laboratory of Natural Gas Hydrates (Grant No. ZDSYS20200421111201738), the SUSTech - Qingdao New Energy Technology Research Institute, and the China Postdoctoral Science Foundation (Grant No. 2020M682830).

Appendix A. Coefficients in the polynomial expressions for Li-ion concentration in the electrolyte phase

The coefficients a_k and b_k used in polynomial expressions for approximating Li-ion concentration in the electrolyte phase are given by [49]:

$$a_1 = -\frac{\varepsilon_n^{-brug} J}{2D_e L_n}, \tag{A.1}$$

$$a_2 = J\left[\varepsilon_n^{1-brug} L_n^2 + 2\varepsilon_p^{1-brug} L_p^2 + 6\varepsilon_p \varepsilon_{sep}^{-brug} L_s L_p + 3\varepsilon_{sep}^{1-brug} L_{sep}^2 + 3\varepsilon_n^{-brug} L_n \left(\varepsilon_p L_p + \varepsilon_{sep} L_{sep}\right)\right] \Big/ 6D_e \left(\varepsilon_n L_n + \varepsilon_p L_p + \varepsilon_{sep} L_{sep}\right) \tag{A.2}$$

$$a_3 = -\frac{\varepsilon_p^{-brug} J}{2D_e L_p}, \tag{A.3}$$

$$a_4 = -J\left[\varepsilon_p^{1-brug} L_p^2 + 2\varepsilon_n^{1-brug} L_n^2 + 6\varepsilon_n \varepsilon_{sep}^{-brug} L_n L_p + 3\varepsilon_{sep}^{1-brug} L_{sep}^2 + 3\varepsilon_n^{-brug} L_p \left(\varepsilon_n L_n + \varepsilon_{sep} L_{sep}\right)\right] \Big/ 6D_e \left(\varepsilon_n L_n + \varepsilon_p L_p + \varepsilon_{sep} L_{sep}\right) \tag{A.4}$$

$$b_1 = -6D_e \varepsilon_n^{brug-1} \varepsilon_p^{brug} \varepsilon_{sep}^{brug} \left(\varepsilon_n L_n + \varepsilon_p L_p + \varepsilon_{sep} L_{sep}\right) \Big/ L_n \left[2\varepsilon_n^{brug} \varepsilon_p \varepsilon_{sep}^{brug} L_p^2 + 2\varepsilon_p^{1+brug} L_p \left(\varepsilon_{sep}^{brug} L_n + 3\varepsilon_n^{brug} L_{sep}\right) + \varepsilon_p^{brug} \varepsilon_{sep} L_{sep} \left(2\varepsilon_{sep}^{brug} L_n + 3\varepsilon_n^{brug} L_{sep}\right)\right] \tag{A.5}$$

$$b_2 = -6D_e \varepsilon_p^{brug-1} \varepsilon_n^{brug} \varepsilon_{sep}^{brug} \left(\varepsilon_n L_n + \varepsilon_p L_p + \varepsilon_{sep} L_{sep}\right) \Big/ L_p \left[2\varepsilon_n \varepsilon_p^{brug} \varepsilon_{sep}^{brug} L_n^2 + 2\varepsilon_n^{1+brug} L_n \left(\varepsilon_{sep}^{brug} L_p + 3\varepsilon_p^{brug} L_{sep}\right) + \varepsilon_n^{brug} \varepsilon_{sep} L_{sep} \left(2\varepsilon_{sep}^{brug} L_p + 3\varepsilon_p^{brug} L_{sep}\right)\right] \tag{A.6}$$

where $brug = 1.5$ is the Bruggeman coefficient; $J = (1 - t_+)\frac{i_{app}}{F}$, in which t_+ is the Li-ion transference number and set as 0.363, and F is the Faraday constant and set as 96,485.33; and $D_e = 3 \times 10^{-10}$ is the diffusion coefficient in the electrolyte phase.

