
PHYSICS-INFORMED MACHINE LEARNING FOR BATTERY DEGRADATION DIAGNOSTICS: A COMPARISON OF STATE-OF-THE-ART METHODS

Sina Navidi¹, Adam Thelen², Tingkai Li¹, Chao Hu^{1,†}

¹Department of Mechanical Engineering, University of Connecticut, Storrs, CT, 06269, USA

²Department of Mechanical Engineering, Iowa State University, Ames, IA 50011, USA

[†]Indicates corresponding author. Email: chao.hu@uconn.edu.

ACKNOWLEDGEMENTS

This research was partly supported by the Iowa Economic Development Authority under the Iowa Energy Center Grant No. 20-IEC-018 and partly by the US National Science Foundation (NSF) Grant No. ECCS-2015710. Any opinions, findings, or conclusions in this paper are those of the authors and do not necessarily reflect the views of the sponsoring agencies. The authors would also like to thank Dr. Gaurav Jain and Dr. Hui Ye at Medtronic for providing the implantable-grade lithium-ion battery cells used to collect the cycling data for this study.

ABSTRACT

Monitoring the health of lithium-ion batteries' internal components as they age is crucial for optimizing cell design and usage control strategies. However, quantifying component-level degradation typically involves aging many cells and destructively analyzing them throughout the aging test, limiting the scope of quantifiable degradation to the test conditions and duration. Fortunately, recent advances in physics-informed machine learning (PIML) for modeling and predicting the battery state of health demonstrate the feasibility of building models to predict the long-term degradation of a lithium-ion battery cell's major components using only short-term aging test data by leveraging physics. In this paper, we present four approaches for building physics-informed machine learning models and comprehensively compare them, considering accuracy, complexity, ease-of-implementation, and their ability to extrapolate to untested conditions. We delve into the details of each physics-informed machine learning method, providing insights specific to implementing them on small battery aging datasets. Our study utilizes long-term cycle aging data from 24 implantable-grade lithium-ion cells subjected to varying temperatures and C-rates over four years. This paper aims to facilitate the selection of an appropriate physics-informed machine learning method for predicting long-term degradation in lithium-ion batteries, using short-term aging data while also providing insights about when to choose which method for general predictive purposes.

Keywords: Lithium-ion battery, Degradation diagnostics, Half-cell model, Physics-informed machine learning, Comparative study

1 Introduction

Monitoring the health of lithium-ion batteries over their lifetime is important for ensuring the safety and reliability of the electric vehicles and portable electronics they power. Common battery health indicators like remaining capacity and direct-current internal resistance (DCIR) can be directly measured through complete diagnostic cycles at low rates (capacity) or hybrid pulse-power characterization (HPPC) tests (resistance) [1]. However, in online applications, battery capacity must be estimated because completing a full discharge cycle would significantly interrupt system operation. Battery capacity and resistance can be predicted online using reduced order algebraic models [2, 3, 4], state observer algorithms (like the extended Kalman filter and particle filters [5, 6, 7, 8]) coupled with physics-based battery models like equivalent circuit models [9, 10, 11] and electrochemical models [12, 13, 14], and machine learning methods [15, 16, 17, 18, 19, 20, 21]. In cases where limited data are available, machine learning models have still been demonstrated to accurately predict battery capacity and resistance using temperature data [22] and partial charging data [23, 24, 25, 26, 27]. Machine learning methods, such as linear regression [15, 24], support vector machines [15], random forest regression [27], neural networks [15, 19, 20, 21, 26], and Gaussian process regression (GPR) or kriging [17, 18, 25], have been successfully applied to estimate the capacity of batteries. These methods train a machine learning model to learn the correlation between features extracted from cell measurements (voltage, current, temperature) and the cell's state of health (SOH). However, none of these methods provide insight into the health of a cell's internal components, which is important for understanding the source of capacity loss and resistance increase in a cell. In real-world scenarios, battery aging occurs under wide ranges of use conditions and cases, often causing cell capacity and resistance to vary non-monotonically over a cell's lifetime [28]. This highlights that, even with similar capacity loss, the level of internal damage and dominant degradation mechanisms vary with operating conditions [29, 30]. To properly quantify cell health, more insight into the health of a cell's major internal components is required.

