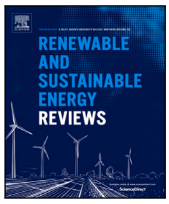
Renewable and Sustainable Energy Reviews 208 (2025) 115045

Contents lists available at ScienceDirect

Renewable and Sustainable Energy Reviews

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



Enhanced Gaussian process dynamical modeling for battery health status forecasting

W.W. Xing [a], Z. Zhang [b,c], A.A. Shah [b,*]

[a] School of Mathematics and Statistics, University of Sheffield, Hounsfield Road, Sheffield S3 7RH, United Kingdom
[b] Key Laboratory of Low-grade Energy Utilisation Technologies and Systems, MOE, Chongqing University, Chongqing 400030, China
[c] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, 518060, China

ARTICLE INFO

Dataset link: https://github.com/PericlesHat/enhanced-GPDM

Keywords:
Li-ion battery degradation
Nonlinear state-space model
Gaussian processes
Markov model
Transfer learning
Features

ABSTRACT

Monitoring the state-of-health of Li-ion batteries is a critical component of battery management systems in electric vehicles. A large number of feature-based machine-learning methods have been introduced in the last decade to improve the accuracy of predictions of the state-of-health and end-of-life, especially early in the lifetime of the battery stack. Unless multiple battery data sets are used for direct and crude predictions of the end-of-life, however, such an approach is infeasible since the features are not known for future cycles. In this study a new nonlinear state-space model that can overcome this limitation is introduced. The powerful Gaussian process dynamical model is extended by generalizing the covariance structure, and therefore permitting more flexible models for the observables and latent variables. The model is further enhanced with transfer learning, to yield accurate early predictions of the future state-of-health of Li-ion batteries up to end-of-life. Experiments conducted on two of the NASA Ames Battery data sets and the Oxford Battery Degradation data set demonstrate the accuracy and superiority of the new model over state-of-the-art benchmarks algorithms, including supervised Gaussian process models, deep convolutional networks, recurrent networks and support vector regression. The root mean square error is reduced by up to 43% on the NASA data sets and by up to 54% on the Oxford data set.

1. Introduction

Batteries are essential components of modern energy systems, providing energy storage capabilities in a broad range of sectors. The most prominent emerging application is electric vehicles, which are forecast to become the primary method of transportation in the developed world beyond 2030. The main technology under consideration for these applications is Li-ion batteries, which undergo a number of degradation processes during operation [1,2], leading to an irreversible loss in capacity and a need to replace the stack when the end-of-life (EOL) is reached. The EOL is normally defined as the charge–discharge cycle number at which the stack reaches 70%–80% of its rated capacity, after which it can be used for second-life applications [3].

One of the challenges in developing electric vehicles, therefore, lies in devising strategies to monitor, control and predict the future health status of the battery stack [4], as part of a broader attempt to develop digital twins [5]. This can aid maintenance, lower the risk of hazards such as fire and explosion, and inform the decision to retire the stack. The health status is usually defined in terms of the stack or battery state-of-health (SOH), which is a normalized capacity at a particular state-of-charge (SOC). An algorithm can be placed inside a battery management system to predict the SOH for future charge–discharge cycles [6,7]. Such algorithms provide running predictions of the EOL and remaining useful life (RUL), namely the number of cycles remaining before the stack reaches EOL.

Algorithms for SOH prediction can be devised in a number of ways. Early attempts relied on physics-based or semi-empirical models of the degradation process. Physics-based models are, however, hampered by a lack of knowledge of the degradation phenomena. As a consequence, data-driven or machine learning approaches have emerged as the preferred choice, especially in the past decade [8]. These approaches can be categorized in terms of (a) the overall strategy, and (b) the machine learning method employed. Strategy 1 seeks a map between the cycle number (input) and the SOH (output) [9]. Strategy 2 augments the input in Strategy 1 with features extracted from current, voltage, temperature, impedance and other data related to the charge–discharge cycles [10,11]. Strategy 3 directly defines the output as the RUL or EOL and uses data from multiple batteries [12,13].

Strategy 1 has the benefit of simplicity in terms of the algorithm and data acquisition, but it leads to poor estimates, especially for early

* Corresponding author.
E-mail address: ashah@cqu.edu.cn (A.A. Shah).

https://doi.org/10.1016/j.rser.2024.115045
Received 13 September 2023; Received in revised form 31 July 2024; Accepted 22 October 2024
Available online 9 November 2024

Nomenclature

Abbreviations

Acronym	Meaning
CNN	Convolutional neural network
DNN	Deep neural network
E-GPDM	Enhanced Gaussian process dynamical model
EOL	End of life
GP	Gaussian process
GPDM	Gaussian process dynamical model
GPLVM	Gaussian process latent variable model
GRU	Gated recurrent unit
LSTM	Long short term memory
RMSE	Root mean square error
RNN	Recurrent neural network
RUL	Remaining useful life
SOH	State of charge
SOH	State of health
SVR	Support vector regression

Symbols

Symbol	Meaning
$a(n)$	Attribute or feature as a function of cycle number n
$\mathbf{A}, \mathbf{B}$	Matrices of basis expansion coefficients
$\mathbf{a}, \mathbf{b}$	Row vectors of $\mathbf{A}, \mathbf{B}$
$\hat{\mathbf{a}}, \hat{\mathbf{b}}$	Column vectors of $\mathbf{A}, \mathbf{B}$
$\mathcal{D}$	Datasets
D	Dimension of the observables (data) space
$\mathbf{K}$	Column or row covariance matrix
$k(\cdot,\cdot\|\cdot)$	Kernel function
$\mathbf{L}$	Cholesky decomposition factor
$\mathcal{L}$	Likelihood function
m	Battery label
n	Cycle number
$\mathbf{n}$	Normal noise
w	Weight factor for noise
Q	Latent space dimension
T	Training point number
$\mathbf{x}$	Latent/hidden variable
$\mathbf{X}$	Matrix of latent variables
$\mathbf{y}$	Observable
$\mathbf{Y}$	Matrix of observed data
$\boldsymbol{\theta}$	vector of hyperparameters
θ	Hyperparameter
$\boldsymbol{\theta}$	Vector of hyperparameter
$\boldsymbol{\Lambda}$	Gaussian process predictive variance
$\boldsymbol{\mu}$	Gaussian process mean estimate of observable/latent variable
σ	Noise variance
$\boldsymbol{\Sigma}$	Covariance matrix
$\boldsymbol{\psi}, \boldsymbol{\phi}$	Vectors of basis functions
ψ, ϕ	Basis functions
$\boldsymbol{\Psi}, \boldsymbol{\Phi}$	Design matrices corresponding to $\boldsymbol{\psi}, \boldsymbol{\phi}$

predictions. Strategy 2 was introduced to improve upon the predictive performance of Strategy 1. Hand-crafted features based on the collected data [10,14] or features extracted during learning [12,15] can exhibit a strong correlation to the SOH. The obvious logical flaw in Strategy 2 is that the features are not known for future cycles, and so cannot be used as inputs to predict the SOH beyond the next cycle. Strategy 3 does not suffer from this logical flaw and it has been used successfully for predicting degradation in batteries [16,17] and other devices or components, such as bearings [18] and supercapacitors [19]. Strategy 3 is also able to exploit multiple data sets to improve predictions. Random variabilities in the performance of batteries, however, means that it can only furnish crude estimates of the EOL or RUL for a batch of batteries (e.g., post manufacturing), as pointed out by Severson et al. [20] and as is evident from the results.

A variety of machine learning methods have been used in Approaches 1–3, including Gaussian process (GP) models [3,9], support vector regression (SVR) [21,22], neural networks (ANN) and deep neural networks (DNN) [17,23]. DNN implementations include recurrent networks (RNNs) [24,25], convolutional networks (CNNs) [26], transformers and encoder–decoder models [17,27], and hybrid networks [28]. DNNs are a popular choice, but they were developed primarily for problems in which the data sets are very large, and their success in such applications is primarily due to their scalability when trained with stochastic gradient descent algorithms. On small- or medium-sized data sets (the case in single battery or single stack SOH prediction), their high model variance and lower accuracy compared to non-parametric methods such as GP models and SVR are major disadvantages. Moreover, DNNs are designed for, and best suited to classification (CNNs) and sequential problems in natural language processing (RNNs); such problems involve categorical outputs arising from discrete probability distributions, rather than continuous sequential data such as a SOH.

Time-series methods are in theory more appropriate for SOH prediction, but only a few examples are found in the literature, in the form of linear state-space models [29,30]. For example, Kim et al. [30] developed an auto-regressive moving average (ARIMA) model with exogenous inputs or features. As with other feature-based models, the features are not available for future cycles. The unsupervised Gaussian process dynamical model (GPDM) [31,32] extends the Gaussian process latent variable model (GPLVM) [33] by including a Markov model for the latent variable, which allows for the prediction of sequential data. In this study, a new approach for SOH prediction based on the GPDM is developed. The GPDM assumption of independence across coordinates is relaxed using a marginalization procedure and kernel substitution, leading to an infinite-dimensional approximating subspace with a far richer covariance structure. This method is termed Enhanced GPDM (E-GPDM) to distinguish it from the original GPDM. To allow for training on multiple SOH data sets, a transfer learning approach is used together with E-GPDM.

E-GPDM overcomes the major issue in Approach 2 regarding the incorporation of features, by recursively predicting features as well as the SOH up to an arbitrary horizon. Thus, the main novelty of E-GPDM is that it is the first method that can incorporate features for predicting the SOH more than one cycle ahead. Transfer learning leverages related battery data sets for training, and further enhances the predictive power of E-GPDM. E-GPDM does not require large data sets and it exhibits a low model variance. As a Bayesian method, it also furnishes uncertainty bounds, which facilitates better-informed decision making on second-life use. Numerical experiments on two NASA AMES data sets and the Oxford Battery Degradation data set demonstrate that E-GPDM is able to provide accurate predictions of the SOH trajectory early in the battery lifetime. Comparisons are made to state-of-the-art supervised GPs, SVR and DNNs, for which transfer learning is also used to ensure fair comparisons. On the NASA data sets, the root mean square error (RMSE) is reduced in 12 out of 15 cases, by up to 43% when using 33% of the data for training, and by up to 45% with a 50% training ratio. On the Oxford data set, the RMSE is reduced in 9 out of 12 cases, by up to 50% with a 33% training ratio and by up to 54% with a 50% training ratio.

2. Data and methods

2.1. Problem definition

It is noted that the use of data from a single cell is valid for the study of stacks, since a battery stack would exhibit a similar SOH trajectory; roughly speaking, an average of the SOH for the individual cells. The discharge capacity $c(n)$ of a battery for a given cycle n can be used as the basis for defining its state-of-health (SOH), namely $\mathrm{SOH}(n) = c(n)/c(0)$. The EOL and RUL for the current cycle are defined as

$$\mathrm{EOL} = \min n \quad \text{such that} \quad \mathrm{SOH}(n) \le \vartheta, \quad \mathrm{RUL} = \mathrm{EOL} - n \tag{1}$$

for some threshold ϑ, normally in the range $\vartheta \in (0.7, 0.8)$.That is, the EOL is the first cycle for which the SOH falls below the threshold, usually 70 to 80% of the rated or initial capacity $c(0)$. The RUL is the number of cycles remaining before the EOL is reached. The primary goal is to predict the curve $\mathrm{SOH}(n)$ accurately and efficiently, especially early in the evolution, namely for cycles $n \le \frac{1}{2}\mathrm{EOL}$, i.e., less than or equal to half the lifetime.

Let $(\mathbf{y}_n)_{n\in\mathbb{N}}$, $\mathbf{y}_n \in \mathbb{R}^D$, be a time-series comprising D attributes of the battery at each cycle n. In the simplest case, $D = 2$ and the only attributes are the SOH and the cycle number, $\mathbf{y}_n = [n, \mathrm{SOH}(n)]^\top$. In the more general case

$$\mathbf{y}_n = [n, a_1(n), \dots, a_{D-1}(n)]^\top$$

with $a_1(n) = \mathrm{SOH}(n)$ and additional scalar or vector attributes $a_2(n), \dots, a_{D-1}(n)$. For example, $a_2(n)$ could be the temperature, $a_3(n)$ the voltage, $a_4(n)$ the humidity and so on, where the values are taken at some fixed SOC on each charge or discharge cycle. Alternatively, whole voltage, temperature, etc. curves can be used, in a vectorized form. The main aim is then to predict $a_1(n) = \mathrm{SOH}(n)$ for $n > T$ given training data $D_T = \{\mathbf{y}_n\}_{n=1}^T$. The data sets used (described in Section 2.6) take the form $D_N = \{\mathbf{y}_n\}_{n=1}^N$ for some cycle number N that exceeds the EOL. $T < N$ cycles are used for training, where the training number T can be varied. Only scalar attributes are considered, although the method allows for the vector case.

