

Research papers

Continual learning for online state of charge estimation across diverse lithium-ion batteries

Jiaqi Yao^{ID}*, Bowen Zheng, Julia Kowal^{ID}

Department of Electrical Energy Storage Technology (EET), Technische Universität Berlin, Einsteinufer 11, 10587, Berlin, Germany

ARTICLE INFO

Keywords:

Lithium-ion batteries
Battery management systems
State of charge estimation
Continual learning
Progressive neural networks

ABSTRACT

An accurate estimation of the state of charge (SOC) ensures the safe and optimized usage of lithium-ion battery systems. With the rapid advances and accelerated iteration of commercial lithium-ion battery cells, the potential benefit for energy storage systems to use different types of battery cells in a mixture emerges. To accommodate this background, this paper proposes a novel continual learning framework for online SOC estimation across diverse lithium-ion batteries. The proposed framework uniquely combines the progressive neural network (PNN) framework with temporal convolutional networks (TCNs), enabling efficient temporal feature extraction while supporting robust forward knowledge transfer and fast adaptation to new battery types without suffering from catastrophic forgetting of the previously learned knowledge. Moreover, in-depth analyses of the interaction between tasks and the influence of task ordering on the model's performance are provided. Experiment results on the three public lithium-ion battery drive cycle datasets showcase that the proposed continual learning framework is not only able to achieve state-of-the-art accuracy for the SOC estimation of different types of lithium-ion batteries simultaneously, but also has proven to significantly reduce model complexity compared to utilizing multiple conventional single-task models. To the best of our knowledge, this work is the first to apply continual learning to SOC estimation of different types of batteries. Therefore, we believe this work sets up a benchmark for the task of SOC estimation of hybrid lithium-ion battery systems.

1. Introduction

With the rapidly growing awareness of environmental protection's importance, clean energy [1–3], and the corresponding energy storage technologies [4–6] are brought under the spotlight. As one of the most popular electrical energy storage, lithium-ion batteries have established their dominance in the market through a variety of applications, ranging from battery energy storage systems [7,8] to portable electronic devices [9,10]. Regardless of the specific applications, an accurate estimation of the state of charge (SOC) is crucial [11–13]. At some time t , the state of charge of a battery SOC_t is defined as:

$$SOC_t = \frac{Q_t}{C_{ref}} \quad (1)$$

where C_{ref} is the reference capacity and Q_t is the remaining quantity of charge at the given time t . SOC serves as an indicator of the remaining available capacity of a battery and is often determined by the so-called battery management system (BMS) [14,15]. A BMS monitors the behavior of the battery system in real time and carries out critical analyses using the measurements taken, based on which it provides

key information about the system to the user and further regulates the battery's operation [16–18], such as energy management [19], guaranteeing safe charging and discharging [14], and carrying out cell balancing [20]. In fact, many of the aforementioned functionalities of BMSs are based on the determination of SOC. Therefore, it is safe to say that the accurate determination of SOC not only ensures safety for the users, but also is the key to an optimized usage of the battery system.

However, SOC has to be estimated since it is not directly measurable [13,21,22]. In general, there are three categories of SOC estimation methods [23,24]: direct measurement approaches, model-based approaches, and data-driven approaches. Direct measurement approaches estimate SOC using the derived knowledge from the measurement only, such as coulomb counting [25,26], open circuit voltage (OCV) lookup [27,28], and electrochemical impedance lookup [29,30]. Although it seems that direct measurement approaches are very intuitive, they each have fatal drawbacks [21]: coulomb counting is easy to implement, but it is prone to error due to the integral operation over time, and the accuracy relies heavily on the determination of the reference capacity; OCV lookup can be very accurate, but in

* Corresponding author.

E-mail address: jiaqi.yao@tu-berlin.de (J. Yao).

real-life applications, the OCV can be rarely accessed; on-board electrochemical impedance spectroscopy measurement devices are often not realistic in practice. Model-based approaches estimate battery SOC using battery models with filtering algorithms [11,31]. The battery models, which can be, e.g., equivalent circuit models [32], empirical models [33], or electrochemical models [34], make predictions of the battery behavior, where filtering algorithms are then applied to update the prediction. Commonly implemented filtering algorithms include linear Kalman filter [35], extended Kalman filter [36], unscented Kalman filter [37] and so on. Many recent works have dedicated themselves to model-based state estimation approaches as they are accurate and robust. In Ref. [38], the authors proposed an improved particle swarm optimization-adaptive square root cubature Kalman filter for accurate SOC estimation of lithium-ion batteries. Test results showed that the proposed model could estimate SOC accurately and robustly. In Ref. [39], a square root-untraced Kalman filter algorithm was proposed to track the SOC of lithium-ion batteries in order to deal with the fluctuated usable capacity caused by the temperatures. Attempts to model the battery behavior more accurately have been made in Ref. [40] as well, where a multiple feature-electrochemical thermal coupling modeling approach was proposed. The proposed modeling method enabled a more robust SOC prediction under complex conditions. However, the drawbacks of model-based SOC estimation methods are apparent: the battery model to apply must be carefully selected and well parameterized with specially designed tests in advance, which requires in-depth domain knowledge regarding the battery system at hand [21].

Data-driven solutions are under the spotlight in many research areas, as well as in the field of battery SOC estimation. Data-driven approaches estimate battery SOC using models that learn from historical battery data. Such approaches are often quite intuitive without the need for in-depth domain knowledge. Powered by the rapidly developing landscape of machine learning and deep learning, data-driven SOC estimation approaches offer enormous potential for further optimization and modification for specific applications, which is why they are chosen to be the fundamental of this work. In Ref. [41], the authors proposed a neural network with long short-term memory (LSTM), a refined recurrent neural network (RNN), for SOC estimation of lithium-ion batteries. The proposed LSTM-RNN takes measurement data, including voltage, current, and temperature, and outputs the estimated SOC at each step. The test results showed that the proposed model achieved the mean absolute error (MAE) of 0.807% and 1.252% on the two test cases at the fixed ambient temperature of 10 °C. In Ref. [42], the authors utilized a deep neural network (DNN) for battery SOC estimation, which maps the input features of voltage, temperature, average current, and average voltage measurements directly into SOC estimates. The proposed DNN was able to achieve the MAE of 1.35% and 1.85% on the two test cases at the fixed ambient temperature of 25 °C. In Ref. [43], the authors combined temporal convolutional networks (TCNs) and gated recurrent units (GRUs) with an attention mechanism for offline SOC estimation of lithium-ion batteries. The proposed model was tested on two public datasets, where it achieved the MAE of 0.0429% and 0.1827% on the two test cycles of the first dataset, and the MAE of 0.0782%, 0.2141%, and 0.1581% on the three test cycles of the second dataset, all under the fixed ambient temperature of 25 °C. In Ref. [44], the authors proposed a framework that combines LSTM with clustering algorithms for SOC estimation. The framework takes voltage, current, battery temperature, and C-rate as input, which was able to achieve an MAE of 0.37% and a root mean square error (RMSE) of 0.47% on the test set. In Ref. [45], the authors explored the possibility of SOC estimation using large-scale pre-trained language models, where a hard prompt generator and a soft prompt adapter were developed for the task. On the Panasonic dataset utilized by the authors, the proposed model achieved the MAE of 1.59%, 0.69%, 0.71%, and 1.80%, respectively, for the four test cycles. Recently, we have also proposed a multi-scale data-driven framework for online SOC estimation [21]. The framework utilizes a multi-scale TCN module

for feature extraction and a cross-scale self-attention technique for 1D vector feature fusion. The proposed framework was tested on one popularly used public dataset from the University of Wisconsin-Madison and the novel public lithium-ion battery drive cycle test set we developed. On the public dataset from the University of Wisconsin-Madison at the fixed ambient temperature of 25 °C, the model was able to achieve an MAE of 0.363% and a RMSE of 0.491% on the test cycle of LA92, and an MAE of 0.494% and a RMSE of 0.770% on the test cycle of UDDS.

Traditional SOC estimation methods require individual modeling, training, and fine-tuning for each type of lithium-ion battery cell since they have more or less different electrochemical parameters and properties. However, with the rapid development of battery technology, the speed of iteration of commercial lithium-ion battery cells has significantly accelerated, and novel models are constantly emerging. In fact, energy storage systems could benefit greatly from the mixed usage of different types of battery cells. For example, as a popular fast mechanical charging method for electric vehicles, battery rental and swapping is now already widely adopted in real-life applications, where a variety of lithium-ion batteries of different types and aging states are available. On the one hand, making full use of the battery cells of older iterations is economical and environmentally friendly. On the other hand, the SOC estimation of cells of newer iterations can benefit from the existing knowledge on the older cells as well. To allow for the mixed usage of different types of lithium-ion batteries, possibly with the same standardized size in the battery systems, implementing one model on board for online SOC estimation of different types of batteries is necessary. In Ref. [46], the authors have made an attempt at model-based SOC estimation of LFP cells in AB hybrid battery packs, where LFP cells are used in a mixture with NMC cells. The proposed approach differentiates different regions of the OCV curve of the LFP cells. In the high-slope interval, an unscented Kalman filter is used for SOC estimation, whereas the SOC of the NMC cells is used for the mapping of the SOC of the LFP cells in the flat-slope interval. Test results showed that the proposed method was able to keep the error of the SOC estimation of the LFP cells under 1%. Meanwhile, when it comes to data-driven SOC estimation approaches, the network is required to learn to deal with input from a data distribution that is constantly shifting because of the introduction of the knowledge of new types of batteries, making it a so-called domain-incremental problem. In an ideal case, the estimation model should be able to not only retain old knowledge on previous battery types but also gain good generalization ability on new battery types. However, neural networks have the problems of catastrophic forgetting by design during sequential learning [47]. Continual learning, a branch of machine learning that is increasingly gaining attention, is designed for such problems [48]. Continual learning aims for the ability to incrementally learn different tasks while preserving the previously acquired knowledge on old tasks and using them to facilitate the learning of new knowledge on future tasks [49]. Continual learning approaches can be generally categorized into three groups [50]: replay-based methods [51,52], regularization-based methods [53,54], and parameter isolation-based methods [55, 56]. Replay-based methods utilize stored historical data from previous tasks and replay them when learning new tasks [50], which is rather naive regarding the applicability in real-life usage scenarios because of the storage cost and possible privacy issues. Regularization-based methods introduce specially designed regularization terms into the loss function to enforce the continual learning ability onto the models [50], but the performance is questionable in regression tasks as online SOC estimation is highly safety-critical, as shown later in this paper. Therefore, we utilize parameter isolation-based continual learning methods and propose a dynamic architecture-based continual learning framework for lifelong learning of online SOC estimation across diverse lithium-ion batteries using progressive neural networks (PNNs) [56] with temporal convolutional neural networks [57]. The PNN framework utilizes an incrementally growing architecture to deal with each new task by adding new trainable network columns with

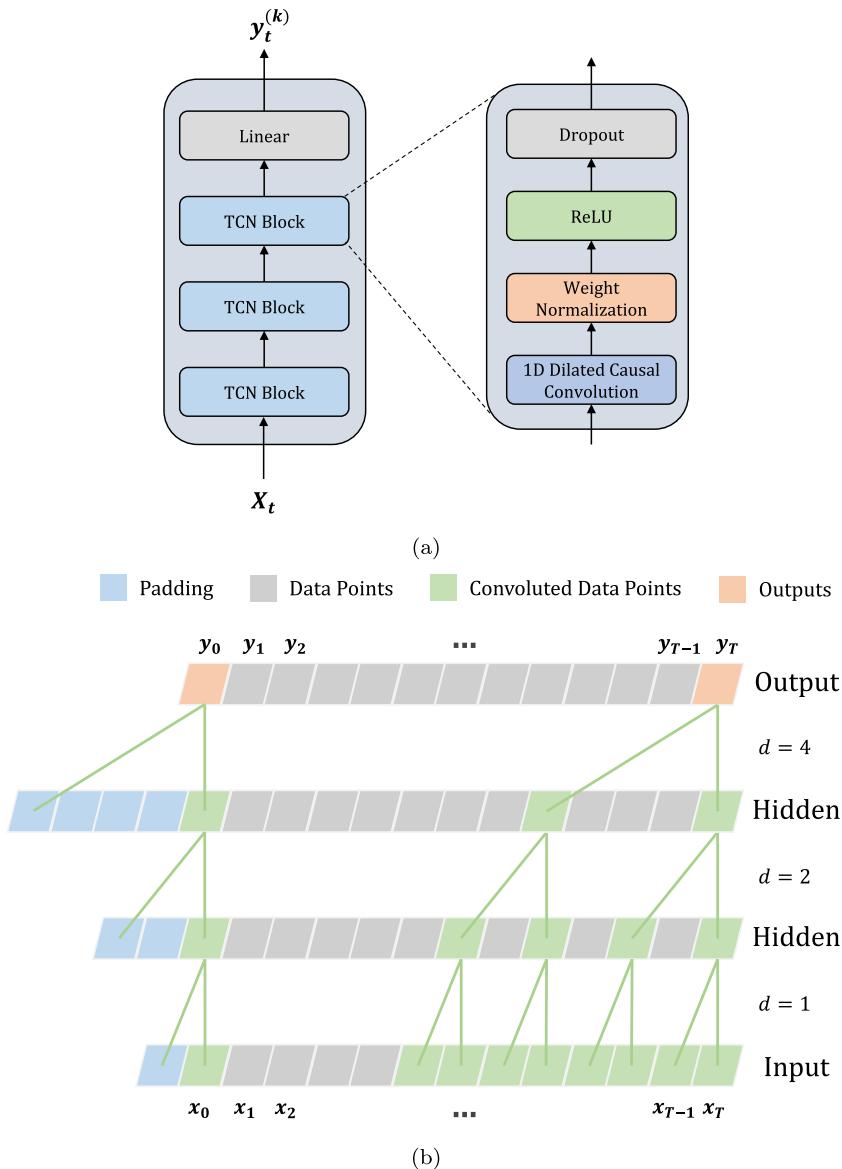


Fig. 1. TCN architecture and work scheme. (a) TCN architecture and block used in our work. (b) Sample TCN of three 1D dilated convolutional layers with a shared kernel size of $k = 2$ and varying dilation factors $d \in \{1, 2, 4\}$.