Three commonly reported degradation modes help to elucidate the root cause of cell capacity loss and resistance increase: they are loss of active materials on the positive and negative electrodes, abbreviated as LAM_{PE} and LAM_{NE}, respectively, and loss of lithium inventory, LLI [30, 31, 32, 33, 34, 35]. The first two modes, LAM_{PE} and LAM_{NE}, describe the loss of active electrode materials, both lithiated and delithiated. Some of the causes for the loss of active materials on the positive and negative electrodes include harsh operating conditions, such as high current, which causes the electrodes to swell and crack. Other mechanisms by which the positive and negative electrodes degrade include structural disordering, gas formation, and the dissolution of metal ions into the electrolyte [31]. On the other hand, the LLI degradation mode is used to track the remaining usable lithium inventory and is closely coupled with cell capacity. The growth of a solid-electrolyte interphase (SEI) layer is one major contributor to LLI [36]. Additionally, other lithium-consuming side reactions occur during normal cell cycling, further decreasing the lithium inventory. For example, reactions between lithium ions and impurities in the electrolyte or electrode materials can lead to the formation of solid compounds that consume lithium ions, causing a reduction in available lithium. Together, LAM_{PE}, LAM_{NE}, and LLI help better explain the component-level causes for cell-level capacity loss and resistance increase.

Generally, these three degradation modes can be experimentally verified by destructively analyzing the cell components. The loss of active electrode materials, both LAM_{PE} and LAM_{NE}, can be quantified by measuring the remaining capacity of the aged electrode materials, typically achieved by cycling half-cells built using the aged electrodes. Measuring the LLI is more subjective since it depends on the upper voltage limit the cell is run at. To measure the LLI capacity, the aged full-cell can be cycled to the maximum upper voltage limit which is typically beyond the real use-case upper voltage limit. However, doing so can significantly damage the cathode material, and is therefore typically only done once the cell has been retired from operation. While these methods for quantifying the state of each degradation mode are certainly effective, they are not preferred because they require retiring the cell from the field and destroying it in the process [37]. There is a great need to develop non-destructive degradation diagnostics methods that can be used

to accurately quantify the state of the LAM_{PE}, LAM_{NE}, and LLI degradation modes online during standard cell operation.

Several non-destructive methods for estimating the state of each degradation mode have been proposed in the past. Han et al. utilized membership functions to quantify the areas under peak locations on the $dQ/dV(V)$ curve and linked them to LLI and LAM_{NE} degradation modes [33]. Birkl et al. developed a diagnostic algorithm to determine the degradation modes and verified its accuracy by reconstructing the pseudo-OCV curve of coin-cells with known levels of degradation [31]. Tian et al. introduced a technique for estimating electrode aging parameters, essentially quantifying the corresponding degradation modes, by selectively sampling segments of daily charging profiles and inputting them directly into a convolution neural network [35]. Another popular approach to estimating the state of degradation modes is by simulating them using physical cell models. For example, an electrochemical model can be built to approximate the voltage response of the cell by incorporating parameters such as active material and binder mass, particle size, electrode thickness, current collector thickness, etc. [38, 39, 40, 41, 42]. Once the model is constructed, the values of the parameters can be changed to alter the model's output such that it matches the voltage data measured from an aged cell, effectively simulating the loss of active materials and lithium inventory. However, aside from being computationally intensive, a significant challenge lies in connecting these physics-based models to real-world aging phenomena, particularly in identifying the state of degradation modes based on standard aging data, such as voltage and capacity measurements obtained from reference performance tests (RPTs).

Recent advances in the field of physics-informed machine learning (PIML), particularly physics-informed neural networks (PINNs), have demonstrated significant potential in bridging the gap between physics-based models and data-driven techniques. PIML combines the strengths of physics-based models and machine learning algorithms to enhance generalization and extrapolation capabilities, even when data is limited. By integrating the principles of physics with the flexibility and scalability of machine learning, PIML approaches offer the advantage of accurately capturing complex physical phenomena with less computational complexity. In a comprehensive overview, Karniadakis et al. highlight the potential applications of PINNs across various scientific domains [43]. PINNs leverage the power of neural networks to incorporate physical principles and constraints directly into the learning process, enabling more precise predictions while maintaining computational efficiency[44].

In the realm of battery applications, a study by Storey et al. explored different architectures for combining physics-based and machine learning models to predict battery health [45]. The study highlights potential architectures for combining these models while discussing the associated limitations. However, these methods have not been widely adopted by battery researchers. In a recent study, Huang et al. [46] proposed a new deep learning architecture, termed model integrated neural network, which combines a recurrent neural network and residual neural network. This model architecture was designed to incorporate physics-based equations of Newman's pseudo-two-dimensional (P2D) model into the recurrent units and compute the residuals of the partial differential equations (PDEs) in an unsupervised manner through the residual connections. They benchmarked their proposed deep learning model against a few other models consisting of a physics-based P2D model solved by an implicit differential-algebraic solver, a purely data-driven deep neural networks model, a PINN model, and a data-driven reduced-order model. In the work by Wen et al. [47], a PINN framework was employed to estimate the rate of capacity degradation with respect to the cycle number and the current SOH of a cell being monitored. An empirical capacity degradation model, which was an improved version of Pierre Verhulst's logistic differential equation [48], was utilized to capture the capacity degradation trend of a battery cell. They used a PINN architecture to fuse the prior information from the empirical model, formulated as PDEs, which took as input features with strong correlations with capacity extracted from the experimental data. Two examples of these input features are peaks of $dQ/dV(V)$ curves and charge time during the constant current charging step. Xue et al. [49] proposed a PINN model to enhance the computational efficiency over a single-particle model while maintaining accuracy under high C-rates. First, they employed this