References

[1] L. Lu, X. Han, J. Li, J. Hua, M. Ouyang, A review on the key issues for lithium-ion battery management in electric vehicles, J. Power Sources 226 (2013) 272–288.
[2] K.S. Ng, C. Moo, Y. Chen, Y. Hsieh, Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries, Appl. Energy 86 (2009) 1506–1511.
[3] S. Zhang, X. Guo, X. Dou, X. Zhang, A data-driven coulomb counting method for state of charge calibration and estimation of lithium-ion battery, Sustain. Energy Technol. Assess. 40 (2020) 100752.
[4] Q. Yu, C. Wan, J. Li, L. E, X. Zhang, Y. Huang, T. Liu, An open circuit voltage model fusion method for state of charge estimation of lithium-ion batteries, Energies 14 (2021) 1797.
[5] S. Tong, M.P. Klein, J.W. Park, On-line optimization of battery open circuit voltage for improved state-of-charge and state-of-health estimation, J. Power Sources 293 (2015) 416–428.
[6] M. Fasahat, M. Manthouri, State of charge estimation of lithium-ion batteries using hybrid autoencoder and Long Short Term Memory neural networks, J. Power Sources 469 (2020) 228375.
[7] R. Xiong, F. Sun, X. Gong, C. Gao, A data-driven based adaptive state of charge estimator of lithium-ion polymer battery used in electric vehicles, Appl. Energy 113 (2014) 1421–1433.
[8] W. Li, Y. Fan, F. Ringbeck, D. Jöst, X. Han, M. Ouyang, D.U. Sauer, Electrochemical model-based state estimation for lithium-ion batteries with adaptive unscented Kalman filter, J. Power Sources 476 (2020) 228534.
[9] B. Jiang, H. Dai, X. Wei, T. Xu, Joint estimation of lithium-ion battery state of charge and capacity within an adaptive variable multi-timescale framework considering current measurement offset, Appl. Energy 253 (2019) 113619.
[10] X. Hu, S. Li, H. Peng, A comparative study of equivalent circuit models for Li-ion batteries, J. Power Sources 198 (2012) 359–367.
[11] J. Yang, Y. Cai, C. Pan, C. Mi, A novel resistor-inductor network-based equivalent circuit model of lithium-ion batteries under constant-voltage charging condition, Appl. Energy 254 (2019) 113726.
[12] C. Li, N. Cui, C. Wang, C. Zhang, Simplified electrochemical lithium-ion battery model with variable solid-phase diffusion and parameter identification over wide temperature range, J. Power Sources 497 (2021) 229900.
[13] A.J. Bard, L.R. Faulkner, H.S. White, Electrochemical Methods: Fundamentals and Applications, John Wiley & Sons, 2022.
[14] W. Li, D. Cao, D. Jöst, F. Ringbeck, M. Kuipers, F. Frie, D.U. Sauer, Parameter sensitivity analysis of electrochemical model-based battery management systems for lithium-ion batteries, Appl. Energy 269 (2020) 115104.
[15] N.A. Chaturvedi, R. Klein, J. Christensen, J. Ahmed, A. Kojic, Algorithms for advanced battery-management systems, IEEE Contr. Syst. Mag. 30 (2010) 49–68.
[16] Y. Wang, J. Tian, Z. Sun, L. Wang, R. Xu, M. Li, Z. Chen, A comprehensive review of battery modeling and state estimation approaches for advanced battery management systems, Renew. Sust. Energ. Rev. 131 (2020) 110015.
[17] P. Shrivastava, T.K. Soon, M.Y.I.B. Idris, S. Mekhilef, Overview of model-based online state-of-charge estimation using Kalman filter family for lithium-ion batteries, Renew. Sust. Energ. Rev. 113 (2019) 109233.
[18] J. Lee, O. Nam, B.H. Cho, Li-ion battery SOC estimation method based on the reduced order extended Kalman filtering, J. Power Sources 174 (2007) 9–15.