lateral connections while preserving previously learned knowledge by freezing the weight parameters of the old columns, so that the network not only benefits from the information flow from old to new tasks, but is also immune to catastrophic forgetting by design. We use TCNs as the basic unit for temporal feature extraction for each PNN column, as they have proven to be powerful and efficient in time series processing. Continual learning has been seen sparsely in the field of battery state estimation in some previous works [58–60], but to the best of our knowledge, our work is the first to apply continual learning to SOC estimation of different batteries. The main contribution of this work is as follows:

1. A continual learning framework is proposed for online SOC estimation of diverse lithium-ion batteries.
2. A detailed study is done on the performance and complexity of the proposed framework.
3. In-depth analyses are conducted regarding the influence of task ordering.

The rest of this paper is structured as follows: Section 2 introduces the proposed continual learning framework for online SOC estimation with a detailed explanation of the consisting parts and the workflow. Section 3 describes the experimental setup for the conducted experiments shown in this work, including the utilized datasets and adopted evaluation metrics. Section 4 showcases the results of different conducted experiments, as well as their analysis and interpretation. The key takeaways are concluded in Section 5.

2. Continual learning framework for online SOC estimation

2.1. Temporal convolutional networks

Temporal convolutional networks are an advanced version of traditional convolutional neural networks and were first introduced by Ref. [57]. They are designed to model sequential data in a synchronous sequence-to-sequence way. In our work, TCN is used as the basic SOC estimator for each battery. Fig. 1(a) shows the overall architecture

of the TCN utilized in our work. For each type of battery, or in other words, each task, one TCN consisting of three TCN blocks and subsequent fully connected layers for mapping is instantiated. Each TCN block is further composed of a 1D dilated causal convolutional layer, a weight normalization layer, a rectified linear unit (ReLU) as the activation function, and a dropout layer. 1D dilated causal convolution is the main feature of TCNs. The use of dilation increases the receptive field rapidly with very few parameters, and causality is ensured by an architectural padding trick. A sample TCN with three 1D dilated causal convolutional layers is shown in Fig. 1(b). The dilated causal convolutional layers have a shared kernel size of $k = 2$ but different dilation factors of $d \in \{1, 2, 4\}$. As shown in the figure, a dilation factor of d means that at that layer, every d th input is taken for calculation. Under the given setup, the output at time step T , namely y_T , is mapped from the input data of the recent eight time steps, namely from x_{T-7} to x_T , because of the use of dilation, despite the small kernel size. It is worth mentioning that the feature at each time step is usually a vector and has multiple channels. More generally, the dilated convolution operation C at time step t is defined as [21,57]:

$$C(t) = (\mathbf{X} *_d \mathbf{G})(t) = \sum_{i=0}^{k-1} \mathbf{G}(i) \cdot \mathbf{x}_{t-d \cdot i} \quad (2)$$

where \mathbf{X} is the input sequence, \mathbf{G} is the filter, k is the kernel size of \mathbf{G} , and d is the dilation factor.

To ensure the fact that the mapping of the output at each time step is only dependent on historically and currently available input information, a certain amount of padding is inserted at the beginning of the input layer and each intermediate layer. Given the dilation factor d and the kernel size k , the receptive field RF of one single dilated causal convolutional layer grows to:

$$RF_{layer} = d \cdot (k - 1) + 1 \quad (3)$$

Therefore, the number of padding p of each layer can be calculated as follows:

$$p_{layer} = (k - 1) \cdot d \quad (4)$$

In practice, stacked dilated causal convolutional layers are used, where the overall receptive field also incrementally grows with each layer stacked. Therefore, the overall receptive field can be seen as the accumulated contribution of each individual layer. Denoting the dilation factor of the n th layer as d_n , the increment of the receptive field of this layer is:

$$\tilde{RF}_{layer} = (k - 1) \cdot d_n \quad (5)$$

By summing up the increment of the receptive field of each layer, plus the initial single data point, it yields the overall receptive field:

$$RF = 1 + \sum_{n=1}^N (k - 1) \cdot d_n \quad (6)$$

We found that by using the dilation factor of k^{n-1} for the n th dilated causal convolutional layer, the overall receptive field RF of an N -layer TCN becomes:

$$RF = 1 + \sum_{n=1}^N (k - 1) \cdot k^{n-1} \quad (7)$$

Since the sum is a geometric series:

$$\sum_{n=1}^N k^{n-1} = 1 + k + k^2 + \dots + k^{N-1} = \frac{k^N - 1}{k - 1} \quad (8)$$

The overall receptive field is thus:

$$RF = 1 + (k - 1) \cdot \frac{k^N - 1}{k - 1} = k^N \quad (9)$$

Therefore, the dilation factor k^{n-1} is used for the n th TCN block in our work for elegance. After the feature extraction with 1D dilated

causal convolution in each TCN block, weight normalization is used for accelerated convergence of training by decoupling the length of the weight vectors from the direction [61]. Following the introduction of nonlinearity from ReLU, dropout is added in order to mitigate potential overfitting by randomly eliminating units during training [62]. Specifically in our setting of SOC estimation, at time step t , the TCN for the k th task takes the input sequence \mathbf{X}_t consisting of historical measurements of voltage, current and temperature, and outputs the current SOC estimate of the k th battery $y_t^{(k)}$. The input data point \mathbf{x} at time step t is therefore $\mathbf{x}_t = [U_t, I_t, T_t]$, where U_t , I_t , and T_t are the voltage, current, temperature measurement at time step t respectively.

2.2. Progressive neural networks

Introduced by Ref. [56], progressive neural networks are a continual learning architecture designed to incrementally learn new tasks while retaining old knowledge. In PNNs, one “column” of the network is assigned for each task. The network is initialized with one column. When a new task arises, a new column of the network is instantiated, with lateral connections to all previous columns added. Fig. 2 shows the PNN architecture used in our work. The framework has three columns, corresponding to the three types of lithium-ion batteries studied in the experiments. Each column is one TCN from Section 2.1. Adapters a are instantiated as lateral connections between the previous columns and the new column. Denoting the mapping function of layer i of column k as $f_i^{(k)}$ and the mapping function of the adapter for the lateral connection from column j of layer $i - 1$ to column k of layer i as $g_i^{(k:j)}$, the output of layer i of column k , $\mathbf{h}_i^{(k)}$, can be calculated as:

$$\mathbf{h}_i^{(k)} = f_i^{(k)}(\mathbf{h}_{i-1}^{(k)} + \sum_{j < k} g_i^{(k:j)}(\mathbf{h}_{i-1}^{(j)})) \quad (10)$$

where \mathbf{h}_0 is the input of the network, and the adapter mapping terms only exist when $i \geq 2$. We use pointwise convolution with the activation function ReLU as adapters in our work to adjust the number of channels:

$$g_i^{(k:j)}(\mathbf{h}_{i-1}^{(j)}) = \text{ReLU}(\mathbf{W}_i^{(k:j)} * \mathbf{h}_{i-1}^{(j)} + \mathbf{b}_i^{(k:j)}) \quad (11)$$

where $\mathbf{W}_i^{(k:j)}$ denotes the 1×1 convolutional kernel that adjusts the number of channels, $\mathbf{b}_i^{(k:j)}$ is the corresponding bias term, and $*$ is the convolution operator. The pointwise convolution ensures that the adapter transforms the feature vectors from column j to match the channel dimensions required by column k with nonlinearity introduced by the ReLU activation function.

By leveraging lateral connections to previous columns, the knowledge transfer from earlier learned representations to the new task is enabled. While training the new column with the corresponding lateral connections on the new task, the weight parameters of the previous columns are frozen. Therefore, PNNs are immune to forgetting by design [56].

Fig. 3 showcases the overall workflow of the proposed continual learning framework. With the PNN architecture, it is possible to use one model to learn the online estimation of different batteries’ SOC incrementally without forgetting previously learned knowledge.

3. Experimental setup

3.1. Data collection

Three public lithium-ion battery drive cycle datasets are used to conduct the experiments and analyses [63–65]. The three utilized lithium-ion battery cells are from three different manufacturers, covering two chemistry types and two standardized formats. The ambient temperatures and drive cycles utilized in the datasets also differ, resulting in a more diverse data distribution.

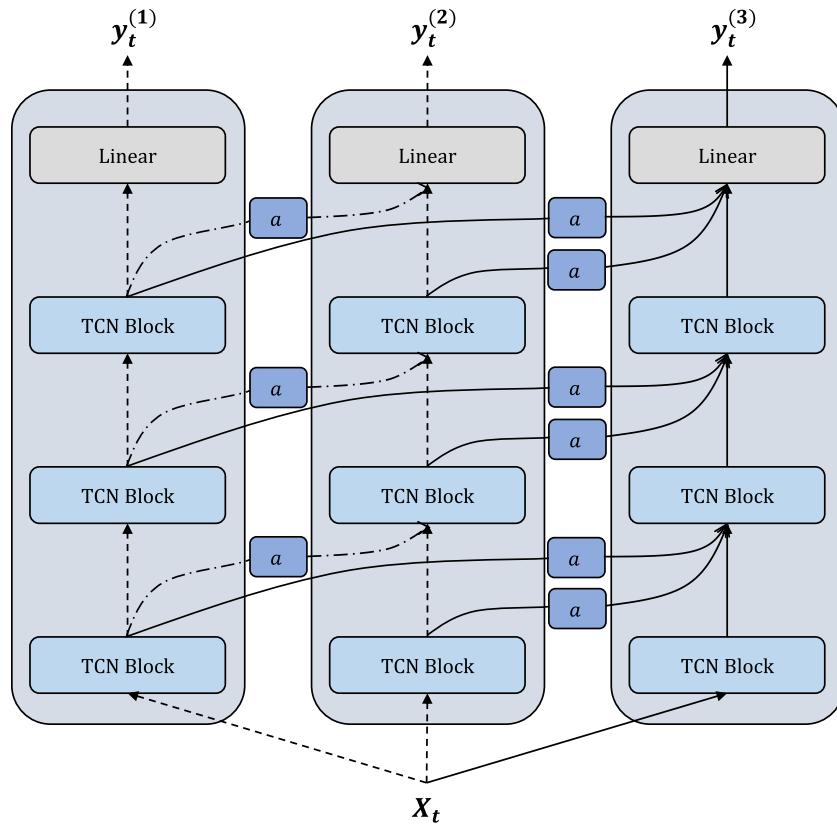


Fig. 2. The PNN architecture used in our work. A third column is augmented for the SOC estimation of the third battery type with lateral connections to the previous columns.

Table 1
Panasonic NCR18650PF cell specifications.

Parameter	Data
Chemistry	NCA
Format	18 650
Nominal capacity	2.90 Ah
Nominal voltage	3.60 V
Nominal energy	9.94 Wh
Max. charge voltage	4.20 V
Min. discharge voltage	2.50 V

3.1.1. Task A: Panasonic 18650PF lithium-ion battery

As one of the most popularly used public datasets by the researchers [21,33,41,42], the battery dataset from the University of Wisconsin-Madison contains dynamic cycling data of a Panasonic 18650PF NCA cell [63], of which the specifications are shown in **Table 1**, at five different ambient temperatures, namely -20°C , -10°C , 0°C , 10°C , and 25°C . At each ambient temperature, nine drive cycle profiles are applied. Four out of nine profiles are standard drive cycles, including Highway Fuel Economy Test (HWFET), California Unified Cycle (LA92), Urban Dynamometer Driving Schedule (UDDS), and US06 Supplemental Federal Test Procedure (US06). Four cycles (Cycle 1–4) are synthesized from the four standard cycles, and one addition cycle (NN) is specifically designed for the training of neural networks. The cycling data at all temperatures are put together as one dataset, so that the generalization ability of the models under varying ambient conditions can be better tested. We use six cycles, namely Cycle 1–4, NN, and US06, at each ambient temperature for training, HWFET at each ambient temperature for validation, and UDDS and LA92 at each temperature for testing. For simplification, the learning of online SOC estimation on this dataset is referred to as task A.