single-particle model to compute the distribution of lithium-ion concentration in the electrolyte given an applied current, disregarding the electrolyte dynamics. Then, the obtained results were used to train a PINN model. This PINN model incorporates a PDE solver in its loss function to solve the one-dimensional diffusion equation. The proposed model could approximate the in-electrolyte distributions of lithium-ion concentration and potential more accurately than the traditional neural network and faster than the single-particle model solved with traditional numerical methods under dynamic charging conditions. Hofmann et al. [50] developed what they called “a sequential PINN” that combined simulation data generated by a P2D model with experimental data from laboratory measurements and vehicle field data to train a neural network for SOH estimation. For the sake of consistency, we classify their PIML approach as data augmentation. When designing the input to the neural network, they utilized time-series signals such as voltage, current, temperature, SOC, and additional internal states from the simulation data (i.e., lithium-ion concentration and potential of the electrodes and electrolyte). Through feature engineering, they calculated various scalar and vectorized features for each time series signal. The final input is represented as a three-dimensional dataframe, encompassing both scalar and vectorized features. To standardize the size of input signals from diverse data sources, a binary decision feature was added, informing the network about data availability. The trained PINN model was evaluated against test datasets from simulation, laboratory, and field sources. The reported performance indicated the highest accuracy for the model when assessed on the simulation data and the lowest on the vehicle field data. In our preliminary study applying the PINN model for degradation diagnostics [51], we imposed half-cell model constraints on the predicted parameters of a shallow neural network. This was achieved by minimizing the difference between an experimentally measured $dQ/dV(V)$ curve and the corresponding generated curve from the half-cell model within the loss function. The proposed approach was evaluated using aging data from a long-term (3.5 years) cycling experiment on 16 implantable-grade lithium-ion cells.

The most common approach in the literature is to use physics-based models to generate training data for machine learning models. A few studies have employed equivalent circuit and electrochemical models to generate training data for deep neural networks, enabling the prediction of terminal voltage, temperature, and state of charge [52, 53, 54]. However, this approach has limitations in terms of model predictive accuracy, computational efficiency, and generalizability. These limitations arise from the fact that the neural network lacks information about the discrepancies between the physical model and the actual internal state of the battery. Moreover, none of the studies have specifically addressed the problem of degradation diagnostics, except for Thelen et al., who developed a lightweight PIML model to estimate cell degradation modes without relying on late-life aging data [30]. By training the model with physics-based simulation data, they reduced estimation errors significantly. However, this approach requires a prior understanding of the model’s inference space, which limits its ability to extrapolate to new conditions.

In this paper, we propose two PIML approaches, namely PINN and co-kriging, specifically designed for diagnosing cell degradation in the late aging stage without relying on late-life aging data for training. To evaluate the effectiveness of these methods, we conduct a comparative study that compares the proposed approaches to two other state-of-the-art PIML techniques: data augmentation and delta learning (more specifically, delta learning with the elastic net). Notably, co-kriging also falls within the broader category of delta learning. We compare the performance of each PIML method using battery aging data obtained from a long-term (> 4 years) cycling experiment involving 24 implantable-grade lithium-ion cells. The motivation behind this study stems from the need to comprehensively compare different PIML methods for battery degradation diagnostics. While PIML approaches have shown promise in enhancing prediction accuracy and generalizability in battery capacity estimation, a thorough investigation and comparison of their performance on battery degradation diagnostics is still lacking in the literature. By comparing PINN, co-kriging, delta learning, and data augmentation techniques, we aim to provide valuable insights into the strengths and limitations of each method. Additionally, we seek to identify the key factors that influence their performance, such as the availability of late-life aging data, the complexity of degradation processes, and the suitability of different machine learning algorithms for deployment in various situations. By

addressing these objectives, our study contributes to the understanding and advancement of PIML methods for degradation diagnostics and general prediction tasks.