Regarding the use of the term 'features', which is the common terminology in the literature on battery degradation, it is pointed out that in feature engineering, features are quantities extracted from data, with the components of the data termed 'attributes'. In particular, physical quantities that are used as inputs to battery degradation models (as opposed to features learned from these quantities) will be referred to as attributes in this study. In contrast, the components of the latent variables in E-GPDM are referred to as features, since they map the attributes in the data to a feature or latent space.

When focusing on the SOH, in contrast to the RUL or EOL, incorporating attributes as inputs is infeasible since they are unknown for future cycles. In E-GPDM, this will no longer present a problem for reasons that were given in the introduction and which will be fully explained in Section 2.3. To enhance the predictive power of E-GPDM, data from other batteries of the same type and operated under the same conditions is exploited during training. This is explained the next section.

2.2. Transfer learning

The approach of Richardson et al. [34] is adopted in order to inject additional battery data into the learning process.Consider data $D_{N_m}^m = \{\mathbf{y}_n^m\}_{n=1}^{N_m}$, $m = 1, 2, \dots, M$, in which N_m is the number of cycles for battery $m = 1, \dots, M$, and $\mathbf{y}_n^m = \mathbb{R}^D$ is the corresponding vector of attributes at cycle n. For simplicity, it is assumed that $N_m = N$ for all batteries, without loss of generality. The task is to predict $a_1^{m_*}(n) = \mathrm{SOH}^{m_*}(n)$ for a given battery of interest $m = m_*$, from cycles $n = T + 1$ to $n = \mathrm{EOL}$. To perform this task, the following data is utilized for training

$$\mathcal{T}^{m_*} = \left(\bigcup_{m\neq m_*} D_N^m \right) \cup D_T^{m_*} \tag{2}$$

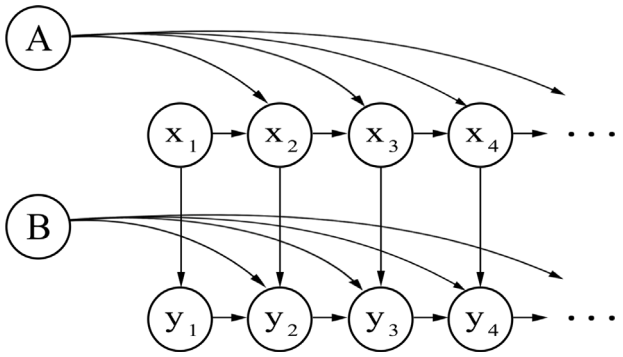


Fig. 1. Graphical representation of GPDM. A and B denote weights for the basis functions.

That is, all of the data for batteries $m \neq m_*$, and only the first T cycles for battery $m = m_*$ (the battery of interest).

To implement the training procedure efficiently, a single output relevant to all batteries is formulated as follows

$$\mathbf{y}_n = [n, m, a_1(n), \dots, a_{D-1}(n)]^\top, \tag{3}$$

In this way, the battery-specific outputs $\mathbf{y}_n^m$ are replaced by $\mathbf{y}_n$, in which the battery label m is used to distinguish between batteries. In the next section, the development of the E-GPDM method is presented.

2.3. Enhanced Gaussian process dynamical models

The GPDM is primarily used to track the dynamics of latent variables, or low-dimensional embeddings. It contains a nonlinear probabilistic mapping from a latent to an observation space, together with a dynamical model in the latent space [31].In contrast to the machine learning approaches usually employed for battery degradation forecasting, it is an unsupervised method, only requiring a singular data point (which could be called an input or an output) consisting of the battery attributes. It can be represented by the graphical model in Fig. 1. Consider training data $\mathbf{Y} = [\mathbf{y}_1 \dots \mathbf{y}_T]^\top \in \mathbb{R}^{T\times D}$, which contains T observations $\mathbf{y}_n \in \mathbb{R}^D$, as described in Section 2.2. Let $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_T]^\top \in \mathbb{R}^{T\times Q}$ be a matrix of the latent variables $\mathbf{x}_n \in \mathbb{R}^Q$, with $Q < D$.The latent variables $\mathbf{x}_n$ represent a 'hidden' or unobservable state of the dynamical system, which in some sense captures the essential dynamics. In contrast, the available data $\mathbf{y}_n$ are the variables that can be observed or measured. In GPDM, the basic principle is to create models of the observable and latent variables: one model propagates the latent variable in time or some other ordered index (cycle number in this case), and a second model is a mapping between the latent variable and observable.

The following model with first-order Markovian dynamics is assumed

$$\begin{aligned} \mathbf{x}_n &= f(\mathbf{x}_{n-1}; \mathbf{A}) + \mathbf{n}_{x,n} = \sum_{i=1}^K \mathbf{a}_i \phi_i(\mathbf{x}_{n-1}) + \mathbf{n}_{x,n} := \mathbf{A}^\top \boldsymbol{\phi}(\mathbf{x}_{n-1}) + \mathbf{n}_{x,n} \\ \mathbf{y}_n &= g(\mathbf{x}_n; \mathbf{B}) + \mathbf{n}_{y,n} = \sum_{j=1}^M \mathbf{b}_j \psi_j(\mathbf{x}_n) + \mathbf{n}_{y,n} := \mathbf{B}^\top \boldsymbol{\psi}(\mathbf{x}_n) + \mathbf{n}_{y,n} \end{aligned} \tag{4}$$

in which the rows of $\mathbf{A} = [\mathbf{a}_1 \dots \mathbf{a}_K]^\top \in \mathbb{R}^{K\times Q}$ and $\mathbf{B} = [\mathbf{b}_1 \dots \mathbf{b}_M]^\top \in \mathbb{R}^{M\times D}$ are weights, while $\{\phi_i(\mathbf{x})\}$ and $\{\psi_j(\mathbf{x})\}$ are basis functions and $\mathbf{n}_{y,n}$, $\mathbf{n}_{x,n}$ are zero-mean Gaussian processes representing noise. The individual models in (4) are essentially generalized linear models.

The use of explicit basis functions with some user specified numbers K and M and associated weights $\mathbf{A}$ and $\mathbf{B}$ is an inconvenience that can be eliminated by using a Bayesian approach that marginalizes out both sets of weights using Gaussian priors. In the standard GPDM, $\mathbf{B}$ is integrated out by placing independent isotropic priors over its columns $\widehat{\mathbf{b}}_d \in \mathbb{R}^M$, $d = 1, \dots, D$

$$p(\widehat{\mathbf{b}}_d \mid w_d) = \mathcal{N}\left(\mathbf{0}, w_d^{-2}\mathbf{I}\right)$$

The model for each column $\hat{\mathbf{y}}_d$ of $\mathbf{Y}$ (the dth attribute) is $\hat{\mathbf{y}}_d = \boldsymbol{\Psi}\widehat{\mathbf{b}}_d + \mathbf{n}_{y,d}$, in which $\boldsymbol{\Psi} = [\boldsymbol{\psi}(\mathbf{x}_1)\dots\boldsymbol{\psi}(\mathbf{x}_T)]^\top$ is a design matrix and $\mathbf{n}_{y,d} \sim \mathcal{N}(\mathbf{0}, w_d^{-2}\sigma_Y^2\mathbf{I})$ is i.i.d. noise. Here, the variance of the error $w_d^{-2}\sigma_Y^2$ is assumed to take a particular form, namely that it is scaled by w_d^{-2} [35]. The distribution over $\hat{\mathbf{y}}_d$ is therefore $p(\hat{\mathbf{y}}_d \mid \mathbf{X}, \widehat{\mathbf{b}}_d, w_d, \sigma_Y) = \mathcal{N}\left(\boldsymbol{\Psi}\widehat{\mathbf{b}}_d, w_d^{-2}\sigma_Y^2\mathbf{I}\right)$. Standard conditional results applied to the Gaussians $p(\hat{\mathbf{y}}_d \mid \mathbf{X}, \widehat{\mathbf{b}}_d, w_d, \sigma_Y)$ and $p(\widehat{\mathbf{b}}_d \mid w_d)$ allow for the elimination of $\widehat{\mathbf{b}}_d$ in order to obtain

$$p(\hat{\mathbf{y}}_d \mid \mathbf{X}, w_d, \sigma_Y) = \mathcal{N}\left(\mathbf{0}, w_d^{-2}\mathbf{K}_Y\right), \quad \mathbf{K}_Y = \boldsymbol{\Psi}\boldsymbol{\Psi}^\top + \sigma_Y^2\mathbf{I} \tag{5}$$

in which $\mathbf{K}_Y$ is an unscaled kernel matrix.

The kernel matrix $\mathbf{K}_Y$ can be generated by an equivalent kernel function, representing covariances between the components $\hat{y}_{d,n}$ of $\hat{\mathbf{y}}_d$, i.e, $\mathrm{cov}(y_{d,n}, y_{d,n'}) = w_d^{-2}k_Y(\mathbf{x}_n, \mathbf{x}_{n'}|\boldsymbol{\theta})$. For example, the following squared exponential kernel is commonly used

$$k_Y(\mathbf{x}_n, \mathbf{x}_{n'}|\boldsymbol{\theta}_Y) = \theta_{Y,1}\exp\left(-\frac{\theta_{Y,2}}{2}\|\mathbf{x}_n - \mathbf{x}_{n'}\|^2\right) + \theta_{Y,3}^{-1}\delta(\mathbf{x}_n - \mathbf{x}_{n'}) \tag{6}$$

in which $\boldsymbol{\theta}_Y = \{\theta_{Y,1}, \theta_{Y,2}, \theta_{Y,3}\}$ and $\theta_{Y,3}^{-1} = \sigma_Y^2$. Note that the noise is absorbed into the kernel using a delta function $\delta(\mathbf{x}_n - \mathbf{x}_{n'})$. The complete joint likelihood is then given by, after using the properties of the trace $\mathrm{tr}(\cdot)$ of a matrix

$$\begin{aligned} p(\mathbf{Y} \mid \mathbf{X}, \boldsymbol{\theta}_Y, \mathbf{W}) &= \prod_{d=1}^{D} p(\hat{\mathbf{y}}_d \mid \mathbf{X}, \boldsymbol{\theta}_Y, w_d) \\ &= \frac{|\mathbf{W}|^T}{\sqrt{(2\pi)^{TD}|\mathbf{K}_Y|^D}}\exp\left\{-\frac{1}{2}\mathrm{tr}\left(\mathbf{K}_Y^{-1}\mathbf{Y}\mathbf{W}^2\mathbf{Y}^\top\right)\right\} \end{aligned} \tag{7}$$

in which $\mathbf{W} = \mathrm{diag}\left(w_1, \dots, w_D\right)$

In E-GPDM, a general matrix Gaussian prior is instead placed over $\mathbf{B}$

$$p(\mathbf{B} \mid \mathbf{K}_D, \mathbf{K}_M) = \frac{1}{\sqrt{(2\pi)^{MD}|\mathbf{K}_D|^M|\mathbf{K}_M|^D}}\exp\left\{-\frac{1}{2}\mathrm{tr}[\mathbf{K}_M^{-1}\mathbf{B}^T\mathbf{K}_D^{-1}\mathbf{B}]\right\} \tag{8}$$

in which $\mathbf{K}_M \in \mathbb{R}^{M\times M}$ and $\mathbf{K}_D \in \mathbb{R}^{D\times D}$ are the row and column covariance matrices. Marginalizing over $\mathbf{B}$ yields

$$p\left(\mathbf{y}_n \mid \mathbf{x}_n, \mathbf{K}_M, \mathbf{K}_D, \sigma_Y\right) = \mathcal{N}\left(\mathbf{0}, \boldsymbol{\psi}^T(\mathbf{x}_n)\mathbf{K}_M\boldsymbol{\psi}(\mathbf{x}_n)\mathbf{K}_D + \sigma_Y^2\mathbf{I}\right) \tag{9}$$

