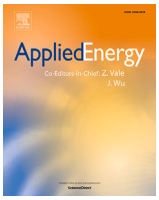
Applied Energy 389 (2025) 125703

Contents lists available at ScienceDirect

Applied Energy

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



Coupling a capacity fade model with machine learning for early prediction of the battery capacity trajectory

Tingkai Li [a], Jinqiang Liu [b], Adam Thelen [c], Ankush Kumar Mishra [c], Xiao-Guang Yang [d,1], Zhaoyu Wang [b], Chao Hu [a,*]

[a] School of Mechanical, Aerospace, and Manufacturing Engineering, University of Connecticut, Storrs, CT, 06269, USA
[b] Department of Electrical and Computer Engineering, Iowa State University, Ames, IA, 50011, USA
[c] Department of Mechanical Engineering, Iowa State University, Ames, IA, 50011, USA
[d] Department of Mechanical Engineering and Electrochemical Engine Center, The Pennsylvania State University, University Park, PA, 16802, USA

HIGHLIGHTS

- Early prediction of capacity fade facilitates aging-focused design and manufacturing.
- Integrating empirical models with ML enhances accuracy and uncertainty calibration.
- Hybrid models achieve <2 % error in-distribution, <4 % error out-of-distribution (OOD).
- Probabilistic predictions yield calibrated uncertainty, even for OOD samples.

ARTICLE INFO

Keywords:
Lithium-ion battery
Capacity degradation
Early prediction
Empirical model
Machine learning
Uncertainty quantification

ABSTRACT

Early prediction of battery capacity degradation, including both the end of life and the entire degradation trajectory, can accelerate aging-focused manufacturing and design processes. However, most state-of-the-art research on early capacity trajectory prediction focuses on developing purely data-driven approaches to predict the capacity fade trajectory of cells, which sometimes leads to overconfident models that generalize poorly. This work investigates three methods of integrating empirical capacity fade models into a machine learning framework to improve the model's accuracy and uncertainty calibration when generalizing beyond the training dataset. A critical element of our framework is the end-to-end optimization problem formulated to simultaneously fit an empirical capacity fade model to estimate the capacity trajectory and train a machine learning model to estimate the parameters of the empirical model using features from early-life data. The proposed end-to-end learning approach achieves prediction accuracies of less than 2 % mean absolute error for in-distribution test samples and less than 4 % mean absolute error for out-of-distribution samples using standard machine learning algorithms. Additionally, the end-to-end framework is extended to enable probabilistic predictions, demonstrating that the model uncertainty estimates are appropriately calibrated, even for out-of-distribution samples.

1. Introduction

Capacity-trajectory prediction using capacity fade models is useful in all areas of battery design and operation. Examples, where capacity-trajectory predictions prove useful, include new materials selection, manufacturing process optimization, charge/discharge protocol optimization, and remaining useful life (RUL) prediction for predictive maintenance and control [1–4]. Predictive maintenance and control are essential to the safe and reliable deployment of Li-ion batteries operating in the field. Furthermore, it is even more useful to estimate a battery cell's capacity trajectory before the cell shows any noticeable capacity fade. Early capacity-trajectory prediction enables researchers to accelerate design and optimization efforts by reducing the time spent testing cells to understand their long-term capacity degradation behaviors and predict the RUL as early as possible. Additionally, there is growing

* Corresponding author.
Email address: chao.hu@uconn.edu (C. Hu).
[1] Present affiliation: School of Mechanical Engineering, Beijing Institute of Technology, Beijing, 100081, China.

https://doi.org/10.1016/j.apenergy.2025.125703
Received 16 November 2024; Received in revised form 4 March 2025; Accepted 9 March 2025
Available online 26 March 2025

interest in repurposing used electric vehicle (EV) batteries for second-life storage applications, an application that requires understanding the entire capacity trajectory of a cell. Accurately predicting the cycle life of a battery cell in its second life requires understanding the entire degradation trajectory of the cell that spans over both its first and second lives [5]. Gaining such an understanding using early-life data offers new and exciting opportunities, such as materials selection [6] and optimal operation [7] for more reliable, cost-effective, and environmentally sustainable battery applications. In summary, early capacity-trajectory prediction tools will enable more efficient battery cell design, manufacturing process optimization, charge/discharge optimization, predictive maintenance, and evaluation for second-life use.

Early work focused on data-driven early prediction of battery life [8]. The focus was to investigate the aging spread between cells due to slight variations in the cell manufacturing process. The researchers cycled 48 lithium nickel manganese cobalt oxide (NMC) cells under the same conditions and used a regression model to group the cells with similar cycle lives together. Their findings showed that more work needed to be done to understand whether it is feasible to determine the cycle life of a cell using only information from its early life. This idea that a cell's cycle life could be estimated using only early-life data came up again when researchers cycled 24 lithium cobalt oxide (LCO) pouch cells and analyzed the failure statistics [9]. They found a weak correlation between a cell's capacity at cycle 80 and its future capacity at cycle 500. These findings have recently spurred a new area of Li-ion battery research, now known as early life prediction. Notable work was done by Severson et al. [6], who built a data-driven regression model for early prediction of cycle life. This model took, as input, early-life statistical features extracted from a cell's voltage (V) vs. discharge capacity (Q) curves in the first 100 cycles and predicted the cell's cycle life. These researchers are the first to demonstrate the concept of early cycle life prediction on cells cycled under various fast-charging protocols. Also, their publicly available dataset [6,10], consisting of 169 lithium iron phosphate (LFP) cells with varying fast-charging protocols but an identical full-depth constant-current discharge protocol, has been widely used in the field for various studies. Some follow-up studies [11–14] aimed to further improve cycle-life predictive performance by examining alternative machine/deep learning models and features to the one originally proposed [6]. Later, early prediction of battery cycle life has been demonstrated in a large dataset consisting of 300 cells with six different types of cathode materials but mostly limited to low C-rate cycling [15]. More recently, researchers have applied early life prediction using features extracted from periodic reference performance tests (RPTs) on a dataset of 225 NMC pouch cells with widely varying charge rates, discharge rates, and depths of discharge (DoD) [16]. Other notable research in this area has investigated estimating the knee point of Li-ion cell capacity fade curves. Researchers created a machine learning pipeline to estimate the knee point of Li-ion cells using many different combinations of early-life statistical features, derived from capacity, current, voltage, and temperature measurements in fashions of both per-cycle and cycle-to-cycle differences [17]. However, all these methods were solely concerned with point predictions like cell end of life (EOL) or RUL. They did not predict the entire capacity trajectory that possesses more information on capacity fade behavior.

One way to tackle the problem of early capacity-trajectory prediction is by building deep learning models based purely on data. Implementation of a sequence-to-sequence model with an encoder and decoder using four stacked long short-term memory (LSTM) recurrent neural networks has been demonstrated to predict both capacity and internal resistance trajectories for 48 NMC 18650 cells [18,19]. However, since the dataset was relatively small and the trajectories to predict were similar due to the identical cycling conditions, training a deep learning model on this small dataset tended to cause overfitting issues, and the model's generalization performance could be poor on a new dataset whose input feature distribution differs from the training data distribution. Alternatively, researchers demonstrated in two recent studies [20,21] reconstructing the capacity trajectory of a cell by first predicting representative points of the trajectory, such as the knee point, knee onset, EOL point, or points with equidistant capacity values. However, in these two studies, the reconstruction was either based on a piecewise cubic Hermite interpolating polynomial (PCHIP) interpolation [21] or a modified cubic spline [20], which lacked interpretability about different degradation trends experienced by cells. Fig. 1 highlights some publications in the past five years that solved some unique problems in the domain of battery early prediction.

In numerous studies on battery degradation modeling or battery prognostics, empirical capacity fade models were shown to capture the capacity fade trend for battery cells tested under a wide range of cycling conditions. However, these studies only focused on offline capacity trajectory fitting and did not tackle challenges associated with integrating empirical modeling with data-driven machine learning for early capacity-trajectory prediction. Hence, there is a gap in our current knowledge of how combining empirical capacity fade modeling and data-driven machine learning can enable early, high-performance prediction of the entire capacity trajectory. Desirable model performance includes higher prediction accuracy, reduced computational burden, better generalization, earlier prediction, and more accurate quantification of predictive uncertainty.

This study aims to fill the knowledge gap by improving data-driven machine learning models by augmenting them with the knowledge of capacity fade from empirical models. Fig. 2 provides an overview of our problem definition, and our contributions to the body of knowledge on early life prediction are elaborated in what follows.

- First, we benchmark three different approaches to combining a machine learning model with an empirical capacity model. A novel approach proposed in this paper is an end-to-end learning framework, which simultaneously fits a selected empirical model to estimate the capacity trajectory and trains a machine learning model to estimate the parameters of the empirical model using early-life data.
- Second, we implement and examine both deterministic and probabilistic configurations of the proposed end-to-end learning

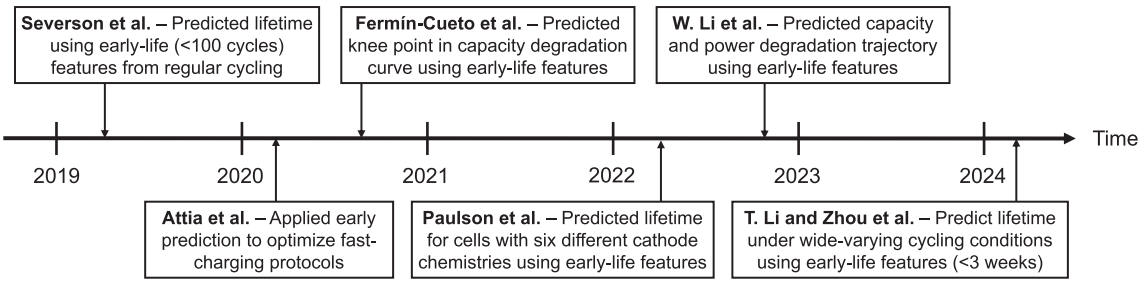


Fig. 1. A timeline highlighting some selected key problems solved in the field of battery early prediction during the past five years. This list is by no means exhaustive, and there may be other important early prediction problems that have been solved but are not reported in this timeline.

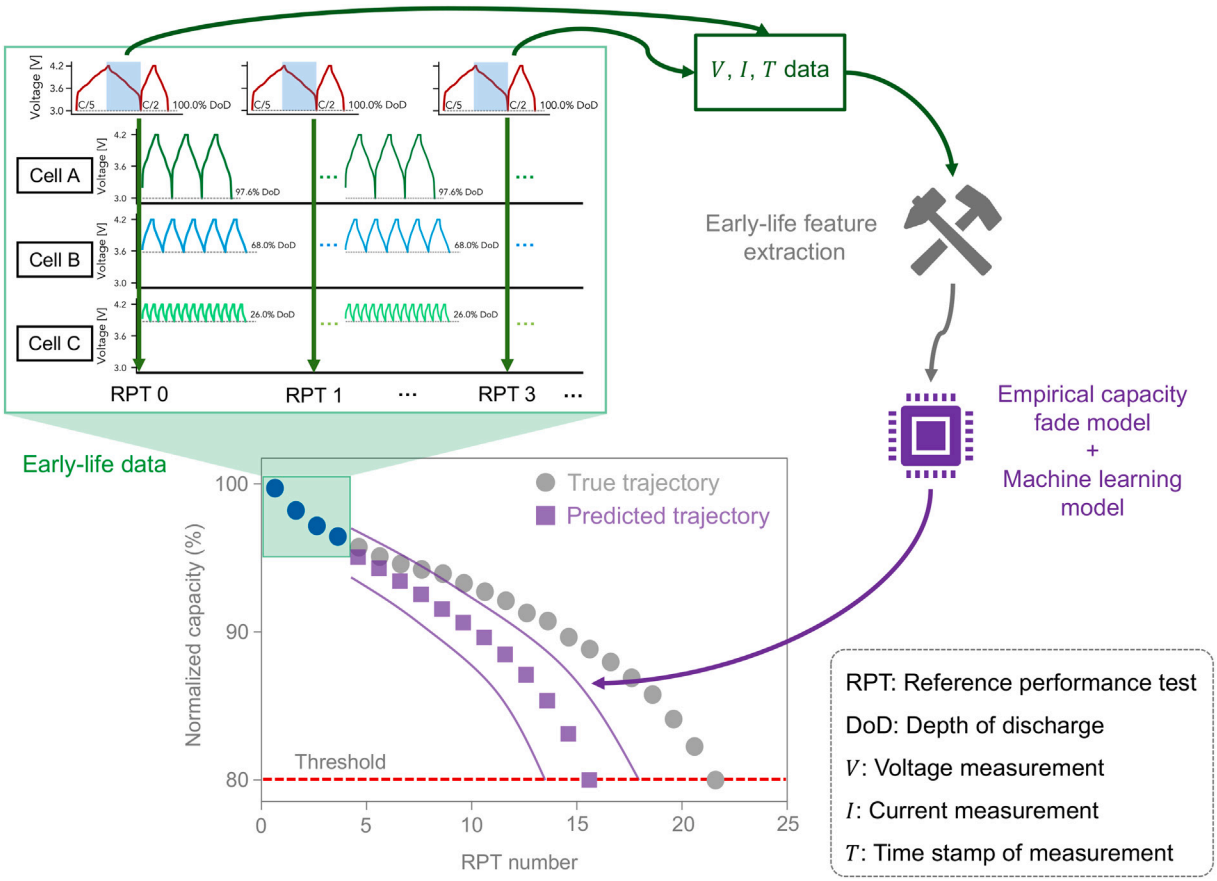


Fig. 2. An overview of the early trajectory prediction problem studied in this work.

framework and study the importance of uncertainty quantification in the context of early capacity-trajectory prediction. We use the neural network ensemble approach to quantify the predictive uncertainty. As a result, the end-to-end learning framework is uniquely flexible, allowing for either deterministic or probabilistic prediction of a capacity trajectory, in addition to the selection of any commonly reported empirical capacity fade model and machine learning model.

The rest of the paper is organized as follows. Section 2 briefly reviews the background of battery prognostics. Section 3 summarizes the battery aging dataset and the empirical capacity model used in this study. Section 4 details the methodology enabling the early capacity-trajectory predictions by coupling machine learning with an empirical model. Section 5 compares and discusses the prediction performances of different prediction approaches. Section 6 presents two benchmarking studies for the proposed end-to-end learning framework. The paper is concluded in Section 7.

2. Background of battery prognostics

Closely related to battery early capacity-trajectory prediction is battery prognostics. Many studies in this area have been conducted over the past decade. These studies attempted capacity-trajectory prediction to estimate a cell's RUL [22]. Battery prognostic methods can be broadly categorized as model-based and data-driven [23,24]. A common feature shared by model-based methods is that they use either a physics-based or an empirical model. Examples include mechanistic models depicting the evolution of degradation mechanisms [25,26], empirical half-cell models quantifying degradation modes [4,27–29], Arrhenius equation-based temperature-dependent capacity degradation models [30], equivalent circuit models depicting electrical performance [31], or empirical capacity fade models depicting the evolution of capacity [2,32–35]. Prognostic methods based on mechanistic models consider electrochemical processes internal to a cell. Because of this, these methods generalize well to new, "unseen" cells that were not used to develop the mechanistic models. However, the practical adoption of these methods may be limited by high computation costs, significant expertise required, and difficulties in identifying model parameters. Some prognostic methods based on equivalent circuit models take into account, to some extent, aging mechanisms by modeling how a cell's internal resistance grows over time/cycle. However, a typical prerequisite for adopting these methods is having access to specialized, often expensive test equipment for electrochemical impedance spectroscopy. Researchers in [36] performed small-scale calendar-life tests and fit empirical aging models to predict cell resistance as a function of time. Prognostic methods based on empirical models, such as those in [36], are popular because they are relatively easy to develop, have shown adequate accuracy, and exhibit good generalization performance. The most widely used way to implement empirical models is to perform recursive filtering that estimates on the fly the parameters of an empirical model using the most recent capacity/resistance measurements from an operating cell. Popular recursive filtering algorithms include extended Kalman filters [37], unscented Kalman filters [38], and particle filters [2,33,39], each with increasing computation cost and estimation capability. One of the most desired attributes of recursive filtering algorithms is their ability to output probabilistic predictions. This attribute allows them the ease of integration into a robust decision-making framework. However, empirical model-based methods work well only when they have access to a large amount of historical aging data (capacity fade or resistance increase) from an online cell that has shown noticeable degradation. Such data will allow for an accurate estimation of the cell's future

degradation trajectory. This is also a common drawback for most existing battery prognostics methods: they generally need access to a major portion (more than the first 40 %) of the entire capacity trajectory from a cell to estimate empirical model parameters [23]. Another drawback is their inability to share information from an offline cell that has reached its EOL with an online cell whose EOL and RUL are unknown and need to be estimated or between online cells. Specifically, the current practice of updating empirical models cannot incorporate extra information about previous best-fit parameters from offline cells or other cells in the online fleet. Given the rapidly growing size of modern battery datasets, this lack of shared information quickly becomes a major issue, making data-driven methods an attractive alternative.

On the other hand, data-driven methods predict the capacity degradation of an online cell based on training data collected from a set of offline cells using machine learning techniques. Some examples of commonly used machine learning techniques are support vector machine [40], relevance vector machine [41], Gaussian process regression [42], and neural networks [43,44]. Also, deep learning models like LSTM models have been widely applied to battery prognostics problems [45–47]. Recently, the battery prognostics community has placed an emphasis on developing prediction models with predictive uncertainty quantification. Gaussian process regression, a classic approach to achieving probabilistic predictions, is applied to forecast the state of health of cells from different starting points [42]. One recent work proposed a Bayesian neural network to learn from battery aging data and predict battery RUL with uncertainty quantification [48]. The proposed method was unique in that the Bayesian neural network could be trained using data from units that had not yet failed, reducing the overall amount of training data required. Another recent work proposed a neural network ensemble to predict the entire capacity trajectory using only early-life data [47]. For cells with extremely long cycle lives, which are considered out-of-distribution samples, the predictive uncertainties from this proposed method are noticeably higher than those of short-life cells. Furthermore, to provide more reliable uncertainty predictions, researchers proposed and demonstrated a framework to jointly calibrate the predictive, aleatory, and epistemic uncertainties while training a Bayesian deep network on predicting battery RUL [49].

3. Overview of the dataset and the empirical capacity fade model

3.1. Battery aging dataset

In this study, the ISU-ILCC battery aging dataset is used to assess the performance of the proposed methodology for predicting capacity trajectories for cells undergoing varying aging conditions [16,50]. This dataset consists of 251 cells cycled under 63 unique combinations of cycling stress factors (charge rate, discharge rate, and DoD). Due to the differences in stress factors, especially the differences in DoD, there is no available measurement from such data to accurately track cells' SOH. Thus, weekly RPTs were used to provide slow-rate, full-depth measurements for evaluating the SOH of cells and extracting early life features. From Fig. 3a, we can observe wide-varying capacity trajectories across different cells due to both the cell-to-cell intrinsic variation and the difference in aging conditions [16]. Note that, these capacity trajectories are constructed using the remaining discharged capacity values measured in the C/5 cycle of RPTs.

3.2. Preprocessing for battery aging data

There are three major steps to preprocess the collected data for empirical model fitting as well as capacity trajectory prediction, which are normalization, removal of fast-aging cells, and interpolation of capacity trajectory.

The need for interpolating trajectories arises from the fact that different cells may have different numbers of RPT data points before reaching the EOL. For example, as shown in Fig. 3b, G57C2 – a long-life cell – has many more RPT measurements than G30C1 – a short-life cell. Our goal for interpolation is to transform the capacity trajectory from any cell into a vector of length m, and stacking multiple trajectories of different cells results in a rectangular matrix with no missing value. This allows an easier implementation of our methodology for capacity trajectory predictions compared with the raw data with different lengths in trajectories.

First, we normalize each cell's capacity data by dividing each capacity measurement by the value of the first capacity measurement. After normalization, the capacity data of each cell started at a normalized capacity of 1.0 (or 100 % on a percentage scale). *Second,* cells are removed if they meet one of the conditions: (1) less than 5 RPTs before reaching the end-of-life threshold (80 % remaining capacity) and (2) less than 10 RPTs before reaching 70 % remaining capacity. *Third,* we interpolate the capacity measurements using a PCHIP interpolation such that the interpolated data points are equidistant for every 1 % capacity drop until the cell reaches 80 % remaining capacity ($m = 21$). From two representative cells shown in Fig. 3c, both G57C2 and G30C1 have the same number of data points for the trajectory after interpolation. After interpolation, capacity trajectories for multiple cells can be easily represented by two rectangular matrices, one for capacity and the other one for Ah-throughput, which are

$$\mathbf{Q}_{n\times 21} = \begin{bmatrix} 1 & 0.99 & \cdots & 0.81 & 0.80 \\ 1 & 0.99 & \cdots & 0.81 & 0.80 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 0.99 & \cdots & 0.81 & 0.80 \end{bmatrix},$$
$$\mathbf{N}_{n\times 21} = \begin{bmatrix} 0 & N_{1,2} & \cdots & N_{1,20} & N_{1,21} \\ 0 & N_{2,2} & \cdots & N_{2,20} & N_{2,21} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & N_{n,2} & \cdots & N_{n,20} & N_{n,21} \end{bmatrix}, \quad (1)$$

where n represents the number of cells to be represented in the matrices, and $N_{i,j}$ is the Ah-throughput corresponding to the element $Q_{i,j}$ in the normalized capacity matrix $\mathbf{Q}$.