Table 2
Samsung INR21700-30T cell specifications.

Parameter	Data
Chemistry	NMC
Format	21 700
Nominal capacity	3.00 Ah
Nominal voltage	3.60 V
Nominal energy	10.80 Wh
Max. charge voltage	4.20 V
Min. discharge voltage	2.50 V

3.1.2. Task B: Samsung INR21700-30T lithium-ion battery

The second dataset used is from McMaster University, which was published by the same author [64]. This dataset contains dynamic cycling data of a Samsung INR21700-30T NMC cell, whose specifications are shown in **Table 2**, at six different ambient temperatures, namely -20°C , -10°C , 0°C , 10°C , 25°C , and 40°C . Twelve cycle profiles are applied at each ambient temperature, of which eight cycles (Mixed 1–8) are synthesized from the rest four standard cycles (HWFET, LA92, UDDS, US06). Same as for task A, the cycling data at all temperatures are put together as one dataset. The eight mixed cycles are used as the training set, and HWFET and US06 are used as the validation set. UDDS and LA92 are used for testing. For simplification, the learning on this dataset is referred to as task B.

3.1.3. Task C: LG INR 21700 M50LT lithium-ion battery

The third dataset is a novel lithium-ion battery drive cycle dataset previously published by us [65]. This dataset contains dynamic cycling data of LG INR 21700 M50LT cells, whose specifications are shown in **Table 3**. The cycling data are from five different ambient temperatures, namely 5°C , 15°C , 25°C , 35°C , and 45°C . At each ambient temperature, twelve distinct drive cycle profiles are applied, including Braunschweig City Driving Cycle (BCDC), City Suburban Heavy Vehicle Cycle (CSHVC), Federal Test Procedure-72 (FTP-72), Federal Test

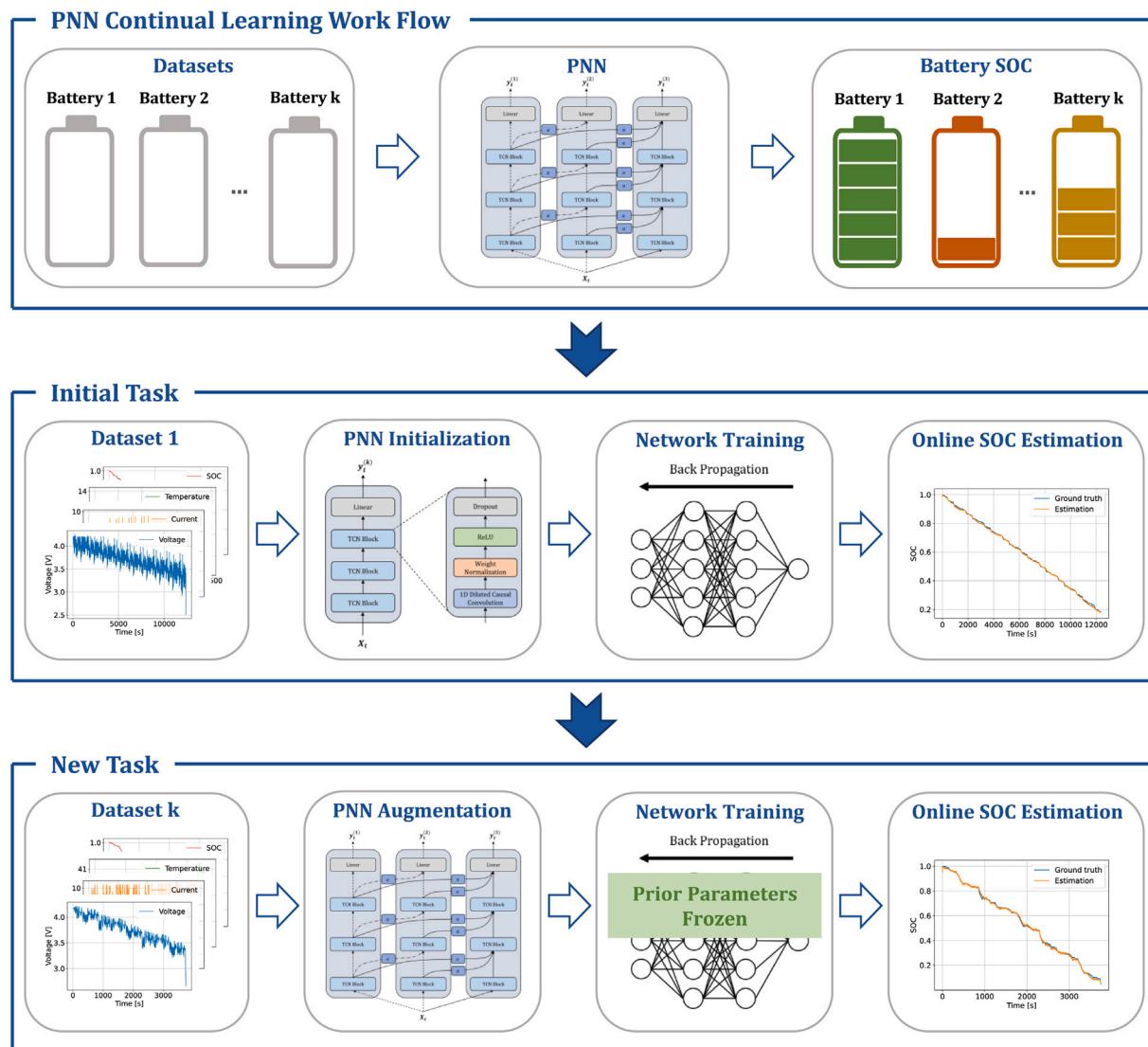


Fig. 3. Workflow of the proposed continual learning framework for online SOC estimation of different batteries.

Table 3
LG INR 21700 M50LT cell specifications.

Parameter	Data
Chemistry	NMC
Format	21700
Nominal capacity	4.93 Ah
Nominal voltage	3.69 V
Nominal energy	18.2 Wh
Max. charge voltage	4.20 V
Min. discharge voltage	2.50 V

Procedure-75 (FTP-75), Heavy Heavy-Duty Diesel Truck Composite Cycle (HHDDT), HWFET, Inspection and Maintenance (IM), LA92, Orange County Transit Bus Cycle (OCTBC), Port Drayage Metro Highway Cycle California (PDMHC), Parcel Delivery Truck Cycle Baltimore (PDTCB), and US06. The distinct drive cycles cover diverse cycling patterns. As with the previous tasks, cycling data at all temperatures are put together as one dataset. We use BCDC, LA92, CSHVC, HWFET, IM, US06, PDTCB, OCTBC for training, HHDDT and FTP-72 for validation, and FTP-75 and PDMHC for testing. The learning on this dataset is referred to as task C.

3.2. Experiment settings

As preprocessing, the data profiles are resampled into strict 10 Hz using linear interpolation to eliminate the uneven time intervals in between caused by the imperfect sampling of the measurement devices. Min–max normalization is performed on the input data to improve the training. In the experiments, the kernel size of all TCNs is 32, and the dilation factor is chosen to be 32^{n-1} for the n th convolutional layer, which results in a receptive field of 3276.8 s for each SOC estimate. The PNNs are always initialized using a TCN with 16, 32, and 64 channels, respectively, for each convolutional layer. The augmented TCN columns of the PNNs have 16, 32, and 64 channels, respectively for each convolutional layer as well by default, which can be different in some experiments shown later. The subsequent two fully connected layers always start with the same number of features as in the final convolutional layer at each time step, and end with one feature, which is the SOC estimate. The automatic hyperparameter optimization framework Optuna [66] is utilized to tune the learning rate. With the pruning function offered in Optuna and early stopping, several models with the most potential are saved based on the fine-tuning on the validation set. We present the results of the best-performing one on the test set for this work to most possibly approximate the optimum.

3.3. Evaluation metrics

The evaluation metrics used in this work originate from two aspects. First, the three common methods for error quantification in regression tasks are used, namely MAE, RMSE, and the coefficient of determination R^2 , which are defined as:

$$MAE(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (12)$$

$$RMSE(\mathbf{y}, \hat{\mathbf{y}}) = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (13)$$

$$R^2(\mathbf{y}, \hat{\mathbf{y}}) = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (14)$$

where \mathbf{y} and $\hat{\mathbf{y}}$ are the sequence of the ground-truth values and the sequence of the estimates with n time steps in total, respectively, and \bar{y} is the mean value of the ground-truth sequence. MAE gives a more intuitive measurement of error, while RMSE is more sensitive to outliers. In practice, as there are multiple sequences in the test set of each task, we take the averaged MAE and RMSE over each sequence as the errors of the respective task. In addition, we offer the evaluation metrics that are often utilized in the field of continual learning as well, namely average accuracy (ACC), backward transfer (BWT), and forward transfer (FWT) [51,67]. Denoting the accuracy the model achieves on the test set of the i th task after the incremental learning from the first task until the k th task as $a_{k,i}$, the evaluation metrics are defined as [51,67]:

$$ACC_k = \frac{1}{k} \sum_{i=1}^k a_{k,i} \quad (15)$$

$$BWT_k = \frac{1}{k-1} \sum_{i=1}^{k-1} (a_{k,i} - a_{i,i}) \quad (16)$$

$$FWT_k = a_{k,k} - \tilde{a}_i \quad (17)$$

where $k \in \{1, 2, \dots, T\}$ in the case of T tasks in total, and \tilde{a}_i is the accuracy of the reference single-task model on the respective task. We define the accuracy $a_{k,i}$ of our regression task simply as:

$$a_{k,i} = 1 - MAE_{k,i} \quad (18)$$

In this way, the overall performance is reflected by ACC. The memory stability, namely the influence of the learning of a new task on prior tasks, is reflected by BWT. FWT is used to evaluate the learning plasticity of the proposed framework, namely the influence of the learning of prior tasks on the new tasks. In this work, we have modified the definition of FWT slightly from the original one [51] for a better evaluation of how prior knowledge better the model's performance on the current task instead of future unlearned ones.

4. Results and discussion

In this section, in-depth experiments and analyses are carried out regarding the proposed framework's performance and complexity, as well as the influence of task ordering.

4.1. Comparison of the models' performance

In the three categories of continual learning [50], replay-based methods are considered naive and impractical for our task of online SOC estimation. Therefore, we compare the performance of our proposed framework with one of the most popular regularization-based methods, namely elastic weight consolidation (EWC) [53], and another intuitive parameter isolation-based method, multi-head networks. EWC aims to mitigate catastrophic forgetting by penalizing the update of the important parameters for prior tasks using regularization terms based

on Fisher information. In the experiment, the EWC model is implemented with a very large importance factor for the old tasks. As for multi-head networks, new network heads are instantiated for new tasks, but the feature extraction of all tasks is done using a shared backbone, which is frozen during the training after the first task. In this work, backbones are the three TCN blocks, and heads refer to the subsequent fully connected layers. In fact, multi-head networks can be considered as a simplified version of PNNs. In addition, we showcase the performance of vanilla incremental learning models, or in other words, transfer learning models, where the model is incrementally trained on the sequence of tasks without specific continual learning designs. The performance of single-task networks, which are used for each task individually, is also given as the baseline for reference. The comparison of the proposed PNN framework with the multi-head model, the vanilla incremental learning model, and the reference single-task model is considered the ablation study of this work. In all models of comparison, the same basic network architecture is used.

As a direct demonstration of continual learning's importance in our task, Fig. 4 shows the estimation results and absolute error of the vanilla incremental learning model on the test cycle UDDS at 10 °C of task A after learning task A, task A and B, task A and B and C, respectively. Not surprisingly, directly after being trained on task A, the model's performance is at its best of an MAE of 0.701%. The estimation follows the ground truth closely, and the absolute error maintains a low level throughout the whole time. After the learning of task B, the model's performance back on task A has significantly deteriorated with a flatter estimation curve, resulting in an overall MAE of 6.944% with a large error in the beginning and towards the end of the sequence. After the learning of task C, the estimation back on the test cycle of task A further loses its conformity with the overall trend, resulting in a significant MAE of 25.364%. Without the implementation of specific continual learning techniques, a vanilla incremental learning model does not have the ability to remember previous knowledge on prior tasks.