$\mathbf{K}_M$ is positive semidefinite (p.s.d.) and therefore possesses a unique p.s.d. square root, so that $\boldsymbol{\psi}^T(\mathbf{x}_n)\mathbf{K}_M\boldsymbol{\psi}(\mathbf{x}_n)$ defines a kernel

$$\langle\widetilde{\boldsymbol{\psi}}(\mathbf{x}_n), \widetilde{\boldsymbol{\psi}}(\mathbf{x}_{n'})\rangle := \langle\boldsymbol{\psi}(\mathbf{x}_n), \boldsymbol{\psi}(\mathbf{x}_{n'})\rangle_{\mathbf{K}_M} \tag{10}$$

in which $\widetilde{\boldsymbol{\psi}}(\mathbf{x}_n) = \sqrt{\mathbf{K}_M}\boldsymbol{\psi}(\mathbf{x}_n)$, $\langle\cdot,\cdot\rangle$ denotes the standard Euclidean inner product and $\langle\cdot,\cdot\rangle_{\mathbf{K}_M}$ is an inner product weighted by $\mathbf{K}_M$. Note that this approach models the $\mathbf{y}_n$ directly, as opposed to the original approach in which the columns of $\mathbf{Y}$ are modeled. Using an equivalent kernel $k_Y(\mathbf{x}_n, \mathbf{x}_{n'}|\boldsymbol{\theta}_Y)$ to replace $\boldsymbol{\psi}^T(\mathbf{x}_n)\mathbf{K}_M\boldsymbol{\psi}(\mathbf{x}_{n'})$ yields the separable model

$$\mathbf{y}_n \mid \mathbf{x}_n, \mathbf{K}_D, \boldsymbol{\theta}_Y, \sigma_Y \sim \mathcal{GP}\left(\mathbf{0}, k_Y(\mathbf{x}_n, \mathbf{x}_{n'}|\boldsymbol{\theta}_Y)\otimes\mathbf{K}_D + \delta(\mathbf{x}_n - \mathbf{x}_{n'})\otimes\sigma_Y^2\mathbf{I}\right) \tag{11}$$

in which $\mathcal{GP}(\cdot,\cdot)$ denotes a GP, with the first and second arguments specifying the mean and covariance functions. $\otimes$ is the Kronecker product. This leads to the likelihood

$$\begin{aligned} p\left(\mathbf{Y} \mid \mathbf{X}, \mathbf{L}_Y, \boldsymbol{\theta}_Y, \sigma_Y\right) &= \frac{1}{\sqrt{(2\pi)^{TD}|\boldsymbol{\Sigma}_Y|^D}} \\ &\times \exp\left\{-\frac{1}{2}\mathrm{tr}\left(\mathrm{vec}(\mathbf{Y})\mathrm{vec}(\mathbf{Y})^\top\boldsymbol{\Sigma}_Y^{-1}\right)\right\} \\ \boldsymbol{\Sigma}_Y &= \mathbf{K}_Y\otimes\mathbf{L}_Y\mathbf{L}_Y^\top + \sigma_Y^2\mathbf{I} \end{aligned} \tag{12}$$

in which $\mathbf{K}_Y$ is generated by $k_Y(\mathbf{x}_n, \mathbf{x}_{n'}|\boldsymbol{\theta}_Y)$, $n, n' = 1, \dots, T$. Since $\mathbf{K}_D$ is p.s.d., the correlations across the dimensions of $\mathbf{y}_n$ can be modeled indirectly using a full-rank or low-rank Cholesky decomposition $\mathbf{K}_D = \mathbf{L}_Y\mathbf{L}_Y^\top$, where $\mathbf{L}_Y$ is lower triangular.

For the latent mapping, the original GPDM places independent isotropic priors over the columns $\hat{\mathbf{a}}_q \in \mathbb{R}^K$, $q = 1, \dots, Q$, of $\mathbf{A}$, this time using standard normals, $p(\hat{\mathbf{a}}_q) = \mathcal{N}(\mathbf{0}, \mathbf{I})$. In E-GPDM, the equivalent of (13) for $\mathbf{A}$ is instead used

$$p(\mathbf{A} \mid \mathbf{K}_Q, \mathbf{K}_K) = \frac{1}{\sqrt{(2\pi)^{KQ}|\mathbf{K}_Q|^K|\mathbf{K}_K|^Q}}\exp\left\{-\frac{1}{2}\mathrm{tr}[\mathbf{K}_K^{-1}\mathbf{A}^T\mathbf{K}_Q^{-1}\mathbf{A}]\right\} \tag{13}$$

in which $\mathbf{K}_K \in \mathbb{R}^{K\times K}$ and $\mathbf{K}_Q \in \mathbb{R}^{Q\times Q}$ are again row and column covariance matrices. The model for $\mathbf{x}_n$ is autoregressive and therefore the distribution over $\mathbf{x}_1$ has to be accounted for separately, given that $\mathbf{x}_0$ is not known. GPDM assumes the model $\hat{\mathbf{x}}_{q\setminus 1} = \boldsymbol{\Phi}_{\setminus T}\hat{\mathbf{a}}_q + \mathbf{n}_{x,q}$ for each column $\hat{\mathbf{x}}_{q\setminus 1}$ of $\mathbf{X}_{\setminus 1}$, where $\mathbf{X}_{\setminus 1}$ is the matrix $\mathbf{X}$ excluding the first row. $\boldsymbol{\Phi}_{\setminus T} = [\boldsymbol{\phi}(\mathbf{x}_1)\dots\boldsymbol{\phi}(\mathbf{x}_{T-1})]^\top$ is a design matrix that excludes the last row and $\mathbf{n}_{x,q} \sim \mathcal{N}(\mathbf{0}, \sigma_X^2\mathbf{I})$ is i.i.d. noise with variance σ_X^2. The distribution over $\hat{\mathbf{x}}_q$ is therefore $p(\hat{\mathbf{x}}_{q\setminus 1} \mid \hat{\mathbf{a}}_q) = \mathcal{N}\left(\boldsymbol{\Phi}_{\setminus T}\hat{\mathbf{a}}_q, \sigma_X^2\mathbf{I}\right)$ and integrating out $\mathbf{a}_q$ as before leads to $p(\hat{\mathbf{x}}_{q\setminus 1} \mid \sigma_X) = \mathcal{N}\left(\mathbf{0}, \mathbf{K}_{X\setminus T}\right)$, in which $\mathbf{K}_{X\setminus T} = \boldsymbol{\Phi}_{\setminus T}\boldsymbol{\Phi}_{\setminus T}^\top + \sigma_X^2\mathbf{I}$ is a kernel matrix that can be generated by any equivalent kernel $k_X(\mathbf{x}_n, \mathbf{x}_{n'}|\boldsymbol{\theta}_X)$, $n, n' = 1, \dots, T-1$, with hyperparameters $\boldsymbol{\theta}_X$. The joint likelihood is then given by

$$\begin{aligned} p(\mathbf{X} \mid \boldsymbol{\theta}_X) &= p(\mathbf{x}_1)\prod_{q=2}^{Q} p(\hat{\mathbf{x}}_{q\setminus 1} \mid \boldsymbol{\theta}_X) \\ &= \frac{p(\mathbf{x}_1)}{\sqrt{(2\pi)^{(T-1)Q}|\mathbf{K}_{X\setminus T}|^Q}}\exp\left\{-\frac{1}{2}\mathrm{tr}\left(\mathbf{K}_{X\setminus T}^{-1}\mathbf{X}_{\setminus 1}\mathbf{X}_{\setminus 1}^\top\right)\right\} \end{aligned} \tag{14}$$

E-GPDM instead marginalizes over $\mathbf{A}$ and defines an equivalent kernel $k_X(\mathbf{x}_{n-1}, \mathbf{x}_{n'-1}|\boldsymbol{\theta}_X)$ to replace $\boldsymbol{\phi}^T(\mathbf{x}_{n-1})\mathbf{K}_K\boldsymbol{\phi}(\mathbf{x}_{n'-1})$, yielding

$$\begin{aligned} \mathbf{x}_n \mid \mathbf{x}_{n-1}, \mathbf{K}_Q, \boldsymbol{\theta}_X, \sigma_X \sim \mathcal{GP}\big(&\mathbf{0}, k_X(\mathbf{x}_{n-1}, \mathbf{x}_{n'-1}|\boldsymbol{\theta}_X)\otimes\mathbf{K}_Q \\ &+ \delta(\mathbf{x}_{n-1} - \mathbf{x}_{n'-1})\otimes\sigma_X^2\mathbf{I}\big) \end{aligned} \tag{15}$$

and the likelihood function

$$\begin{aligned} p\left(\mathbf{X} \mid \mathbf{L}_X, \boldsymbol{\theta}_X, \sigma_X\right) &= \frac{p(\mathbf{x}_1)}{\sqrt{(2\pi)^{(T-1)Q}|\boldsymbol{\Sigma}_X|^Q}} \\ &\times \exp\left\{-\frac{1}{2}\mathrm{tr}\left(\mathrm{vec}(\mathbf{X}_{\setminus 1})\mathrm{vec}(\mathbf{X}_{\setminus 1})^\top\boldsymbol{\Sigma}_X^{-1}\right)\right\} \\ \boldsymbol{\Sigma}_X &= \mathbf{K}_{X\setminus T}\otimes\mathbf{L}_X\mathbf{L}_X^\top + \sigma_X^2\mathbf{I} \end{aligned} \tag{16}$$

with $\mathbf{K}_{X\setminus T}$ generated by $k_X(\mathbf{x}_n, \mathbf{x}_{n'}|\boldsymbol{\theta}_X)$, $n, n' = 1, \dots, T-1$ and $\mathbf{L}_X$ being the equivalent of $\mathbf{L}_Y$. $\mathbf{x}_1$ can be modeled by an isotropic Gaussian prior but is irrelevant as far as minimizing the negative log posterior is concerned. It is noted that neither (14) nor (16) is Gaussian.

For dynamical problems, common kernels such as the squared exponential or Matérn are often combined with a linear kernel

$$k_X(\mathbf{x}_n, \mathbf{x}_{n'}|\boldsymbol{\theta}_X) = \theta_{1,X}\exp\left(-\frac{\theta_{2,X}}{2}\left\|\mathbf{x} - \mathbf{x}'\right\|^2\right) + \theta_{3,X}\mathbf{x}^T\mathbf{x}' \tag{17}$$

in which $\boldsymbol{\theta}_X = \left\{\theta_{1,X}, \theta_{2,X}, \theta_{3,X}\right\}$. In this work, multi-kernels formed by linear combinations of up to six different kernels (detailed later) were also considered; they can be defined as

$$k_X(\mathbf{x}_n, \mathbf{x}_{n'}|\boldsymbol{\theta}_X) = \sum_{l=1}^{L} w_l k_l(\mathbf{x}_n, \mathbf{x}_{n'}|\boldsymbol{\theta}_l) \tag{18}$$

in which $k_l(\mathbf{x}_n, \mathbf{x}_{n'}|\boldsymbol{\theta}_l)$ are different kernels with hyperparameters $\boldsymbol{\theta}_l$, and $\boldsymbol{\theta}_X = \{\boldsymbol{\theta}_l\}$.

2.4. Training via maximum likelihood

To learn $\mathbf{X}$, the posterior $p(\mathbf{X} \mid \mathbf{Y})$ must be inferred according to Bayes' rule (assuming that the hyperparameters have already been inferred)

$$p(\mathbf{X} \mid \mathbf{Y}) = \frac{p(\mathbf{X}, \mathbf{Y})}{p(\mathbf{Y})} \propto p(\mathbf{Y} \mid \mathbf{X})p(\mathbf{X}) \tag{19}$$

This can be achieved by approximately sampling from the posterior using Monte Carlo methods, or by using a point estimate, such as maximizing the log of the posterior, treating the denominator as a

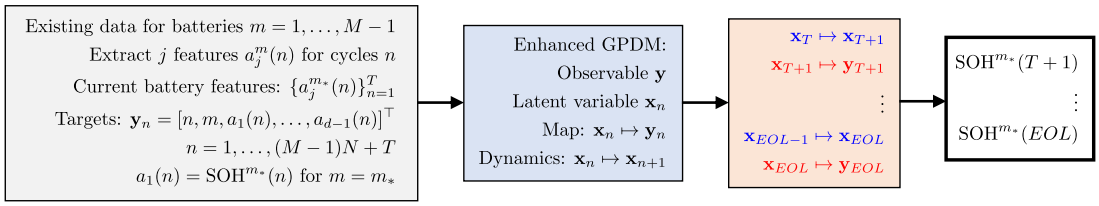


Fig. 2. Flowchart of the proposed E-GPDM method with transfer learning.

constant. A joint maximization, in which the hyperparameters are simultaneously learned, is simpler. To avoid overfitting, inverse priors are placed over the kernel hyperparameters and variances

$$p(\boldsymbol{\theta}_j) \propto \prod_i \theta_{i,j}^{-1}, \quad p(\sigma_j^2) \propto \sigma_j^{-2}, \quad j \in \{X, Y\}, \tag{20}$$

so that the posterior in Eq. (19) becomes

$$p(\mathbf{X}, \boldsymbol{\theta}_X, \boldsymbol{\theta}_Y, \sigma_X, \sigma_Y \mid \mathbf{Y}, \mathbf{L}_X, \mathbf{L}_Y) \propto p(\mathbf{Y} \mid \mathbf{X}, \mathbf{L}_Y, \boldsymbol{\theta}_Y, \sigma_Y) p(\mathbf{X} \mid \mathbf{L}_X, \boldsymbol{\theta}_X, \sigma_X) \times \prod_j p(\boldsymbol{\theta}_j) p(\sigma_j^2) \tag{21}$$