There is a clear rationale behind establishing the aforementioned criteria for removing fast-aging cells. The first condition is imposed because early-life features were extracted from the initial and week 3 RPTs (i.e., the first and fourth RPTs in the trajectory, or the first and fifth if the cells undergo an additional week 0.5 RPT). As a result, performing early capacity trajectory predictions on these cells would hold no value because they would have already reached EOL even before the predictions could be made. The second condition ensures a sufficient number of data points are available to represent the overall trajectory during the interpolation process. When the true measurements on a capacity trajectory are too few, the underlying relationship between charge throughput and remaining capacity becomes underrepresented. Including interpolated curves based on such insufficient data could introduce unnecessary noises and errors into the modeling process. Additionally, a cell with fewer than 10 RPTs before reaching 70 % remaining capacity would hit EOL (i.e., 80 % remaining capacity) even earlier. In this case, the early prediction point – occurring at the fourth RPT (or fifth if an additional week 0.5 RPT is included) – would be too close to, or even beyond, 50 % of the cell's total lifetime. This proximity undermines the validity of the prediction as an "early prediction".

Although removing fast-aging cells might raise concerns about subjective biases, early life prediction is a one-time process, and its practical relevance increases when made further away from EOL. Thus, removing these cells based on the above-outlined criteria is expected to have a minimal impact on the generalizability and practical significance of the early prediction models.

3.3. Data partition for trajectory prediction

Once the preprocessing is done, cells are partitioned into three subsets based on their associated groups defined by the three stress factors, following the methodology of the original study on this aging dataset

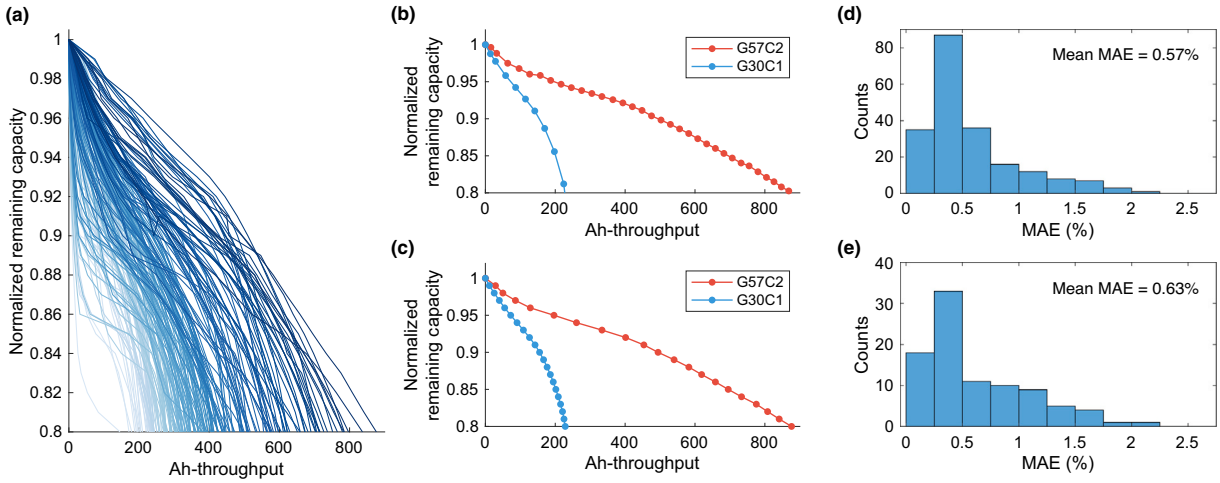


Fig. 3. Overview of the trajectory from cells in the dataset. (a) Capacity trajectories for all cells; (b) the data collected in RPTs for constructing the capacity trajectories for a long-life cell (G57C2) and a short-life cell (G30C1); (c) The capacity trajectories for the two cells on (b) constructed by interpolated data points using equidistant remaining capacity points; (d) An error histogram of fitting the empirical capacity fade model in Eq. (2) on the entire dataset; (e) An error histogram of fitting the empirical capacity fade model in Eq. (2) on the training set.

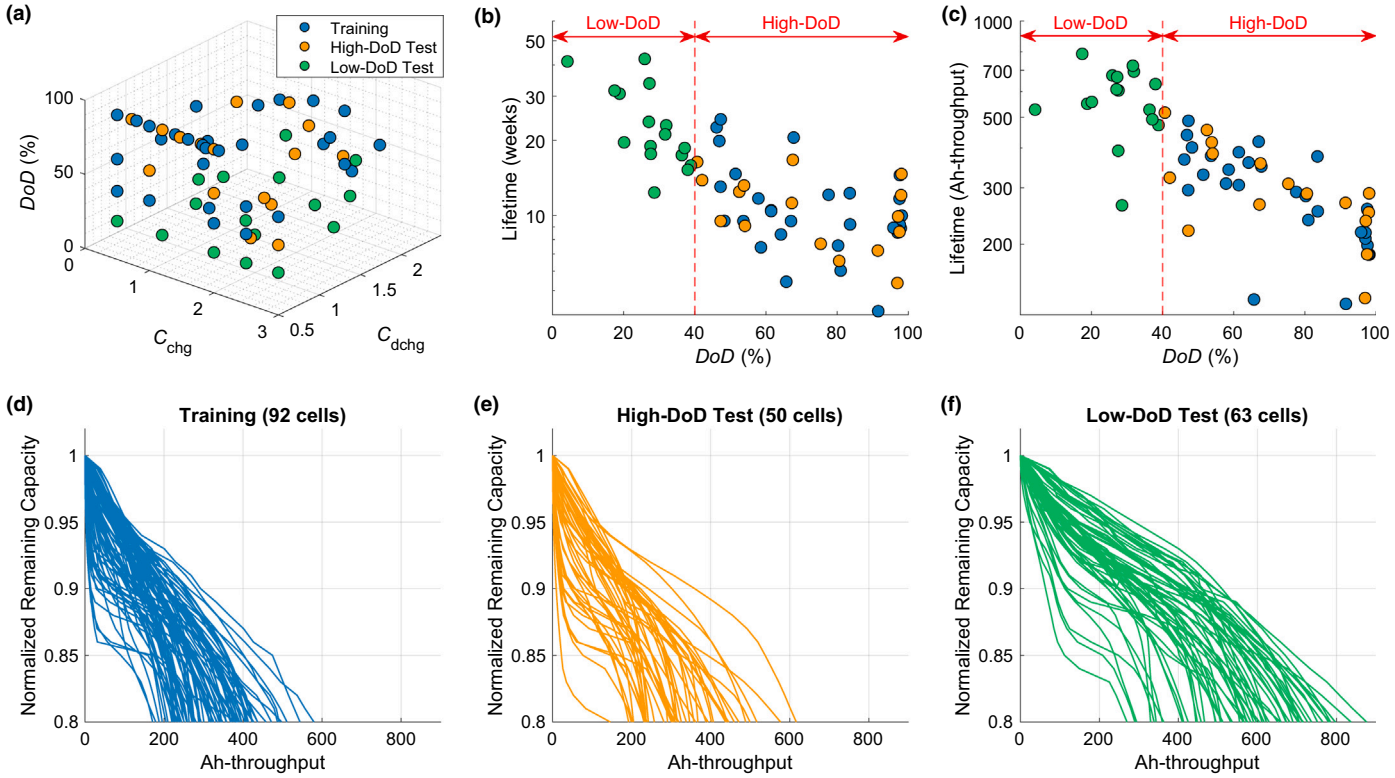


Fig. 4. Overview of the dataset partition. (a) Scatter plot of cycling conditions for each group, colored by its assigned subset after partitioning. (b) Group mean lifetime (in weeks) against DoD, measured by time on test. (c) Group mean lifetime against DoD, measured by total Ah-throughput. (d–f) Capacity trajectories for cells in three different subsets: (d) training set, (e) high-DoD test set, and (f) low-DoD test set.

[16]. Fig. 4a shows the results of this group-based partition. This partitioning approach is designed to simulate a battery modeling workflow, starting from the experimental design phase. When we want to collect some run-to-EOL aging data to build an early prediction model, the only controllable parameters are cycling conditions. The trained model is then used to predict aging trajectories under unknown conditions after observing the cells for a short duration under these controlled cycling conditions. *First*, the dataset is split into two subsets based on a DoD threshold of 40 %, and the subset with DoD less than 40 % is designated as the low-DoD test set. As shown in Fig. 4b, cells in the low-DoD region can take up to 50 weeks of cycling to reach EOL at 80 % remaining capacity, while most groups in the high-DoD region reached EOL within 25 weeks of cycling. Such a partition strategy simulates an accelerated modeling approach where groups reaching EOL faster are used for training. This approach can reduce the time and cost of the overall modeling process. Also, despite differences in cycling C-rates, groups

cycled at lower DoDs tend to withstand higher total Ah-throughput (or more full equivalent cycles) before reaching EOL, as depicted in Fig. 4c. *Second*, the subset with DoD ≥ 40 % is further divided into a training set and a high-DoD test set. The training groups are randomly picked to adequately cover the test design space above 40 % DoD, ensuring that the high-DoD test serves as an interpolation test set with respect to cycling conditions. After partitioning, the dataset contains 30 groups in the training set, 16 groups in the low-DoD test set, and 16 groups in the high-DoD test. The capacity trajectories for the valid cells in each subset are shown in Fig. 4d–f. By comparing different subsets, we observe that capacity trajectories in the training set exhibit patterns similar to those of the high-DoD test set, where most cells exhibit a three-stage capacity fade behavior. Unlike the training and high-DoD test set, the low-DoD test set contains more cells with longer lifetimes, showing only two-stage trajectories. This comparison signifies that (1) the high-DoD test set represents in-distribution test samples and (2) the low-DoD test set serves as an out-of-distribution test case, allowing evaluation of the extrapolation capability of different prediction methods. It is important to note that the feature engineering and trajectory prediction approaches discussed later in this work learn exclusively from the training set to avoid potential data leakage from test data.

3.4. Empirical capacity fade model

The battery research community has reported several empirical models with different algebraic expressions that accurately model Li-ion battery capacity fade trends at different stages of their lifespan, such as the linear term [2,51], the exponential term [2,34,41], the power-law term [35,41], and the sigmoid term [51,52]. To encode the idea that separate terms carry information related to cell degradation modes and only cause capacity drops in the empirical capacity fade model, we force all parameters to be positive real numbers. In this paper, we consider a hybrid empirical model blending a power-law term and a sigmoid term to better capture the three-stage degradation trend in the ISU-ILCC battery aging dataset. The empirical model is expressed as

$$Q(N; a, b_1, b_2, b_3) = 1 - b_1 N^a - \frac{1}{1 + \exp\frac{b_2 - N}{b_3}}, \tag{2}$$

where N represents the Ah-throughput of the cell, a is the power-law coefficient that is fitted to the entire dataset as a global parameter, and b_1, b_2, and b_3 are local parameters that are fitted to each individual cell. The power-law term with $a < 1$ intends to capture the initial rapid degradation mainly due to the SEI formation and the following close-to-linear slow degradation, and the sigmoid term is designed to capture the degradation when the "knee point" effect [53] (i.e., the accelerating degradation trend after cells age to a certain level) kicks in.

After that, we utilize a multi-level empirical model fitting strategy inspired by the work of Gasper et al. [52], in which we first identify the best global parameter a over the entire dataset and then fit local parameters (b_1, b_2, and b_3) to each cell individually. The fitting error of the entire dataset consisting of 205 valid cells is 0.57 % mean absolute error (MAE), and Fig. 3d shows the fitting accuracy over the entire dataset. Also, to estimate the empirical model parameters for cells in the training set, the model is fitted only to the training set of 92 cells, which yields a mean fitting MAE of 0.63 %.

One limitation of this empirical model, and its resulting trajectory predictions, is that it considers only cyclic aging effects on the cells while excluding calendar aging effects. The main reason for excluding calendar aging effects is the lack of a well-designed calendar aging test campaign for the type of cell used in the cycling aging test. Also, this empirical model is not a one-solution-fits-all, particularly in cases where cells do not exhibit a knee-point effect or experience a sudden shift in cycling conditions in the middle of their lifespan. Although the model can be adapted to accommodate different pre-knee aging behaviors by adjusting the exponent of the power-law term – resulting in linear degradation if $a = 1$ or continuously accelerating degradation if $a > 1$ – the presence of the sigmoid term still limits the model's effectiveness on datasets lacking a distinct knee-point effect. The benchmarking dataset presented in Section 6.1, for example, contains cells that mostly do not exhibit an apparent knee-point transition, necessitating the removal of the sigmoid term from the empirical model used for that dataset. Furthermore, the model does not inherently account for abrupt changes in cycling conditions, which could occur in scenarios such as second-life repurposing. In these scenarios, cells may be first subjected to EV duty cycles simulating first-life usage before transitioning to grid storage duty cycles simulating second-life usage. Such shifts in operating conditions introduce complex degradation behaviors that may not be adequately captured by the current empirical modeling framework, requiring additional modifications or hybrid modeling approaches to improve predictive accuracy.

4. Empirical model-informed trajectory prediction

In this section, we present different approaches to early capacity-trajectory prediction, with the assistance of an empirical capacity model and features extracted from early-life data. First, we discuss the method for feature engineering from early-life data and the rationale behind choosing it. Following that, we break down the details for capacity-trajectory prediction by coupling the empirical model with machine learning in three distinct ways. In addition, we introduce the ensemble learning approach for the novel end-to-end framework to enable a probabilistic prediction of the capacity trajectory.

4.1. Feature engineering from early-life data

In the original study on this dataset [16], the researchers extracted a set of 29 early-life features derived from cycling conditions, discharge capacity values, constant-voltage charging curves, capacity–voltage curves, incremental capacity curves, and differential voltage curves. A detailed description of all 29 extracted features is included in Table A.1. The previous study focused exclusively on predicting the EOL, and the complete set of features was down-selected using a stepwise forward search method, optimizing for prediction performance based on a linear regression model. However, in this study, where the focus is on predicting the capacity trajectory via three diverging utilizations of empirical capacity models, a supervised feature selection method (e.g., ranking using Pearson's correlation coefficients, a stepwise forward/backward search) is challenging to implement because the input/output pairs are different for each method. Thus, a principal component analysis (PCA) model is applied to the training dataset to perform feature decomposition. The main purpose of PCA is to identify orthogonal bases that can explain most of the variance in the data while reducing the dimensionality. PCA is achieved by performing a linear transformation on the feature space [54], which can be expressed as

$$\mathbf{P}\mathbf{X}_{\text{ob}} = \mathbf{X}_{\text{pca}}, \tag{3}$$

where $\mathbf{P}$ is the transformation matrix mapping the feature data represented on the original basis ($\mathbf{X}_{\text{ob}}$) to the principal basis ($\mathbf{X}_{\text{pca}}$). Since the available RPT data is collected at fixed cycling time intervals instead of cycle counts, the early-life dataset includes the periodic RPTs from the first three weeks of aging. Early-life features are extracted from data collected during the initial RPT at the beginning of life and the RPT after three weeks of cycling, following the same methodology as the preceding work [16]. Also, to avoid data leakage, PCA is only applied to the training set to obtain the transformation matrix, and only the top 10 PCA features are selected, which explain over 95 % of the variance of the features in the training set.

4.2. Knot point-based battery capacity trajectory prediction

The first approach we benchmark for predicting the capacity trajectories is inspired by the state-of-the-art, in which some specific points in a trajectory are predicted to reconstruct the entire trajectory [20,21]. More specifically, the work by Kim et al. [21] predicts the number of

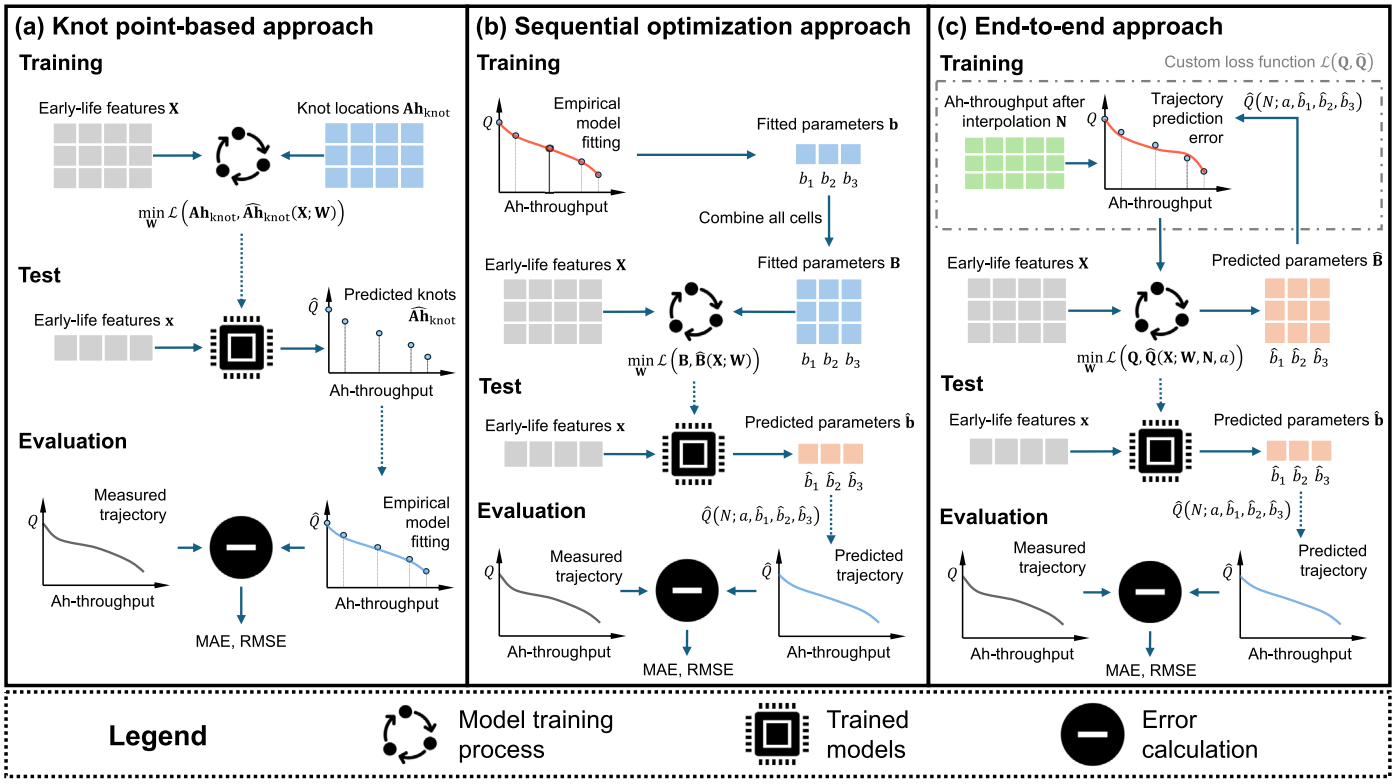


Fig. 5. Overview of capacity trajectory prediction approaches. (a) Knot point-based prediction of empirical model parameters; (b) sequential prediction of empirical model parameters; (c) end-to-end prediction of empirical model parameters.

cycles between several predetermined SOH values and reconstructs the trajectory using PCHIP interpolation. Meanwhile, the work by Ibraheem et al. [20] simultaneously predicts the cycle numbers and capacity values at the knee onset and the knee point, and the reconstruction is done by fitting modified three-stage quadratic splines, with each term connecting between a pair of consecutive points. In our study, we formulate the first approach for capacity trajectory prediction following two steps: (1) predicting the location of four equidistant knot points selected based on normalized remaining capacity values and (2) fitting the predicted points with the empirical model (see Fig. 5a).

The five equidistant knot points to be predicted are at [96 %, 92 %, 88 %, 84 %, 80 %] remaining capacity, and the target to predict from the machine learning model is the difference of Ah-throughput values between each location. The reason for predicting the difference between each location instead of the location directly is to ensure the monotonicity of the predicted trajectory, which is easily enforced by using ReLU activation functions at the output layer. After predicting the five points and combining these with the beginning of life (i.e., 100 % remaining capacity with an Ah-throughput of 0), the empirical model in Eq. (2) is fitted to a total of six points, and the empirical model parameters are obtained for each cell. Then, the prediction error is evaluated using the empirical model with the obtained parameters, which is a univariate function of Ah-throughput for capacity, at each Ah-throughput value of the capacity data.

4.3. Sequential optimization for battery capacity trajectory prediction

Early prediction of a battery cell's capacity trajectory using the sequential optimization method involves two optimization processes: (1) fitting the empirical model to a cell's trajectory and obtaining the optimal set of parameters; (2) training a machine learning model to estimate these parameters using PCA features extracted from early-life data (see the overview in Fig. 5b). In this study, we consider two machine learning algorithms for mapping early-life features to empirical capacity model parameters: (1) elastic-net regularized linear regression (ENR) and (2) multi-layer perceptron neural network (MLP). To better illustrate this approach and draw differences from the end-to-end optimization method introduced in the following subsection, we present the mathematical formulation for sequential optimization using an elastic net regression model.