2 Dataset Description

The battery aging dataset used in this work consists of 24 implantable-grade lithium-cobalt-oxide/graphite (LCO) cells tested in groups of four cells under six different operating conditions. The testing conditions of each group are outlined in Table 1. All cells are charged at C/3 to the upper voltage cutoffs, followed by constant voltage charging until the current is below C/50. Then, the cells are discharged with constant current to the lower voltage limit at 3.4 V. The aging tests were run for nearly five years in temperature-controlled thermal chambers. The capacity trajectories for this dataset can be seen in Fig. 1. An RPT was performed every 2 weeks during the first 3 months and every 4 weeks thereafter, to check the cells' capacity under a common discharge rate and obtain data for half-cell model analysis and SoH estimation. During RPT, a common procedure has been applied across all groups. First, cells are charged to 4.075 V using a constant-current-constant-voltage protocol with a constant current at C/3 and a constant-voltage cutoff current at C/50. Second, a constant-current discharge step and a constant-current charge step, both at C/50 and followed by a half-hour rest, are applied to obtain data at a slow rate. Third, cells are discharged to 3.4 V at C/10, with a one-hour rest for every 10% decrease in the state of charge. The temperature of thermal chambers is adjusted to 40 °C before every RPT.

Table 1: Summary of Aging Test Conditions

Group	Charge rate	Discharge rate	Temperature	Upper voltage cutoff	No. of cells
G1	C/3	C/24	37 °C	4.075 V	4
G2	C/3	C/24	55 °C	4.075 V	4
G3	C/3	C/3	37 °C	4.075 V	4
G4	C/3	C/3	55 °C	4.075 V	4
G5	C/3	C/10	37 °C	4.075 V	4
G6	C/3	C/24	37 °C	4.175 V	4

3 Half-Cell Model

The half-cell model is a non-destructive method for quantifying three dominant degradation modes in a lithium-ion battery (i.e., LAM_{PE}, LAM_{NE}, and LLI) by reconstructing a pseudo-OCV curve of the cell [30, 31]. The full-cell OCV curve is obtained by taking the potential difference between the positive and negative electrode OCV curves over the full-cell operating voltage range. At low rates ($\ll C/10$), kinetic and thermal effects during charge and discharge have minimal impact on the measured OCV of the cell, making it possible to closely model full-cell OCV as the potential difference between the two individual positive and negative electrode half-cells [31, 32, 55, 56].

The remaining active mass on the positive electrode (m_p) and the negative electrode (m_n) define the width of the positive and negative electrode's QV curves, respectively (Fig. 2a). The remaining capacity of the electrodes is obtained by multiplying the specific capacity (q_p and q_n) with the remaining active mass ($Q_p = m_p q_p$ and $Q_n = m_n q_n$). Then, the relative horizontal positions of these two half-cell curves are adjusted with respect to the left endpoint of the full-cell curve ($Q_C = 0$) by two slippage parameters, δ_p and δ_n (Fig. 2b). The reconstructed full-cell curve is obtained by limiting the voltage range of the full-cell curve to the usable voltage window (i.e., 3.4 V to 4.075 V for RPTs). With that, we can also calculate the remaining usable capacity of the cell

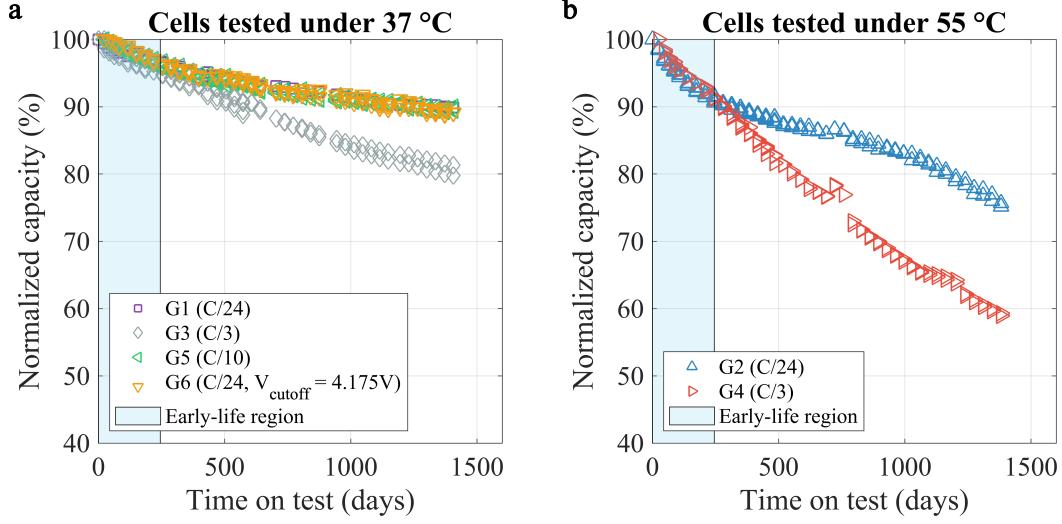


Figure 1: Overview of capacity fade curves for cells cycled under two distinct temperatures: **a.** 37°C ; **b.** 55°C . The discharge rate for each group of cells appears in the legend in parentheses (\cdot). The standard upper voltage cutoff (V_{cutoff}) is set at 4.075V . The early-life region includes approximately 8 months of experimental data.