Fig. 5 shows the estimation results of the different models on the test cycle LA92 of task B after learning tasks A, B, and C at different ambient temperatures. Unlike the vanilla incremental learning model, models with parameter isolation-based continual learning techniques generally can maintain their knowledge of old tasks to some level. The single-task reference model achieves a very low error in general but its estimation still shows some discrepancies compared with the ground truth, possibly because of the complicated test scenario. However, not only immune to forgetting, the estimation of the proposed PNN is able to converge to the ground truth even better than the reference model at all temperatures because of the forward knowledge transfer from its previously learned task A.

Fig. 6 shows the box plots of the MAE of the models' estimation on each test cycle of the respective task after learning tasks A, B, and C. As shown in Fig. 6(a), the EWC and incremental models show very high errors when tested back on task A due to the forgetting of the old knowledge, whereas EWC shows less scattered MAEs, possibly because of the imposed constraint. The PNN, multi-head, and reference models show similarly low MAEs, as PNN and the multi-head model are immune to forgetting. However, on the test set of task B, as shown in Fig. 6(b), the estimation error of the multi-head model gets higher as the architecture limits the complete feature extraction of the new task. The PNN model generally achieves smaller MAEs than the reference model, which is due to the positive influence of the FWT from task A to task B. When it comes to task C, as shown in Fig. 6(c), the multi-head model performs the worst because of the architectural limitation. The EWC model also has generally higher MAEs on the test cycles because of the constrained learning of new tasks. The incremental learning model performs slightly better than the reference model in this case due to the positive influence introduced by transfer learning. The PNN model outperforms the rest four models by exploiting previously learned knowledge using the lateral connections.

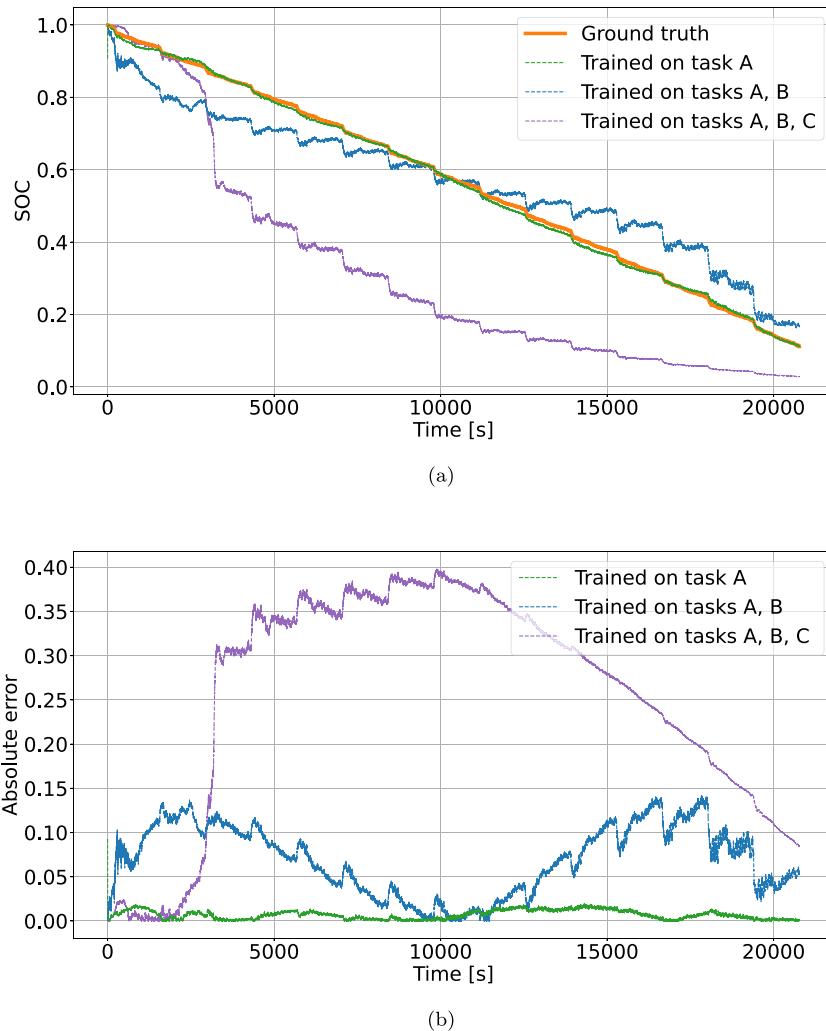


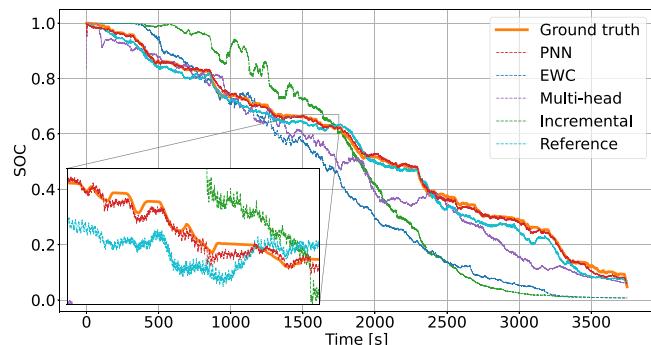
Fig. 4. Test results of the vanilla incremental learning model on the test cycle UDDS at 10°C of task A after learning task A, task A and B, task A and B and C, respectively. (a) Estimation results. (b) Absolute error.

Table 4
Performance comparison of the different models on the test sets under the learning order of tasks A, B, C.

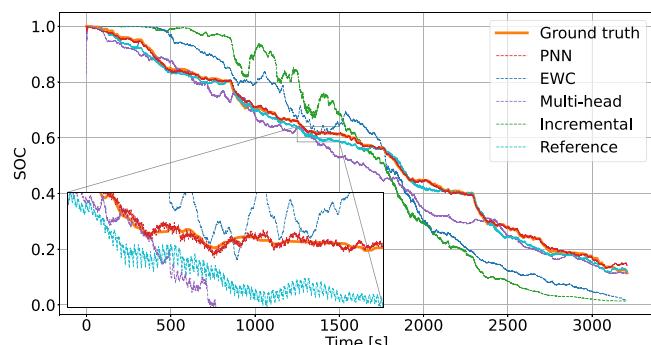
Model	Trained tasks	Test accuracy on task [%]			Evaluation metrics		
		A	B	C	ACC [%]	BWT [%]	FWT [%]
PNN	A	98.696	–	–	98.696	–	–
	AB	98.696	98.844	–	98.770	0	0.520
	ABC	98.696	98.844	99.462	99.001	0	0.556
EWC	A	98.721	–	–	98.721	–	–
	AB	93.464	98.146	–	95.805	-5.257	-0.178
	ABC	69.596	86.651	97.893	84.713	-20.310	-1.013
Multi-Head	A	98.784	–	–	98.784	–	–
	AB	98.784	95.387	–	97.086	0	-2.937
	ABC	98.784	95.387	95.524	96.565	0	-3.382
Incremental	A	98.786	–	–	98.786	–	–
	AB	91.878	98.272	–	95.075	-6.908	-0.052
	ABC	70.432	84.566	99.110	84.703	-21.030	0.204
Reference	A/B/C	98.701	98.324	98.906	98.644	–	–

Table 4 summarizes the performance of the different models on the test sets under the learning order of tasks A, B, and C. Because of the randomness intentionally brought into the training process for more generalized results, the models start with a slightly different accuracy on task A. Because of the special network design, PNN and the multi-head model are immune to forgetting and thus do not have BWT. However, for other networks, the learning of task B introduces

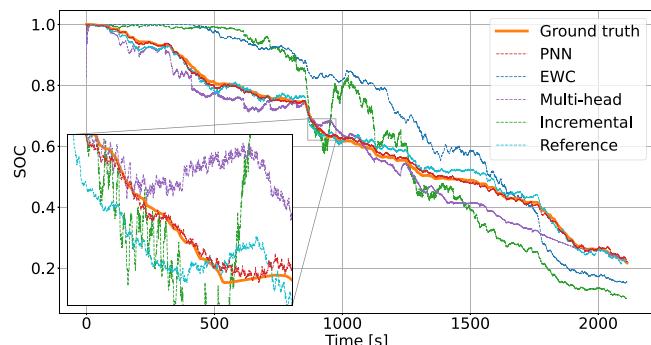
a negative BWT with a relatively small absolute value because of the similar cell specifications and data distribution. BWT becomes more negative after the training of task C, which fits our expectation as the cycling data is more distinct, and the cell used in task C has a significantly larger capacity. The use of EWC mitigates the forgetting of old tasks slightly compared with the vanilla incremental learning model. On the other hand, using EWC causes a lower FWT of



(a)



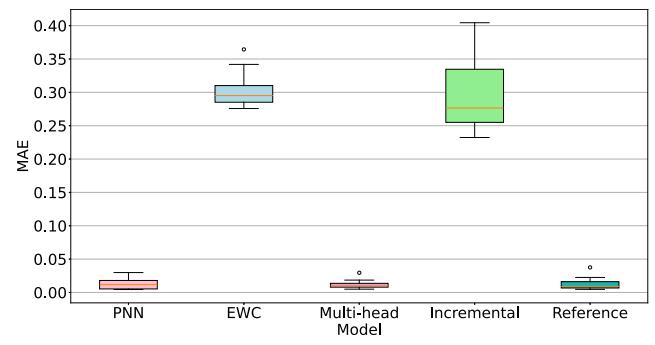
(b)



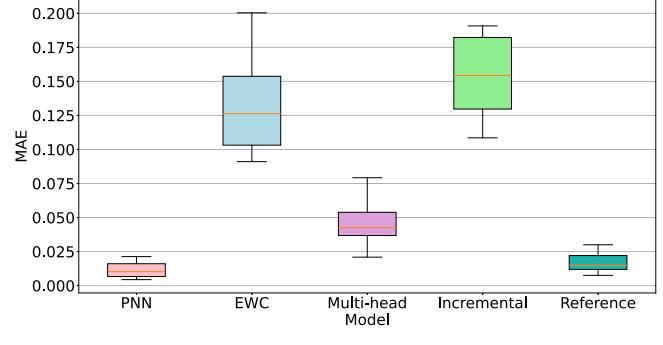
(c)

Fig. 5. Estimation results of the models on the test cycle LA92 of task B at different ambient temperatures after learning tasks A, B, and C. (a) 40 °C. (b) 10 °C. (c) -20 °C.

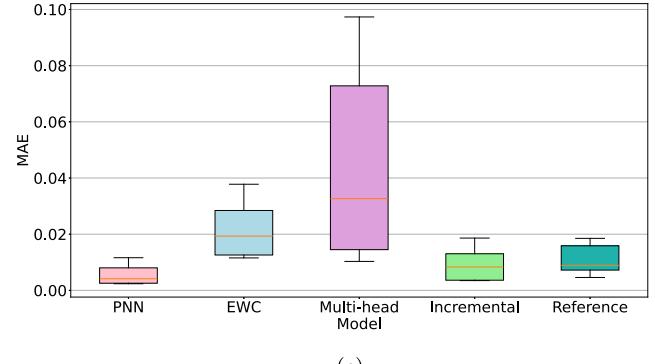
-0.178% and -1.013% compared with vanilla incremental learning, as the update of parameters is restrained, so as for multi-head models since the backbone is frozen for the training of new tasks. Vanilla incremental learning, or transfer learning, does not show significant forward knowledge transfer for task B but improves the performance of task C with a FWT of 0.204%, which indicates that learning of task A has no significant influence on task B, but the learning of task B helps task C. PNN has the highest FWT on both task B (0.520%) and task C (0.556%), which showcases the great positive effects introduced by the lateral connections by allowing the information flow from old columns to the new column. ACC is majorly influenced by BWT since forgetting dominates the overall performance in this case. As shown in Fig. 7(b), EWC achieves a better overall performance than vanilla incremental learning, followed by multi-head and PNN. Because of the immunity to forgetting and the positive knowledge transfer from old



(a)



(b)



(c)

Fig. 6. Box plots of the MAE of the models' estimation on each test cycle of the respective task after learning tasks A, B, and C. (a) On the test set of task A. (b) On the test set of task B. (c) On the test set of task C.

to new tasks, PNN achieves a better overall performance even than reference single-task models.

Fig. 8 showcases the training losses of reference single-task models, PNN, and the vanilla incremental learning model under the incremental learning order of tasks A, B, and C, on task B and task C, respectively. In both cases, reference single-task models are the slowest to converge during training, with the largest residual error towards the end. Both PNN and transfer learning help with an accelerated learning of new tasks.