The first step is a curve-fitting problem that minimizes the mean squared error (MSE), formulated for **one cell** as

$$\min_{a,b_1,b_2,b_3} \frac{1}{2m} \Bigg\| \overbrace{\mathbf{1}_{1\times m} - \overbrace{b_1 \mathbf{N}_{1\times m}^{a}}^{\text{power-law term}} - \mathbf{1}_{1\times m} \oslash \left(\mathbf{1} + \exp \frac{b_2 \mathbf{1}_{1\times m} - \mathbf{N}_{1\times m}}{b_3}\right)}^{\text{sigmoid term}} - \underbrace{\mathbf{q}_{1\times m}}_{\text{true trajectory}} \Bigg\|_2^2 ,$$
$$\text{fitted trajectory}$$

$$\text{s.t.} \quad 0.4 \le a \le 0.6 ,$$
$$10^{-6} \le b_1 \le 1 ,$$
$$100 \le b_2 \le 2000 ,$$
$$10 \le b_3 \le 500 ,$$
(4)

where b_1, b_2, and b_3 are scalars representing the local empirical model parameters for the given cell, and a is a scalar representing the global parameter shared across the entire dataset. The first three terms in Eq. (4) are the fitted trajectory using the empirical capacity fade model in Eq. (2), and the fourth term represents the true capacity measurements that the empirical model is fitted to. Here, since we only consider one cell, the Ah-throughput measurement $\mathbf{N} \in \mathbb{R}^m$, the corresponding capacity $\mathbf{q} \in \mathbb{R}^m$, and a unity vector $\mathbf{1} \in \mathbb{R}^m$ are all vectors of m elements, where m denotes the number of interpolated measurements on a capacity trajectory.

This optimization problem is non-convex (i.e., no analytical global optimum can be derived). To improve convergence, we provide initial parameter guesses to the optimization algorithm and set bounds based on expected behavior and the numerical values of Ah-throughput. For the exponent of the power-law term (global parameter a), an expected value of 0.5 is widely reported in the capacity modeling literature, primarily capturing SEI formation and other surface-related degradation mechanisms during initial cycles [52,55]. To accommodate dataset flexibility, we set a bound between 0.4 and 0.6. The local parameters b_2 and b_3 control the onset and slope of the expected knee-point effect, respectively. The bounds for these two parameters, along with the coefficient of the power-law term (b_1), are determined based on numerical experiments on the dataset to ensure expected model behavior.

Expanding this optimization problem to the case with **multiple cells**, the minimization of curve-fitting MSE becomes, in a matrix form,

$$\min_{a,\mathbf{b}} \quad \frac{1}{2nm}\Big\| \mathbf{1}_{n\times m} - \overbrace{\mathbf{b}_1\mathbf{1}_{1\times m}\otimes \mathbf{N}_{n\times m}^{a}}^{\text{power-law terms}} - \overbrace{\mathbf{1}_{n\times m}\oslash\left(\mathbf{1}_{n\times m}+\exp\left(\mathbf{b}_2\mathbf{1}_{1\times m}-\mathbf{N}_{n\times m}\right)\oslash\mathbf{b}_3\mathbf{1}_{1\times m}\right)}^{\text{sigmoid terms}} - \overbrace{\mathbf{Q}_{n\times m}}^{\text{true trajectories}} \Big\|_{\mathrm{F}}^{2}, \tag{5}$$

$$\text{s.t.} \quad 0.4 \le a \le 0.6\,,$$
$$10^{-6} \le b_{1,j} \le 1, b_{1,j} \in \mathbf{b}_1 = [b_{1,1}, b_{1,2}, \dots, b_{1,n}]\,,$$
$$100 \le b_{2,j} \le 2000, b_{2,j} \in \mathbf{b}_2 = [b_{2,1}, b_{2,2}, \dots, b_{2,n}]\,,$$
$$10 \le b_{3,j} \le 500, b_{3,j} \in \mathbf{b}_3 = [b_{3,1}, b_{3,2}, \dots, b_{3,n}]\,,$$

where n is the number of cells. When considering n cells together, instead of in the form of a vector for one cell, the Ah-throughput for n cells is a matrix $\mathbf{N} \in \mathbb{R}^{n\times m}$ and the corresponding capacity is a matrix $\mathbf{Q} \in \mathbb{R}^{n\times m}$. Thus, each term inside the norm is a matrix with a shape of $n\times m$. Here, since $\mathbf{b}_i \in \mathbb{R}^n|_{i=1,2,3}$ are column vectors of empirical model parameters of n cells, a unity vector $\mathbf{1}_{1\times m}$ is introduced to broadcast the empirical model parameter vectors into $n\times m$ matrices for element-wise operations. The subscript in $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm of a matrix, which is equivalent to the L2-norm of a vector.

Once the empirical model parameters are obtained from all cells in the training set, a multi-task elastic net regularized linear regression model is trained to predict the local parameters using early life features. The training process by minimizing the MSE of labels (i.e., the fitted empirical model parameters) is specified as

$$\min_{\mathbf{W}} \quad \frac{1}{2\,n}\left\|\mathbf{X}_{n\times k}\mathbf{W}_{k\times 3}-\mathbf{B}_{n\times 3}\right\|_{\mathrm{F}}^{2} + \alpha\left(\frac{\rho}{2}\|\mathbf{W}_{k\times 3}\|_{\mathrm{F}}^{2} + (1-\rho)\|\mathbf{W}_{k\times 3}\|_{1}\right), \tag{6}$$

where $\mathbf{X} \in \mathbb{R}^{n\times k}$ is the extracted from early-life data, $\mathbf{B} \in \mathbb{R}^{n\times 3}$ is the optimized empirical model parameters of all training cells, and $\mathbf{W} \in \mathbb{R}^{k\times 3}$ is the weight assigned to each feature for three tasks separately (i.e., predicting the three empirical parameters). Here, k represents the number of early-life features. Two hyperparameters, ρ and α, are the ratio between L1 and L2 penalization and the regularization parameter, respectively. Similarly, this process of training an elastic net regression model can be replaced with the backpropagation algorithm for training an MLP network.

4.4. End-to-end optimization for battery capacity trajectory prediction

Instead of following a two-step optimization for predicting the empirical model parameters as discussed in Section 4.3, the distinct difference of an *end-to-end optimization* framework is that there is only one optimization that performs both learning tasks simultaneously (see Fig. 5c). Specifically, when we train a model using the end-to-end optimization regardless of the machine learning approach, the loss is evaluated based on the predicted trajectories from the empirical model and the true capacity trajectories are considered as the ground truth. By embedding the empirical model into the loss function, both curve fitting and empirical parameter learning can be achieved within a single optimization process. The idea of an end-to-end optimization framework was introduced in a brief conference proceeding [56], along with some preliminary results obtained using the publicly available 124-cell LFP dataset [6]. However, in this work, we aim to present a more complete study on a dataset with a more significant divergence in observed capacity trajectories, which means a more challenging early capacity-trajectory prediction problem.

In the case of utilizing a multi-task elastic net regressor as the machine learning model in the end-to-end optimization framework and neglecting constraints on bounds for the empirical model parameters, the optimization problem is formulated with a single objective

$$\min_{\hat{\mathbf{b}}_1,\hat{\mathbf{b}}_2,\hat{\mathbf{b}}_3} \quad \frac{1}{2nm}\Big\| \mathbf{1}_{n\times m} - \overbrace{\hat{\mathbf{b}}_1\mathbf{1}_{1\times m}\otimes \mathbf{N}_{n\times m}^{a}}^{\text{predicted power-law terms}} - \overbrace{\mathbf{1}_{n\times m}\oslash\left(\mathbf{1}_{n\times m}+\exp\left(\hat{\mathbf{b}}_2\mathbf{1}_{1\times m}-\mathbf{N}_{n\times m}\right)\oslash\hat{\mathbf{b}}_3\mathbf{1}_{1\times m}\right)}^{\text{predicted sigmoid terms}} - \overbrace{\mathbf{Q}_{n\times m}}^{\text{true trajectories}} \Big\|_{\mathrm{F}}^{2} + \overbrace{\alpha\left(\rho\left\|\mathbf{W}_{k\times 3}\right\|_{\mathrm{F}}^{2}+(1-\rho)\left\|\mathbf{W}_{k\times 3}\right\|_{1}\right)}^{\text{elastic net regularization}}, \tag{7}$$

$$\text{s.t.} \quad \hat{\mathbf{b}}_1 = \mathbf{X}\mathbf{w}_1\,,$$
$$\hat{\mathbf{b}}_2 = \mathbf{X}\mathbf{w}_2\,,$$
$$\hat{\mathbf{b}}_3 = \mathbf{X}\mathbf{w}_3\,.$$

In this single objective, the empirical model parameters ($\hat{\mathbf{B}} = [\hat{\mathbf{b}}_1, \hat{\mathbf{b}}_2, \hat{\mathbf{b}}_3]$) are predicted from a multi-task elastic net regression model originated from Eq. (6) with a weight matrix $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3]$. The weights for each individual parameter $\mathbf{w}_i|_{i=1,2,3}$ are column vectors with a shape of $k\times 1$. Then, the MSE loss (overall prediction performance) is calculated between the predicted trajectories and the true capacity trajectories $\mathbf{Q}$, which requires the Ah-throughput of equidistant capacity $\mathbf{N}_{n\times m}$ defined in Eq. (1) as an additional input to the loss function. This formulation can then be simplified into one equation with constraints as

$$\min_{\mathbf{W}} \quad \frac{1}{2nm}\Big\| \mathbf{1}_{n\times m} - \mathbf{X}\mathbf{w}_1\mathbf{1}_{1\times m}\otimes \mathbf{N}_{n\times m}^{a} - \mathbf{1}_{n\times m}\oslash\left(\mathbf{1}_{n\times m}+\exp\left(\mathbf{X}\mathbf{w}_2\mathbf{1}_{1\times m}-\mathbf{N}_{n\times m}\right)\oslash\mathbf{X}\mathbf{w}_3\mathbf{1}_{1\times m}\right) - \mathbf{Q}_{n\times m}\Big\|_{\mathrm{F}}^{2} + \alpha\left(\rho\left\|\mathbf{W}_{k\times 3}\right\|_{\mathrm{F}}^{2}+(1-\rho)\left\|\mathbf{W}_{k\times 3}\right\|_{1}\right), \tag{8}$$

$$\text{s.t.} \quad \min\left(\mathbf{X}\mathbf{w}_1\right) > 0\,,$$
$$\min\left(\mathbf{X}\mathbf{w}_2\right) > 0\,,$$
$$\min\left(\mathbf{X}\mathbf{w}_3\right) > 0\,.$$

Compared to the sequential optimization that can be easily implemented via various curve fitting and machine learning toolboxes, the implementation of an end-to-end framework is not as straightforward, especially in the case of elastic net regression. The end-to-end optimization problem in Eq. (8) is hand-coded into a non-linear optimizer to find the weights $\mathbf{W}$. So, to avoid convergence issues from enforcing too many non-differentiable constraints (as shown in Eq. 5), we only implement three constraints to ensure all predicted empirical model parameters during the training process are strictly positive.

In both the sequential optimization and end-to-end approaches, machine learning models (e.g., an elastic net regressor or an MLP network) take PCA-transformed early-life features as the input and predict the empirical model parameters. The key distinction between these two early trajectory prediction approaches lies in the loss function design. Specifically, the end-to-end approach employs a custom loss function, as shown in Eq. (8) for an elastic net regressor, which directly embeds the empirical model to guide the optimization process. This eliminates the need for an intermediate step to generate labels for machine

learning (i.e., obtaining empirical model parameters via curve fitting). Disregarding the regularization terms, the prediction error of the end-to-end framework should always be lower than that of its sequential optimization counterpart. A more detailed mathematical proof is shown in Section A.2, which yields

$$\begin{aligned}&\frac{1}{2nm}\Big\|\mathbf{1}_{n\times m}-\mathbf{X}\mathbf{w}_1^{\mathrm{e2e}}\mathbf{1}_{1\times m}\otimes\mathbf{N}^a\\&\qquad-\mathbf{1}_{n\times m}\oslash\left(\mathbf{1}_{n\times m}+\exp\left(\mathbf{X}\mathbf{w}_2^{\mathrm{e2e}}\mathbf{1}_{1\times m}-\mathbf{N}\right)\oslash\mathbf{X}\mathbf{w}_3^{\mathrm{e2e}}\mathbf{1}_{1\times m}\right)-\mathbf{Q}\Big\|_F^2\\\le&\frac{1}{2nm}\Big\|\mathbf{1}_{n\times m}-\mathbf{X}\mathbf{w}_1^{\mathrm{seq}}\mathbf{1}_{1\times m}\otimes\mathbf{N}^a\\&\qquad-\mathbf{1}_{n\times m}\oslash\left(\mathbf{1}_{n\times m}+\exp\left(\mathbf{X}\mathbf{w}_2^{\mathrm{seq}}\mathbf{1}_{1\times m}-\mathbf{N}\right)\oslash\mathbf{X}\mathbf{w}_3^{\mathrm{seq}}\mathbf{1}_{1\times m}\right)-\mathbf{Q}\Big\|_F^2,\end{aligned}\tag{9}$$

where the superscript e2e denotes the weights obtained through the end-to-end framework, and the superscript seq denotes the weights obtained through the sequential optimization.

4.5. Neural network ensemble for probabilistic trajectory prediction

To promote real-world decision-making applications that incorporate the early prediction of capacity trajectories, it is essential to understand the uncertainty of such methods. Typically, machine learning model uncertainty is classified into two conceptual categories, namely aleatory and epistemic uncertainties. Aleatory uncertainty (also known as data uncertainty) is a measure of deviation within the dataset distribution; thus, it is irreducible. Epistemic uncertainty is uncertainty from our imperfect understanding and modeling of the data, and because of this, it is reducible. Therefore, we select ensemble learning based on neural networks to better quantify the epistemic uncertainty associated with our end-to-end learning framework.

For the neural network ensemble (NNE) applied in a general machine learning problem, each predicted mean has its corresponding predicted variance, which makes each prediction an independent Gaussian distribution. However, in our application, the predictions from machine learning models are the three empirical model parameters; providing three Gaussian distributions each for one empirical parameter makes it hard to propagate from uncertainties of parameters to the uncertainty of final trajectory prediction. So, we design the neural network in the ensemble to output three mean predictions and only one prediction related to the variance. Specifically, the three mean predictions are the empirical model parameters used to construct the predicted capacity trajectory, and the fourth output from the model should reflect the variance of overall trajectory prediction in our application. We adopt the variance prediction for a trajectory such that, for a given cell i, the predicted standard deviation $\hat{\boldsymbol{\sigma}}_i$ is defined as

$$\hat{\boldsymbol{\sigma}}_i=\widehat{CV}_i\times\left(\mathbf{1}-\hat{\mathbf{q}}_i\right),\tag{10}$$

where $\widehat{CV}_i$ is a scalar value for the coefficient of variation predicted by a model, and $\hat{\mathbf{q}}_i$ are vectors of the predicted capacity values at a given sequence of Ah-throughput for the cell i. An assumption made in this adaptation is that the predictive uncertainty should increase as the predicted capacity fade increases (i.e., predicting further into the future), and the prediction for the beginning of life (at $N = 0$ with no capacity fade) should ideally be 0. The ensemble method used to quantify the uncertainty of the end-to-end framework neural network models is summarized in Fig. 6. In an ensemble of M neural networks, each network outputs means for the three empirical model parameters ($\widehat{\mu_{b_1}}$, $\widehat{\mu_{b_2}}$, and $\widehat{\mu_{b_3}}$) and a coefficient of variation ($\widehat{CV}$) for the predicted trajectory. The network outputs can then be used to construct a Gaussian distribution which describes the mean and variance of the predicted trajectory.

Conventionally, a single neural network in an ensemble is trained by minimizing the overall negative log-likelihood (NLL) between true and predicted values, with a general form for the mean NLL of

$$\mathrm{NLL}=\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{2}\log(2\pi\hat{\sigma}_i^2)+\frac{(y_i-\hat{\mu}_i)^2}{2\hat{\sigma}_i^2}\right),\tag{11}$$

where n is the number of samples. In our implementation, the NLL loss function is simplified and adapted as

$$\mathrm{NLL}=\frac{1}{2mn}\sum_{i=1}^{n}\left(\left(\mathbf{q}_i-\hat{\mathbf{q}}_i\right)^2\oslash\hat{\boldsymbol{\sigma}}_i^2+\log\hat{\boldsymbol{\sigma}}_i^2\right).\tag{12}$$

However, as widely reported in the literature, training a mean-variance estimation network by simply minimizing NLL can result in a biased model towards data points where the model predicts well at the beginning of the training process [57–59]. Thus, we utilize a β-NLL loss function, which is defined to scale the gradient based on the prediction performance of each sample [58]. The β-NLL loss function adapted for the end-to-end optimization is defined as

$$\beta\text{-NLL}=\frac{1}{2mn}\sum_{i=1}^{n}\left(\left(\left(\mathbf{q}_i-\hat{\mathbf{q}}_i\right)^2\oslash\hat{\boldsymbol{\sigma}}_i^2+\log\hat{\boldsymbol{\sigma}}_i^2\right)\otimes\lfloor\hat{\boldsymbol{\sigma}}_i^{2\beta}\rfloor\right),\tag{13}$$

where β is a hyperparameter to balance between MSE and NLL, which is set at 0.5 based on the suggestion from the original work [58]. The symbol $\lfloor\cdot\rfloor$ denotes the operation of stopping gradient for the back-propagation during the training process. In addition to utilizing the modified NLL loss function, warm-up training epochs are also introduced to ensure a better starting model before learning the mean and variance simultaneously, which has been proven effective in improving mean predictions [59]. During the warm-up, all variance values are set to unity and the model solely learns the mean prediction because the β-NLL loss then becomes the MSE loss.

Once M individual models are trained, the ensemble prediction is obtained via a Gaussian mixture of a mean $\hat{\mathbf{q}}_i^*$ and variance $\hat{\boldsymbol{\sigma}}_i^{2^*}$ [60], which is formulated as

$$\hat{\mathbf{q}}_i^*=\frac{1}{M}\sum_{j=1}^{M}\hat{\mathbf{q}}_{i,j},\tag{14}$$

$$\hat{\boldsymbol{\sigma}}_i^{2^*}=\frac{1}{M}\sum_{j=1}^{M}\left(\hat{\boldsymbol{\sigma}}_{i,j}^2+\hat{\mathbf{q}}_{i,j}^2-\hat{\mathbf{q}}_i^{*^2}\right).\tag{15}$$

Strictly speaking, the probability distribution assumed in the NLL loss function for model training (Eqs. 12 and 13) and the Gaussian mixture for ensemble prediction (Eqs. 14 and 15) should be a truncated Gaussian distribution, where $\hat{\mathbf{q}}$ can neither be greater than one (i.e., the capacity fade with respect to the initial capacity cannot be negative) nor less than zero (i.e., the capacity of a cell cannot be negative). However, we still assume an unbounded Gaussian distribution for our implementation since this study focuses on the presentation of combining an empirical capacity model with machine learning algorithms for trajectory prediction, and our evaluation and comparison of the model performance are mainly based on the mean predictions. Also, by monitoring the training process, both physical constraints are rarely violated for training samples, but the violation may appear in some test samples, especially for the out-of-distribution (low-DoD) test samples. The implementation of NNE serves as a proof of concept that our proposed end-to-end optimization can be probabilistic, and NNE is just one approach to enable probabilistic prediction and uncertainty quantification. Standard probabilistic machine learning approaches, such as bootstrapping or Monte Carlo dropout, can also be implemented with the deterministic end-to-end optimization models introduced in the previous section for obtaining probabilistic predictions [61,62]. Thus, a more complete and thorough benchmark study on the probabilistic early-life trajectory prediction and uncertainty quantification is outside the scope of this study.

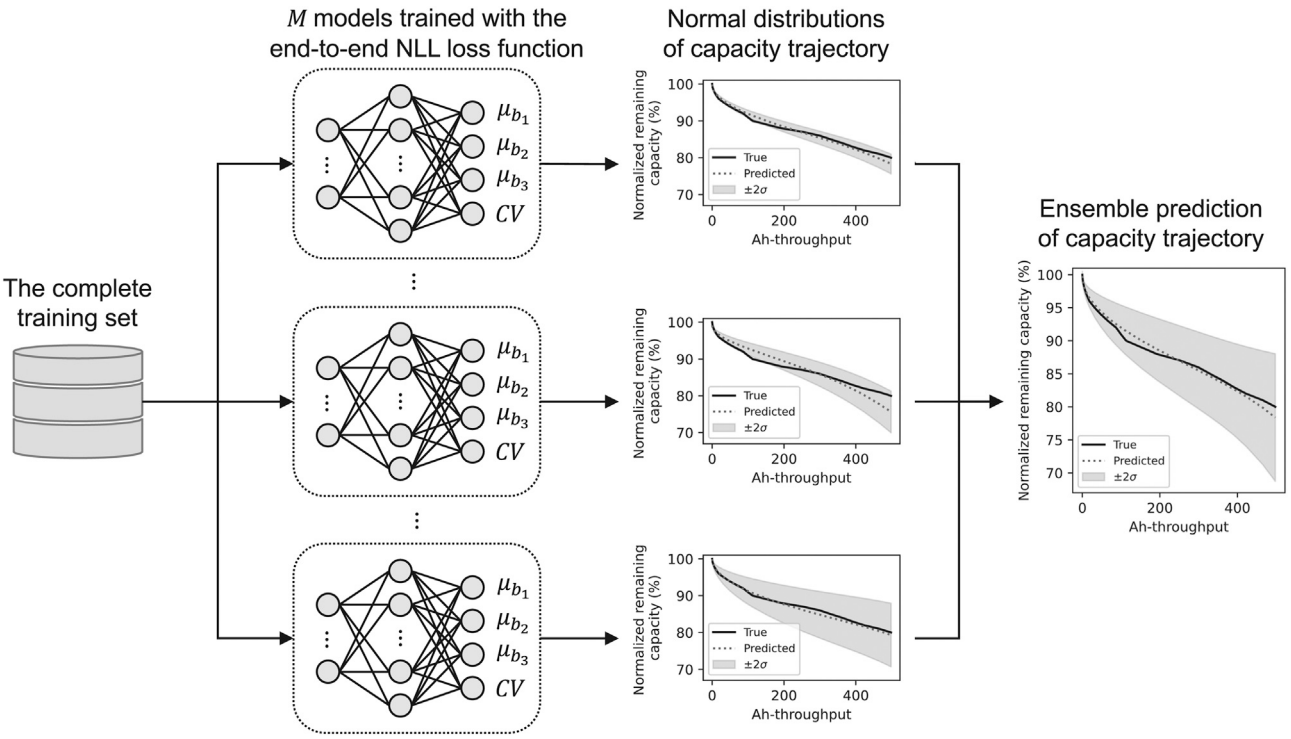


Fig. 6. A neural network ensemble, with M individual mean-variance estimation networks, for probabilistic predictions of the capacity trajectory.