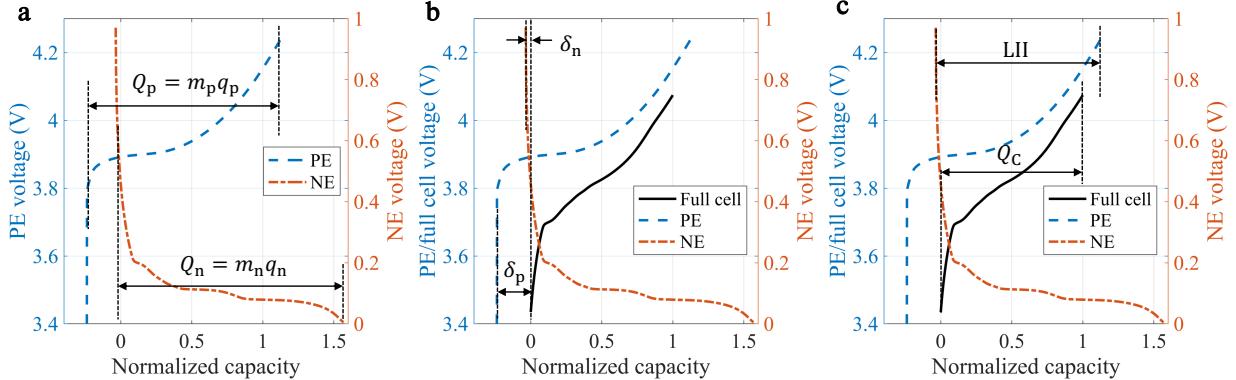


Figure 2: Overview of the half-cell model parameters. **a.** remaining active masses; **b.** slippage parameters; **c.** derived degradation parameters.

(Q_C) and the lithium inventory indicator (LII), graphically shown in Fig. 2c. LII can be calculated mathematically as $LII = Q_p - (\delta_p - \delta_n)$.

3.1 Degradation Mode Quantification Using the Half-Cell Model

The half-cell model is employed to quantify the true values of each degradation mode over the life of the cell. This is accomplished by manually fitting the half-cell model parameters (i.e., m_p , m_n , δ_p , and δ_n) to the measured C/50 voltage curves obtained from each RPT. Half-cell data from fresh electrodes was collected at C/50 to match the C-rate used in the RPT of the full-cell aging test. As the cell undergoes degradation, the widths of the positive and negative electrodes' QV curves, reflecting the remaining active mass on the electrodes (m_p and m_n), decrease. Simultaneously, the relative horizontal positions of the half-cell curves in relation to the left endpoint of the full-cell

curve (defined by two slippage parameters, δ_p and δ_n) also change due to the stoichiometric offset and loss of lithium inventory (LLI). Consequently, the four parameters of the half-cell model, m_p , m_n , δ_p and δ_n , influence the lengths and shapes of the two half-cell curves in specific ways, as illustrated in Fig. 2 and experimentally validated in [31]. Thus, adjusting these parameters allows us to simulate a pseudo-OCV full-cell curve that agrees well with an experimental full-cell curve. To make the fitting process more accurate in reflecting the underlying cell dynamics, we also considered the fitting of width and horizontal position (corresponding to m_n and δ_n) and magnitude (corresponding to m_p and δ_p) of certain features (e.g., peaks) on the differential voltage curves ($dV/dQ(Q)$) [30, 51, 56]. Manual fitting was often preferred to strike a balance between achieving a lower fitting error and leveraging the known physical meanings of certain features on the $dV/dQ(Q)$ curve, compared to automatic fitting [56]. However, this manual process is neither efficient nor practical for deployment in use cases where human intervention is not feasible (e.g., online health diagnostics inside a battery management system onboard an electric vehicle). To overcome these challenges and automate the process of degradation mode quantification, we proposed a PIML framework by leveraging both data-driven machine learning and physics-based modeling (i.e., the half-cell modeling in our earlier studies on degradation diagnostics [30, 51, 56]). Furthermore, a few cells (full-cells) were removed during the aging tests to fabricate aged positive and negative half-cells whose capacity was measured through charge/discharge cycling. This effort was made to experimentally estimate the true active mass in the electrodes of each aged full-cell, serving as an experimental validation of the parameter estimates from half-cell model fitting [56]. For groups G1 and G3, cells C1 and C2 were removed for analysis at day 573, and for groups G2 and G4, cells C3 and C4 were removed at day 484. These cells were disassembled, and half-cells were fabricated from their aged electrodes. The aged half-cells were then cycled, and the measured capacities were used to confirm the loss of active materials.