[19] F. Yang, D. Shi, K. Lam, Modified extended Kalman filtering algorithm for precise voltage and state-of-charge estimations of rechargeable batteries, J. Energy Storage 56 (2022) 105831.
[20] X. Han, M. Ouyang, L. Lu, J. Li, Simplification of physics-based electrochemical model for lithium ion battery on electric vehicle. Part I: diffusion simplification and single particle model, J. Power Sources 278 (2015) 802–813.
[21] J. Zhang, C. Xia, State-of-charge estimation of valve regulated lead acid battery based on multi-state Unscented Kalman Filter, Int. J. Elec. Power 33 (2011) 472–476.
[22] Z. Chen, L. Yang, X. Zhao, Y. Wang, Z. He, Online state of charge estimation of Li-ion battery based on an improved unscented Kalman filter approach, Appl. Math. Model. 70 (2019) 532–544.
[23] Z. Chen, H. Sun, G. Dong, J. Wei, J.I. Wu, Particle filter-based state-of-charge estimation and remaining-dischargeable-time prediction method for lithium-ion batteries, J. Power Sources 414 (2019) 158–166.
[24] S. Schwunk, N. Armbruster, S. Straub, J. Kehl, M. Vetter, Particle filter for state of charge and state of health estimation for lithium–iron phosphate batteries, J. Power Sources 239 (2013) 705–710.
[25] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444.
[26] Y. Liu, Y. He, H. Bian, W. Guo, X. Zhang, A review of lithium-ion battery state of charge estimation based on deep learning: directions for improvement and future trends, J. Energy Storage 52 (2022) 104664.
[27] Y. Zhang, Y. Li, Prognostics and health management of Lithium-ion battery using deep learning methods: a review, Renew. Sust. Energ. Rev. 161 (2022) 112282.
[28] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (1989) 359–366.
[29] T. Chen, H. Chen, Approximations of continuous functionals by neural networks with application to dynamic systems, IEEE Trans. Neural Netw. 4 (1993) 910–918.
[30] T. Chen, H. Chen, Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems, IEEE Trans. Neural Netw. 6 (1995) 911–917.
[31] L. Lu, P. Jin, G. Pang, Z. Zhang, G.E. Karniadakis, Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators, Nat. Mach. Intell. 3 (2021) 218–229.
[32] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.
[33] S. Cai, Z. Wang, F. Fuest, Y.J. Jeon, C. Gray, G.E. Karniadakis, Flow over an espresso cup: inferring 3-D velocity and pressure fields from tomographic background oriented Schlieren via physics-informed neural networks, J. Fluid Mech. 915 (2021) A102.
[34] M. Liu, L. Liang, W. Sun, A generic physics-informed neural network-based constitutive model for soft biological tissues, Comput. Methods Appl. Mech. Eng. 372 (2020) 113402.
[35] K. Shukla, P.C. Di Leoni, J. Blackshire, D. Sparkman, G.E. Karniadakis, Physics-informed neural network for ultrasound nondestructive quantification of surface breaking cracks, J. Nondestruct. Eval. 39 (2020) 1–20.
[36] D.P. Finegan, J. Zhu, X. Feng, M. Keyser, M. Ulmefors, W. Li, M.Z. Bazant, S. J. Cooper, The application of data-driven methods and physics-based learning for improving battery safety, Joule 5 (2021) 316–329.
[37] M. Aykol, C.B. Gopal, A. Anapolsky, P.K. Herring, B. van Vlijmen, M.D. Berliner, M. Z. Bazant, R.D. Braatz, W.C. Chueh, B.D. Storey, Perspective—combining physics and machine learning to predict battery lifetime, J. Electrochem. Soc. 168 (2021) 30525.
[38] W. Li, J. Zhang, F. Ringbeck, D. Jöst, L. Zhang, Z. Wei, D.U. Sauer, Physics-informed neural networks for electrode-level state estimation in lithium-ion batteries, J. Power Sources 506 (2021) 230034.