5. Results and discussion

In this section, we present the capacity trajectory prediction accuracy and compare six different methods from the aforementioned early capacity-trajectory prediction approaches, which are informed by the empirical capacity model. The six methods are listed below:

- Knot-point based reconstruction with an MLP network (Knot-point)
- Sequential optimization with an elastic net regression (Seq-ENR)
- Sequential optimization with an MLP network (Seq-MLP)
- End-to-end optimization with an elastic net regression (E2E-ENR)
- End-to-end optimization with an MLP network (E2E-MLP)
- End-to-end optimization with a neural network ensemble of 5 MLP networks (E2E-NNE)

5.1. Implementation details of trajectory prediction methods

For all methods with MLP networks as the machine learning algorithm, we utilize Optuna framework [63], which is a comprehensive hyperparameter optimization framework, to help determine the number of layers, number of neurons, learning rate, weight decay, and batch size. For the E2E-NNE specifically, an additional hyperparameter of the warmup epoch number has been included in the optimization process. In the hyperparameter optimization, we perform 10-fold cross-validation based on groups (i.e., cells from a given group are only in the training or validation subset for each fold) and use the average of cross-validated errors as the objective to minimize. Since we have a very small training set of 92 samples, overfitting becomes a concern during network training for any given set of hyperparameters. To prevent overfitting, on top of limiting the maximum number of layers and number of neurons for each layer, the validation set of each fold is tracked to stop the training process when the validation loss is no longer decreasing (i.e., the early-stopping strategy for neural network training). The hyperparameter optimization results are listed in Section A.3. The adaptive moment estimation (Adam) algorithm is used as the optimizer for the training process.

For each cross-validation fold, the validation set is excluded from the training set, resulting in a different training subset for updating the weights in each fold. Therefore, we include the results of all 10 models (or 10 ensembles for E2E-NNE), each trained with the training subset of a given fold, for later evaluations in Section 5.3. For methods with elastic net regression as the machine learning algorithm (i.e., Seq-ENR and E2E-ENR), there is no need to exclude a validation set for early-stopping or any other evaluation during the training process once the hyperparameters are defined. So, we only train one model for these methods by fully utilizing the entire training set of 92 cells.

5.2. Error metrics

We utilize two commonly used error metrics for regression problems to evaluate the prediction performance of different empirical-model-assisted approaches on the capacity trajectory, namely the mean absolute error (MAE) and the root mean squared error (RMSE). These two metrics are in the same unit as the data, which is the normalized remaining capacity on a percentage scale. The MAE and RMSE for a given cell i are formulated as

$$\mathrm{MAE}_i = \frac{1}{m}\left\|\mathbf{q}_i - \hat{\mathbf{q}}_i\right\|_1 , \tag{16}$$

$$\mathrm{RMSE}_i = \sqrt{\frac{1}{m}\left\|\mathbf{q}_i - \hat{\mathbf{q}}_i\right\|_2^2}, \tag{17}$$

where $\mathbf{q}$ and $\hat{\mathbf{q}}$ are vectors of length m for the true and predicted capacity values at a given sequence of Ah-throughput for a cell, respectively. The error metrics are evaluated for each cell individually, and then the overall performance of a given model is taken by the mean of the error metric for all cells in a given subset (i.e., the mean MAE and the mean RMSE of a given subset). Mathematically, they can be expressed as

$$\overline{\mathrm{MAE}} = \frac{1}{n}\sum_{i=1}^{n}\mathrm{MAE}_i , \tag{18}$$

$$\overline{\text{RMSE}} = \frac{1}{n}\sum_{i=1}^{n} \text{RMSE}_i\,, \tag{19}$$

where n represents the number of cells in a given subset (training set, high-DoD test set, or low-DoD test set).

For probabilistic predictions from neural network ensembles, we include the continuous ranked probability score (CRPS) as an additional error metric. The CRPS measures the discrepancy between a predicted probability distribution and the true observed value by comparing their cumulative distributions. A lower CRPS indicates a more accurate probabilistic prediction. For a given cell i, CRPS is computed as

$$\text{CRPS}_i = \frac{1}{m}\sum_{j=1}^{m}\int_{-\infty}^{\infty}\left(F(x;\hat{q}^*_{i,j},\hat{\sigma}^*_{i,j}) - H(q_{i,j})\right)^2 dx\,, \tag{20}$$

where $F(x)$ is the cumulative distribution function defined by the predicted mean and standard deviation, and $H(\cdot)$ is the Heaviside step function centered at the true value $q_{i,j}$. Similarly, the average CRPS across all cells within a given data subset can be expressed as

$$\overline{\text{CRPS}} = \frac{1}{n}\sum_{i=1}^{n} \text{CRPS}_i\,. \tag{21}$$

5.3. Trajectory prediction performance

The overall prediction performance for all six methods is summarized in Table 1 and Fig. 7. As mentioned in Section 5.1, for those methods with MLP networks as machine learning algorithms, the mean values of prediction error for each method across 10 models/ensembles, one for each fold, are reported in Table 1, and the prediction errors for all 10 models/ensembles are visualized in Fig. 7 as box plots. For methods based on elastic net regression, only one model is trained for each method. Thus, no variation is shown in Fig. 7. Predicted trajectories from different methods for selected cells are included in Section A.5.

The first benchmarking method we consider is the knot point-based prediction with an MLP network. The neural network in this method learns the nonlinear relationship between features and the degradation rate within different health ranges. However, we observe a larger variation in the degradation trends in this dataset compared to other public datasets, which makes this approach hard to perform well. From Fig. 7, we can see a noticeable fold-to-fold variation in the training error compared to all other methods, which is even worse for the test errors over the two test sets. This observation indicates three issues that hinder the overall performance of this approach. First and foremost, we have a small aging dataset with wide-varying aging trends, which requires more data to allow sufficient learning of the degradation rates within different ranges with respect to the early life features. Second, there is a lack of information about the dependency of the degradation rate of a given health range on the history because the neural network outputs the Ah-throughput difference between each pair of subsequent knot points independently. Third, there is an issue with curve-fitting on a very small amount of points, especially when a fitted curve needs to be extrapolated beyond the data points it was fitted to.

Table 1
Summary of overall capacity trajectory prediction errors from different empirical-model-informed methods, reported in the unit of normalized capacity at % scale.

Method	$\overline{\text{MAE}}$			$\overline{\text{RMSE}}$		
	Training	High DoD	Low DoD	Training	High DoD	Low DoD
Knot-point	2.77 %	3.82 %	15.99 %	4.25 %	5.86 %	22.05 %
Seq-ENR	2.32 %	2.63 %	13.62 %	3.08 %	3.46 %	19.28 %
Seq-MLP	2.32 %	3.00 %	15.98 %	3.06 %	4.05 %	21.73 %
E2E-ENR	**1.43 %**	1.97 %	**3.77 %**	**1.71 %**	2.46 %	**4.39 %**
E2E-MLP	1.61 %	**1.96 %**	5.92 %	1.93 %	**2.38 %**	7.62 %
E2E-NNE	1.70 %	2.00 %	6.44 %	2.10 %	2.45 %	8.05 %

The sequential optimization approach is implemented by using two different machine learning algorithms, one is a multitask elastic net model, and the other one is an MLP network, to map from the early-life features to the fitted empirical model parameters. From the overall results listed in Table 1 and Fig. 7, these two methods also exhibit high prediction errors towards the low-DoD test set, at a similar magnitude to the Knot-point method but with a much smaller fold-to-fold variation for the Seq-MLP. Both training errors and test errors on the high-DoD test sets are lower in both overall magnitude and variation compared to the Knot-point method but are still higher than the proposed end-to-end approach. There are multiple sources where the sequential optimization methods are limited in performance and could fail when extrapolating to out-of-distribution test samples. For the Seq-ENR method, a linear regression model cannot sufficiently learn the relationship between PCA features and individually fitted parameters. Meanwhile, for the Seq-MLP method, given that a simple MLP network is trained with mini-batches and the early-stopping mechanism, it is likely that the feature-label relationship learned from the training subset doesn't reflect well on that of both test sets. Also, the machine learning loss that is minimized during the sequential optimization training process is solely on predicting parameters, which is not directly scalable or mappable to the error of the entire trajectory (e.g., 1 % error in parameters may result in more

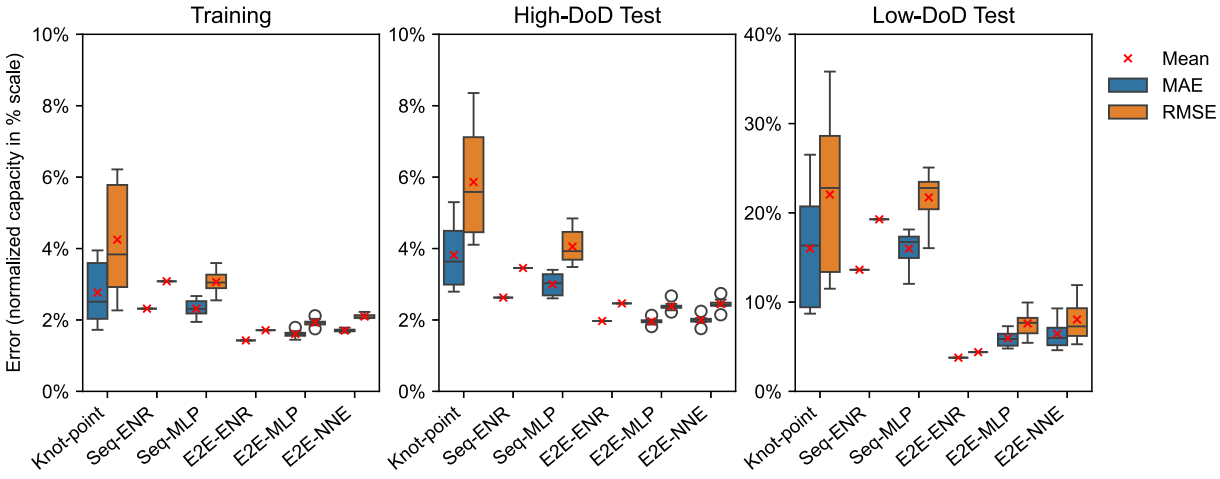


Fig. 7. Capacity trajectory prediction error of different empirical-model-informed methods for cells in three subsets. The red cross indicates the mean.

than 1 % error in the trajectory prediction). Furthermore, in the first optimization problem that minimizes the empirical model fitting error on individual cells, the optimal set of empirical parameters for each cell is not necessarily the global optimum given the non-convex nature of fitting the empirical model in Eq. (2). Under an assumption that the empirical parameters should correlate with the early life features, either in a linear or a nonlinear fashion, the obtained local minima may introduce more errors or deviations in the training labels (fitted parameters of the training set) compared to an "ideal" relationship. Or, in an intuitive way, the locally optimal fitted parameters may result in a hard-to-learn pattern with respect to the extracted features.

In contrast, the end-to-end optimization has shown better performance during the training and over the two test cases. The E2E-ENR method targets to balance optimizing a linear relationship between the early-life features and the empirical parameters and optimizing the capacity trajectory prediction from the predicted parameters. It is also similar for the E2E-MLP and E2E-NNE methods while the relationship to find can be non-linear. For the test accuracy on the high-DoD test set, all three methods have similar performance based on the two defined error metrics, where the mean MAE is less than 2 % and the mean RMSE is less than 3 %. However, by considering the out-of-distribution test performance, E2E-ENR outperforms all its peers and shows a higher stability in extrapolation. This can be attributed to the benefit of a simple linear model where the performance of extrapolation is better than a complicated, larger, non-linear NN model. In addition, since all machine learning models in this study have a very small training set to learn from, a key consideration for building a model is the ratio between the number of samples and the number of trainable parameters. A practical rule of thumb for achieving good generalization performance and avoiding overfitting to the training data on a small dataset is that the number of parameters should be limited to 10 % of the number of training data points (or 10-to-1 data points-to-parameters rule) [64]. Such a rule is to ensure that a model is provided with a sufficient amount of data to span through every dimension. In this study, the linear model has much fewer parameters to optimize during model training (only 30 weights plus three intercepts for E2E-ENR) compared to its MLP-based counterparts, and the ratio between the number of training samples and the number of trainable parameters for each output is close to the 10-to-1 rule.

Furthermore, we can see that the E2E-NNE has a relatively larger fold-to-fold variation than E2E-MLP from Fig. 7. This can be partially attributed to the difference in the loss functions, where an MSE loss function minimized in the E2E-MLP method solely focuses on providing accurate mean predictions, but an NLL loss function minimized in the E2E-NNE method focuses on providing accurate predictions for probability distributions. Also, during the training process, more randomness is introduced in the E2E-NNE due to the random initialization of weights and the random shuffling of mini-batches for each individual model. If the randomness results in multiple poorly-performed individual models, the randomness can aggregate through the ensemble, resulting in an even higher error. This can be backed up by the observation in Section 5.5, in which we have an ensemble of 10 individually trained models for each fold, and the overall performance in terms of the mean predictions worsens.

5.4. Analysis of empirical model parameter predictions

For all the methods included in this study, the ultimate goal of incorporating machine learning is to obtain a prediction of the battery capacity trajectory via the predicted empirical model parameters (as shown in Fig. 5). To better understand the performance of all six methods, it is worth visualizing the space of predicted parameters for each of them. It is worth noting that, as shown in Fig. 8, there are distinct differences. Both Seq-ENR and E2E-ENR have narrower distributions with clear linearity due to the nature of the machine learning algorithm. However, the trajectory prediction performances are significantly different, where E2E-ENR has a very stable performance in both test cases. This observation signifies that having a learnable pattern enables simple models to perform well on a complicated prediction problem, which is the main benefit provided by the end-to-end optimization framework. For Knot-point and Seq-MLP methods where MLP networks are used, the parameter space is sparse, and such sparsity mainly comes from the training dataset, which can be backed up by the lowest training errors among all methods. When the parameter space is sparse and the dataset is small, it is challenging for a machine learning model to learn the underlying relationships well. On the other hand, the E2E-MLP method shows a very learnable pattern that is almost perfectly linear, but this linear pattern is totally different from the linear pattern from E2E-ENR. One contributor to this difference is that, for the E2E-MLP method, the output of the neural network (empirical parameters) is two-sided constrained by using scaled sigmoid activation function, while the E2E-ENR only has one-sided constraint (empirical parameters are positive). However, these two methods share very similar training accuracy, which highlights the nature of a non-convex objective for the end-to-end optimization.

5.5. A parametric study on neural network ensemble for probabilistic predictions

To better understand the performance of the neural network ensemble, additional experiments were conducted to evaluate the performance of the individual models trained with the β-NLL loss function ($M = 1$) and an ensemble of 10 individual models ($M = 10$) for each fold. The results from this experiment are included in Table 2, and box plots in Fig. 9 showcase the fold-to-fold variation on performance metrics among all three cases ($M = 1$, $M = 5$, and $M = 10$). In addition to comparing the trajectory prediction accuracy using the mean of the probability distribution, we include two more metrics specifically for evaluating probabilistic predictions: one is the CRPS, and the other one is the expected calibration error (ECE) of a calibration curve. The CRPS allows for a general evaluation of how well the predicted Gaussian distribution aligns with the observation. Then, the calibration curve (also known as the reliability curve) is introduced to better analyze the probabilistic predictions from these three models with different confidence levels, as shown in Fig. 10. A calibration curve consists of K discrete confidence levels, and at each level, the ratio of samples within the confidence interval on predicted distributions (the observed confidence) is plotted against the expected confidence. A calibration curve plot can be interpreted with two regions: one is the overconfident region where the observed confidence is less than the expected confidence (i.e., the lower right region to the ideal prediction line), and the other one is the underconfident region where the observed confidence is higher than the expected confidence (i.e., the upper left region to the ideal prediction line) [61]. Or, as an alternative way of understanding the calibration curve, an underconfident calibration means the model predicts a wider prediction interval (an upper bound and a lower bound for a given set of input) at a given confidence level than it is supposed to be, while an overconfident calibration means the model predicts a narrower prediction interval at a given confidence interval than it is supposed to be. ECE is used to quantify the calibration error, which can be calculated as $ECE = \sum_{i=1}^{K}(|c_i - \hat{c}_i|/K)$, where c_i and $\hat{c}_i$ are the expected confidence and the observed confidence at the i-th confidence threshold from the K discrete cutoffs, respectively.

By ensemble of multiple models, we observe improvements over the two test sets. Even though the mean prediction accuracy values, measured by MAE and averaged across all 10 folds, are similar, the fold-to-fold variation shrinks for the two ensembles. However, by comparing the MAE of $M = 5$ and $M = 10$ cases on the low-DoD test set, we notice that an ensemble of more individual models does not necessarily

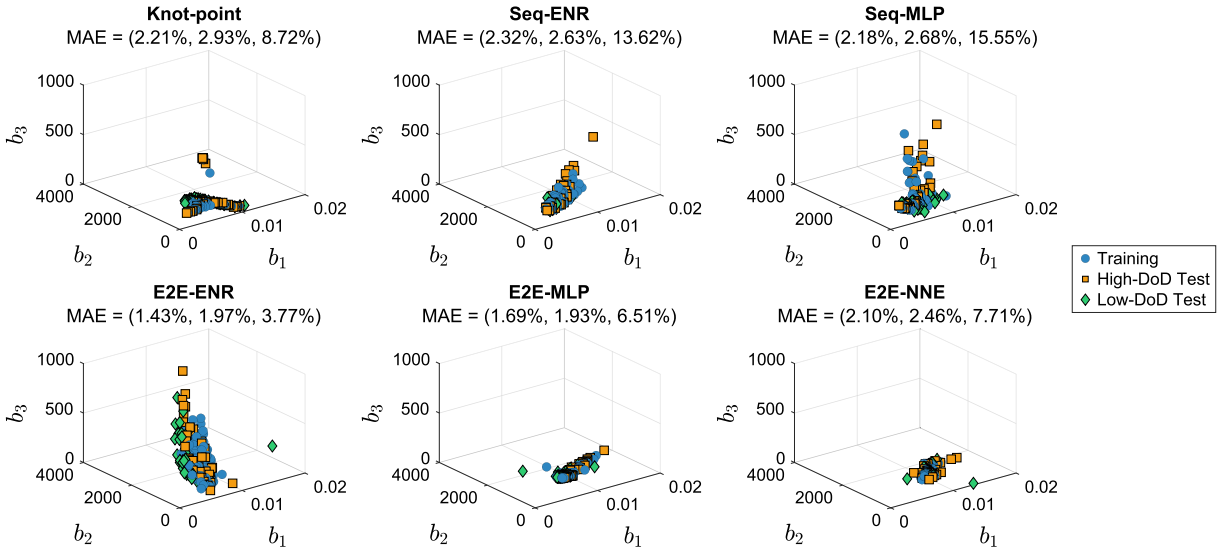


Fig. 8. Distribution of predicted empirical model parameters from six different methods. For MLP-based methods, the predicted parameters from one fold are visualized in this figure. The MAE values of trajectory predictions from a given distribution are listed in the parentheses for the training set, the high-DoD test set, and the low-DoD test set, respectively.

Table 2
Summary of capacity trajectory prediction errors and expected calibration errors for end-to-end neural network ensembles (E2E-NNE) with varying ensemble sizes M. Results are averaged over 10-fold cross-validation.