Lithium-ion battery degradation is complex, and rarely is a single degradation mode responsible for the observed capacity fade. Researchers have long used incremental capacity analysis (ICA) and differential voltage analysis (DVA) to qualitatively assess cell degradation and identify probable degradation modes. Both ICA and DVA are useful for visualizing degradation by tracking the changes in magnitude and location of peaks and valleys that correspond to Li phase changes inside the cell. Shown in Fig. 3a, we use the half-cell model to simulate full-cell OCV and plot the incremental capacity curves ($dQ/dV(V)$) corresponding to a 20% loss in a single degradation mode. The loss of active materials (LAM_{PE} and LAM_{NE}) causes changes in the magnitude and width of the peaks in the incremental capacity curves while a loss of lithium inventory LLI primarily decreases the magnitude of the peaks. Additionally, another set of simulations is run to visualize the effect of two simultaneous degradation modes on the shape of the incremental capacity curve. Shown in Fig. 3b, a loss of either active material combined with a loss of lithium inventory significantly reduces the magnitude of the peaks in the incremental capacity curve, especially the larger peak around 3.8 V.

3.2 Simulating Aging Data Using the Half-Cell Model

The second way we leverage the half-cell model is for simulating cell degradation. Just as we were able to show how each degradation parameter affects a cell's incremental capacity curve in Fig. 3, we can generate many different combinations of degradation parameters and their corresponding capacity-voltage curves. Many thousands of different input-output data pairs can be simulated and used for training machine learning models. Specifically, simulation is performed by selecting a wide range of half-cell model parameters and simulating the full-cell $Q(V)$ and $dQ/dV(V)$ curves with respect to different degradation parameters from the half-cell model.

4 Problem Definition

Existing approaches to estimating the degradation modes present in a battery cell have a few limitations that make them impractical for deployment in the field. They generally use expensive-to-

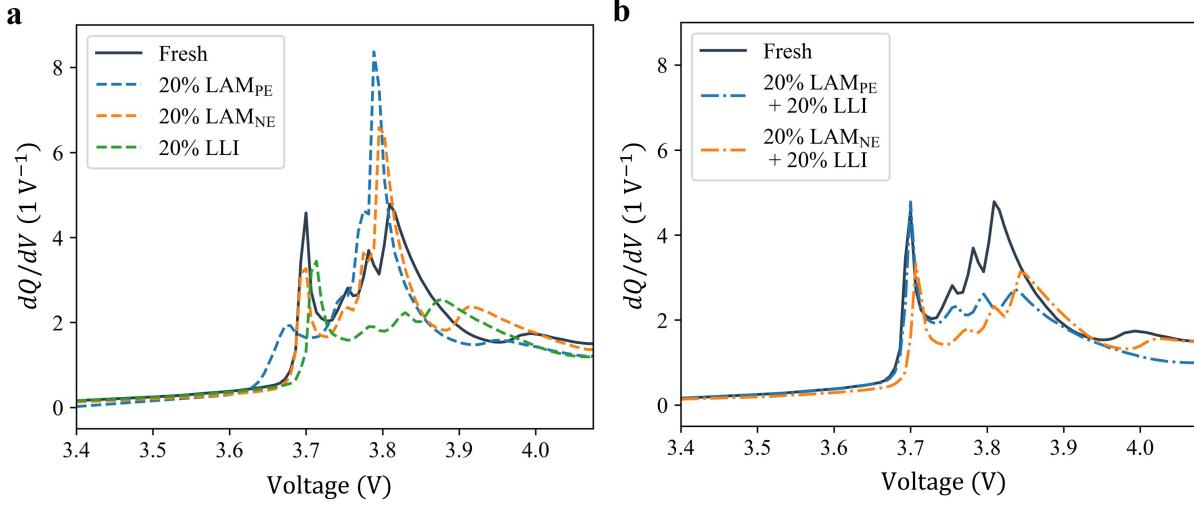


Figure 3: Simulated incremental capacity curves from the half-cell model with: **a.** one degradation mode; **b.** two degradation modes.

evaluate physical battery models [57, 58] or require collecting long-term aging test data [33, 59, 60], the latter of which makes deploying the methods extremely time and cost-intensive.