4.2. Evaluation of model complexity

Although the proposed PNN framework performs greatly on the task of online SOC estimation across different lithium-ion batteries, we would like to know if it is cost-effective to use it, especially compared with simply having multiple single-task models on board. Therefore,

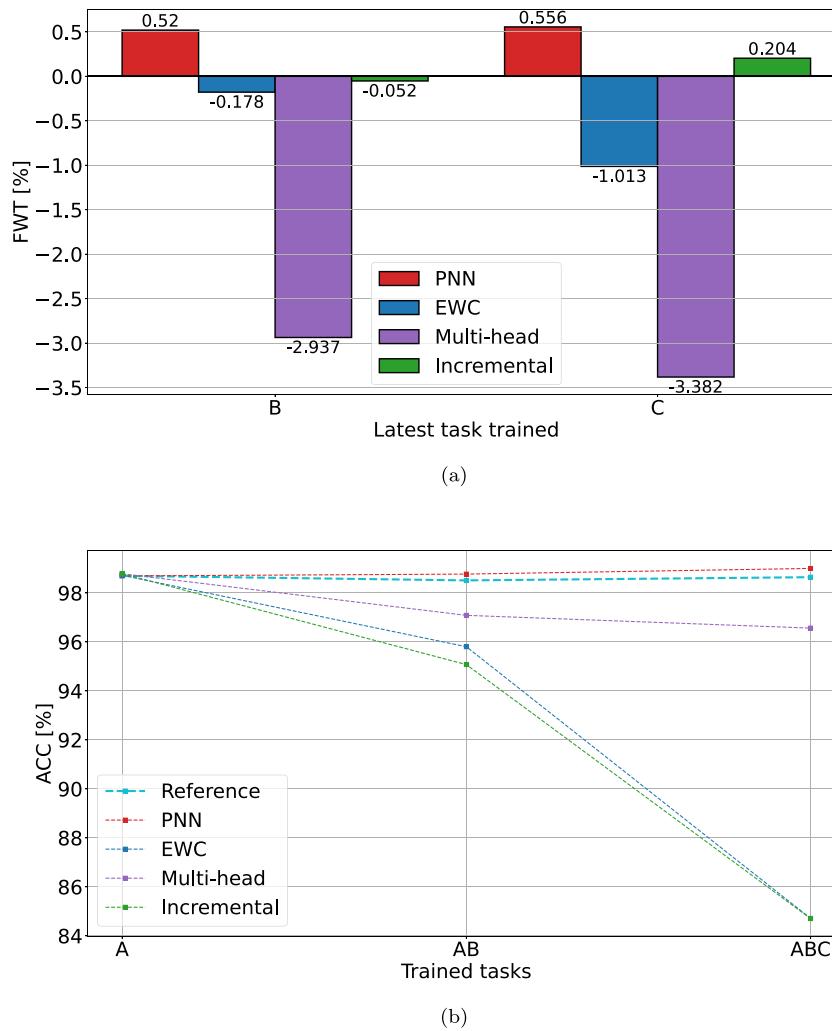


Fig. 7. ACC and FWT of the models under the training order of tasks A, B, C. (a) FWT of the models on the latest task trained after the incremental training until task B and task C. (b) Change of ACC throughout the incremental training process.

we study the performance of PNNs augmented with TCN columns of different numbers of channels in Table 5. The studied PNNs all initialize with the same number of channels, namely 16, 32, and 64, for task A, but are augmented with TCN columns of different numbers of channels for tasks B and C. At the same time, we train reference single-task models with the respective number of channels on task B and task C for comparison. Since the number of floating point operations (FLOPs) is not easy to get for our PNN with dynamic architectures, we use the number of parameters as a proxy for the models' complexity. Although the number of parameters itself only indicates the space complexity, we use it to approximate the time complexity as well since usually larger models require more computations. We show the increment of parameters in each case, namely how many more parameters are needed for carrying out tasks B and C. For PNNs, the parameter increment is from the augmented two columns with the corresponding lateral connections, while for the reference, the parameter increment is from the two single-task models for task B and task C, respectively.

We generally observe that the PNNs augmented with more parameters perform better. The same goes for the single-task models. In most cases, PNNs perform significantly better than reference models with the same number of channels. This improvement does not originate merely from the extra number of parameters introduced by the lateral connections but from the knowledge transfer enabled by them. As shown in Fig. 9, PNN with two extra TCN columns of 2, 4, and 8

channels, namely 4.552×10^3 extra parameters, has an average test MAE on task B and task C of 1.2205%, which is even better than the average MAE of 1.385% of two single-task TCN models with 16, 32, and 64 channels with 1.71586×10^5 parameters, directly indicating the superiority of the proposed framework. However, it is observed that in the case of augmentation with an extremely small TCN column, namely 1, 1, and 2 channels, the performance of PNN deteriorates greatly, which could be because the previous column has used significantly more parameters to learn task A, which is very different than the new task C, so the first column dominates the feature extraction.

As a typical example, Fig. 10 showcases the regression performance of PNNs and reference models with different number of channels on test cycle LA92 at 40 °C of task B. With the same TCN architecture, PNNs perform better than the single-task models in this case. PNN augmented with the TCN column of 2, 4, and 8 channels achieves a R^2 of 0.996, which is lower than the R^2 of 0.991 of the reference model with 16, 32, and 64 channels. In the case of extremely few extra parameters, PNN augmented with the TCN column of 1, 1, and 2 channels is still able to maintain a relatively accurate estimation with a R^2 of 0.980 due to the forward knowledge transfer from the learning of the prior task, while the single-task model with the same number of channels demonstrates unstable regression results with a R^2 of 0.782. Therefore, in fact, the adoption of PNN significantly reduces model complexity while maintaining the same overall performance.

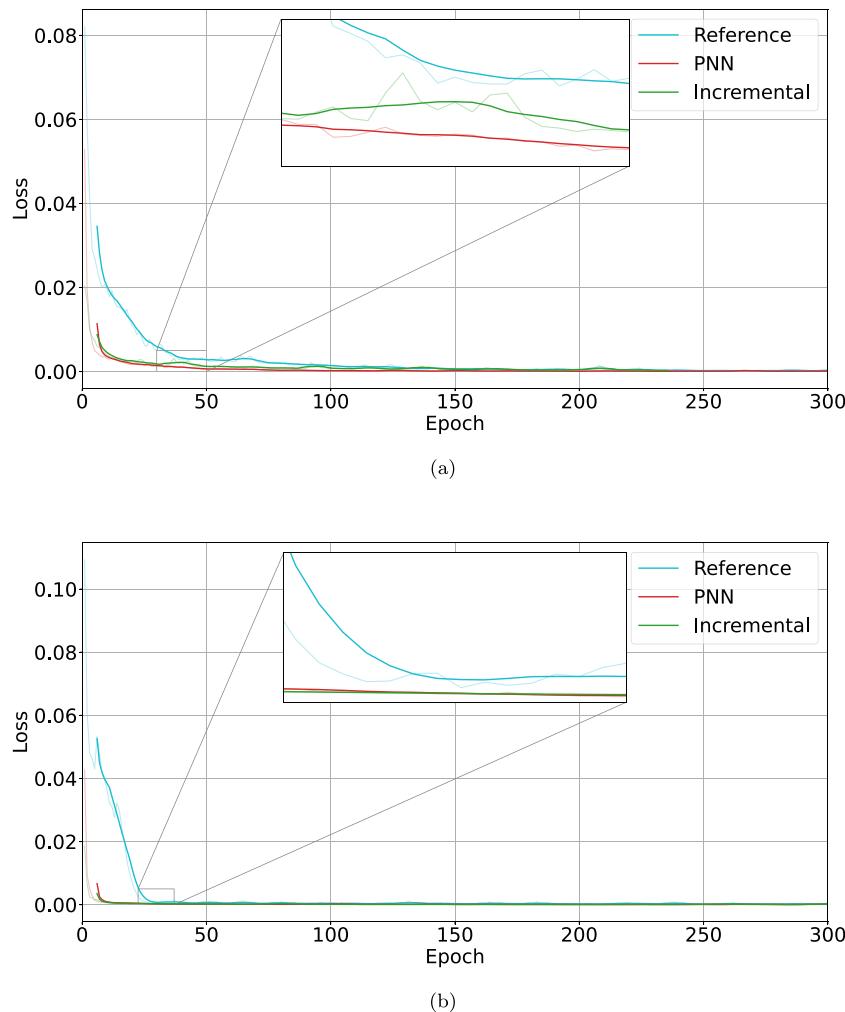


Fig. 8. Training losses of reference single-task models, PNN, and the vanilla incremental learning model under the incremental learning order of tasks A, B, C. (a) Training losses on task B. (b) Training losses on task C.

Table 5
Comparison of performance and complexity of PNNs and reference single-task models with different numbers of channels for new tasks.

Channel Num.	Model	Param. Incr.	Test case			
			Task B		Task C	
			MAE [%]	RMSE [%]	MAE [%]	RMSE [%]
16, 32, 64	PNN	188.050K	1.156	1.583	0.538	0.668
	Reference	171.586K	1.676	2.076	1.094	1.328
8, 16, 32	PNN	50.698K	1.357	1.792	0.822	1.030
	Reference	43.810K	1.732	2.161	1.229	1.512
4, 8, 16	PNN	14.518K	1.366	1.853	0.940	1.190
	Reference	11.410K	1.883	2.354	1.258	1.610
2, 4, 8	PNN	4.552K	1.573	2.005	0.868	1.027
	Reference	3.082K	2.214	2.722	1.263	1.562
2, 2, 6	PNN	2.532K	1.684	2.145	1.259	1.547
	Reference	1.498K	2.904	3.485	1.438	1.788
1, 2, 4	PNN	1.600K	2.044	2.568	1.377	1.749
	Reference	0.886K	3.407	4.122	1.627	2.042
1, 1, 2	PNN	0.780K	2.092	2.833	2.475	2.922
	Reference	0.410K	10.806	12.768	1.874	2.238

4.3. Analysis on the influence of task ordering

Fig. 11 shows the test results of the vanilla incremental learning model on the test cycle PDMHC at 15 °C of task C after learning different tasks. Not surprisingly, the model has the best performance directly

after being trained on task C with an MAE of 0.954%. The two models, one trained on tasks C and B, and the other one trained on tasks C, A, and B, share similar overall performance, with the MAE of 6.181% and 5.676%, respectively. The same goes for the rest two models, namely the model trained on tasks C and A, and the model trained

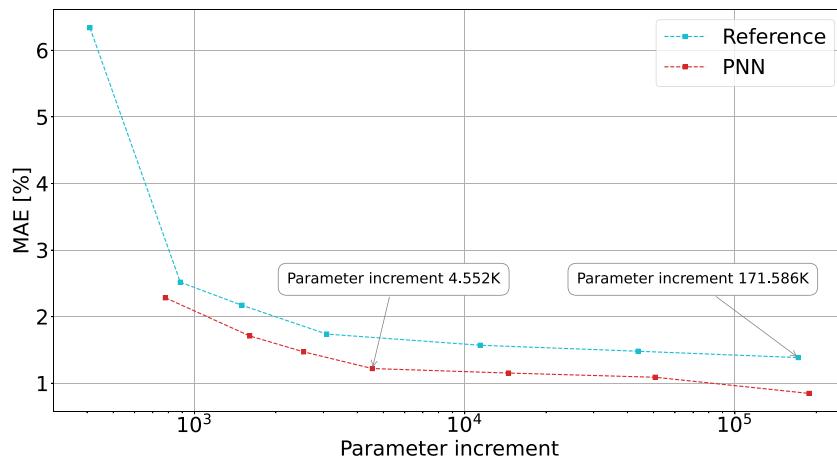


Fig. 9. The average test MAE of tasks B and C of the reference single-task models and PNNs with different numbers of extra parameters introduced for the two new tasks.

Table 6
Performance comparison of PNNs trained under different orders.

Order	Trained tasks	Test accuracy on task [%]			Evaluation metrics		
		No. 1	No. 2	No. 3	ACC [%]	BWT [%]	FWT [%]
ABC	A	98.696	–	–	98.696	–	–
	AB	98.696	98.844	–	98.770	0	0.520
	ABC	98.696	98.844	99.462	99.001	0	0.556
ACB	A	98.696	–	–	98.696	–	–
	AC	98.696	99.290	–	98.993	0	0.384
	ACB	98.696	99.290	99.011	98.999	0	0.687
BAC	B	98.333	–	–	98.333	–	–
	BA	98.333	99.050	–	98.692	0	0.349
	BAC	98.333	99.050	99.265	98.883	0	0.359
BCA	B	98.333	–	–	98.333	–	–
	BC	98.333	99.387	–	98.860	0	0.481
	BCA	98.333	99.387	98.988	98.903	0	0.287
CAB	C	98.991	–	–	98.991	–	–
	CA	98.991	98.805	–	98.898	0	0.104
	CAB	98.991	98.805	98.901	98.899	0	0.577
CBA	C	98.991	–	–	98.991	–	–
	CB	98.991	99.034	–	99.013	0	0.710
	CBA	98.991	99.034	98.738	98.921	0	0.037

on tasks C, B, and A, whose estimation has the MAE of 20.468% and 18.862%, respectively. Observing the results, it can be concluded that learning different tasks has completely different influences on some other tasks. In this case, the training order with task A in the end causes significantly larger forgetting on previously learned task C compared with the training order with task B in the end. Therefore, we train the PNN model and the vanilla incremental learning model under all six possible training orders in order to study the interaction between tasks and the influence of task ordering.