M	$\overline{\text{MAE}}$			$\overline{\text{CRPS}}$			ECE		
	Training	High DoD	Low DoD	Training	High DoD	Low DoD	Training	High DoD	Low DoD
1	1.85 %	2.43 %	7.84 %	1.46	1.86	5.72	0.058	0.067	0.223
5	1.70 %	2.00 %	6.44 %	1.41	1.63	4.16	0.100	0.078	0.094
10	1.73 %	2.05 %	7.14 %	1.44	1.66	4.65	0.102	0.077	0.087

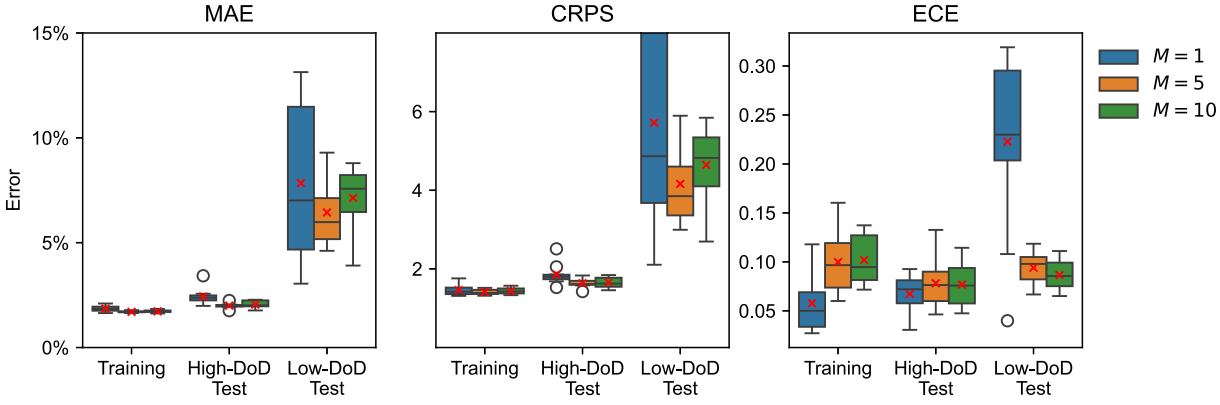


Fig. 9. Performance of end-to-end neural network ensembles (E2E-NNE) using different ensemble sizes (M). Each ensemble consists of M individual mean-variance networks.

improve the prediction performance, where the interquartile range of MAE for the low-DoD test set shifts up (worsen) on $M = 10$. The quality of probabilistic predictions evaluated by CRPS and ECE also improves significantly for ensembles of multiple individual mean-variance estimation models, especially for the low-DoD test set that provides out-of-distribution samples. In other words, neural network ensembles are able to capture higher uncertainty towards out-of-distribution samples and provide more reliable probabilistic predictions.

From the calibration curves, it is clear that neural network ensembles ($M = 5$ and $M = 10$) tend to be underconfident for the training and high-DoD test sets (i.e., more samples observed in a given confidence interval than expected) but slightly overconfident on the low-DoD test sets when we set the expected confidence to a low level (i.e., fewer samples observed in a given confidence interval than expected). This is mainly due to the limited size of training samples and an ideal assumption of a constant coefficient of variation when we formulate the customized NLL loss function. As a main takeaway of this parametric study, choosing a proper number of individual models for the ensemble is needed to balance between the accuracy of mean predictions for a regression problem and the accuracy of the prediction interval. In practice, if the model is given a large enough training set that covers most of the test

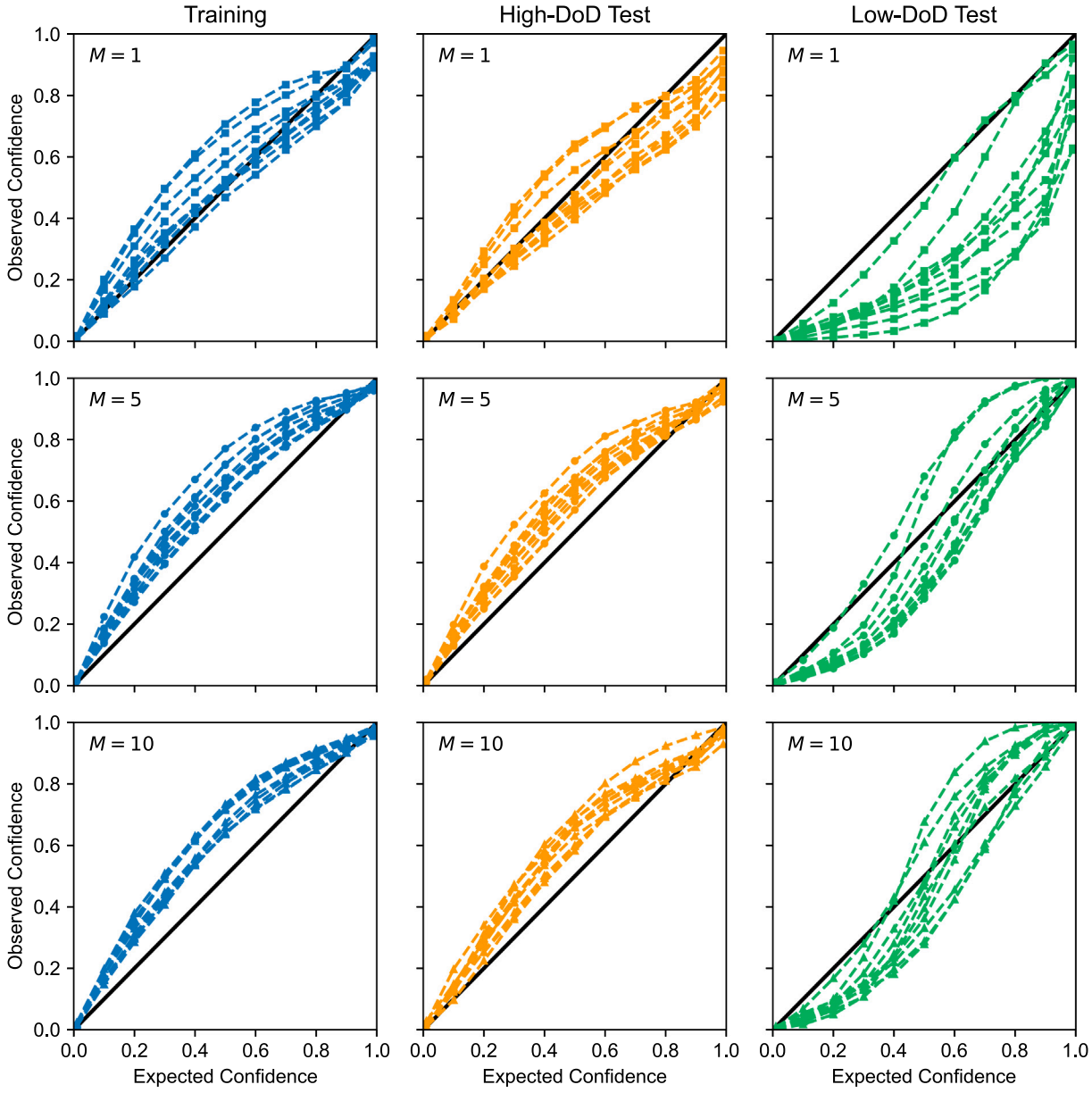


Fig. 10. Calibration curves for the E2E-NNE method with different numbers of individual models M on the three subsets.

sample ranges, the proposed E2E-NNE method should provide robust probabilistic predictions and uncertainty quantification.

5.6. A parametric study on feature extraction intervals

Another parametric study we investigated is the effect of different RPT intervals for early-life feature extraction, denoted as $w_i - w_0$, on the performance of capacity trajectory prediction. In particular, we intend to answer the question – how early can our proposed end-to-end optimization framework be to achieve an accurate prediction of the capacity trajectory. So, we applied the identical feature extraction and decomposition approach described in Section 4.1 but from the RPT after one week of cycling (w_1) up to the RPT after 6 weeks of cycling (w_6). It is worth noting that the RPT after 4 weeks of cycling (w_4) is omitted due to missing data for a batch of cells. The results obtained for this parametric study are based on the E2E-ENR method, and the prediction performance on three data subsets is visualized in Fig. 11.

From the results, we can easily observe that the training accuracy is consistent for all five different intervals. Both $w_2 - w_0$ and $w_3 - w0$ have slightly higher overall accuracy compared with the other three intervals on the high-DoD test samples. However, for the $w_2 - w_0$ interval, there is a significant increase in the prediction error on the low-DoD test samples. To better understand the difference in using different feature sets, we visually inspect the predicted parameter spaces, as shown in Fig. 12. It is clear that, for each predicted space, the predicted empirical model parameters for the high-DoD test samples are distributed with a similar pattern as the distribution of training samples. However, for $w_2 - w_0$, the distribution of parameters for low-DoD samples are much concentrated in which b_2 and b_3 are smaller, while for other models, the values of b_2 and b_3 for low-DoD samples are either similar to or larger than those for the training samples. Smaller b_2 and b_3 compared to the high-DoD samples are not expected for the low-DoD samples given that a larger b_2 delays the onset of the knee effect and a larger b_3 enables a more gentle degradation trend beyond the knee point. Such an observation indicates that the weights (or the input–output relationship

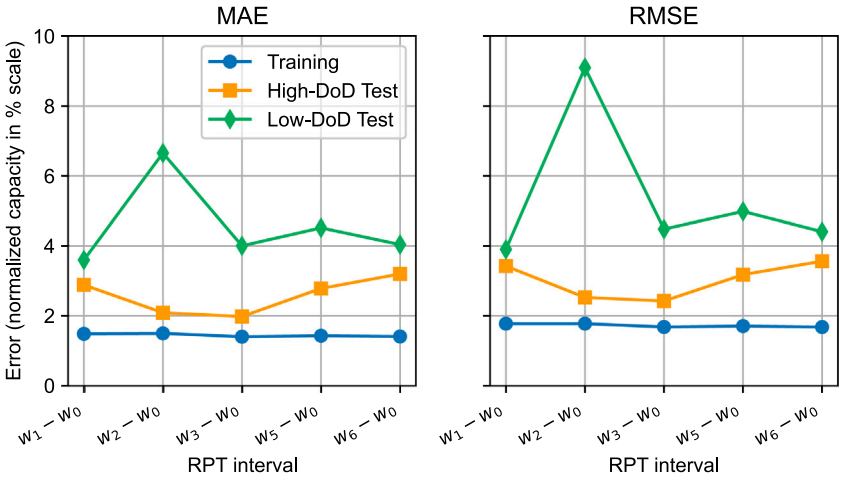


Fig. 11. Prediction performance of the E2E-ENR method with PCA features extracted from different RPT intervals.

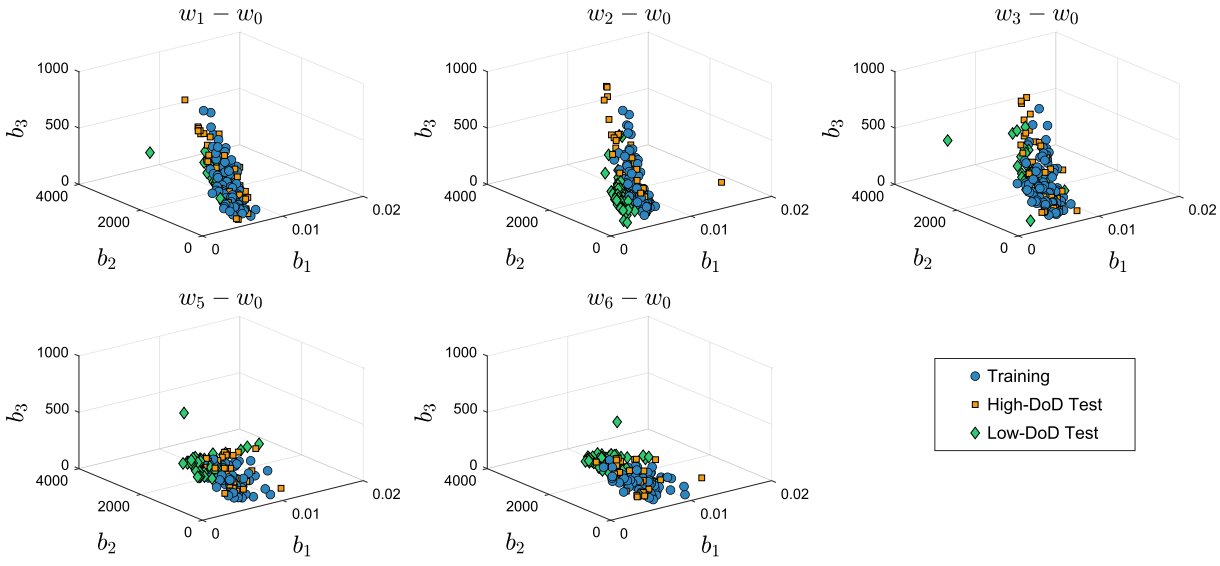


Fig. 12. Distribution of predicted empirical model parameters from five different RPT intervals for feature extraction using the E2E-ENR method.

in general) learned are not always guaranteed to extrapolate well to a different feature distribution.

Although there is no significant difference in the overall training accuracy, $w_5 - w_0$ and $w_6 - w_0$ have distinct differences in the pattern compared with the other three intervals. This observation can be mainly attributed to an inevitable change in early-life feature distributions. After five to six weeks of cycling, some fast-aging cells may have already transitioned to a different stage of degradation (e.g., reaching the knee onset or even beyond the knee point) and start experiencing different degradation mechanisms (e.g., particle cracking, loss of electrical contact, etc.) due to prolonged exposure to strenuous cycling conditions compared to their slow-aging peers. These changes in underlying degradation dynamics would significantly alter the degradation information captured by the extracted early-life features, thereby affecting the relationship between features and the empirical model parameters.

In general, unlike the original study of this dataset on EOL, where the model performance improves steadily when the RPT interval for feature extraction expands [16], the proposed methodology in this study is relatively agnostic to the RPT interval between RPTs. There are two factors contributing to the agnostic performance. First, using PCA to decompose the raw features derived from measurements can extract as much information as possible from the entire feature space to describe the variance in training samples. Second, the proposed end-to-end optimization can find its best way to minimize the overall error aggregated from (1) the machine-learning error between input features and empirical model parameters and (2) the curve-fitting error between empirical model parameters and capacity trajectories.

6. Additional benchmarking studies

In this section, we present two benchmarking studies: (1) comparing the proposed end-to-end method with a state-of-the-art method and (2) demonstrating the generalizability of our proposed method to a dataset with a different battery chemistry and form factor.

6.1. A benchmarking study comparing with existing early prediction approaches

Our first benchmarking study compares the proposed neural network ensemble method (E2E-NNE) with an LSTM ensemble method proposed by Rieger et al. [47]. Both methods use an NNL loss function,

allowing individual models in the ensemble to learn and predict a normal distribution for each target observation. However, they differ in input and output representations and model structure. This LSTM ensemble method (LSTM-E) considers two sets of input: a fixed-length sequence of capacity measurements immediately preceding the prediction timestamp t and early-life features, specifically PCA features from the first three weeks of cycling. The training process of an individual LSTM model is designed as follows. *First*, the sequence data are passed through an LSTM layer before being concatenated with early-life features. *Second*, the concatenated features undergo transformation through a hidden layer with the ReLU activation. *Finally*, the model outputs the predicted mean and variance of a normal distribution describing the capacity ratio between the current step and the next step (Q_{t+1}/Q_t). In this problem specifically, the model is trained for one-step-ahead prediction, and the capacity trajectories are interpolated based on a fixed step size of 5 Ah-throughput. Multi-step predictions are recursively generated for a cell until the cell reaches its EOL: the predicted capacity value at each given step is appended to the observed trajectory and used as input for the subsequent step. After multiple individual models are trained, the final ensemble prediction is obtained by combining predictions from the individual models in the form of a Gaussian mixture, as described in Eqs. (14) and (15).

Both ensemble methods in this benchmarking study are trained using 10-fold cross-validation with ensembles of five individual models. For each fold, the held-out validation subset will be used to trigger early stopping if the validation error stagnates or increases. The mean prediction performance of both E2E-NNE and LSTM-E across all 10 folds is summarized in Table 3, and the fold-to-fold variation of these performance metrics is visualized in Fig. 13. Overall, the prediction accuracy of E2E-NNE, evaluated using MAE and CRPS, is comparable to that of the state-of-the-art LSTM-E method. E2E-NNE outperforms in the high-DoD test set, and LSTM-E outperforms in the low-DoD test set, both marginally. From an uncertainty calibration standpoint, E2E-NNE exhibits a slightly lower calibration error (ECE) than LSTM-E across all three subsets. Moreover, when looking at the calibration curves of both methods in Fig. 14, it is clear that LSTM-E tends towards overconfidence, while E2E-NNE is generally underconfident on both training and high-DoD test sets. In practice, underconfident models are often preferred for early prediction tasks, given the limited input information and long prediction horizons relative to available observations. Although the neural network ensemble variant of our proposed end-to-end approach does not significantly improve prediction accuracy over LSTM-E, one key benefit of E2E-NNE is its flexibility. Unlike LSTM-E, the proposed method can accommodate different machine learning algorithms, which can be as simple as linear regression, enhancing its adaptability to diverse predictive settings with varying data quantities.

6.2. A benchmarking study on battery with a different chemistry and form factor

In Section 5, we demonstrate the proposed end-to-end optimization framework on the ISU-ILCC battery aging dataset with diverse degradation trends. To better showcase that the end-to-end framework can be applied to different test cases, we use a dataset published by Stroebl et al. [65], which uses cells with different chemistry and form factor. This dataset consists of 279 Samsung INR21700-50E cylindrical cells with a nominal capacity of 4.9 Ah and lithium nickel cobalt aluminum oxide (NCA) as the active material on the positive electrode [66]. Among all cells, 49 groups of cells (three cells per group) were subjected to various cycling conditions defined by five design variables, namely the charge rate, discharge rate, ambient temperature, maximum SOC, and DoD. Since cells were not fully discharged in each cycle, RPTs were performed at predefined time intervals to obtain universal full-DoD capacity measurements for analysis. Also, the experimental design for this dataset consists of two batches of cells (referred to as Stage 1 and Stage 2 in the data descriptor), each following a distinct sampling strategy for defining cycling conditions. More details about the aging study design and experimental procedures can be found in the data descriptor [65].

With a wide range of design variables, the observed capacity trajectories in this dataset exhibit significant diversity. However, since the charge rate and discharge rate are within the manufacturer's specifications and the SOC level during cycling is constrained between 20 % and 80 %, the overall degradation rate is much slower than in other accelerated aging datasets such as the ISU-ILCC dataset [50] and the 124-cell LFP dataset [6]. As a result, cycling tests were terminated at various remaining capacity levels, with many cells stopped from testing

Table 3
Comparison of capacity trajectory prediction errors and expected calibration errors between LSTM ensembles (LSTM-E) and neural network ensembles (E2E-NNE), both with five individual models $M = 5$ (mean of the 10 folds).

Method	$\overline{\text{MAE}}$			$\overline{\text{CRPS}}$			ECE		
	Training	High DoD	Low DoD	Training	High DoD	Low DoD	Training	High DoD	Low DoD
LSTM-E	1.44 %	2.30 %	5.14 %	1.06	1.67	3.52	0.120	0.141	0.107
E2E-NNE	1.70 %	2.00 %	6.44 %	1.41	1.63	4.16	0.100	0.078	0.094

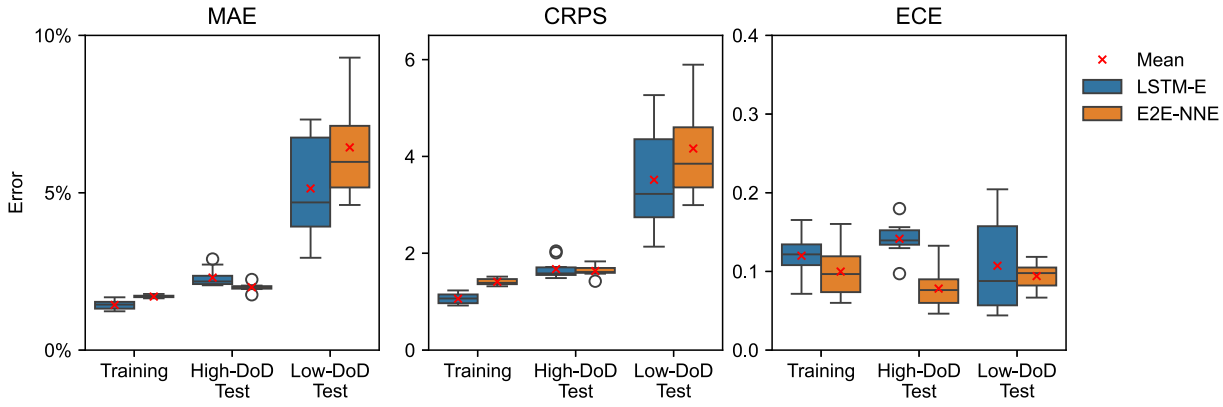


Fig. 13. Fold-to-fold performance variation of LSTM-E and E2E-NNE, evaluated by three metrics for probabilistic predictions.