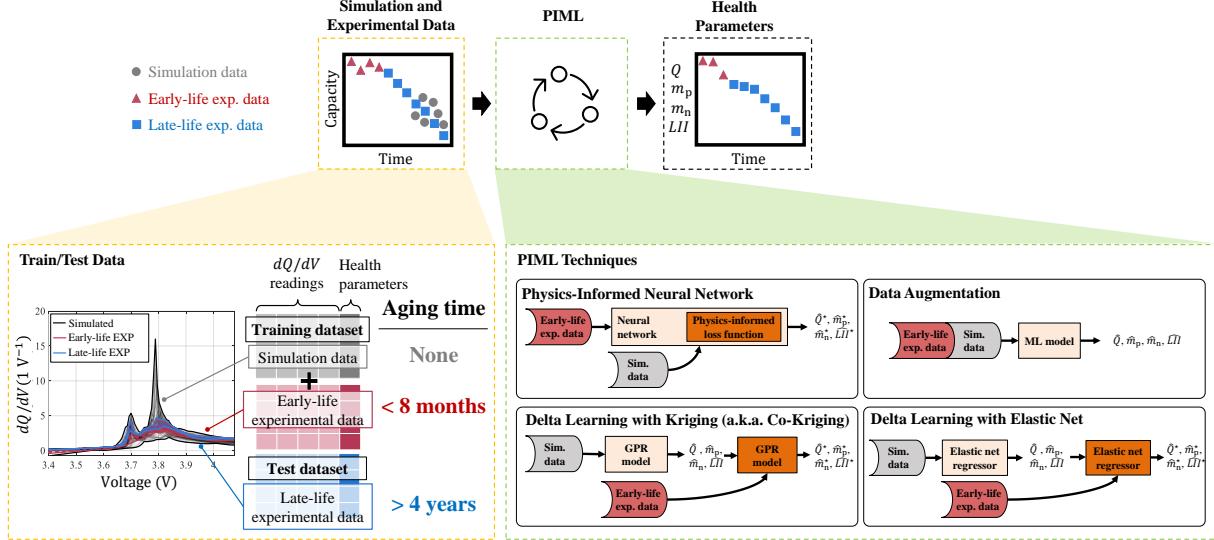


Figure 4: An overview of degradation diagnostics using PIML techniques.

Initial research by Thelen et al. showed that lightweight machine learning models could be trained to predict late-life degradation modes without using long-term aging data from experimental tests [30]. The models learned from a combination of experimental data and physics-based simulation data to estimate cell capacity and three degradation modes. We further investigate this line of research, and in this work, comprehensively study new state-of-the-art physics-informed machine learning (PIML) methods for degradation diagnostics.

We explore the integration of physics into lightweight machine learning models through four unique approaches. Specifically, we train physics-informed machine learning models to estimate cell

capacity and the state of three primary degradation modes (LAM_{PE} , LAM_{NE} , and LLI) using the measured incremental capacity curve $dQ/dV(V)$ as input data. Using the differential capacity curve as input is common practice and eliminates the need for manual feature selection and enhances the applicability of the models and methods to other battery chemistries. The primary focus of this research is to conduct a comparative analysis of four distinct methods that incorporate physics-based knowledge into machine learning models. We aim to identify the most effective approach in terms of model accuracy and performance, ultimately providing insight into the optimal integration of physics concepts into machine learning frameworks for battery degradation diagnostics.

We study the degradation diagnostics problem using a framework inspired by Thelen et al. This framework leverages both early-life experimental data and simulation data from a half-cell model to construct the training dataset for our PIML methods. The process involves fine-tuning four adjustable parameters (m_p , m_n , δ_p , δ_n) in the half-cell model to match the reconstructed pseudo-OCV curves with those obtained from RPTs. This yields the $dQ/dV(V)$ curves, their corresponding fitted half-cell model parameters, and four health parameters (cell's capacity (Q) and degradation parameters (m_p , m_n , and LLI)) for both the early-life and late-life stages of the aging tests. The early-life experimental data serve as the training set, while the late-life experimental data act as the testing set for our models. We employ the half-cell model to simulate data points on the $dQ/dV(V)$ curves, along with their corresponding health parameters, for both the medium and late-life stages of each cell. These simulated data points are then incorporated into the training process alongside experimental data. The PIML methods are then trained on both the early-life experimental data and the simulated data to predict the capacity and degradation parameters for the late-life stage. Finally, we evaluate the performance of our models using late-life experimental data. Fig. 4 provides an overview of the framework.

5 Physics-Informed Machine Learning (PIML) Techniques

The following sections highlight the key characteristics of the four PIML techniques employed in our study. These PIML methods are (1) physics-informed neural networks (PINN), (2) data augmentation, (3) delta learning with kriging (a.k.a., multi-fidelity co-kriging), and (4) delta learning with the elastic net. Fig. 5 provides a visual representation of these methods. A common objective among these techniques is to utilize simulation data from the late-life stage, thereby simplifying the task of extrapolating beyond the early-life experimental data toward late-life regions for the machine learning model.