Tables 6 and 7 show the performance of PNN and the vanilla incremental learning model under different training orders. By observing the scenarios where two tasks are trained in each training order of Table 7, some conclusions can be drawn regarding the interaction between every two tasks:

- In the first training order (ABC), the model's accuracy on the test set of task A after being trained on tasks A and B drops 6.908% compared with after being trained on task A only, which is marginal compared with the average BWT. Meanwhile, learning task A anteriorly does not show significant influences on the model's accuracy on task B with merely an FWT of -0.052% . On the other hand, in the third training order (BAC), the model's accuracy on task B after being trained on tasks B and A has an average decrease of -15.761% , while its accuracy on task

A has an FWT of 0.239% . Combining the observations on the two cases, it can be concluded that learning task B has positive influences on task A, while learning task A does not show significant influences on task B. The reason behind this could be that while the lithium-ion battery cells have similar nominal capacities, the dynamic cycling dataset used for task B covers more ambient conditions, making the data more diverse and, thus, the task more complicated so that the model learns more generic patterns from task B that can be applied for task A. On the other hand, the model is not able to learn such generic knowledge from the relatively simple task A.

- In the second training order (ACB), the model's accuracy on task A has a major decrease of 28.158% after learning task C, and learning task A anteriorly has an average negative influence of -0.192% on task C. Similar phenomena can be observed in the fifth training order (CAB). It can be concluded that learning task C has a great negative impact on task A, and learning task A affects task C negatively as well. This interaction between tasks A and C is caused by the great difference in the data distribution. Not only do the lithium-ion battery cells utilized for the two tasks have a completely different format, capacity, and chemistry, but the drive cycles applied also have very few overlaps.

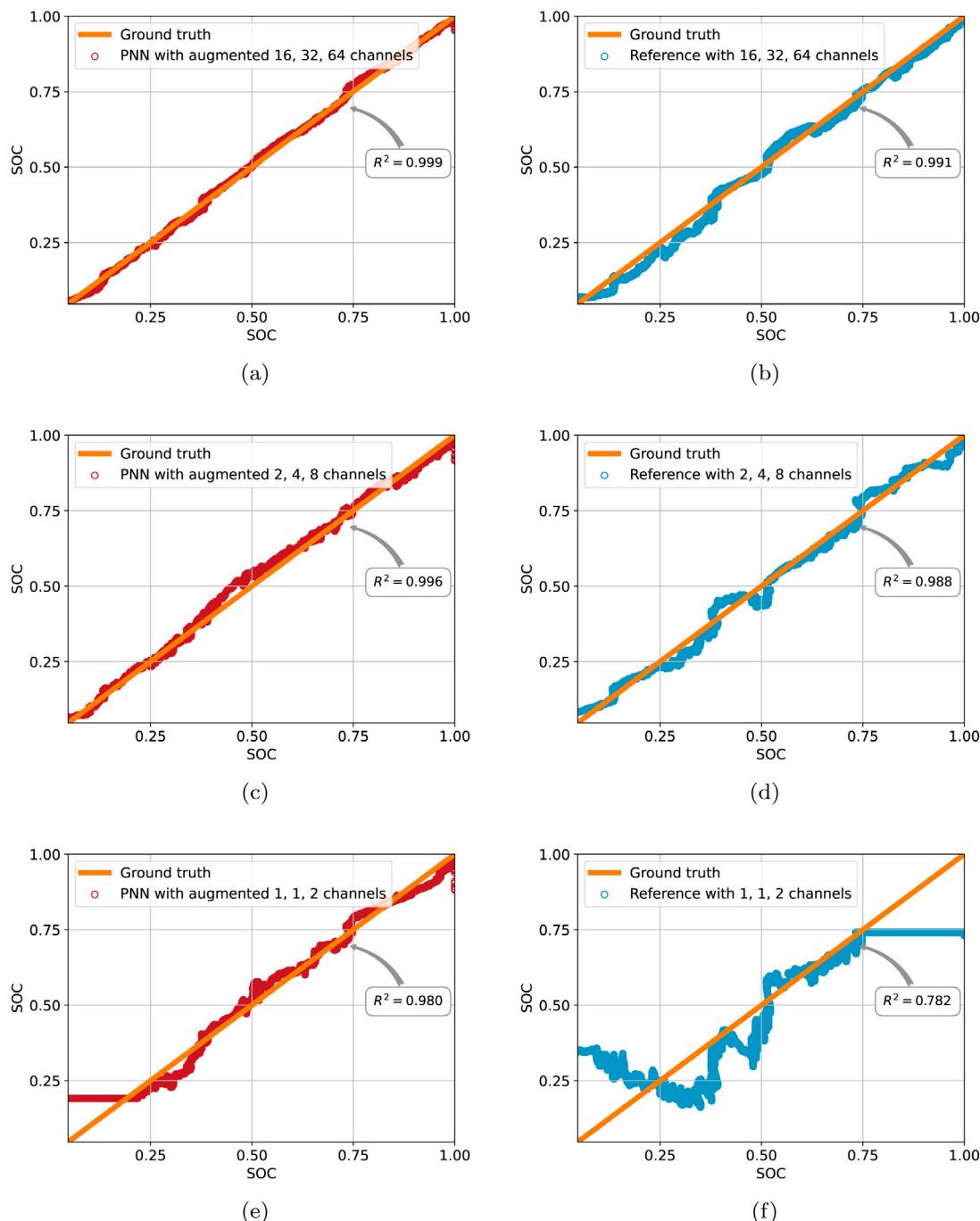


Fig. 10. Regression performance of PNNs and reference models with different number of channels on test cycle LA92 at 40 °C of task B. (a) PNN augmented with the TCN column of 16, 32, 64 channels. (b) Reference single-task TCN model of 16, 32, 64 channels. (c) PNN augmented with 2, 4, 8 channels. (d) Reference of 2, 4, 8 channels. (e) PNN augmented with 1, 1, 2 channels. (f) Reference of 1, 1, 2 channels.

3. Similarly, observing the results from the training orders of BCA and CBA, it can be concluded that the learning of task B has a significant positive influence on task C since in the fourth training order (BCA), learning task B anteriorly increases the model's accuracy on task C significantly with 0.668%, and in the sixth training order (CBA), the model's accuracy on task C only drops slightly with 5.795% after learning task B. In addition, the learning of task C shows a relatively small positive influence on task B. The reason behind this could be that task B shares similar features with task C, such as both cell types have the format of 21700 and the chemistry of NMC. In fact, the features of task

B seem to stand in the middle of task A and task C, making the transfer learning from task B easier for both tasks.

PNNs outperform the vanilla incremental learning models in most cases. Interestingly, in the third and fourth training orders, where task B is learned first, the performance of the transfer learning models is better than PNNs on task C and close to PNNs on task A. The reason behind this could be that the parameters of the transfer learning model after being trained on task B lie in the middle of the optimums of task A and task C. Therefore, it can easily reach the new optimum when trained on a new task. In fact, it also indicates the difference in the mechanisms of forward knowledge transfer between PNN and transfer learning.

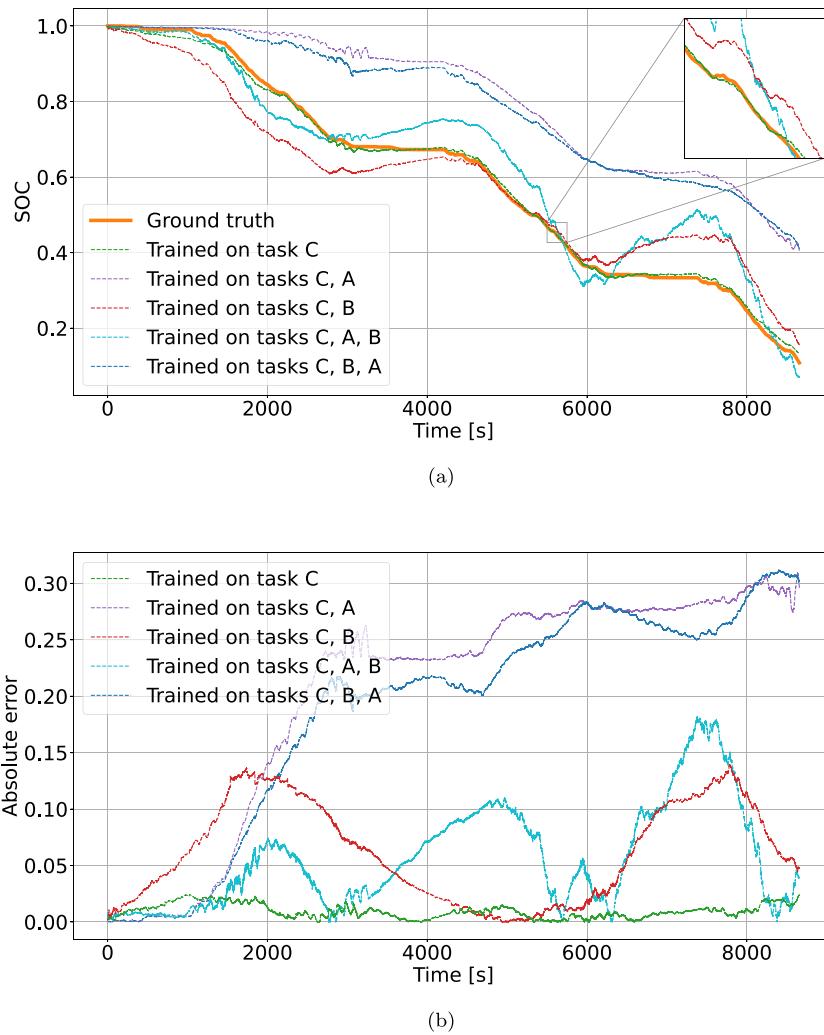


Fig. 11. Test results of the vanilla incremental learning model on the test cycle PDMHC at 15°C of task C after learning different tasks. (a) Estimation results. (b) Absolute error.

Table 7
Performance comparison of vanilla incremental learning models trained under different orders.

Order	Trained tasks	Test accuracy on task [%]			Evaluation metrics		
		No. 1	No. 2	No. 3	ACC [%]	BWT [%]	FWT [%]
ABC	A	98.786	–	–	98.786	–	–
	AB	91.878	98.272	–	95.075	-6.908	-0.052
	ABC	70.432	84.566	99.110	84.703	-21.030	0.204
ACB	A	98.786	–	–	98.786	–	–
	AC	70.628	98.714	–	84.671	-28.158	-0.192
	ACB	89.784	83.981	98.182	90.649	-11.868	-0.142
BAC	B	98.235	–	–	98.235	–	–
	BA	82.474	98.940	–	90.707	-15.761	0.239
	BAC	89.766	80.328	99.613	89.902	-13.541	0.707
BCA	B	98.235	–	–	98.235	–	–
	BC	87.192	99.574	–	93.383	-11.043	0.668
	BCA	81.119	81.282	98.909	87.103	-17.704	0.208
CAB	C	98.995	–	–	98.995	–	–
	CA	82.691	97.619	–	90.155	-16.304	-1.082
	CAB	92.389	82.469	98.238	91.032	-10.878	-0.086
CBA	C	98.995	–	–	98.995	–	–
	CB	93.200	98.417	–	95.809	-5.795	0.093
	CBA	84.356	80.382	98.920	87.886	-16.337	0.219

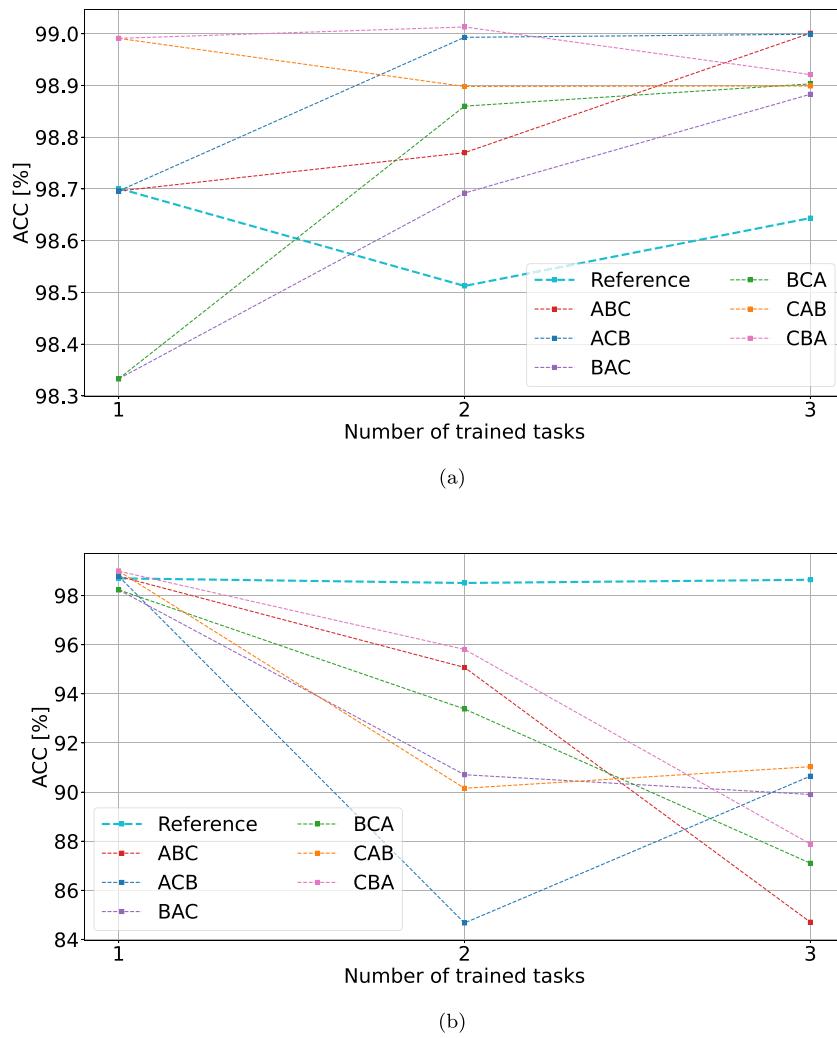


Fig. 12. Change of ACC of PNNs and vanilla incremental learning models under different learning orders. (a) PNNs. (b) Vanilla incremental learning models.