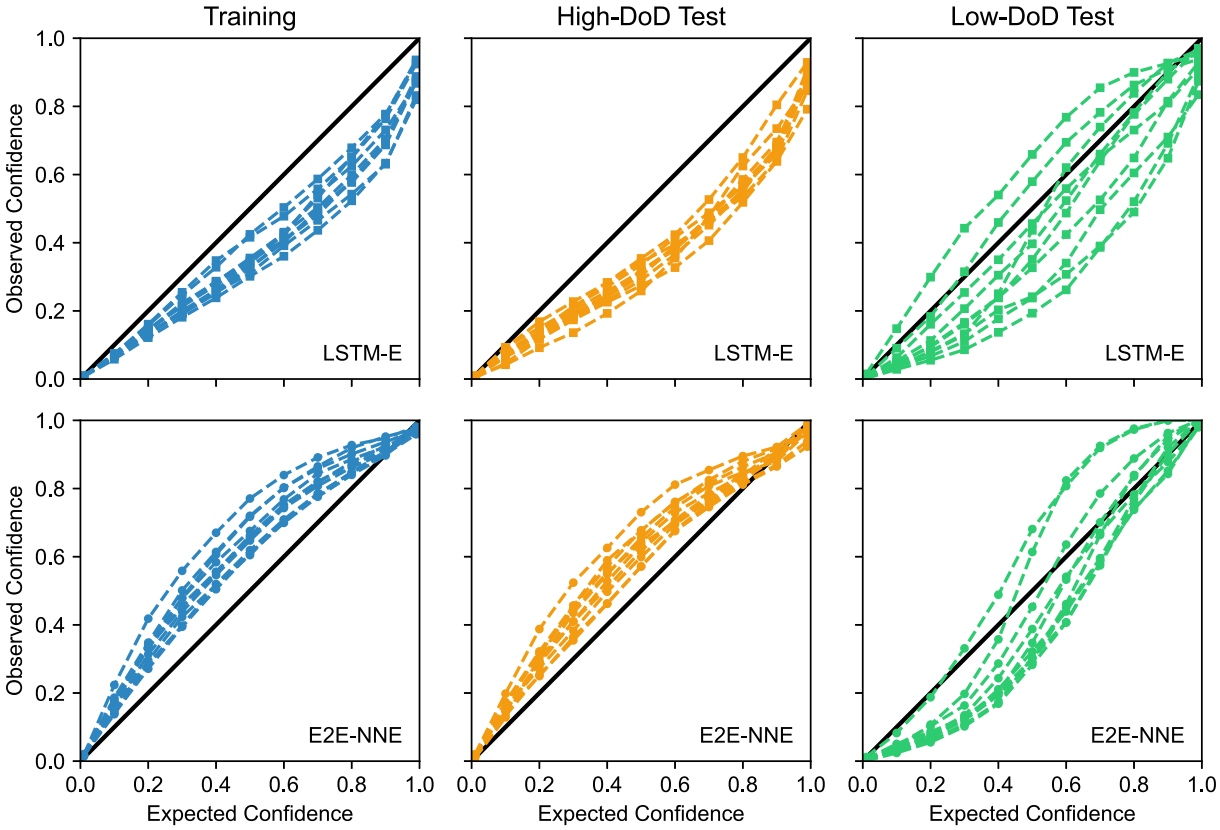


Fig. 14. Calibration curves of LSTM-E and E2E-NNE on the three subsets.

before reaching 90 % remaining capacity. So, to retain as many cells as possible for the benchmark study while relying solely on true RPT measurements for remaining capacity, we set the EOL for this dataset at 95 % and filtered all cells with no capacity measurements beyond the EOL. All remaining cells from Stage 1 form the training set, and cells from Stage 2 are considered the test samples. The observed trajectories from both subsets of cells can be found in Fig. 15. For this dataset, the end-to-end framework utilizes a power-law empirical model, which can be expressed as

$$Q(N; c_1, c_2) = 1 - c_1 N^{c_2}, \tag{22}$$

where c_1 and c_2 are cell-specific parameters that machine learning models predict using input features.

Before training early prediction models, the early-life data must be defined. Given that the earliest available RPT occurs after one week of cycling, we extract five early-life features from data collected during the initial RPT and the week 1 RPT. We also use the five cycling condition variables as additional input features. The first two early-life features come from the well-known capacity–voltage curve difference [6], which are $\log(|\text{mean}(\Delta Q_{\text{w1}-\text{w0}}(V))|)$ and $\log(|\text{var}(\Delta Q_{\text{w1}-\text{w0}}(V))|)$. The other three early-life features are from the voltage-windowed incremental capacity curve differences [16], which are $\log(|\text{mean}(\Delta dQ/dV^{2.5\text{ V}-3.4\text{ V}}_{\text{w1}-\text{w0}}(V))|)$, $\log(|\text{mean}(\Delta dQ/dV^{3.4\text{ V}-3.7\text{ V}}_{\text{w1}-\text{w0}}(V))|)$, and $\log(|\text{mean}(\Delta dQ/dV^{3.7\text{ V}-4.0\text{ V}}_{\text{w1}-\text{w0}}(V))|)$. The mid-voltage range between 3.4 V and 3.7 V aligns with the major peak on the incremental capacity curve (see Fig. A.2). At the time of this work, no published early prediction models exist for this dataset. Since this benchmarking study aims to conduct a preliminary investigation on the feasibility of the proposed end-to-end method on an aging dataset with different chemistry, feature engineering on this dataset is not extensively optimized, and we acknowledge that the results of this benchmarking study are by no means the best of what we can get from this dataset.

In this preliminary benchmarking study, we consider three test cases with different features for the elastic-net-based end-to-end framework (E2E-ENR).

- The first case uses only the five parameters characterizing the cycling conditions, excluding any cell-specific measurements.
- The second case directly uses the five early-life features defined above.
- For the third case, we concatenate the five cycling condition parameters with the five early-life features and reduce the dimensions of features from ten to five using PCA transformation.

To ensure a sufficient sample-to-feature ratio, we limit the number of PCA-transformed features to five, following the 10-to-1 sample-to-feature ratio rule [64]. Limiting the feature size is particularly critical for this benchmark study due to the smaller training sample size than the ISU-ILCC dataset. The prediction results from these three test cases are summarized in Table 4, and the predicted trajectories for selected cells are shown in Fig. A.8. The end-to-end approach achieves good accuracy by only utilizing the cycling condition parameters as inputs. However, a comparison of the training and test errors reveals significant differences in both MAE and RMSE, suggesting potential overfitting. This can likely be attributed to the discrete and sparse nature of cycling conditions, as the experimental design follows a grid-based sampling strategy – only 25 samples are available in a 5-dimensional design space. Incorporating the early-life features largely reduces the discrepancy between the training and test errors, but the overall error magnitude becomes slightly higher. The minimal improvement observed when adding the cell-specific features suggests that the selected five early-life features are not optimized for predictive performance in this new dataset. Further refinement of feature engineering is left for future studies.

It is important to note that the error magnitudes reported in Table 4 are not directly comparable to those reported in other tables within

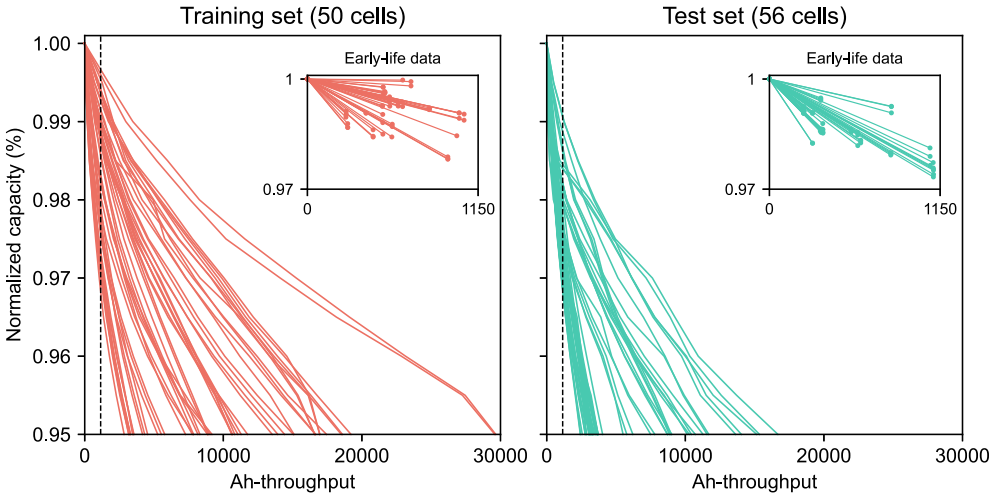


Fig. 15. Capacity trajectories of cells from the public NCA dataset after preprocessing. The black dashed line highlights the Ah-throughput value of 1150, indicating the threshold before which all early-life data are collected.

Table 4
Summary of overall capacity trajectory prediction errors by using different feature sets, reported in the unit of normalized capacity at % scale.

Features	$\overline{\text{MAE}}$		$\overline{\text{RMSE}}$	
	Training	Test	Training	Test
Cycling conditions	0.27 %	0.61 %	0.33 %	0.76 %
Early-life features	0.60 %	0.70 %	0.72 %	0.83 %
PCA features	0.46 %	0.69 %	0.55 %	0.87 %

this paper (MAE and RMSE reported in Table 1 as well as MAE reported in Tables 2 and 3). For instance, a 1 % error in the unit of normalized capacity in Table 4 corresponds to 20 % of the targeted prediction range (from 100 % to 95 % normalized capacity). Conversely, the EOL threshold for the ISU-ILCC dataset is at 80 % normalized capacity, meaning that an equivalent 1 % error in the NCA dataset corresponds to a 4 % error in the ISU-ILCC dataset. Considering this difference in the EOL threshold, the prediction accuracy of E2E-ENR on the NCA test subset is at a similar level to that of E2E-ENR on the low-DoD test set of the ISU-ILCC dataset.

7. Conclusion

This work explores three approaches incorporating empirical capacity fade models for machine learning-based early-life battery capacity trajectory prediction. The empirical capacity fade model used in this study incorporates long-term degradation trajectory information into the machine learning model fitting process, leading to improved extrapolation on out-of-distribution test samples. The proposed end-to-end learning framework achieves less than 2 % MAE on in-distribution samples (high-DoD test set) and less than 4 % on out-of-distribution samples (low-DoD test set). At the same time, comparable baseline approaches (the knot point-based approach and the sequential optimization approach to trajectory prediction) exhibit greater than 10 % MAE on out-of-distribution test samples. Such distinct performance differences highlight the benefits of coupling empirical battery capacity fade models with machine learning during the training process.

However, there are still some limitations to this work. First, the capacity fade model we use can only capture the cyclic aging behaviors, and the calendar aging part is neglected, which is also an important contributor to battery degradation. Second, a thorough study of the probabilistic prediction of capacity trajectory and uncertainty quantification using the proposed end-to-end framework is yet to be conducted. Third, the proposed end-to-end framework is currently limited to accommodating one empirical capacity model at a time, which partially contributes to lower prediction accuracy towards out-of-distribution samples. An ideal early prediction model should be more generalizable and robust, distinguishing between different degradation trends (e.g., two-stage vs. three-stage degradation) based on early-life data and providing more accurate predictions across varying degradation trends. This remains an open problem that merits further investigation by researchers in the field. In addition, a critical knowledge gap exists in the application domain of early life prediction methods. Specifically, there is a need to further our understanding of utilizing early trajectory prediction to optimize the design of aging experiments for accelerating the process of capacity fade modeling. We will explore this new application in future work.

CRediT authorship contribution statement

Tingkai Li: Writing – original draft, validation, software, methodology, investigation, and formal analysis. **Jinqiang Liu:** Writing – review & editing, software, methodology, investigation, and formal analysis. **Adam Thelen:** Writing – review & editing, methodology, and investigation. **Xiao-Guang Yang:** Writing – review & editing and methodology. **Zhaoyu Wang:** Writing – review & editing, supervision, and conceptualization. **Chao Hu:** Writing – review & editing, supervision, project administration, methodology, funding acquisition, and conceptualization.

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

This work was supported in part by Iowa Economic Development Authority under the Iowa Energy Center Grant Number (20-IEC-018 and in part by the US National Science Foundation under Grant ECCS-2015710. Any opinions, findings, or conclusions in this paper are those of the authors and do not necessarily reflect the sponsors' views.

A. Appendix

A.1. Details of early-life features extracted from RPT data

A total of 29 features were extracted from six different sources for the ISU-ILCC dataset. A more complete description and discussion of these features can be found in the previous work [16]. For the features extracted from curves windowed by voltage cutoffs, there are three sets of bounds: a low-voltage window between 3.0 V and 3.6 V, a mid-voltage window between 3.6 V and 3.9 V, and a high-voltage window between 3.9 V and 4.2 V (Fig. A.1)

A.2. Proof of end-to-end optimization performance

For the end-to-end framework problem, we can obtain that

$$\begin{aligned}
&\min_{\mathbf{b_1},\mathbf{b_2},\mathbf{b_3}} \Big[\frac{1}{2nm}\|\mathbf{1}_{n\times m} - \mathbf{b_1}\mathbf{1}_{1\times m}\otimes\mathbf{N}^a \\
&\quad -\mathbf{1}_{n\times m}\oslash\left(\mathbf{1}_{n\times m}+\exp\left(\mathbf{b_2}\mathbf{1}_{1\times m}-\mathbf{N}\right)\oslash\mathbf{b_3}\mathbf{1}_{1\times m}\right)-\mathbf{Q}\|_F^2\Big] \\
&\le \min_{\mathbf{W}} \frac{1}{2nm}\|\mathbf{1}_{n\times m} - \mathbf{X}\mathbf{w}_1\mathbf{1}_{1\times m}\otimes\mathbf{N}^a \\
&\quad -\mathbf{1}_{n\times m}\oslash\left(\mathbf{1}_{n\times m}+\exp\left(\mathbf{X}\mathbf{w}_2\mathbf{1}_{1\times m}-\mathbf{N}\right)\oslash\mathbf{X}\mathbf{w}_3\mathbf{1}_{1\times m}\right)-\mathbf{Q}\|_F^2 \\
&= \frac{1}{2nm}\|\mathbf{1}_{n\times m} - \mathbf{X}\mathbf{w}_1^{\mathrm{e2e}}\mathbf{1}_{1\times m}\otimes\mathbf{N}^a \\
&\quad -\mathbf{1}_{n\times m}\oslash\left(\mathbf{1}_{n\times m}+\exp\left(\mathbf{X}\mathbf{w}_2^{\mathrm{e2e}}\mathbf{1}_{1\times m}-\mathbf{N}\right)\oslash\mathbf{X}\mathbf{w}_3^{\mathrm{e2e}}\mathbf{1}_{1\times m}\right)-\mathbf{Q}\|_F^2, \qquad (A.1)
\end{aligned}$$

where the inequality indicates that the end-to-end optimization results (right-hand side of the inequality) cannot achieve higher accuracy than fitting empirical models to individual cells (left-hand side of the inequality).

Furthermore, as a feasible solution to the end-to-end optimization problem, the sequential optimization cannot achieve a better solution. This is due to the optimality of end-to-end optimization that any possible better solution obtained by sequential optimization can be achieved by end-to-end optimization, which yields

$$\begin{aligned}
&\min_{\mathbf{W}} \frac{1}{2nm}\|\mathbf{1}_{n\times m} - \mathbf{X}\mathbf{w}_1\mathbf{1}_{1\times m}\otimes\mathbf{N}^a \\
&\quad -\mathbf{1}_{n\times m}\oslash\left(\mathbf{1}_{n\times m}+\exp\left(\mathbf{X}\mathbf{w}_2\mathbf{1}_{1\times m}-\mathbf{N}\right)\oslash\mathbf{X}\mathbf{w}_3\mathbf{1}_{1\times m}\right)-\mathbf{Q}\|_F^2 \\
&\le \frac{1}{2nm}\|\mathbf{1}_{n\times m} - \mathbf{X}\mathbf{w}_1^{\mathrm{seq}}\mathbf{1}_{1\times m}\otimes\mathbf{N}^a \\
&\quad -\mathbf{1}_{n\times m}\oslash\left(\mathbf{1}_{n\times m}+\exp\left(\mathbf{X}\mathbf{w}_2^{\mathrm{seq}}\mathbf{1}_{1\times m}-\mathbf{N}\right)\oslash\mathbf{X}\mathbf{w}_3^{\mathrm{seq}}\mathbf{1}_{1\times m}\right)-\mathbf{Q}\|_F^2. \qquad (A.2)
\end{aligned}$$

By combining the two inequalities in Eqs. (A.1) and (A.2), we can obtain the relation in Eq. (9).

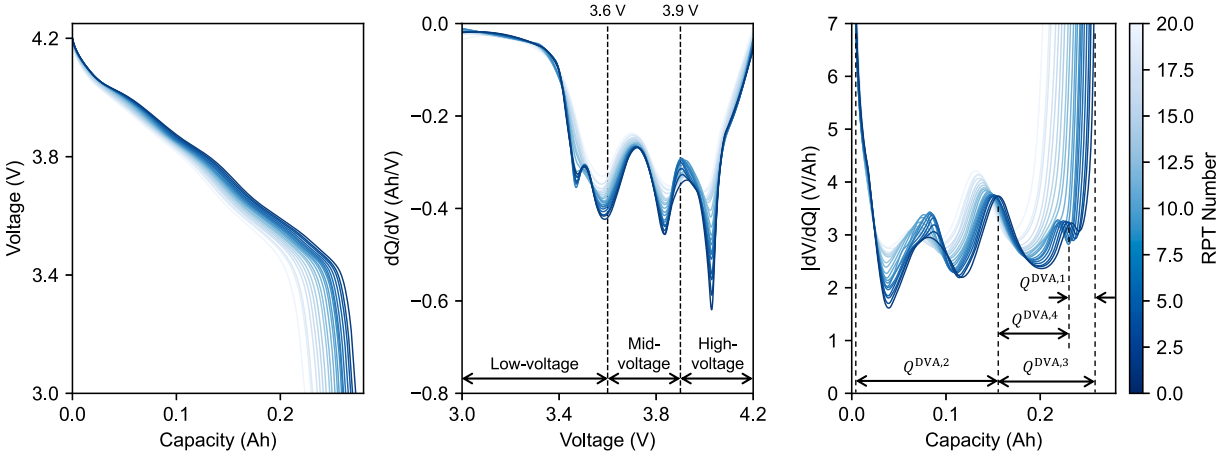


Fig. A.1. The evolution of capacity–voltage curves, incremental capacity curves, and differential voltage curves as the NMC cell ages.

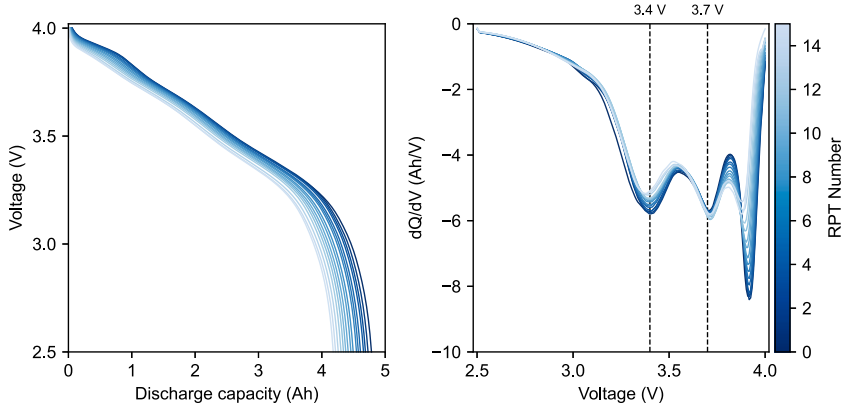


Fig. A.2. The evolution of capacity–voltage curves and incremental capacity curves as the NCA cell ages.

Table A.1
Source, mathematical formula, and description of early-life features.

Source	Feature	Description
Cycling conditions	C_{chg}	Charge C-rate
	C_{dchg}	Discharge C-rate
	DoD	Depth of discharge
	${C_{\text{chg}}}^{0.5} DoD^{0.5}$	Charging stress
	${C_{\text{dchg}}}^{0.5} DoD^{0.5}$	Discharging stress
	$({C_{\text{chg}}}^{0.5} DoD^{0.5} + {C_{\text{dchg}}}^{0.5} DoD^{0.5})/2$	Mean cycling stress
	${C_{\text{chg}}}^{0.5} DoD^{0.5} \times {C_{\text{dchg}}}^{0.5} DoD^{0.5}$	Multiplicative cycling stress
Incremental capacity curves ($dQ/dV(V)$)	$\log\left(\left\lvert\text{mean}\left(\Delta dQ/dV_{\text{w3-w0}}(V)\right)\right\rvert\right)$	Mean of the incremental curve difference between two RPTs
	$\log\left(\left\lvert\text{var}\left(\Delta dQ/dV_{\text{w3-w0}}(V)\right)\right\rvert\right)$	Variance of the incremental curve difference between two RPTs
	$\log\left(\left\lvert\text{mean}\left(\Delta dQ/dV_{\text{w3-w0}}^{3.0\text{ V}-3.6\text{ V}}(V)\right)\right\rvert\right)$	Mean of the low-voltage $dQ/dV(V)$ curve difference between two RPTs
	$\log\left(\left\lvert\text{mean}\left(\Delta dQ/dV_{\text{w3-w0}}^{3.6\text{ V}-3.9\text{ V}}(V)\right)\right\rvert\right)$	Mean of the mid-voltage $dQ/dV(V)$ curve difference between two RPTs
	$\log\left(\left\lvert\text{mean}\left(\Delta dQ/dV_{\text{w3-w0}}^{3.9\text{ V}-4.2\text{ V}}(V)\right)\right\rvert\right)$	Mean of the high-voltage $dQ/dV(V)$ curve difference between two RPTs
	$\log\left(\left\lvert\text{var}\left(\Delta dQ/dV_{\text{w3-w0}}^{3.0\text{ V}-3.6\text{ V}}(V)\right)\right\rvert\right)$	Variance of the low-voltage $dQ/dV(V)$ curve difference between two RPTs
	$\log\left(\left\lvert\text{var}\left(\Delta dQ/dV_{\text{w3-w0}}^{3.6\text{ V}-3.9\text{ V}}(V)\right)\right\rvert\right)$	Variance of the mid-voltage $dQ/dV(V)$ curve difference between two RPTs
	$\log\left(\left\lvert\text{var}\left(\Delta dQ/dV_{\text{w3-w0}}^{3.9\text{ V}-4.2\text{ V}}(V)\right)\right\rvert\right)$	Variance of the high-voltage $dQ/dV(V)$ curve difference between two RPTs
Capacity–voltage curves ($Q(V)$)	$\log\left(\left\lvert\text{mean}\left(\Delta Q_{\text{w3-w0}}(V)\right)\right\rvert\right)$	Mean of the QV curve difference between two RPTs
	$\log\left(\left\lvert\text{var}\left(\Delta Q_{\text{w3-w0}}(V)\right)\right\rvert\right)$	Variance of the QV curve difference between two RPTs
Constant-voltage charging curves	$\log\left(\text{CV Time}_{\text{w0}}\right)$	CV hold time of initial RPT
	$\log\left(\text{CV Time}_{\text{w3}}\right)$	CV hold time of the third week RPT
	$\log\left(\lvert\Delta\text{CV Time}_{\text{w3-w0}}\rvert\right)$	Change in CV hold time between two RPTs
Discharge capacity values	$\log\left(Q_{\text{w0}}\right)$	Initial capacity
	$\log\left(Q_{\text{w0}}^{3.0\text{ V}-3.6\text{ V}}\right)$	Initial capacity at low-voltage
	$\log\left(Q_{\text{w0}}^{3.6\text{ V}-3.9\text{ V}}\right)$	Initial capacity at mid-voltage
	$\log\left(Q_{\text{w0}}^{3.9\text{ V}-4.2\text{ V}}\right)$	Initial capacity at high-voltage
	$\log\left(\Delta Q_{\text{w3-w0}}\right)$	Capacity fade between two RPTs
Differential voltage curves ($dV/dQ(Q)$)	$\Delta Q_{\text{w3-w0}}^{\text{DVA},1}$	Change of the capacity measured between the third peak and the end of discharge on $dV/dQ(Q)$ curve
	$\Delta Q_{\text{w3-w0}}^{\text{DVA},2}$	Change of the capacity measured between the beginning of discharge and the second peak on $dV/dQ(Q)$ curve
	$\Delta Q_{\text{w3-w0}}^{\text{DVA},3}$	Change of the capacity measured between the second peak and the end of discharge on $dV/dQ(Q)$ curve
	$\Delta Q_{\text{w3-w0}}^{\text{DVA},4}$	Change of the capacity measured between the second peak and the third peak on $dV/dQ(Q)$ curve

Table A.2
Hyperparameters for the MLP networks in various approaches.