The negative joint log posterior $-\ln p(\mathbf{X}, \boldsymbol{\theta}_X, \boldsymbol{\theta}_Y, \sigma_X, \sigma_Y \mid \mathbf{Y}, \mathbf{L}_X, \mathbf{L}_Y)$ (ignoring any constants) is then minimized

$$\begin{aligned}
&\mathbf{X}^*, \boldsymbol{\theta}_X^*, \boldsymbol{\theta}_Y^*, \mathbf{L}_X^*, \mathbf{L}_Y^*, \sigma_X^*, \sigma_Y^* = \underset{\mathbf{X}, \boldsymbol{\theta}_X, \boldsymbol{\theta}_Y, \mathbf{L}_X, \mathbf{L}_Y, \sigma_X, \sigma_Y}{\arg\min} -\mathcal{L}(\mathbf{X}, \boldsymbol{\theta}_X, \boldsymbol{\theta}_Y, \mathbf{L}_X, \mathbf{L}_Y, \sigma_X, \sigma_Y) \\
&\mathcal{L}(\mathbf{X}, \boldsymbol{\theta}_X, \boldsymbol{\theta}_Y, \mathbf{L}_X, \mathbf{L}_Y, \sigma_X, \sigma_Y) = -\frac{Q}{2} \ln \left|\boldsymbol{\Sigma}_X\right| - \frac{1}{2} \mathrm{tr}\left(\mathrm{vec}(\mathbf{X}_{\setminus 1}) \right. \\
&\qquad \left. \times\ \mathrm{vec}(\mathbf{X}_{\setminus 1})^\top \boldsymbol{\Sigma}_X^{-1}\right) - \frac{D}{2} \ln \left|\boldsymbol{\Sigma}_Y\right| \\
&\qquad - \frac{1}{2} \mathrm{tr}\left(\mathrm{vec}(\mathbf{Y}) \mathrm{vec}(\mathbf{Y})^\top \boldsymbol{\Sigma}_Y^{-1}\right) - \sum_{i,j} \ln \theta_{i,j} \\
&\qquad - 2 \sum_j \ln \sigma_j
\end{aligned} \tag{22}$$

$\mathbf{X}$ can be initialized and a value of Q selected using a principal component analysis, selecting the Q features (principal components) with the largest associated eigenvalues.

2.5. Prediction with observations

To predict future values of the observables $\mathbf{y}$, this study adopts the simple yet effective method of mean-prediction [32]; the mean value in the GP predictive posterior is used. Based on the first-order Markovian dynamics, the model predicts $\mathbf{x}_n$ conditioned on $\mathbf{x}_{n-1}$ using Gaussian conditioning rules, as in the original GPDM

$$\begin{aligned}
&p(\mathbf{x}_n \mid \mathbf{x}_{n-1}, \boldsymbol{\theta}_X, \mathbf{L}_X) = \mathcal{N}\left(\boldsymbol{\mu}_X\left(\mathbf{x}_{n-1}\right), \boldsymbol{\Lambda}_X\left(\mathbf{x}_{n-1}\right)\right) \\
&\boldsymbol{\mu}_X(\mathbf{x}) = \left(\mathbf{L}_X \mathbf{L}_X^\top \otimes \mathbf{k}_{X\setminus T}(\mathbf{x})\right)^\top \boldsymbol{\Sigma}_X^{-1}\ \mathrm{vec}(\mathbf{X}_{\setminus 1}) \\
&\boldsymbol{\Lambda}_X(\mathbf{x}) = k_X(\mathbf{x}, \mathbf{x} | \boldsymbol{\theta}_X) \mathbf{L}_X \mathbf{L}_X^\top - \left(\mathbf{L}_X \mathbf{L}_X^\top \otimes \mathbf{k}_{X\setminus T}(\mathbf{x})\right)^\top \\
&\quad \times\ \boldsymbol{\Sigma}_X^{-1} \left(\mathbf{L}_X \mathbf{L}_X^\top \otimes \mathbf{k}_{X\setminus T}(\mathbf{x})\right)
\end{aligned} \tag{23}$$

in which $\mathbf{k}_{X\setminus T}(\mathbf{x}) = [k_X(\mathbf{x}, \mathbf{x}_1 | \boldsymbol{\theta}_X), \ldots, k_X(\mathbf{x}, \mathbf{x}_{T-1} | \boldsymbol{\theta}_X)]^\top$. To be clear, the forecasting consists of iteratively using the mean estimate $\mathbf{x}_{n-1} = \boldsymbol{\mu}_X(\mathbf{x}_{n-2})$ to estimate $\mathbf{x}_n$ via Eq. (23). Similarly, $\mathbf{y}_n$ is inferred by the mean of the following posterior predictive distribution using the mean estimate of $\mathbf{x}_n$

$$\begin{aligned}
&p(\mathbf{y}_n \mid \mathbf{x}_n, \boldsymbol{\theta}_Y, \mathbf{L}_Y) = \mathcal{N}\left(\boldsymbol{\mu}_Y\left(\mathbf{x}_n\right), \boldsymbol{\Lambda}_Y\left(\mathbf{x}_n\right)\right), \\
&\boldsymbol{\mu}_Y(\mathbf{x}) = \left(\mathbf{L}_Y \mathbf{L}_Y^\top \otimes \mathbf{k}_Y(\mathbf{x})\right)^\top \boldsymbol{\Sigma}_Y^{-1}\ \mathrm{vec}(\mathbf{Y}), \\
&\boldsymbol{\Lambda}_Y(\mathbf{x}) = k_Y(\mathbf{x}, \mathbf{x} | \boldsymbol{\theta}_Y) \mathbf{L}_Y \mathbf{L}_Y^\top - \left(\mathbf{L}_Y \mathbf{L}_Y^\top \otimes \mathbf{k}_Y(\mathbf{x})\right)^\top \boldsymbol{\Sigma}_Y^{-1} \left(\mathbf{L}_Y \mathbf{L}_Y^\top \otimes \mathbf{k}_Y(\mathbf{x})\right),
\end{aligned} \tag{24}$$

in which $\mathbf{k}_Y(\mathbf{x}) = [k_Y(\mathbf{x}, \mathbf{x}_1 | \boldsymbol{\theta}_Y), \ldots, k_Y(\mathbf{x}, \mathbf{x}_T | \boldsymbol{\theta}_Y)]^\top$. The entire process is illustrated in Fig. 2.

2.6. Datasets

E-GPDM is assessed on two NASA Ames Prognostics Center of Excellence Battery Datasets from Saha and Goebel [36] and the Oxford Battery Degradation Dataset of Birkl [37]. Both datasets record the charging and discharging performance of Li-ion batteries. The Oxford data set contains a significant level of noise, and also shows evidence of the regeneration phenomenon in Li-ion batteries. It is therefore considered a gold-standard in degradation forecasting.

Data pertaining to seven batteries from two different groups in the NASA data set were used. The batteries are labeled B0005, B0006, B0007 (group 1), and B0029, B0030, B0031, B0032 (group 2). All batteries were charged using a constant current of 1.5 A until the voltage reached the limit 4.2 V, following which a constant voltage was applied until the current reached 20 mA. The operating temperature for the group 1 tests was 24 °C, while the temperature for the group 2 tests was 43 °C. Batteries in group 1 were discharged at a constant current of 2 A until the voltage fell to 2.7 V, 2.5 V and 2.2 V for B0005, B0006 and B0007, respectively. Batteries in group 2 were discharged at a constant current of 4 A to 2.0 V, 2.2 V, 2.5 V and 2.7 V for B0029, B0030, B0031 and B0032, respectively. For the group 1 batteries, 168 charge–discharge cycles were performed, whereas for the group 2 batteries, 39 cycles were performed. The data in each case consists of temperature, voltage, current and capacity measurements. Although the data also contains impedance measurements, they were not used in this study due to the extremely high levels of noise they contain.

The Oxford dataset contains measurements from Li-ion pouch cells, all tested in a thermal chamber at 40 °C. The cells were exposed to a constant-current and constant-voltage charging profile, followed by a drive cycle discharging profile derived from the urban Artemis profile, with measurements taken every 100 cycles. The batteries are labeled Ox1, Ox2, Ox3 and Ox4. Again, the data contains measurements of temperature, voltage, current and capacity on each cycle.

2.7. Data preprocessing

The raw data was pre-processed as follows:

1. *Cut off*. Voltage and temperature sequences during the discharge cycles were truncated after the voltage reached the threshold (cut-off) value as set in the experiment.
2. *Interpolation*. For the NASA dataset, cubic splines were used to obtain voltage and temperature data on a fixed grid, with 200 measurements for each cycle. The Oxford dataset did not require interpolation since there are an equal number of measurements for each cycle.
3. *Rescaling*. The capacity was rescaled to define the SOH as described earlier, while the temperature, voltage and other attributes were scaled using a min–max normalization.

Fig. 3 shows the SOH trajectories for the batteries considered.

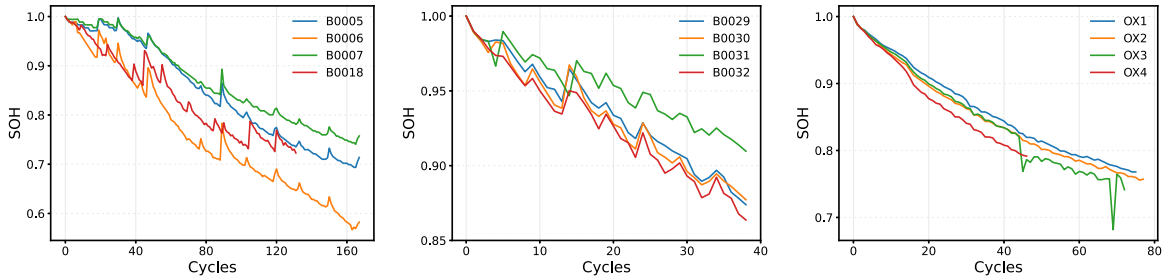


Fig. 3. SOH trajectories for 3 groups of batteries. Left: B0005, B0006, B0007, and B0018 from the NASA dataset. Middle: B0029, B0030, B0031 and B0032 from the NASA dataset. Right: Ox1, Ox2, Ox3 and Ox4 from the Oxford dataset.

3. Results and discussion

3.1. Model training and evaluation

In this section, the results of E-GPDM are compared to those from a number of baseline methods: a GP model, a CNN, stacked long short term memory (LSTM) networks, stacked gated recurrent unit (GRU) networks, and support vector regression (SVR). After testing stacked LSTMs and GRUs with up to 10 recurrent layers, 5 dense layers and dropout layers, it was found that a bi-LSTM and a single GRU worked best, and therefore only results for these architectures are presented. All models were evaluated with 5 different random seeds, and the average errors were calculated.

The GP models were tested with combinations of 6 different kernel functions, including the linear, squared-exponential (RBF), polynomial, rational quadratic, and Matérn kernels. Amongst these, a combination of Matérn 3 and Matérn 5 gave the best performance for the GP model, while a mixed RBF and linear kernel gave the best performance for E-GPDM. The latent variables $\mathbf{x}_n$ are embeddings of the data $\mathbf{y}_n$. To initialize $\mathbf{X}$, a PCA on $\mathbf{Y}$ was conducted, and all principal components were used to define the latent variables $\mathbf{x}$. Stochastic gradient descent (SGD) was used for the MLE solution. The SVR algorithm was implemented using an $\epsilon-$insensitive error and a Gaussian kernel.

The bi-LSTM layers used a tanh activation and a sigmoid recurrent activation, and were followed by a dense layer of size 64 neurons and a final output layer. Each LSTM layer was followed by a dropout layer of rate 0.2. The GRU used a single GRU layer of size 64 followed by two dense layers of sizes 64 and 32, followed by the output layer. ReLU activation functions were used. Large networks (high model complexity) led to deteriorating performance due to the exponential growth in the number of parameters and, therefore, the difficulty in minimizing the loss function with the small data sets available. For this reason, regression versions of large networks such as VGG [38] and Faster-r-CNN [39] are not used for battery degradation forecasting and similar problems.

Various CNN architectures were investigated and the best performing network used two CNN layers followed by two dense layers of sizes 64 and 32 neurons, along with an output layer. The CNN layers employed padding, strides of 1, and 16 filters each. Pooling was not used since it was found to deteriorate performance. A dropout layer of ratio 0.25 was used after each CNN layer and ReLu activation functions were used in the dense layers. Again, larger architectures of up to 10 CNN layers and 5 dense layers led to worse performance. Both of the DNNs were trained using the ADAM algorithm [40], with a learning rate of 0.5×10^{-4} and a decay rate of 0.5×10^{-5}.