In transfer learning, the parameters trained on the original task are directly updated during the training of the new task. Therefore, the knowledge transfer depends on the distance between the optima of the old and new tasks. When there is a large distance, transfer learning would only have a negative influence on the learning of the new task. On the contrary, PNNs' performance on new tasks usually would not be worse than that of reference single-task models, as new columns are instantiated for new tasks, and the knowledge transfer is controlled by the trainable adapters. However, the gradient propagation and parameter updating could become more difficult because of the extra model complexity introduced by the lateral connections, such that the new columns could possibly be influenced by the strong representation learned by the prior columns.

Fig. 12 shows the change of ACC of PNNs and vanilla incremental learning models under different learning orders. PNNs outperform reference single-task models in all possible learning orders. The best two scenarios for PNNs are the training order of ABC and ACB, where task A is the first task learned. On the other hand, the overall performance of transfer learning models is always worse than reference single-task models because of the forgetting of prior knowledge. The best two scenarios for transfer learning models are the training order of ACB and CAB, where task B is the last task learned. It can be concluded that PNN achieves the best performance when starting with the task with the least data so that the new tasks would not be affected by the strong representations learned by the previous columns. In addition, the vanilla incremental learning model achieves the best performance

when ending with the task with the most centered characteristics so that the performance on all tasks could reach a final compromise since old knowledge has been forgotten.

5. Conclusion

An accurate estimation of SOC ensures safety for the users and optimized usage of the battery systems. With the rapid development of battery technology and accelerated iteration of commercial lithium-ion battery cells, the potential benefit for energy storage systems to use different types of battery cells in a mixture emerges. To accommodate this background, this paper proposes a continual learning framework for online SOC estimation across diverse lithium-ion batteries. The proposed framework relies on the progressive neural network architecture and utilizes temporal convolutional networks as the fundamental unit for feature extraction. In the PNN architecture, one TCN column is instantiated for each new task, together with the lateral connections between prior columns and the new one, enabling forward knowledge transfer from the old columns. While training the new column with the corresponding lateral connections on the new task, the weight parameters of the previous columns are frozen, so the forgetting of previously learned knowledge is avoided by design. Three public lithium-ion battery drive cycle datasets were used for the experiments and analyses, covering three manufacturers, two chemistry types, and two standardized formats with diverse cycling data and ambient conditions. A variety of evaluation metrics were applied to

study the model's regression accuracy, memory stability, and learning plasticity. Under the learning order of tasks A, B, and C, the proposed PNN model achieved the highest FWT on both task B (0.520%) and task C (0.556%) compared with other models, which demonstrates the positive knowledge transfer from old to new tasks enabled by the lateral connections. Meanwhile, the proposed PNN always had a BWT of 0%, as they are immune to forgetting by design. As ACC is majorly influenced by BWT, the proposed PNN also achieved the best ACC of 99.001% on the three tasks. Detailed studies of the proposed framework's performance and complexity were also carried out, where PNN with two extra TCN columns of 2, 4, and 8 channels, namely 4.552×10^3 extra parameters, had an average test MAE on task B and task C of 1.2205%, which was even better than the average MAE of 1.385% of two single-task TCN models with 16, 32, and 64 channels with 1.71586×10^5 parameters. Therefore, the proposed PNN framework has not only proven to have state-of-the-art accuracy for the SOC estimation of different types of lithium-ion batteries simultaneously but also significantly reduces model complexity compared to utilizing multiple conventional single-task models while maintaining the same overall performance. In addition, in-depth analyses were conducted regarding the influence of task ordering on PNNs and transfer learning models, with the interaction between tasks carefully discussed as well. Results show that PNNs outperform reference single-task models in all possible learning orders and achieve the best performance of an ACC of 99.001% (ABC) and an ACC of 98.999% (ACB) when starting with the task with the least data so that the new tasks would not be affected by the strong representations learned by the previous columns. So far, only lithium-ion battery chemistries with similar OCV curves have been studied, so it would be interesting for future works to expand the cell types further. By applying continual learning and setting up a benchmark for the task of SOC estimation of hybrid lithium-ion battery systems, we believe this extensive work brings great novelty to the research field and has tremendous potential in future industrial applications.

Abbreviations

The following abbreviations are used in this manuscript:

BCDC	Braunschweig City Driving Cycle
BMS	Battery Management System
CSHVC	City Suburban Heavy Vehicle Cycle
DNN	Deep Neural Network
EWC	Elastic Weight Consolidation
FLOPs	Floating Point Operations
FTP-72	Federal Test Procedure-72
FTP-75	Federal Test Procedure-75
GRU	Gated Recurrent Unit
HHDDT	Heavy Heavy-Duty Diesel Truck Composite Cycle
HWFET	Highway Fuel Economy Test
IM	Inspection and Maintenance
LA92	California Unified Cycle
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
OCTBC	Orange County Transit Bus Cycle
OCV	Open Circuit Voltage
PDMHC	Port Drayage Metro Highway Cycle California
PDTCB	Parcel Delivery Truck Cycle Baltimore
PNN	Progressive Neural Network
ReLU	Rectified Linear Unit
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
TCN	Temporal Convolutional Network
SOC	State of Charge
UDDS	Urban Dynamometer Driving Schedule
US06	Supplemental Federal Test Procedure

CRediT authorship contribution statement

Jiaqi Yao: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Bowen Zheng:** Methodology, Investigation. **Julia Kowal:** Writing – review & editing, Supervision.

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.

Acknowledgments

We thank the High-Performance Computing Cluster of TU Berlin ZECM for the GPU resources and the Open Access Publication Fund of TU Berlin for the support.

Data availability

The data utilized in this paper is from three public datasets [63–65].

References

- [1] B. Lin, Z. Li, Towards world's low carbon development: the role of clean energy, Appl. Energy 307 (2022) 118160, <http://dx.doi.org/10.1016/j.apenergy.2021.118160>.
- [2] C. Tarhan, M.A. Çil, A study on hydrogen, the clean energy of the future: hydrogen storage methods, J. Energy Storage 40 (2021) 102676, <http://dx.doi.org/10.1016/j.est.2021.102676>.
- [3] S.E. Hosseini, M.A. Wahid, Hydrogen from solar energy, a clean energy carrier from a sustainable source of energy, Int. J. Energy Res. 44 (6) (2020) 4110–4131, <http://dx.doi.org/10.1002/er.4930>.
- [4] K.M. Tan, T.S. Babu, V.K. Ramachandaramurthy, P. Kasinathan, S.G. Solanki, S.K. Raveendran, Empowering smart grid: A comprehensive review of energy storage technology and application with renewable energy integration, J. Energy Storage 39 (2021) 102591, <http://dx.doi.org/10.1016/j.est.2021.102591>.
- [5] A.A. Kebede, T. Kalogiannis, J. Van Mierlo, M. Berecibar, A comprehensive review of stationary energy storage devices for large scale renewable energy sources grid integration, Renew. Sustain. Energy Rev. 159 (2022) 112213, <http://dx.doi.org/10.1016/j.rser.2022.112213>.
- [6] M.M. Rahman, A.O. Oni, E. Gemechu, A. Kumar, Assessment of energy storage technologies: A review, Energy Convers. Manage. 223 (2020) 113295, <http://dx.doi.org/10.1016/j.enconman.2020.113295>.
- [7] K. Marnell, M. Obi, R. Bass, Transmission-scale battery energy storage systems : a systematic literature review, Energies 12 (23) (2019) 4603, <http://dx.doi.org/10.3390/en12234603>.
- [8] M. Killer, M. Farrokhseresht, N.G. Paterakis, Implementation of large-scale Li-ion battery energy storage systems within the EMEA region, Appl. Energy 260 (2020) 114166, <http://dx.doi.org/10.1016/j.apenergy.2019.114166>.
- [9] Y. Liang, C.-Z. Zhao, H. Yuan, Y. Chen, W. Zhang, J.-Q. Huang, D. Yu, Y. Liu, M.-M. Titirici, Y.-L. Chueh, H. Yu, Q. Zhang, A review of rechargeable batteries for portable electronic devices, InfoMat 1 (1) (2019) 6–32, <http://dx.doi.org/10.1002/inf2.12000>.
- [10] G. Zubí, R. Dufo-López, M. Carvalho, G. Pasaoglu, The lithium-ion battery: State of the art and future perspectives, Renew. Sustain. Energy Rev. 89 (2018) 292–308, <http://dx.doi.org/10.1016/j.rser.2018.03.002>.
- [11] D.N.T. How, M.A. Hannan, M.S. Hossain Lipu, P.J. Ker, State of charge estimation for lithium-ion batteries using model-based and data-driven methods: A review, IEEE Access 7 (2019) 136116–136136, <http://dx.doi.org/10.1109/ACCESS.2019.2942213>.
- [12] J. Rivera-Barrera, N. Muñoz-Galeano, H. Sarmiento-Maldonado, Soc estimation for lithium-ion batteries: Review and future challenges, Electronics 6 (4) (2017) 102, <http://dx.doi.org/10.3390/electronics6040102>.
- [13] M. Hannan, M. 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. Sustain. Energy Rev. 78 (2017) 834–854, <http://dx.doi.org/10.1016/j.rser.2017.05.001>.
- [14] K.W.E. Cheng, B.P. Divakar, H. Wu, K. Ding, H.F. Ho, Battery-management system (BMS) and SOC development for electrical vehicles, IEEE Trans. Veh. Technol. 60 (1) (2011) 76–88, <http://dx.doi.org/10.1109/TVT.2010.2089647>.