Hyperparameter	Knot-point	Seq-NN	E2E-NN	E2E-NNE
Number of layers	4	3	3	4
Number of neurons	6	16	13	5
Learning rate	4.868×10^{-2}	1.148×10^{-4}	1.076×10^{-2}	1.028×10^{-3}
Weight decay	5.041×10^{-7}	5.704×10^{-8}	9.713×10^{-3}	3.572×10^{-7}
Batch size	35	23	53	43
Warm-up epochs	–	–	–	667

Table A.3
Training time averaged across 5 runs for each method, reported in the unit of seconds.

Method	Knot-point	Seq-ENR	Seq-NN	E2E-ENR	E2E-NN	E2E-NNE ($M = 5$)
Package used	PyTorch	Scikit-learn	PyTorch	Scipy	PyTorch	PyTorch
Training time	1.41	0.42	2.45	20.86	1.45	29.50

A.3. Hyperparameter optimization results for all approaches with MLP neural networks involved

See Table A.2.

A.4. Computational time results

For all methods presented in this work, the model training processes were completed on a local PC with an Intel® Core™ i5-10505 6-core CPU, 16GB of RAM, and no external GPU. The training time for a single model or a single ensemble of 5 models is listed in Table A.3.

A.5. Predicted trajectories for selected cells

See Figs. A.3, A.4, A.5, A.6, A.7 and A.8.

Data availability

The aging dataset used for this study has been shared publicly [50, 65]; the code for this study can be found on GitHub (https://github.com/tingkai-li/empirical_early_prediction).

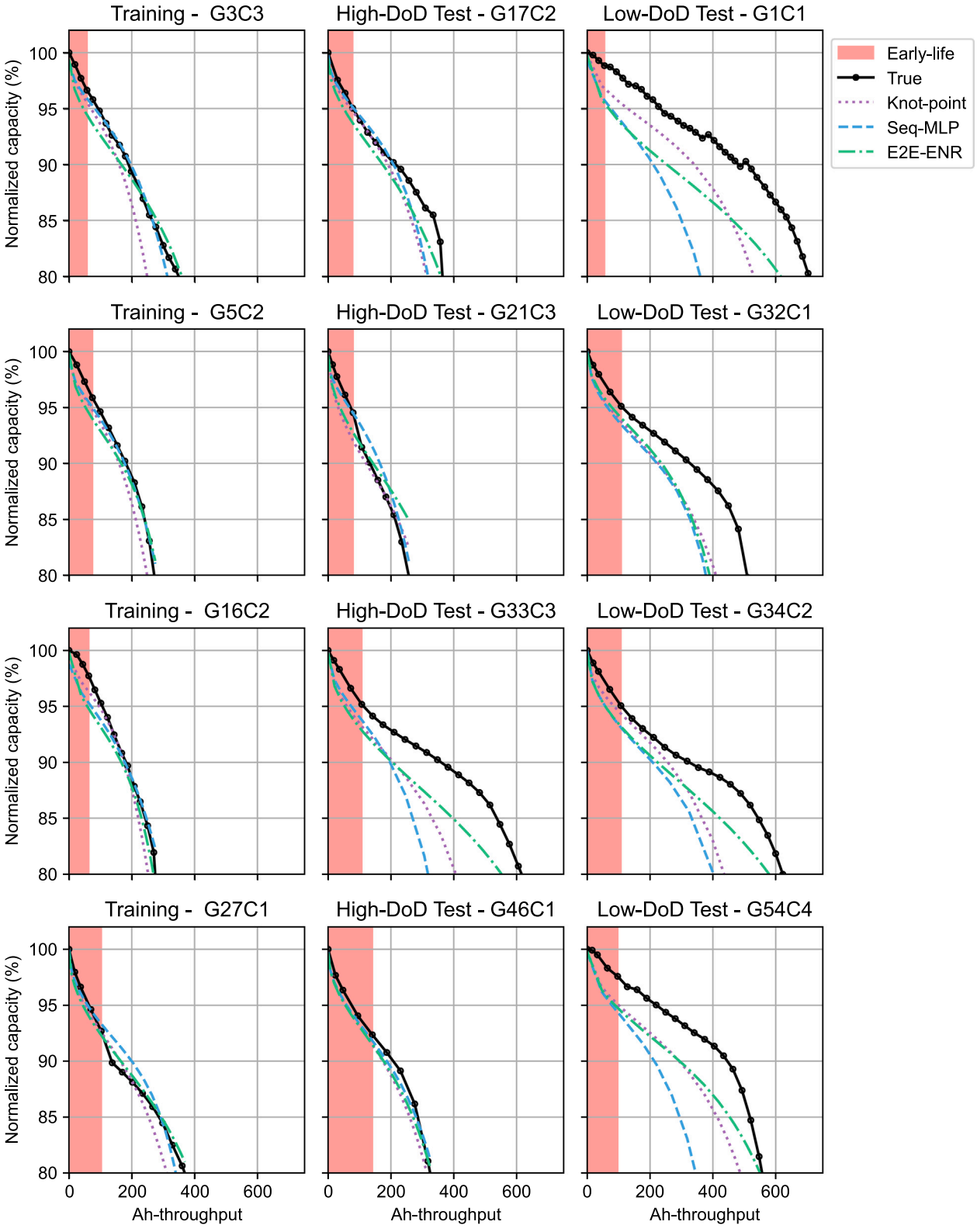


Fig. A.3. Predicted trajectories from selected deterministic methods for selected cells.

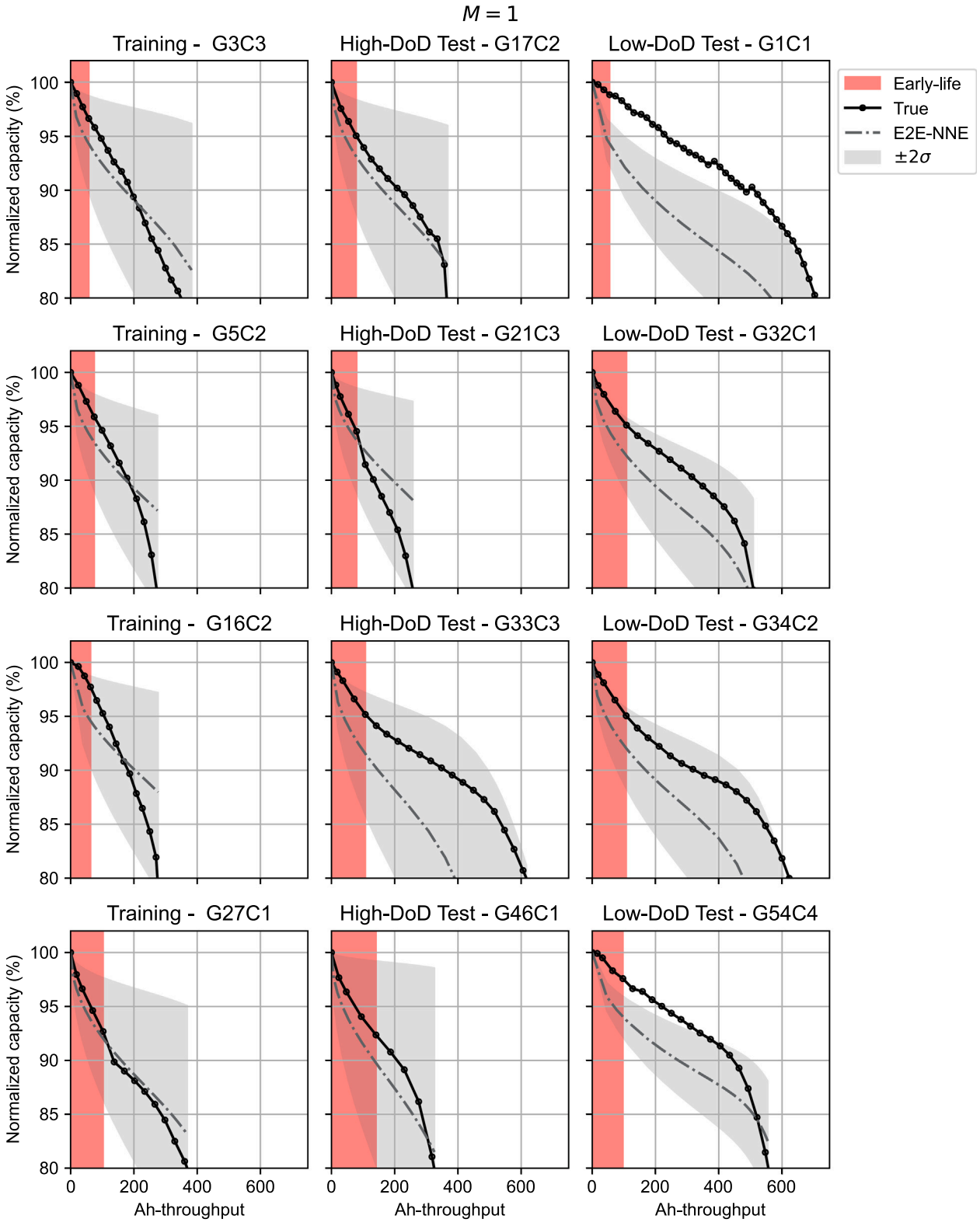


Fig. A.4. Predicted trajectories from E2E-NNE ($M = 1$) for selected cells.

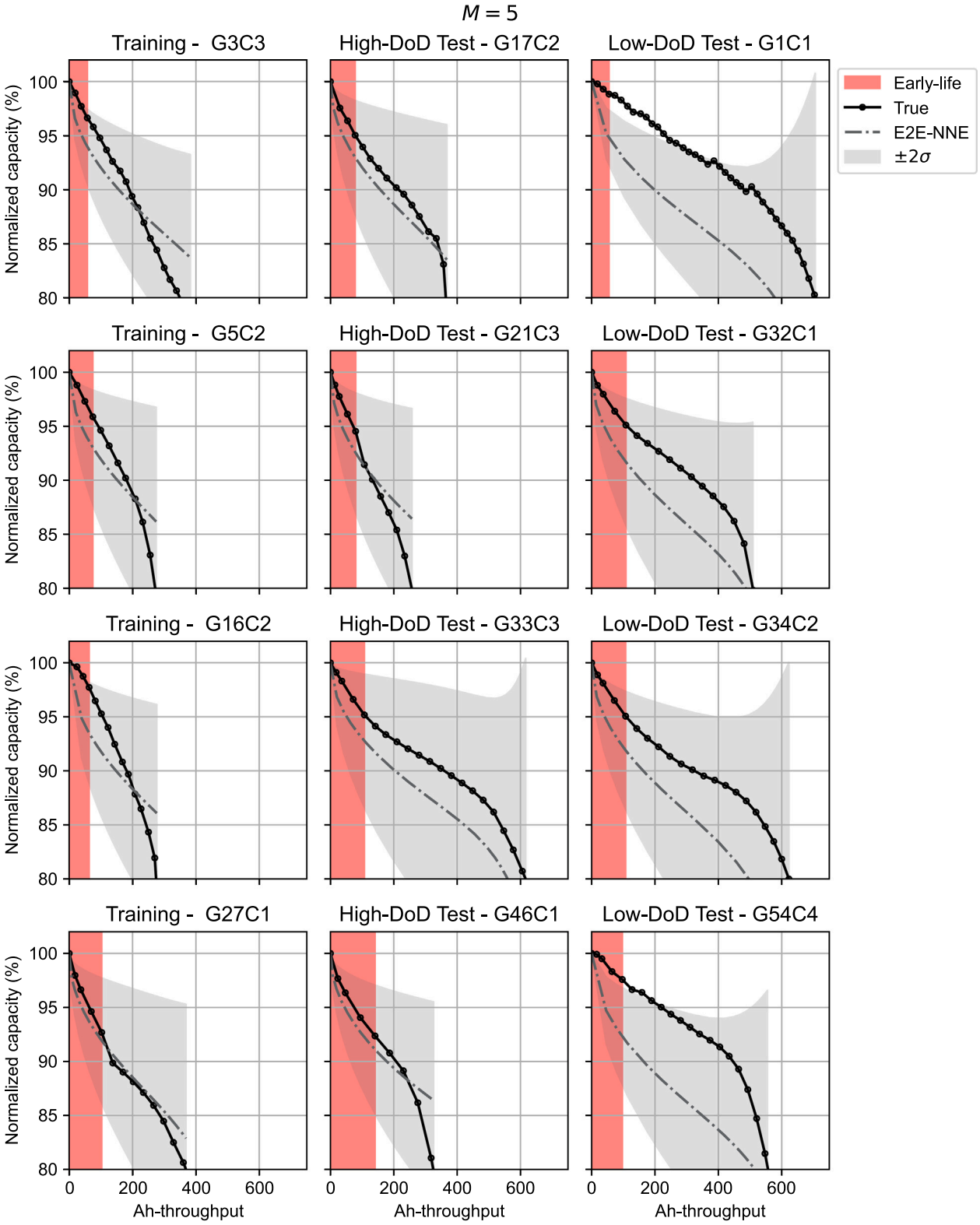


Fig. A.5. Predicted trajectories from E2E-NNE ($M = 5$) for selected cells.

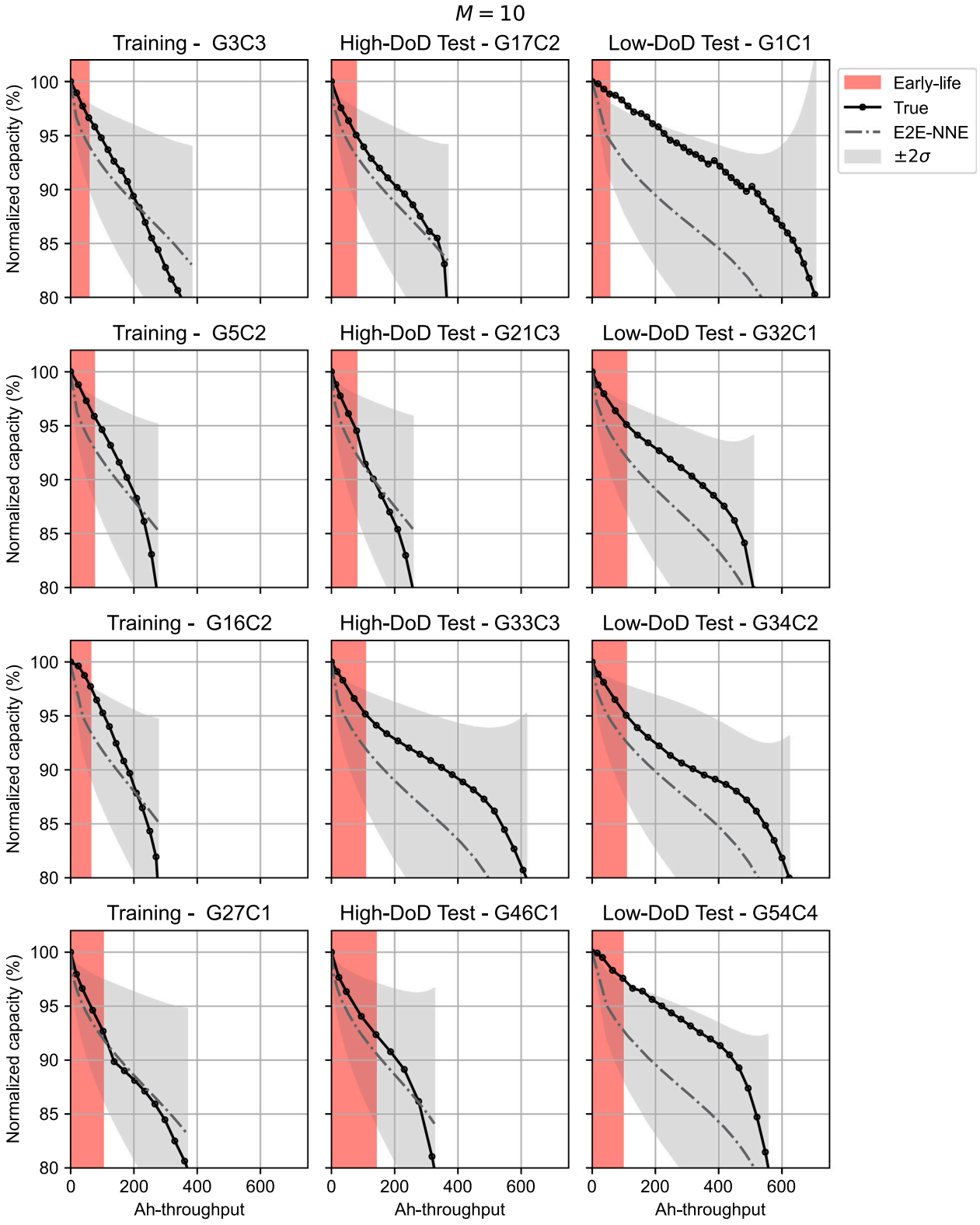


Fig. A.6. Predicted trajectories from E2E-NNE ($M = 10$) for selected cells.

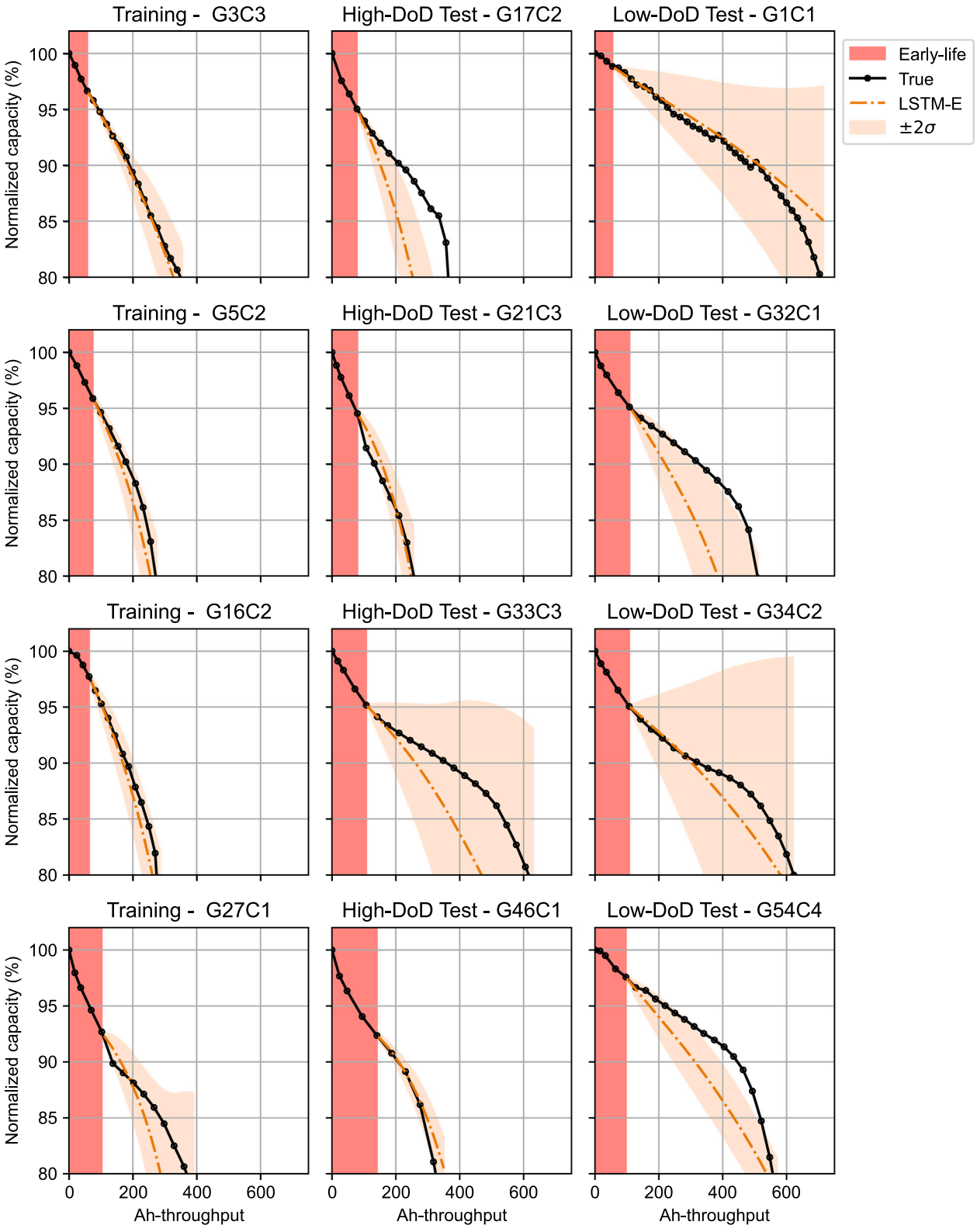


Fig. A.7. Predicted trajectories from LSTM-E method for selected cells in the public NCA dataset.