To assess the accuracy, the root mean square error (RMSE) is employed

$$\mathrm{RMSE} = \sqrt{\frac{1}{N_t}\sum_n (y(n) - y_t(n))^2} \tag{25}$$

in which $y(n)$ is the prediction, $y_t(n)$ is the test value at cycle n, and N_t is the number of test points.The RMSE is used over the MAE since it is a more robust measure of error. The RMSE upper bounds the MAE and is more sensitive the presence of outliers by virtue of the squaring of the residuals. In the next section, results without the use of attributes other than the SOH and battery label are presented.

Table 1
RMSE values relating to the SOH predictions using all methods on the NASA data sets. % Train refers to the percentage of the total data used for training.

B0005							
% Train	E-GPDM	E-GPDM[a]	GP	CNN	**bi-LSTM**	GRU	SVR
33	**0.0147**	0.0588	0.0630	0.0313	0.0376	0.0330	0.0168
50	**0.0095**	0.0227	0.0324	0.0289	0.0312	0.0215	0.0173
70	0.0113	0.0640	0.0278	0.0350	**0.0101**	0.0265	0.0236
B0006							
33	**0.0189**	0.0321	0.0499	0.0330	0.0702	0.0350	0.0414
50	0.0378	0.0408	0.0322	0.0319	**0.0211**	0.0247	0.0335
70	**0.0188**	0.0430	0.0353	0.0311	0.0287	0.0217	0.0194
B0007							
33	**0.0184**	0.0800	0.0774	0.0550	0.0479	0.0281	0.0198
50	**0.0113**	0.0558	0.0399	0.0441	0.0292	0.0289	0.0151
70	**0.0128**	0.0314	0.0473	0.0228	0.0298	0.0213	0.0178
B0029							
33	**0.0172**	0.0533	0.0215	0.0207	0.0184	0.0189	0.0315
50	0.0226	0.0415	0.0280	**0.0175**	0.0195	0.0242	0.0340
70	**0.0145**	0.0299	0.0194	0.0190	0.0154	0.0223	0.0269
B0032							
33	**0.0124**	0.0225	0.0201	0.0186	0.0144	0.0168	0.0209
50	**0.0203**	0.0216	**0.0205**	0.0229	**0.0201**	0.0260	0.0253
70	**0.0112**	0.0145	0.0132	0.0154	0.0132	0.0212	0.0130

[a] Without transfer learning.

3.2. Comparison to baseline methods without additional attributes

In the first set of experiments, the data points are considered to only contain the cycle number n, battery label m and SOH. In the supervised models, the inputs and outputs are $\mathbf{z}_n^m = [n, m]^\top$ and $\mathbf{y}_n^m = \mathrm{SOH}^m(n)$, respectively. For the unsupervised E-GPDM, there are only attributes, so that the data takes the form (3), i.e.

$$\mathbf{y}_n = [n, m, \mathrm{SOH}^m(n)]^\top$$

For each group of batteries, all of the data related to the group (e.g., B0029-B0032) was used, except that for battery under consideration (e.g., B0029), for which 30, 50 or 70% was used.

3.3. Predictions on the NASA data set

Table 1 shows the RMSE values for predictions related to B0005, B0006, B0007, B0029 and B0032. The results from E-GPDM and three of the other methods are displayed in Fig. 4 for B0005 with 30%, 50% and 70% training ratios (percentage of the total data used for training).

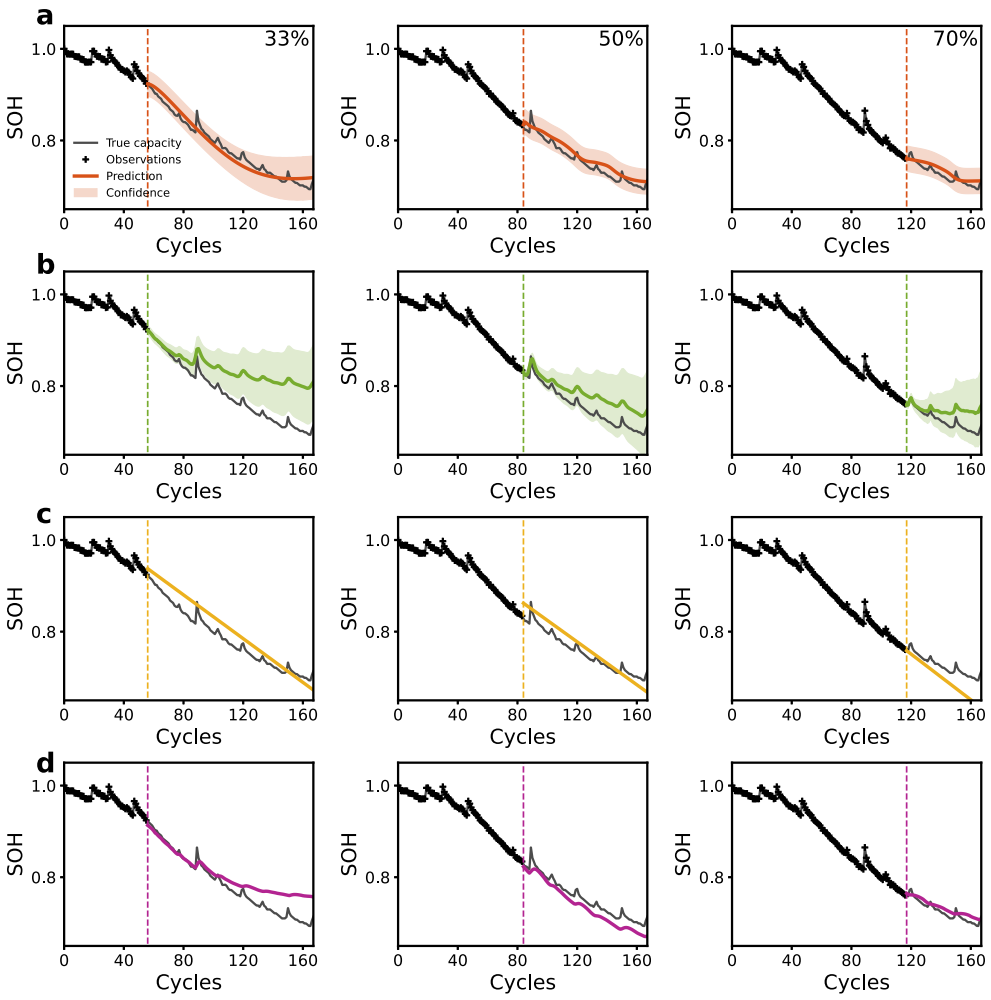


Fig. 4. SOH predictions for battery B0005. (a) E-GPDM (with transfer learning), (b) GP, (c) CNN, (d) bi-LSTM. The figures from left to right correspond to 33%, 50% and 70% training ratios.

The GP models include uncertainty estimates, namely 95% confidence intervals shown by the shaded regions in Fig. 4 and defined by

$$\mu(n) - 1.96\sigma(n) \leq \mathrm{SOH}(n) \leq \mu(n) + 1.96\sigma(n) \tag{12}$$

in which $\mu(n)$ and $\sigma(n)$ are the mean value and predictive variance at cycle n.

The RMSE values without transfer learning for E-GPDM are also shown in Table 1, from which it is clear that transfer learning significantly improves the accuracy. In the case of B0007, the improvement is 77%, 79% and 59% for 30, 50 and 70% training ratios, respectively. Similar dramatic improvements are seen for B0005, B0006 and B0029, while the improvements on B0032 are less significant at 50% and 70% training ratios. To ensure fair comparisons, transfer learning was used for all methods to generate the results in Table 1, and similar improvements were seen for all methods. As seen in Table 1, for the group 1 batteries, E-GPDM (with transfer learning) yields the lowest RMSE in 7 out of 9 cases. Consistent with these results, the predictions for group 2 batteries (B0029 and B0032) are superior with E-GPDM in almost all (5 out of 6) cases. Notably, the earliest prediction (33% training ratio) is most accurate with E-GPDM for all of the 5 batteries.

The direct GP model (Fig. 4(b)) with transfer learning yields a poor estimate of the rate of decline in SOH, even when presented with 70% of the data. The CNN and GRU perform reasonably well, except at a 70% training ratio, while the bi-LSTM performs well at both 50% and 70% training ratios. In fact, the bi-LSTM is the most accurate at a 70% training ratio (10% lower RMSE than E-GPDM). SVR also performs well at 30% and 50% ratios, but has low accuracy for the 70% ratio. From Table 1, it can be seen that E-GPDM is superior to all of the other methods for 33% and 50% training ratios on the B0005 battery. The overall trend of the SOH is captured well with E-GPDM (Fig. 4(a)), and the EOL prediction is close to the true value.

Predictions related to the B0007 battery are shown in Fig. 5. The direct GP model performance is again poor. The CNN and bi-LSTM (Fig. 5(c)&(d)) essentially fail in this case, while the GRU performance improves compared to the B0005 data set, as can be seen in Table 1. E-GPDM (Fig. 5(a)) is clearly superior to all of the other methods, achieving high accuracy for all training point numbers and again capturing the overall trend of the SOH well. Table 1 reveals that the performance of SVR is also impressive for B0007. For B0006, the E-GPDM predictions are significantly more accurate than any of the other methods at a 30% training ratio. SVR is accurate only with a 70% training ratio. At a 50% training ratio, the bi-LSTM yields the lowest RMSE, with the GRU yielding the second best performance.

For B0029, while E-GPDM yields the lowest RMSE at 30% and 70% training ratios, the differences between E-GPDM and other methods not as pronounced, as seen in Table 1 and Fig. 6. In particular, the bi-LSTM performance is close to that of E-GPDM. The CNN yields marginally the best performance at the 50% training ratio, and the GRU closely matches the performance of E-GPDM and the bi-LSTM at the 30% ratio. For the B0032 battery, E-GPDM yields the best performance in all cases, although at a 50% training ratio, the performances of E-GPDM, the GP and the bi-LSTM are indistinguishable. In this case, the GP does rather

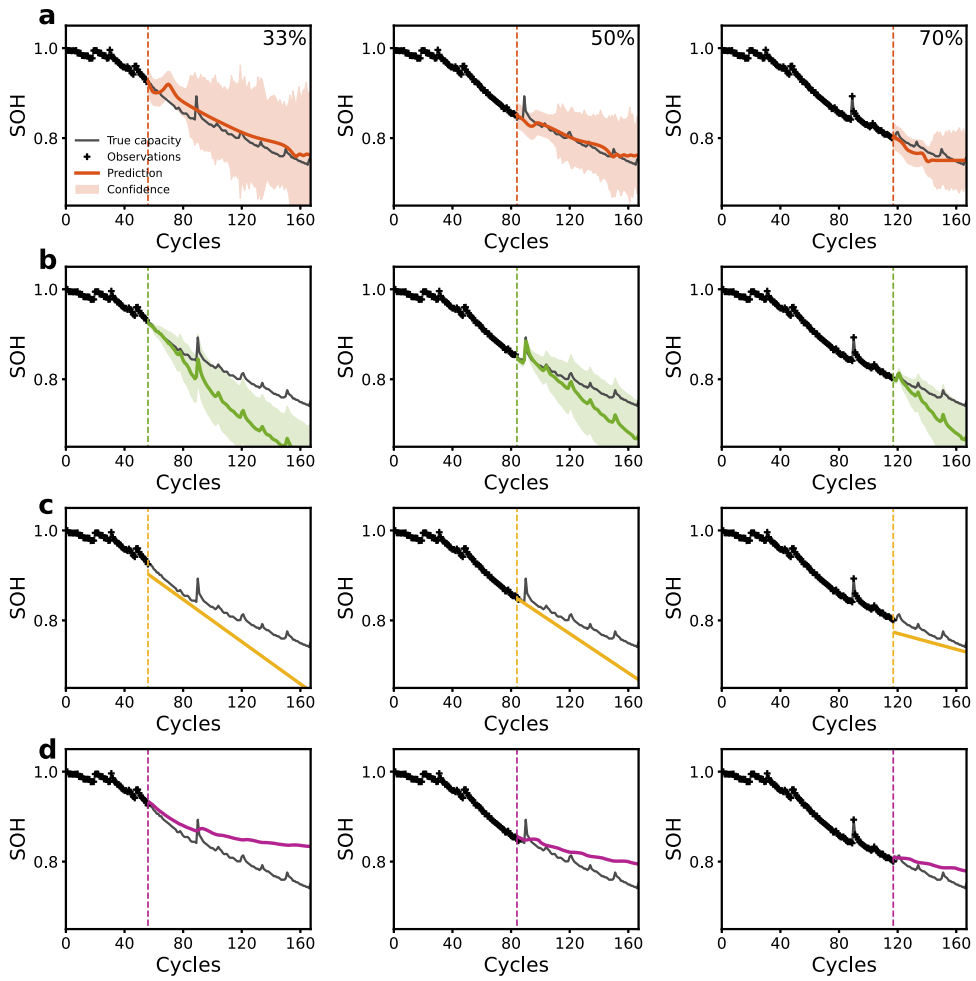


Fig. 5. SOH predictions for battery B0007. (a) E-GPDM (with transfer learning), (b) GP, (c) CNN, (d) bi-LSTM. The figures from left to right correspond to 33%, 50% and 70% training ratios.

well, especially at 50% and 70% training ratios. Furthermore, the bi-LSTM again closely matches the performance of E-GPDM for all ratios. The comparatively good performance of bi-LSTM on the group 1 and group 2 batteries is not surprising, since it is designed for sequence problems. What is not obvious is why the GRU performance is relatively poor, given that it is also designed for such problems.