[39] S. Wang, H. Wang, P. Perdikaris, Learning the solution operator of parametric partial differential equations with physics-informed DeepONets, Sci. Adv. 7 (2021) i8605.
[40] Q. Zheng, X. Yin, D. Zhang, Inferring electrochemical performance and parameters of Li-ion batteries based on deep operator networks, J. Energy Storage 65 (2023) 107176.
[41] P. Jin, S. Meng, L. Lu, MIONet: learning multiple-input operators via tensor product, SIAM J. Sci. Comput. 44 (2022) A3490–A3514.
[42] X. Hu, Z. Deng, X. Lin, Y. Xie, R. Teodorescu, Research directions for next-generation battery management solutions in automotive applications, Renew. Sust. Energ. Rev. 152 (2021) 111695.
[43] J. Newman, W. Tiedemann, Porous-electrode theory with battery applications, AICHE J. 21 (1975) 25–41.
[44] W. Lai, F. Ciucci, Mathematical modeling of porous battery electrodes—revisit of Newman's model, Electrochim. Acta 56 (2011) 4369–4377.
[45] C. Wang, W.B. Gu, B.Y. Liaw, Micro-macroscopic coupled modeling of batteries and fuel cells: I. Model development, J. Electrochem. Soc. 145 (1998) 3407.
[46] M.Z. Bazant, Theory of chemical kinetics and charge transfer based on nonequilibrium thermodynamics, Acc. Chem. Res. 46 (2013) 1144–1160.
[47] S. Santhanagopalan, Q. Guo, P. Ramadass, R.E. White, Review of models for predicting the cycling performance of lithium ion batteries, J. Power Sources 156 (2006) 620–628.
[48] S. Santhanagopalan, Q. Guo, R.E. White, Parameter estimation and model discrimination for a lithium-ion cell, J. Electrochem. Soc. 154 (2007) A198.
[49] J. Li, N. Lotfi, R.G. Landers, J. Park, A single particle model for lithium-ion batteries with electrolyte and stress-enhanced diffusion physics, J. Electrochem. Soc. 164 (2017) A874.
[50] X. Zhang, W. Shyy, A.M. Sastry, Numerical simulation of intercalation-induced stress in Li-ion battery electrode particles, J. Electrochem. Soc. 154 (2007) A910.
[51] S.J. Julier, J.K. Uhlmann, Unscented filtering and nonlinear estimation, Proc. IEEE 92 (2004) 401–422.
[52] L. Ren, G. Zhu, J. Kang, J.V. Wang, B. Luo, C. Chen, K. Xiang, An algorithm for state of charge estimation based on a single-particle model, J. Energy Storage 39 (2021) 102644.
[53] H. Fang, Y. Wang, Z. Sahinoglu, T. Wada, S. Hara, State of charge estimation for lithium-ion batteries: an adaptive approach, Control. Eng. Pract. 25 (2014) 45–54.
[54] M. Torchio, L. Magni, R.B. Gopaluni, R.D. Braatz, D.M. Raimondo, Lionsimba: a matlab framework based on a finite volume model suitable for li-ion battery design, simulation, and control, J. Electrochem. Soc. 163 (2016) A1192.
[55] S. Han, Y. Tang, S.K. Rahimian, A numerically efficient method of solving the full-order pseudo-2-dimensional (P2D) Li-ion cell model, J. Power Sources 490 (2021) 229571.
[56] P. Ramachandran, B. Zoph, Q.V. Le, Searching for Activation Functions, arXiv preprint arXiv:1710.05941, 2017.
[57] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, arXiv preprint arXiv:1412.6980, 2014.
[58] M.A. Hannan, M.H. Lipu, A. Hussain, A. Mohamed, A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: challenges and recommendations, Renew. Sust. Energ. Rev. 78 (2017) 834–854.
[59] L.H. Saw, K. Somasundaram, Y. Ye, A. Tay, Electro-thermal analysis of lithium iron phosphate battery for electric vehicles, J. Power Sources 249 (2014) 231–238.