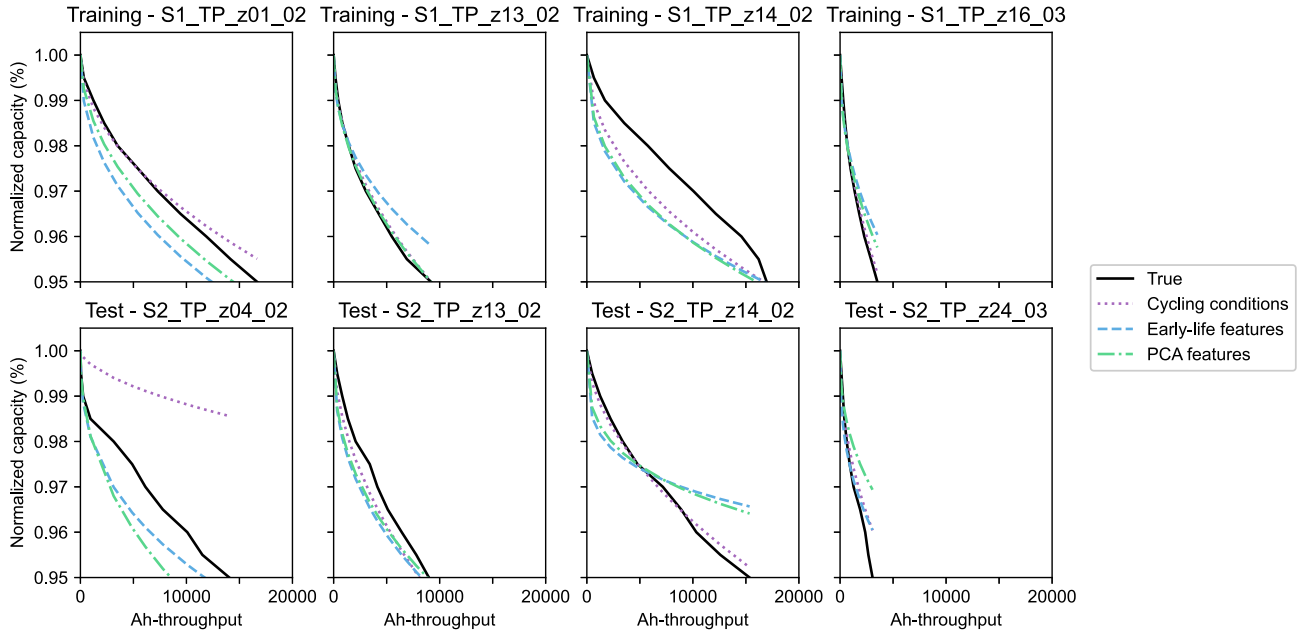


Fig. A.8. Predicted trajectories from E2E-ENR method for selected cells in the public NCA dataset.

References

[1] Guo J, Zhaojun L, Pecht M. A Bayesian approach for Li-ion battery capacity fade modeling and cycles to failure prognostics. J Power Sources 2015 May; 281:173–84. doi:https://doi.org/10.1016/j.jpowsour.2015.01.164

[2] Chao H, Hui Y, Jain G, Schmidt C. Remaining useful life assessment of lithium-ion batteries in implantable medical devices. J Power Sources 2018 January; 375:118–30. doi:https://doi.org/10.1016/j.jpowsour.2017.11.056

[3] Downey A, Lui Y-H, Chao H, Laflamme S, Shan H. Physics-based prognostics of lithium-ion battery using non-linear least squares with dynamic bounds. Reliab Eng Syst Saf 2019 February; 182:1–12. doi:https://doi.org/10.1016/j.ress.2018.09.018

[4] Hui Lui Y, Li M, Downey A, Shen S, Nemani VP, Hui Y, et al. Physics-based prognostics of implantable-grade lithium-ion battery for remaining useful life prediction. J Power Sources 2021;485:229327. doi:https://doi.org/10.1016/j.jpowsour.2020.229327

[5] Mathews I, Bolun X, Wei H, Barreto V, Buonassisi T, Marius Peters I. Technoeconomic model of second-life batteries for utility-scale solar considering calendar and cycle aging. Appl Energy 2020 July; 269:115127. doi:https://doi.org/10.1016/j.apenergy.2020.115127

[6] Severson KA, Attia PM, Jin N, Perkins N, Jiang B, Yang Z, et al. Data-driven prediction of battery cycle life before capacity degradation. Nat Energy 2019;4(5):383–91. doi:https://doi.org/10.1038/s41560-019-0356-8

[7] Liu X, Zhuoyuan Zheng {EB, Zhou Z, Wang P. Battery asset management with cycle life prognosis. Reliab Eng Syst Saf 2021 December; 216:107948. doi:https://doi.org/10.1016/j.ress.2021.107948.

[8] Baumhöfer T, Brühl M, Rothgang S, Uwe Sauer D. Production caused variation in capacity aging trend and correlation to initial cell performance. J Power Sources 2014 February; 247:332–38. doi:https://doi.org/10.1016/j.jpowsour.2013.08.108

[9] Harris SJ, Harris DJ, Li C. Failure statistics for commercial lithium ion batteries: a study of 24 pouch cells. J Power Sources 2017 February; 342:589–97. doi:https://doi.org/10.1016/j.jpowsour.2016.12.083

[10] Attia PM, Grover A, Jin N, Severson KA, Markov TM, Liao Y-H, et al. Closed-loop optimization of fast-charging protocols for batteries with machine learning. Nature 2020;578(7795):397–402. doi:https://doi.org/10.1038/s41586-020-1994-5

[11] Yang F, Wang D, Fan X, Huang Z, Tsui K-L. Lifespan prediction of lithium-ion batteries based on various extracted features and gradient boosting regression tree model. J Power Sources 2020 November; 476:228654. doi:https://doi.org/10.1016/j.jpowsour.2020.228654

[12] Attia PM, Severson KA, Witmer JD. Statistical learning for accurate and interpretable battery lifetime prediction. J Electrochem Soc 2021 September; 168(9):090547. doi:https://doi.org/10.1149/1945-7111/ac2704

[13] Fei Z, Yang F, Tsui K-L, Lishuai L, Zhang Z. Early prediction of battery lifetime via a machine learning based framework. Energy 2021 June; 225:120205. doi:https://doi.org/10.1016/j.energy.2021.120205

[14] Zhang Y, Peng Z, Guan Y, Lifeng W. Prognostics of battery cycle life in the early-cycle stage based on hybrid model. Energy 2021 April; 221:119901. doi:https://doi.org/10.1016/j.energy.2021.119901

[15] Paulson NH, Kubal J, Ward L, Saxena S, Lu W, Babinec SJ. Feature engineering for machine learning enabled early prediction of battery lifetime. J Power Sources 2022 April; 527:231127. doi:https://doi.org/10.1016/j.jpowsour.2022.231127

[16] Tingkai L, Zhou Z, Thelen A, Howey DA, Chao H. Predicting battery lifetime under varying usage conditions from early aging data. Cell Rep Phys Sci 2024 March; 101891. doi:https://doi.org/10.1016/j.xcrp.2024.101891

[17] Fermín-Cueto P, McTurk E, Allerhand M, Medina-Lopez E, Anjos MF, Sylvester J, et al. Identification and machine learning prediction of knee-point and knee-onset in capacity degradation curves of lithium-ion cells. Energy AI 2020 August; 1:100006. doi:https://doi.org/10.1016/j.egyai.2020.100006

[18] Weihan L, Sengupta N, Dechent P, Howey D, Annaswamy A, Uwe Sauer D. One-shot battery degradation trajectory prediction with deep learning. J Power Sources 2021 September; 506:230024. doi:https://doi.org/10.1016/j.jpowsour.2021.230024

[19] Weihan L, Zhang H, Van Vlijmen B, Dechent P, Uwe Sauer D. Forecasting battery capacity and power degradation with multi-task learning. Energy Storage Mater 2022 December; 53:453–66. doi:https://doi.org/10.1016/j.ensm.2022.09.013

[20] Ibraheem R, Strange C, Dos Reis G. Capacity and internal resistance of lithium-ion batteries: full degradation curve prediction from voltage response at constant current at discharge. J Power Sources 2023 February; 556:232477. doi:https://doi.org/10.1016/j.jpowsour.2022.232477

[21] Kim S, Jung H, Lee M, Young Choi Y, Choi J-I. Model-free reconstruction of capacity degradation trajectory of lithium-ion batteries using early cycle data. ETransportation 2023 July; 17:100243. doi:https://doi.org/10.1016/j.etran.2023.100243

[22] Tang T, Yuan H. A hybrid approach based on decomposition algorithm and neural network for remaining useful life prediction of lithium-ion battery. Reliab Eng Syst Saf 2022 January; 217:108082. doi:https://doi.org/10.1016/j.ress.2021.108082

[23] Xiaosong H, Xu L, Lin X, Pecht M. Battery lifetime prognostics. Joule 2020 February; 4(2):310–46. doi:https://doi.org/10.1016/j.joule.2019.11.018.

[24] Jiabei H, Tian Y, Lifeng W. A hybrid data-driven method for rapid prediction of lithium-ion battery capacity. Reliab Eng Syst Saf 2022 October; 226:108674. doi:https://doi.org/10.1016/j.ress.2022.108674

[25] Ning G, White RE, Popov BN. A generalized cycle life model of rechargeable Li-ion batteries. Electrochimica Acta 2006 February; 51(10):2012–22. doi:https://doi.org/10.1016/j.electacta.2005.06.033

[26] Ramadesigan V, Boovaragavan V, Arabandi M, Chen K, Tsukamoto H, Braatz R, et al. Parameter estimation and capacity fade analysis of lithium-ion batteries using first-principles-based efficient reformulated models. ECS Trans 2009 October; 19(16):11–19. doi:https://doi.org/10.1149/1.3245868

[27] Birkl CR, Roberts MR, McTurk E, Bruce PG, Howey DA. Degradation diagnostics for lithium ion cells. J Power Sources 2017 February; 341:373–86. ISSN 03787753. doi:https://doi.org/10.1016/j.jpowsour.2016.12.011

[28] Thelen A, Hui Lui Y, Shen S, Laflamme S, Shan H, Hui Y, et al. Integrating physics-based modeling and machine learning for degradation diagnostics of lithium-ion batteries. Energy Storage Mater 2022;50:668–95. doi:https://doi.org/org/10.1016/j.ensm.2022.05.047

[29] Navidi S, Thelen A, Tingkai L, Chao H. Physics-informed machine learning for battery degradation diagnostics: a comparison of state-of-the-art methods. Energy Storage Mater 2024 April; 68:103343. doi:https://doi.org/10.1016/j.ensm.2024.103343

[30] Xiaodong X, Tang S, Chuanqiang Y, Xie J, Han X, Ouyang M. Remaining useful life prediction of lithium-ion batteries based on wiener process under time-varying temperature condition. Reliab Eng Syst Saf 2021 October; 214:107675. doi:https://doi.org/10.1016/j.ress.2021.107675

[31] Guha A, Patra A. Online estimation of the electrochemical impedance spectrum and remaining useful life of lithium-ion batteries. IEEE Trans Instrum Meas 2018 August; 67(8):1836–49. doi:https://doi.org/10.1109/TIM.2018.2809138

[32] Broussely M, Herreyre S, Biensan P, Kasztejna P, Nechev K, Staniewicz RJ. Aging mechanism in Li ion cells and calendar life predictions. J Power Sources 2001 July; 97-98:13–21. doi:https://doi.org/10.1016/S0378-7753(01)00722-4

[33] Saha B, Goebel K, Poll S, Christophersen J. Prognostics methods for battery health monitoring using a Bayesian framework. IEEE Trans Instrum Meas 2009 February; 58(2):291–96. doi:https://doi.org/10.1109/TIM.2008.2005965

[34] Wei H, Williard N, Osterman M, Pecht M. Prognostics of lithium-ion batteries based on Dempster–Shafer theory and the Bayesian Monte Carlo method. J Power Sources 2011 December; 196(23):10314–21. doi:https://doi.org/10.1016/j.jpowsour.2011.08.040

[35] Diao W, Saxena S, Pecht M. Accelerated cycle life testing and capacity degradation modeling of LiCoO2-graphite cells. J Power Sources 2019 September; 435:226830. doi:https://doi.org/10.1016/j.jpowsour.2019.226830

[36] Thomas EV, Bloom I, Christophersen JP, Battaglia VS. Statistical methodology for predicting the life of lithium-ion cells via accelerated degradation testing. J Power Sources 2008 September; 184(1):312–17. doi:https://doi.org/10.1016/j.jpowsour.2008.06.017

[37] Pradeep Lall A, Hao Zhang B, Rahul Lall C. PHM of state-of-charge for flexible power sources in wearable electronics with EKF. In: 2018 IEEE International Reliability Physics Symposium (IRPS); Burlingame, CA: IEEE; 2018 March, p. P-SR.2-1-P-SR.2-6. doi:https://doi.org/10.1109/IRPS.2018.8353695

[38] Walker E, Rayman S, White RE. Comparison of a particle filter and other state estimation methods for prognostics of lithium-ion batteries. J Power Sources 2015 August; 287:1–12. doi:https://doi.org/10.1016/j.jpowsour.2015.04.020

[39] Sai L, Fang H, Shi B. Remaining useful life estimation of lithium-ion battery based on interacting multiple model particle filter and support vector regression. Reliab Eng Syst Saf 2021 June; 210:107542. doi:https://doi.org/10.1016/j.ress.2021.107542

[40] Nuhic A, Terzimehic T, Soczka-Guth T, Buchholz M, Dietmayer K. Health diagnosis and remaining useful life prognostics of lithium-ion batteries using data-driven methods. J Power Sources 2013 October; 239:680–88. doi:https://doi.org/10.1016/j.jpowsour.2012.11.146

[41] Wang D, Miao Q, Pecht M. Prognostics of lithium-ion batteries based on relevance vectors and a conditional three-parameter capacity degradation model. J Power Sources 2013 October; 239:253–64. doi:https://doi.org/10.1016/j.jpowsour.2013.03.129

[42] Richardson RR, Osborne MA, Howey DA. Gaussian process regression for forecasting battery state of health. J Power Sources 2017 July; 357:209–19. doi:https://doi.org/10.1016/j.jpowsour.2017.05.004

[43] Liu J, Saxena A, Goebel K, Saha B, Wang W. An adaptive recurrent Neural network for remaining useful life prediction of lithium-ion batteries. Proc Annu Conf Progn Health Manag Soc 2010;2(1) October. doi:https://doi.org/10.36001/phmconf.2010.v2i1.1896.

[44] Zhang Y, Xiong R, Hongwen H, Pecht MG. Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries. IEEE Trans Veh Technol 2018 July; 67(7):5695–705. doi:https://doi.org/10.1109/TVT.2018.2805189

[45] Bae J, Zhimin X. Learning of physical health timestep using the LSTM network for remaining useful life estimation. Reliab Eng Syst Saf 2022 October; 226:108717. doi:https://doi.org/10.1016/j.ress.2022.108717

[46] Rouhi Ardeshiri R, Liu M, Chengbin M. Multivariate stacked bidirectional long short term memory for lithium-ion battery health management. Reliab Eng Syst Saf 2022 August; 224:108481. doi:https://doi.org/10.1016/j.ress.2022.108481

[47] Hannemose Rieger L, Flores E, Frellesen Nielsen K, Norby P, Ayerbe E, Winther O, et al. Uncertainty-aware and explainable machine learning for early prediction of battery degradation trajectory. Digital Discovery 2023;2(1):112–22. doi:https://doi.org/10.1039/D2DD00067A

[48] Zhu R, Chen Y, Peng W, Zhi-Sheng Y. Bayesian deep-learning for RUL prediction: an active learning perspective. Reliab Eng Syst Saf 2022 December; 228:108758. doi:https://doi.org/10.1016/j.ress.2022.108758

[49] Lin Y-H, Gang-Hui L. A bayesian deep learning framework for RUL prediction incorporating uncertainty quantification and calibration. IEEE Trans Ind Inf 2022 October; 18(10):7274–84. doi:https://doi.org/10.1109/TII.2022.3156965

[50] Thelen A, Tingkai L, Liu J, Tischer C, Chao H. ISU-ILCC battery aging dataset. Iowa State University DataShare; 2023. doi:https://doi.org/10.25380/IASTATE.22582234

[51] Johnen M, Pitzen S, Kamps U, Kateri M, Dechent P, Uwe Sauer D. Modeling long-term capacity degradation of lithium-ion batteries. J Energy Storage 2021 February; 34:102011. doi:https://doi.org/10.1016/j.est.2020.102011

[52] Gasper P, Gering K, Dufek E, Smith K. Challenging practices of algebraic battery life models through statistical validation and model identification via machine-learning. J Electrochem Soc 2021 February; 168(2):020502. doi:https://doi.org/10.1149/1945-7111/abdde1

[53] Attia PM, Bills A, Brosa Planella F, Dechent P, Dos Reis G, Dubarry M, et al. Review—"knees" in lithium-ion battery aging trajectories. J Electrochem Soc 2022 June; 169(6):060517. doi:https://doi.org/10.1149/1945-7111/ac6d13

[54] Shlens J. A tutorial on Principal component analysis. arXiv, 2014 April. doi:https://doi.org/10.48550/arXiv.1404.1100

[55] Smith K, Gasper P, Colclasure AM, Shimonishi Y, Yoshida S. Lithium-ion battery life model with electrode cracking and early-life break-in processes. J Electrochem Soc 2021 October; 168(10):100530. doi:https://doi.org/10.1149/1945-7111/ac2ebd

[56] Liu J, Thelen A, Chao H, Yang X-G. An end-to-end learning framework for battery capacity-fade trajectory prediction using early life data. Ann Conf PHM Society 2021; 13(1) November. doi:https://doi.org/10.36001/phmconf.2021.v13i1.3053.

[57] Nix DA, Weigend AS. Estimating the mean and variance of the target probability distribution. In: Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94), vol. 1. Orlando, FL, USA: IEEE; 1994. p. 55–60. doi:https://doi.org/10.1109/ICNN.1994.374138

[58] Seitzer M, Tavakoli A, Antic D, Martius G. On the pitfalls of heteroscedastic uncertainty estimation with probabilistic neural networks. 2022 April.

[59] Sluijterman L, Cator E, Heskes T. Optimal training of mean variance estimation neural networks. Neurocomputing 2024 September; 597:127929. doi:https://doi.org/10.1016/j.neucom.2024.127929

[60] Lakshminarayanan B, Pritzel A, Blundell C. Simple and scalable predictive uncertainty estimation using deep ensembles. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17; Red Hook, NY, USA: Curran Associates Inc; 2017. p. 6405–16.

[61] Nemani V, Biggio L, Huan X, Zhen H, Fink O, Tran A, et al. Uncertainty quantification in machine learning for engineering design and health prognostics: a tutorial. Mech Syst And Signal Process 2023 December; 205:110796. doi:https://doi.org/10.1016/j.ymssp.2023.110796

[62] Thelen A, Huan X, Paulson N, Onori S, Zhen H, Chao H. Probabilistic machine learning for battery health diagnostics and prognostics—review and perspectives. Npj Mater Sustain 2024 June; 2(1):14. doi:https://doi.org/10.1038/s44296-024-00011-1

[63] Akiba T, Sano S, Yanase T, Ohta T, Koyama M. Optuna: a next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2019.

[64] Sendek AD, Ransom B, Cubuk ED, Pellouchoud LA, Nanda J, Reed EJ. Machine learning modeling for accelerated battery materials design in the small data regime. Adv Energy Mater 2022 August; 12(31):2200553. doi:https://doi.org/10.1002/aenm.202200553

[65] Stroebl F, Petersohn R, Schricker B, Schaeufl F, Bohlen O, Palm H. A multi-stage lithium-ion battery aging dataset using various experimental design methodologies. Sci Data 2024 September; 11(1):1020. doi:https://doi.org/10.1038/s41597-024-03859-z

[66] Popp H, Zhang N, Jahn M, Arrinda M, Ritz S, Faber M, et al. Ante-mortem analysis, electrical, thermal, and ageing testing of state-of-the-art cylindrical lithium-ion cells. Elektrotech Informationstechnik 2020 August; 137(4–5):169–76. doi:https://doi.org/10.1007/s00502-020-00814-9