3.4. Predictions on the Oxford data set

Table 2 shows the equivalent RMSE values for the Oxford battery data set, with transfer learning. Again, E-GPDM is generally superior (9 out of 12 cases), with the bi-LSTM and GP occasionally providing the best result (2 out of 12 cases for the GP and 1 out of 12 cases for the bi-LSTM). As with the NASA data set, the bi-LSTM does well on the short-term predictions corresponding to a 70% training ratio, with the GP now also performing well on Ox2 and Ox3. The early predictions (33% training ratio) are still, however, superior with E-GPDM, which is respectively 31%, 17%, 44% and 50% more accurate than the next most accurate method (bi-LSTM for Ox1-3 and SVR for Ox4). The equivalent numbers for a 50% training ratio are 29%, 33%, 20% and 54%. Fig. 7 shows the predictions corresponding to Ox3, clearly demonstrating the superiority of E-GPDM at the 33% training ratio. The excellent performance of bi-LSTM and the GP at a 70% ratio is also evident.

Table 2
RMSE values relating to the SOH predictions using all methods on the Oxford data set. % Train refers to the percentage of the total data used for training.

Ox1							
% Train	E-GPDM	E-GPDM[a]	GP	CNN	**bi-LSTM**	GRU	SVR
33	**0.0093**	0.0226	0.0668	0.0199	0.0135	0.0197	0.0288
50	**0.0135**	0.0244	0.0207	0.0226	0.0191	0.0219	0.0308
70	0.0147	0.0263	0.0261	0.0448	**0.0105**	0.0237	0.0327
Ox2							
33	**0.0044**	0.0769	0.0092	0.0330	0.0053	0.0209	0.0202
50	**0.0058**	0.0325	0.0087	0.0276	0.0118	0.0240	0.0220
70	0.0042	0.0082	**0.0037**	0.0316	0.0101	0.0269	0.0234
Ox3							
33	**0.0086**	0.0364	0.0134	0.0207	0.0117	0.0140	0.0186
50	**0.0099**	0.0118	0.0123	0.0236	0.0158	0.0118	0.0182
70	0.0101	0.0089	**0.0061**	0.0096	0.0078	0.0109	0.0183
Ox4							
33	**0.0101**	0.0189	0.0411	0.0327	0.0299	0.0424	0.0204
50	**0.0086**	0.0360	0.0345	0.0337	0.0185	0.0330	0.0188
70	**0.0073**	0.0133	0.0219	0.0394	0.0175	0.0236	0.0233

[a] Without transfer learning.

3.5. Predictions with additional attributes

The previous experiments evaluated the predictive power of E-GPDM and the other methods without attributes other than the SOH.

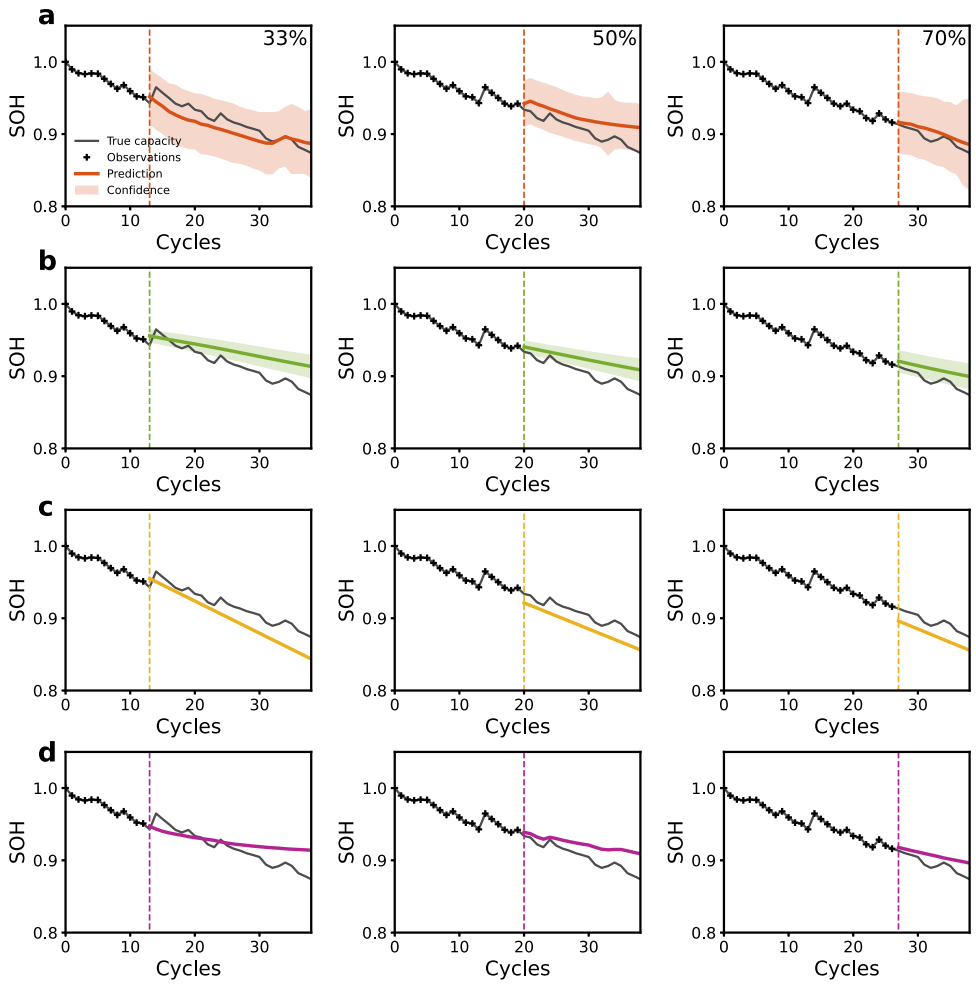


Fig. 6. SOH predictions for battery B0029. (a) E-GPDM (with transfer learning), (b) GP, (c) CNN, (d) bi-LSTM. The figures from left to right correspond to 33%, 50% and 70% training ratios.

E-GPDM was superior (lower RMSE) when the number of training points was low, corresponding to early predictions. In other cases (50% and 70% training ratios) it yielded the lowest RMSE in 67% (12 out of 18) cases. Overall, E-GPDM was more accurate or as accurate in 19 out of 27 (70%) of the cases. Although these results are impressive, the main advantage of E-GPDM is that it allows for the incorporation of an arbitrary number of attributes, which, in theory, can improve the results. In this section, the following attributes are now incorporated for each battery m

1. Attribute 1. Temperature at the midpoint of each discharge cycle, $T_m(n)$
2. Attribute 2. Voltage at the midpoint of each discharge cycle, $V_m(n)$
3. Attribute 3. Energy delivered by the battery (unnormalized), given by $I_m(n) = \int_t V_m(n)dt$, in which $V_m(n)$ is the voltage curve (a function of time t) for cycle n. The integral was evaluated using a trapezoidal rule

Attributes 1 and 2 will clearly be correlated with the SOH, and are two of the most frequently used attributes. The (unnormalized) energy delivered by the battery will exhibit a decline as the SOH declines, so is again correlated with the SOH. The choice of these attributes is based on their similarity to those used previously in the literature as well as the extent of the information contained in the three data sets under consideration. More attributes can be added and optimized for best performance. Explicitly, the data now takes the form (3)

$$\mathbf{y}_n = [n, m, \mathrm{SOH}(n), T_m(n), V_m(n), I_m(n)]^\top \tag{26}$$

in which $a_1(n) = \mathrm{SOH}(n)$, $a_2(n) = T_m(n)$, $a_3(n) = V_m(n)$ and $a_4(n) = I_m(n)$.

The other methods of the previous section cannot be used in this case, for the obvious reason that the attributes will not be known for future cycles. Using known attribute values from the given data set to perform a multi-cycle lookahead would not lead to a practical algorithm. In comparison, E-GPDM propagates the dynamic hidden variable $\mathbf{x}_n$, from which the observable attributes $\mathbf{y}_n$ (including the SOH) can be predicted for an arbitrary number of future cycles. It is possible to compare E-GPDM with a GPLVM [41], which is able to reconstruct a complete data set from partially known features or attributes, a problem that is encountered in many areas, e.g., human motion tracking. This property of GPLVM can be exploited to reconstruct the complete SOH curve from 33%, 50% and 70% of the data. The E-GPDM latent space embedding is again initialized using all principal components from a PCA. A combination of a linear and RBF kernel was used, together with the conjugate gradient method to solve the maximum likelihood problem for both E-GPDM and GPLVM.

Table 3 shows the RMSE values on all three data sets, which are to be compared with the values in Tables 1 and 2. For convenience, the E-GPDM results with transfer learning from the latter tables are reproduced in Table 3. Also reproduced is the best result for each battery (from Tables 1 and 2). In 23 out of 27 cases (85%), the RMSE values are lower when additional attributes are included. In three of the other cases (B0005/50%, B0032/33% and Ox3/50%) the values are virtually indistinguishable. In the case of Ox2/33%, the error with features is 36% higher, although the result is still very accurate. In particular, there is generally a vast decrease in the RMSE at 50% and 70% training

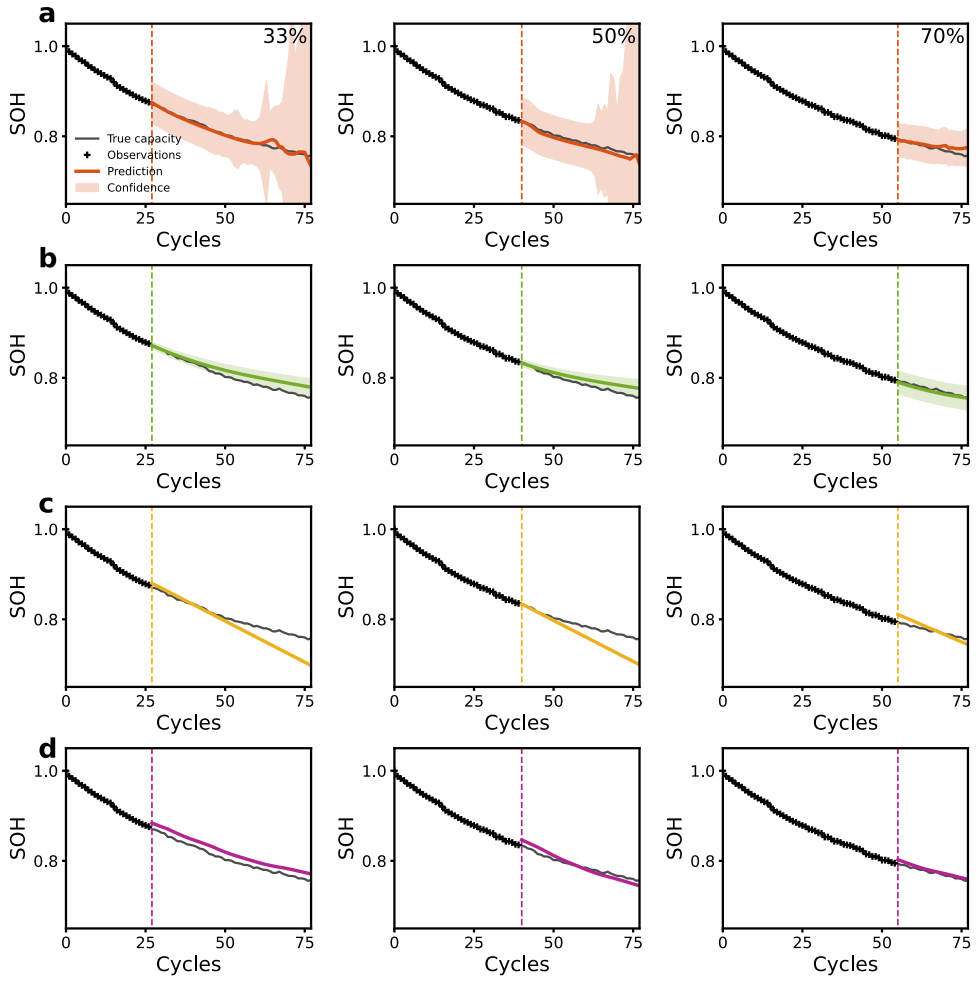


Fig. 7. SOH predictions for the Ox3 battery. (a) E-GPDM (with transfer learning), (b) GP, (c) CNN, (d) bi-LSTM. The figures from left to right correspond to 33%, 50% and 70% training ratios.