- [15] Y. Jin, W. Zhao, Z. Li, B. Liu, K. Wang, SOC estimation of lithium-ion battery considering the influence of discharge rate, *Energy Rep.* 7 (2021) 1436–1446, <http://dx.doi.org/10.1016/j.egyr.2021.09.099>.
- [16] M. Lelie, T. Braun, M. Knips, H. Nordmann, F. Ringbeck, H. Zappen, D.U. Sauer, Battery management system hardware concepts: An Overview, *Appl. Sci.* 8 (4) (2018) 534, <http://dx.doi.org/10.3390/app8040534>.
- [17] B. Balasingam, M. Ahmed, K. Pattipati, Battery management systems—Challenges and some solutions, *Energies* 13 (11) (2020) 2825, <http://dx.doi.org/10.3390/en13112825>.
- [18] R. Xiong, L. Li, J. Tian, Towards a smarter battery management system: A critical review on battery state of health monitoring methods, *J. Power Sources* 405 (2018) 18–29, <http://dx.doi.org/10.1016/j.jpowsour.2018.10.019>.
- [19] H. Gabbar, A. Othman, M. Abdussami, Review of battery management systems (BMS) development and industrial standards, *Technologies* 9 (2) (2021) 28, <http://dx.doi.org/10.3390/technologies9020028>.
- [20] N. Khan, C.A. Ooi, A. Alturki, M. Amir, Shreasth, T. Alharbi, A critical review of battery cell balancing techniques, optimal design, converter topologies, and performance evaluation for optimizing storage system in electric vehicles, *Energy Rep.* 11 (2024) 4999–5032, <http://dx.doi.org/10.1016/j.egyr.2024.04.041>.
- [21] J. Yao, J. Kowal, A multi-scale data-driven framework for online state of charge estimation of lithium-ion batteries with a novel public drive cycle dataset, *J. Energy Storage* 107 (2025) 114888, <http://dx.doi.org/10.1016/j.est.2024.114888>.
- [22] N. Watrin, B. Blunier, A. Miraoui, Review of adaptive systems for lithium batteries state-of-charge and state-of-health estimation, in: 2012 IEEE Transportation Electrification Conference and Expo, ITEC, IEEE, Dearborn, MI, USA, 2012, pp. 1–6, <http://dx.doi.org/10.1109/ITEC.2012.6243437>.
- [23] S. Neupert, J. Kowal, Model-based state-of-charge and state-of-health estimation algorithms utilizing a new free lithium-ion battery cell dataset for benchmarking purposes, *Batteries* 9 (7) (2023) 364, <http://dx.doi.org/10.3390/batteries9070364>.
- [24] J. Yao, S. Neupert, J. Kowal, Cross-stitch networks for joint state of charge and state of health online estimation of lithium-ion batteries, *Batteries* 10 (6) (2024) 171, <http://dx.doi.org/10.3390/batteries10060171>.
- [25] I. Baccouche, S. Jemmali, A. Maylah, B. Manai, N.E.B. Amara, Implementation of an improved Coulomb-counting algorithm based on a piecewise SOC-ovc relationship for SOC estimation of li-IonBattery, 2018, <http://dx.doi.org/10.48550/ARXIV.1803.10654>.
- [26] L. He, D. Guo, An improved Coulomb counting approach based on numerical iteration for SOC estimation with real-time error correction ability, *IEEE Access* 7 (2019) 74274–74282, <http://dx.doi.org/10.1109/ACCESS.2019.2921105>.
- [27] S. Sundaresan, B. Devabattini, P. Kumar, K. Pattipati, B. Balasingam, Tabular open circuit voltage modelling of li-ion batteries for robust SOC estimation, *Energies* 15 (23) (2022) 9142, <http://dx.doi.org/10.3390/en15239142>.
- [28] P. Pillai, S. Sundaresan, P. Kumar, K.R. Pattipati, B. Balasingam, Open-circuit voltage models for battery management systems: A Review, *Energies* 15 (18) (2022) 6803, <http://dx.doi.org/10.3390/en15186803>.
- [29] K. McCarthy, H. Gullapalli, K.M. Ryan, T. Kennedy, Review—Use of impedance spectroscopy for the estimation of li-ion battery state of charge, state of health and internal temperature, *J. Electrochem. Soc.* 168 (8) (2021) 080517, <http://dx.doi.org/10.1149/1945-7111/ac1a85>.
- [30] A. Cuadras, O. Kanoun, Soc li-ion battery monitoring with impedance spectroscopy, in: 2009 6th International Multi-Conference on Systems, Signals and Devices, IEEE, Djerba, Tunisia, 2009, pp. 1–5, <http://dx.doi.org/10.1109/SSD.2009.4956761>.
- [31] 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. Sustain. Energy Rev.* 113 (2019) 109233, <http://dx.doi.org/10.1016/j.rser.2019.06.040>.
- [32] X. Lai, Y. Zheng, T. Sun, A comparative study of different equivalent circuit models for estimating state-of-charge of lithium-ion batteries, *Electrochim. Acta* 259 (2018) 566–577, <http://dx.doi.org/10.1016/j.electacta.2017.10.153>.
- [33] T. Uhm, S. Kim, State-of-charge estimation for remaining flying time prediction of small UAV using adaptive robust extended Kalman filter, *IEEE Trans. Aerosp. Electron. Syst.* (2024) 1–20, <http://dx.doi.org/10.1109/TAES.2024.3449273>.
- [34] Y. Liu, R. Ma, S. Pang, L. Xu, D. Zhao, J. Wei, Y. Huangfu, F. Gao, A nonlinear observer SOC estimation method based on electrochemical model for lithium-ion battery, *IEEE Trans. Ind. Appl.* 57 (1) (2021) 1094–1104, <http://dx.doi.org/10.1109/TIA.2020.3040140>.
- [35] Z. Yu, R. Huai, L. Xiao, State-of-charge estimation for lithium-ion batteries using a Kalman filter based on local linearization, *Energies* 8 (8) (2015) 7854–7873, <http://dx.doi.org/10.3390/en8087854>.
- [36] Z. Chen, Y. Fu, C.C. Mi, State of charge estimation of lithium-ion batteries in electric drive vehicles using extended Kalman filtering, *IEEE Trans. Veh. Technol.* 62 (3) (2013) 1020–1030, <http://dx.doi.org/10.1109/TVT.2012.2235474>.
- [37] M. Shehab El Din, A.A. Hussein, M.F. Abdel-Hafez, Improved battery SOC estimation accuracy using a modified UKF with an adaptive cell model under real EV operating conditions, *IEEE Trans. Transp. Electrification* 4 (2) (2018) 408–417, <http://dx.doi.org/10.1109/TTE.2018.2802043>.
- [38] S. Wang, S. Zhang, S. Wen, C. Fernandez, An accurate state-of-charge estimation of lithium-ion batteries based on improved particle swarm optimization-adaptive square root Cubature Kalman filter, *J. Power Sources* 624 (2024) 235594, <http://dx.doi.org/10.1016/j.jpowsour.2024.235594>.
- [39] S. Wang, Q. Dang, Z. Gao, B. Li, C. Fernandez, F. Blaabjerg, An innovative square root - untraced Kalman filtering strategy with full-parameter online identification for state of power evaluation of lithium-ion batteries, *J. Energy Storage* 104 (2024) 114555, <http://dx.doi.org/10.1016/j.est.2024.114555>.
- [40] S. Wang, H. Gao, P. Takyi-Aninakwa, J.M. Guerrero, C. Fernandez, Q. Huang, Improved multiple feature-electrochemical thermal coupling modeling of lithium-ion batteries at low-temperature with real-time coefficient correction, *Prot. Control. Mod. Power Syst.* 9 (3) (2024) 157–173, <http://dx.doi.org/10.23919/PCMP.2023.000257>.
- [41] E. Chemali, P.J. Kollmeyer, M. Preindl, R. Ahmed, A. Emadi, Long short-term memory networks for accurate state-of-charge estimation of li-ion batteries, *IEEE Trans. Ind. Electron.* 65 (8) (2018) 6730–6739, <http://dx.doi.org/10.1109/TIE.2017.2787586>.
- [42] E. Chemali, P.J. Kollmeyer, M. Preindl, A. Emadi, State-of-charge estimation of li-ion batteries using deep neural networks: A machine learning approach, *J. Power Sources* 400 (2018) 242–255, <http://dx.doi.org/10.1016/j.jpowsour.2018.06.104>.
- [43] K. Yang, Y. Wang, Y. Tang, S. Zhang, Z. Zhang, A temporal convolution and gated recurrent unit network with attention for state of charge estimation of lithium-ion batteries, *J. Energy Storage* 72 (2023) 108774, <http://dx.doi.org/10.1016/j.est.2023.108774>.
- [44] M.K. Al-Alawi, A. Jaddoa, J. Cugley, H. Hassanin, A novel enhanced SOC estimation method for lithium-ion battery cells using cluster-based LSTM models and centroid proximity selection, *J. Energy Storage* 97 (2024) 112866, <http://dx.doi.org/10.1016/j.est.2024.112866>.
- [45] C. Bian, Z. Duan, Y. Hao, S. Yang, J. Feng, Exploring large language model for generic and robust state-of-charge estimation of li-ion batteries: A mixed prompt learning method, *Energy* 302 (2024) 131856, <http://dx.doi.org/10.1016/j.energy.2024.131856>.
- [46] X. Cheng, X. Liu, H. Deng, J. Lu, Q. Yu, State of charge estimation of LiFePO4 battery in AB hybrid battery packs, *J. Energy Storage* 108 (2025) 115070, <http://dx.doi.org/10.1016/j.est.2024.115070>.
- [47] M. McCloskey, N.J. Cohen, Catastrophic interference in connectionist networks: the sequential learning problem, in: *Psychology of Learning and Motivation*, Vol. 24, Elsevier, 1989, pp. 109–165, [http://dx.doi.org/10.1016/S0079-7421\(08\)60536-8](http://dx.doi.org/10.1016/S0079-7421(08)60536-8).
- [48] R. Hadsell, D. Rao, A.A. Rusu, R. Pascanu, Embracing change: Continual Learning in deep neural networks, *Trends Cogn. Sci.* 24 (12) (2020) 1028–1040, <http://dx.doi.org/10.1016/j.tics.2020.09.004>.
- [49] E.L. Aleixo, J.G. Colonna, M. Cristo, E. Fernandes, Catastrophic forgetting in deep learning: A comprehensive taxonomy, 2023, <http://dx.doi.org/10.48550/ARXIV.2312.10549>.
- [50] M.D. Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, T. Tuytelaars, A continual learning survey: Defying forgetting in classification tasks, *IEEE Trans. Pattern Anal. Mach. Intell.* (2021) 1, <http://dx.doi.org/10.1109/TPAMI.2021.3057446>, arXiv:1909.08383.
- [51] D. Lopez-Paz, M. Ranzato, Gradient episodic memory for continual learning, 2022, <http://dx.doi.org/10.48550/arXiv.1706.08840>, arXiv:1706.08840.
- [52] D. Rolnick, A. Ahuja, J. Schwarz, T.P. Lillicrap, G. Wayne, Experience replay for continual learning, 2018, <http://dx.doi.org/10.48550/ARXIV.1811.11682>.
- [53] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A.A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, R. Hadsell, Overcoming catastrophic forgetting in neural networks, *Proc. Natl. Acad. Sci.* 114 (13) (2017) 3521–3526, <http://dx.doi.org/10.1073/pnas.1611835114>, arXiv:1612.00796.
- [54] A. Chaudhry, P.K. Dokania, T. Ajanthan, P.H.S. Torr, Riemannian walk for incremental learning: Understanding forgetting and intransigence, 2018, <http://dx.doi.org/10.48550/arXiv.1801.10112>, arXiv:1801.10112.
- [55] A. Mallya, S. Lazebnik, PackNet: Adding multiple tasks to a single network by iterative pruning, 2018, <http://dx.doi.org/10.48550/arXiv.1711.05769>, arXiv:1711.05769.
- [56] A.A. Rusu, N.C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, R. Hadsell, Progressive neural networks, 2016, <http://dx.doi.org/10.48550/ARXIV.1606.04671>.
- [57] S. Bai, J.Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, 2018, <http://dx.doi.org/10.48550/arXiv.1803.01271>, arXiv:1803.01271.
- [58] M. Lee, S. Kim, S. Kim, J.-I. Choi, Bilevel-optimized continual learning for predicting capacity degradation of lithium-ion batteries, *J. Energy Storage* 86 (2024) 111187, <http://dx.doi.org/10.1016/j.est.2024.111187>.
- [59] Y. Che, Y. Zheng, S. Onori, X. Hu, R. Teodorescu, Increasing generalization capability of battery health estimation using continual learning, *Cell Rep. Phys. Sci.* 4 (12) (2023) 101743, <http://dx.doi.org/10.1016/j.xcrp.2023.101743>.
- [60] J.P. Ortiz, G.P. Ayabaca, A.R. Cardenas, D. Cabrera, J.D. Valladolid, Continual reinforcement learning using real-world data for intelligent prediction of SOC consumption in electric vehicles, *IEEE Lat. Am. Trans.* 20 (4) (2022) 624–633, <http://dx.doi.org/10.1109/TIA.2022.9675468>.

- [61] T. Salimans, D.P. Kingma, Weight normalization: A simple reparameterization to accelerate training of deep neural networks, 2016, <http://dx.doi.org/10.48550/arXiv.1602.07868>, arXiv:1602.07868.
- [62] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, *J. Mach. Learn. Res.* 15 (56) (2014) 1929–1958.
- [63] P. Kollmeyer, Panasonic 18650PF li-ion battery data, 2018, <http://dx.doi.org/10.17632/WYKHT8Y7TG.1>.
- [64] P. Kollmeyer, Samsung INR21700 30T 3Ah Li-ion battery data, 2023, <http://dx.doi.org/10.17632/9XYVY2NJJ3.2>.
- [65] J. Yao, Lithium-ion battery drive cycle dataset, 2024, <http://dx.doi.org/10.14279/DEPOSITIONCE-21133>.
- [66] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, 2019, <http://dx.doi.org/10.48550/arXiv.1907.10902>, arXiv:1907.10902.
- [67] L. Wang, X. Zhang, H. Su, J. Zhu, A comprehensive survey of continual learning: Theory, method and application, *IEEE Trans. Pattern Anal. Mach. Intell.* 46 (8) (2024) 5362–5383, <http://dx.doi.org/10.1109/TPAMI.2024.3367329>.