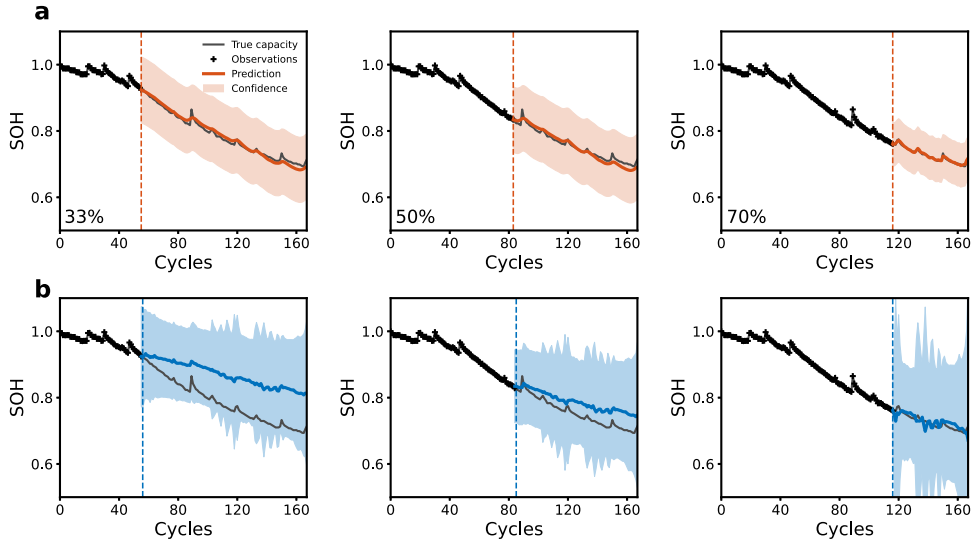


Fig. 8. SOH predictions for B0005 when employing additional features. (a) E-GPDM (with transfer learning), (b) GPLVM. The figures from left to right correspond to 33%, 50% and 70% training ratios.

ratios. The decrease for a 70% training ratio compared to the best result in Tables 1 and 2 is: 71% (B0005), 65% (B0006), 29% (B0007), 44% (B0029), 6% (B0032), 80% (Ox1), 38% (Ox2), 46% (Ox3) and 75% (Ox4). For 50%, the improvement is: negligible (B0005), 34% (B0006), negligible (B0007), 54% (B0029), 38% (B0032), 63% (Ox1), 40% (Ox2), 12% (Ox3) and 58% (Ox4).

Figs. 8 and 9 show the predictions for B0005 and B0006 for different training ratios using E-GPDM (with transfer learning) and GPLVM. It

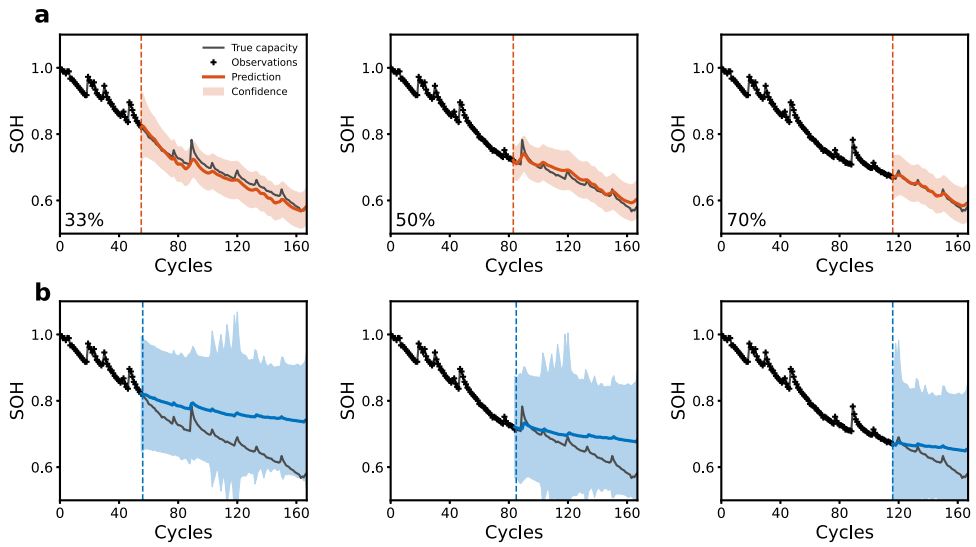


Fig. 9. SOH predictions for B0006 when employing additional features. (a) E-GPDM (with transfer learning), (b) GPLVM. The figures from left to right correspond to 33%, 50% and 70% training ratios.

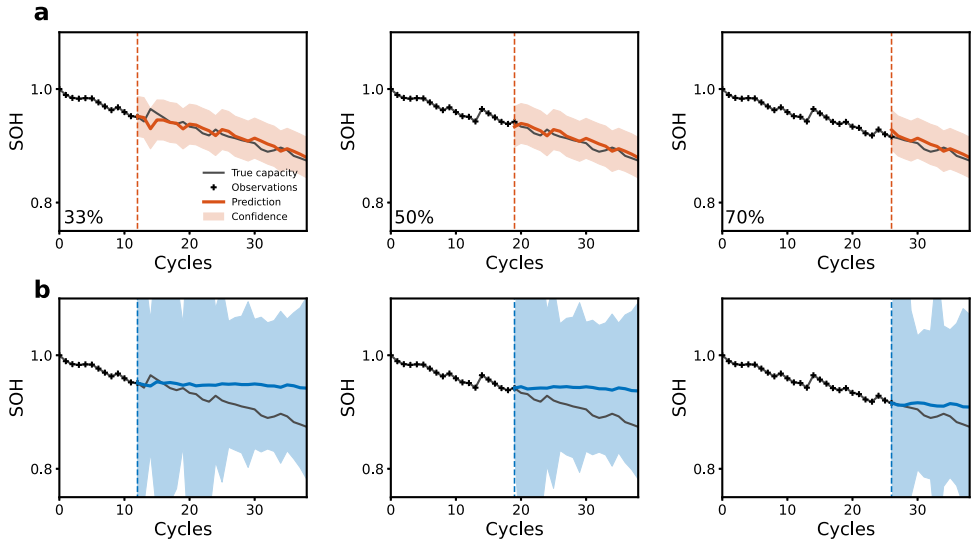


Fig. 10. SOH predictions for B0029 when employing additional features. (a) E-GPDM (with transfer learning), (b) GPLVM. The figures from left to right correspond to 33%, 50% and 70% training ratios.

is clear that E-GPDM tracks the overall trend of the curve as well as the fluctuations with good accuracy. GPLVM, however, fails both qualitatively and quantitatively. Fig. 8 should be compared with Fig. 4 (without additional attributes), from which the improvement in performance is obvious. It is recalled that for B0005 at a 70% training ratio, the RMSE was 0.0113 without features, higher than the bi-LSTM value of 0.0101. The RMSE for E-GPDM with features is now 0.0029, far lower than that for the bi-LSTM. For B0006, the RMSE for E-GPDM was 0.0378 at 50%, while that for the bi-LSTM was 0.0211. The RMSE for E-GPDM with features is 0.0140, again lower than that for the bi-LSTM.

Fig. 10, which can be compared to Fig. 6, shows the predictions for B0029. Again, the accuracy improves markedly. In this case, the fluctuations are not as well captured, since they appear not to follow any trend, in comparison to those in the B0005-B0007 datasets, which are caused by a regeneration phenomenon. Finally, in Fig. 11 (to be compared with Fig. 7) the predictions are shown for Ox3. In this case there is a less significant improvement since the result without features was already very accurate. What is noticeable, however, is the improvement near the end of the cycling at all training ratios, although at the start of the prediction interval the results are slightly less accurate.

3.6. Direct comparisons with other methods in the literature

A vast number of methods in the literature use features. Tagade et al. [42] employed a deep GP with features on the B0006 data set to obtain mean absolute error (MAE) values of ca. 0.009 and 0.008 (Fig. 3 of their paper) for 100 training points (60% training ratio) and 120 training points (71% training ratio), respectively. E-GPDM yields MAE values of 0.0096 and 0.0045 for 50% and 70% training ratios. Yang et al. [10] employed a supervised GP model with four features. The RMSE values on the B0006 and B0007 data sets using 80 training points (ca. 50% training ratio) were 0.0149 and 0.0078, respectively. These values can be compared to 0.0140 and 0.0111, respectively, using E-GPDM. Yang et al. however, discarded what they considered to be outliers data, without specifying the number of these points. Chen et al. [14] used features with an LSTM. They discarded 60 unspecified outliers of the 168 B0006 data points and were able to

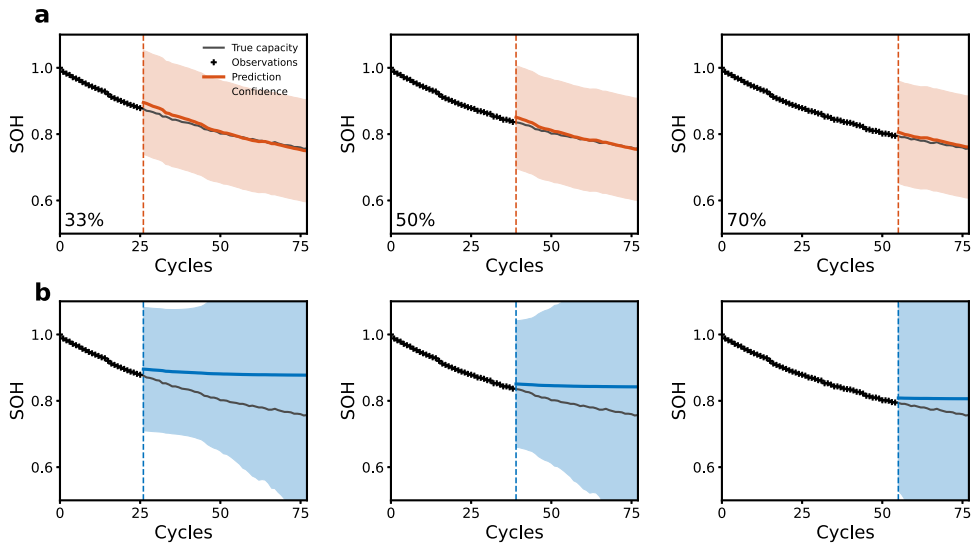


Fig. 11. SOH predictions for Ox3. (a) E-GPDM (with transfer learning), (b) GPLVM. The figures from left to right correspond to 33%, 50% and 70% training ratios.

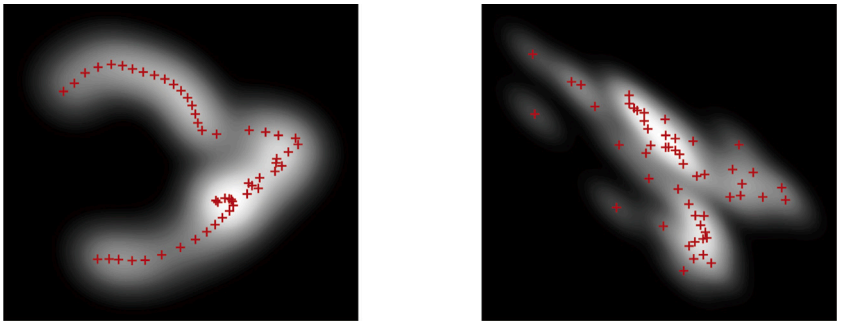


Fig. 12. 2-d latent trajectories for B0006 using E-GPDM (left) and GPLVM (right). The axes correspond to the two latent coordinates and the intensity is proportional to the predictive variance (lighter regions correspond to smaller variance).

obtain RMSE values between 0.0025 and 0.0012 for B0006, depending on the proportion of data used for training.

In this study, no points were discard as outliers, which make comparisons to the latter two studies difficult. The use of features in [10,14,42] and other similar approaches can provide a virtually arbitrary level of accuracy (with virtually any method) because the features follow the same pattern as the SOH, including the fluctuations. The predictions, however, are invalid for anything other than the next cycle because the feature values are not known in the future, i.e., all of the values past the Tth cycle, where T is the training point number. The only meaningful error would be the cumulative error from a sequence of one-step predictions, which is not what is calculated in these studies, and which would be far higher than the quoted errors.

3.7. Latent trajectories

One important component of the proposed method is the underlying dynamics captured by Eqs. (4). If this component is removed, the proposed method is reduced to the GPLVM. Here, the effects of incorporating the dynamical process are demonstrated by comparing 2-d latent variable reconstructions from GPLVM and E-GPDM in the case of B0006 (with attributes). The 2-d latent trajectories $\{\mathbf{x}_n\}$ from the two methods are shown in Fig. 12, with the shading intensity proportional to the predictive variance (brighter regions indicate smaller variance). Fig. 12 (left) illustrates that E-GPDM produces a much smoother configuration of latent positions than GPLVM (right), which contains a high degree of scatter. This illustrates that the evolution underlying the state of the battery is much better captured by E-GPDM, as a consequence of including the dynamics. The latent embeddings provide a good representation of the state of the system.

4. Conclusions

Accurate predictions of the SOH of a Li-ion battery and its EOL are crucial for the management and control of stacks in electric vehicles. One of the issues that has hampered the development of algorithms for early prediction of the EOL relates to the incorporation of prior knowledge from other batteries, while also using data that is specific to the battery under consideration. Another issue is in how to incorporate other attributes (or features) of the battery performance in order to improve predictions, given that they are not known for future cycles. In this study, a method that overcomes both of these issues in an efficient manner was presented. This method (E-GPDM) leads to an algorithm with a very low computational burden compared to DNNs, and the added benefit of providing a confidence interval.

The results on three data sets show that without additional attributes, E-GPDM is more accurate in the majority of cases. With the inclusion of additional attributes such as temperature, the accuracy is further increased except in a small number of cases. Key to these results is the incorporation of additional data through transfer learning. Without transfer learning it is difficult to capture an overall SOH trend early in the battery life when the trajectory is complex.

Future research will focus on two aspects: better incorporation of prior knowledge and data for more complex SOH trajectories, and the performance of E-GPDM with more features. The volume and nature (whether under the same conditions or not) of the data that

Table 3
RMSE values relating to the SOH predictions using E-GPDM and GPLVM on all data sets when employing additional features. % Train refers to the percentage of the total data used for training.

B0005					
% Train	Best[b]	E-GPDM[c]	E-GPDM	E-GPDM[a]	GPLVM [41]
33	0.0147	0.0147	**0.0091**	0.1682	0.0884
50	**0.0095**	**0.0095**	0.0101	0.0321	0.0348
70	0.0101	0.0113	**0.0029**	0.0224	0.0141
B0006					
33	0.0189	0.0189	**0.0185**	0.0410	0.0944
50	0.0211	0.0378	**0.0140**	0.0325	0.0535
70	0.0188	0.0188	**0.0065**	0.0166	0.0432
B0007					
33	0.0184	0.0184	**0.0137**	0.0824	0.0343
50	0.0113	0.0113	**0.0111**	0.0428	0.0329
70	0.0128	0.0128	**0.0091**	0.0284	0.0292
B0029					
33	0.0172	0.0172	**0.0101**	0.0424	0.0378
50	0.0175	0.0226	**0.0080**	0.0436	0.0389
70	0.0145	0.0145	**0.0081**	0.0214	0.0203
B0032					
33	**0.0124**	**0.0124**	0.0128	0.0284	0.0391
50	0.0203	0.0203	**0.0125**	0.0342	0.0414
70	0.0112	0.0112	**0.0105**	0.0223	0.0193
Ox1					
33	0.0093	0.0093	**0.0050**	0.0158	0.0603
50	0.0135	0.0135	**0.0050**	0.0061	0.0409
70	0.0105	0.0147	**0.0021**	0.0033	0.0218
Ox2					
33	**0.0044**	**0.0044**	0.0060	0.0115	0.0569
50	0.0058	0.0058	**0.0035**	0.0132	0.0606
70	0.0037	0.0042	**0.0023**	0.0059	0.0186
Ox3					
33	**0.0086**	**0.0086**	0.0089	0.0439	0.0630
50	0.0099	0.0099	**0.0087**	0.0090	0.0409
70	0.0061	0.0101	**0.0033**	0.0039	0.0213
Ox4					
33	0.0101	0.0101	**0.0071**	0.0152	0.0300
50	0.0086	0.0086	**0.0036**	0.0160	0.0289
70	0.0073	0.0073	**0.0018**	0.0058	0.0078

[a] Without transfer learning.
[b] The best result from Tables 1 and 2 for each battery.
[c] The results for E-GPDM without features in Tables 1 and 2.

best provides accurate learning of the SOH trajectory requires a more thorough investigation. Additionally, characterization of the number and types of features for optimal accuracy requires more extensive data sets combined with rigorous ablation studies. It is suspected that the optimal data and features will be specific to the type of battery. Both of these extensions are beyond the scope of this study, which serves as a proof of concept for E-GPDM combined with transfer learning.

CRediT authorship contribution statement

W.W. Xing: Conceptualization, Methodology, Software, Writing – review & editing. **Z. Zhang:** Software, Writing – review & editing. **A.A. Shah:** Conceptualization, Methodology, Writing – original draft, 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.

Data availability

All source codes for the work contained in this paper are available on Github at https://github.com/PericlesHat/enhanced-GPDM.

References

[1] Li AG, West AC, Preindl M. Towards unified machine learning characterization of lithium-ion battery degradation across multiple levels: A critical review. Appl Energy 2022;316:119030.
[2] McBrayer JD, Rodrigues MTF, Schulze MC, Abraham DP, Apblett CA, Bloom I, Carroll GM, Colclasure AM, Fang C, Harrison KL, et al. Calendar aging of silicon-containing batteries. Nat Energy 2021;6(9):866–72.
[3] Cheng M, Zhang X, Ran A, Wei G, Sun H. Optimal dispatch approach for second-life batteries considering degradation with online soh estimation. Renew Sustain Energy Rev 2023;173:113053.
[4] Rauf H, Khalid M, Arshad N. Machine learning in state of health and remaining useful life estimation: Theoretical and technological development in battery degradation modelling. Renew Sustain Energy Rev 2022;156:111903.
[5] Naseri F, Gil S, Barbu C, Cetkin E, Yarimca G, Jensen A, Larsen P, Gomes C. Digital twin of electric vehicle battery systems: Comprehensive review of the use cases, requirements, and platforms. Renew Sustain Energy Rev 2023;179:113280.
[6] Xiong R, Pan Y, Shen W, Li H, Sun F. Lithium-ion battery aging mechanisms and diagnosis method for automotive applications: Recent advances and perspectives. Renew Sustain Energy Rev 2020;131:110048.
[7] Tian H, Qin P, Li K, Zhao Z. A review of the state of health for lithium-ion batteries: Research status and suggestions. J Clean Prod 2020;261:120813.
[8] Berecibar M. Machine-learning techniques used to accurately predict battery life. Nature 2019;568(7752):325–7.
[9] Richardson R, Osborne M, Howey D. Gaussian process regression for forecasting battery state of health. J Power Sources 2017;357:209–19.
[10] Yang D, Zhang X, Pan R, Wang Y, Chen Z. A novel gaussian process regression model for state-of-health estimation of lithium-ion battery using charging curve. J Power Sources 2018;384:387–95.
[11] Wang Z, Song C, Zhang L, Zhao Y, Liu P, Dorrell DG. A data-driven method for battery charging capacity abnormality diagnosis in electric vehicle applications. IEEE Trans Transp Electrif 2021;1.
[12] Hong J, Lee D, Jeong ER, Yi Y. Towards the swift prediction of the remaining useful life of lithium-ion batteries with end-to-end deep learning. Appl Energy 2020;278:115646.
[13] Hsu CW, Xiong R, Chen NY, Li J, Tsou NT. Deep neural network battery life and voltage prediction by using data of one cycle only. Appl Energy 2022;306:118134.
[14] Chen JC, Chen TL, Liu WJ, Cheng C, Li MG. Combining empirical mode decomposition and deep recurrent neural networks for predictive maintenance of lithium-ion battery. Adv Eng Inform 2021;50:101405.
[15] Couture J, Lin X. Image-and health indicator-based transfer learning hybridization for battery rul prediction. Eng Appl Artif Intell 2022;114:105120.
[16] Shu X, Li G, Zhang Y, Shen S, Chen Z, Liu Y. Stage of charge estimation of lithium-ion battery packs based on improved cubature kalman filter with long short-term memory model. IEEE Trans Transp Electrif 2020;7(3):1271–84.
[17] Li W, Sengupta N, Dechent P, Howey D, Annaswamy A, Sauer DU. One-shot battery degradation trajectory prediction with deep learning. J Power Sources 2021;506:230024.
[18] Yang L, Liao Y, Duan R, Kang T, Xue J. A bidirectional recursive gated dual attention unit based rul prediction approach. Eng Appl Artif Intell 2023;120:105885.
[19] Lou G, Lin W, Huang G, Xiang W. A two-stage online remaining useful life prediction framework for supercapacitors based on the fusion of deep learning network and state estimation algorithm. Eng Appl Artif Intell 2023;123:106399.
[20] Severson KA, Attia PM, Jin N, Perkins N, Jiang B, Yang Z, Chen MH, Aykol M, Herring PK, Fraggedakis D, et al. Data-driven prediction of battery cycle life before capacity degradation. Nat Energy 2019;4(5):383–91.
[21] Zhao Q, Qin X, Zhao H, Feng W. A novel prediction method based on the support vector regression for the remaining useful life of lithium-ion batteries. Microelectron Reliab 2018;85:99–108.
[22] Ni Y, Xu J, Zhu C, Pei L. Accurate residual capacity estimation of retired lifepo4 batteries based on mechanism and data-driven model. Appl Energy 2022;305:117922.
[23] Wang F, Cheng X, Hsiao K. Stacked long short-term memory model for proton exchange membrane fuel cell systems degradation. J Power Sources 2020;448:227591.
[24] Zhang Y, Xiong R, He H, Pecht MG. Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries. IEEE Trans Veh Technol 2018;67(7):5695–705.
[25] Lu J, Xiong R, Tian J, Wang C, Hsu CW, Tsou NT, Sun F, Li J. Battery degradation prediction against uncertain future conditions with recurrent neural network enabled deep learning. Energy Storage Mater 2022;50:139–51.

[26] Qian C, Xu B, Chang L, Sun B, Feng Q, Yang D, Ren Y, Wang Z. Convolutional neural network based capacity estimation using random segments of the charging curves for lithium-ion batteries. Energy 2021;227:120333.
[27] Chen D, Hong W, Zhou X. Transformer network for remaining useful life prediction of lithium-ion batteries. IEEE Access 2022;10:19621–8.
[28] Zraibi B, Okar C, Chaoui H, Mansouri M. Remaining useful life assessment for lithium-ion batteries using cnn-lstm-dnn hybrid method. IEEE Trans Veh Technol 2021;70(5):4252–61.
[29] Duong PLT, Raghavan N. Heuristic kalman optimized particle filter for remaining useful life prediction of lithium-ion battery. Microelectron Reliab 2018;81:232–43.
[30] Kim S, Lee PY, Lee M, Kim J, Na W. Improved state-of-health prediction based on auto-regressive integrated moving average with exogenous variables model in overcoming battery degradation-dependent internal parameter variation. J Energy Storage 2022;46:103888.
[31] Wang J, Hertzmann A, Fleet DJ. Gaussian process dynamical models. Adv Neural Inf Process Syst 2005;18.
[32] Wang JM, Fleet DJ, Hertzmann A. Gaussian process dynamical models for human motion. IEEE Trans Pattern Anal Mach Intell 2007;30(2):283–98.
[33] Lawrence N, Hyvärinen A. Probabilistic non-linear principal component analysis with gaussian process latent variable models. J Mach Learn Res 2005;6(11).
[34] Richardson RR, Osborne MA, Howey DA. Gaussian process regression for forecasting battery state of health. J Power Sources 2017;357:209–19.
[35] Grochow K, Martin SL, Hertzmann A, Popović Z. Style-based inverse kinematics. In: ACM SIGGRAPH 2004 papers. 2004, p. 522–31.
[36] Saha B, Goebel K. Battery data set. In: NASA aMES prognostics data repository. 2007.
[37] Birkl C. Oxford battery degradation dataset 1. 2017.
[38] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014, arXiv preprint arXiv:1409.1556.
[39] Ren S, He K, Girshick R, Sun J. Faster r-cnn: Towards real-time object detection with region proposal networks. Adv Neural Inf Process Syst 2015;28.
[40] Kingma DP, Ba J. Adam: A method for stochastic optimization. 2014, arXiv preprint arXiv:1412.6980.
[41] Lawrence ND. Learning for larger datasets with the gaussian process latent variable model. 2007, p. 243–50.
[42] Tagade P, Hariharan K, Ramachandran S, Khandelwal A, Naha A, Kolake S, Han S. Deep gaussian process regression for lithium-ion battery health prognosis and degradation mode diagnosis. J Power Sources 2020;445:227281.