5.1 Physics-Informed Neural Network (PINN)

Physics-informed neural networks (PINNs) offer the advantage of fully integrating the problem's physics into their loss function, thereby enhancing generalization performance through the utilization of data and mathematical operators based on underlying physical principles [43]. In our implementation, we leverage the underlying physics of the half-cell model discussed in Sec. 3 to guide the training of a neural network. This enables the network to learn the relationship between the input (a $dQ/dV(V)$ curve) and the health parameters (Q , m_p , m_n , and LLI) throughout the battery's lifetime. As illustrated in Fig. 5a, the late-life simulation data, which includes data points corresponding to the highest 20% degradation in health parameters, is combined with the early-life experimental data to train the PINN model. This approach ensures that heavily-aged $dQ/dV(V)$ curves in the late-life region, along with their corresponding health parameters, are incorporated into the training process. To improve cell degradation estimation accuracy, particularly during the late-aging stage, we incorporate knowledge of the relationship between health parameters and $dQ/dV(V)$ into the model by introducing two physics-informed loss terms in addition to the standard data-driven loss. Specifically, to harness the known physics embedded in the half-cell model within the PINN model, we adopt a two-step process of first predicting the half-cell model parameters and then mapping them into the capacity and a degradation parameter, as shown in

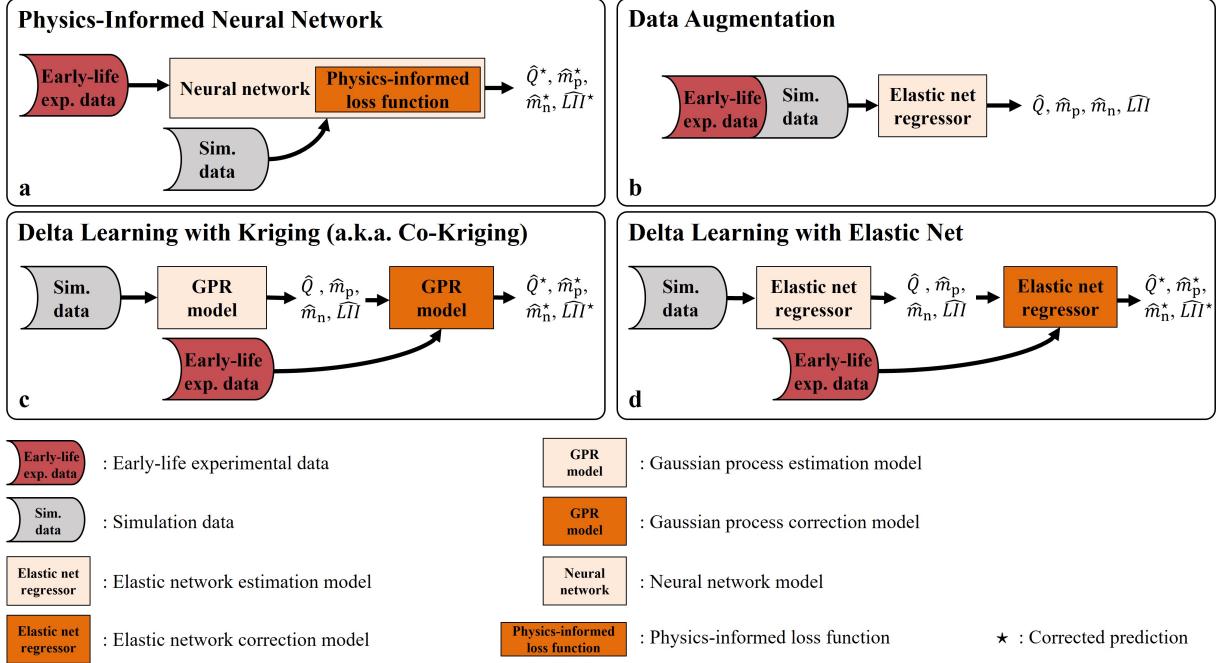


Figure 5: Overview of physics-informed machine learning (PIML) techniques adopted in this comparative study.

Fig. 6b, rather than directly predicting the health parameters, as implemented in a baseline neural network (depicted in Fig. 6a).

- Initially, a shallow neural network predicts the half-cell model parameters (m_p , m_n , δ_p , and δ_n) for a given measurement of the $dQ/dV(V)$ curve from a fresh/aged full-cell. This neural network handles the reverse process of half-cell modeling, where the half-cell model parameters are estimated by matching the model-simulated and experimentally measured full-cell QV curves. This model fitting process using the physics-based half-cell model is slow and computationally demanding due to its iterative nature, prompting us to train a neural network to perform it more efficiently.
- Then, to translate these predicted half-cell model parameters into the battery capacity (Q) and the lithium inventory indicator (LII), we integrate a surrogate of the half-cell model (f_{hc}) into the loss function. In defining the physics-informed loss terms, we leverage the forward mapping of the physics-based half-cell model, which proves to be straightforward and fast.

As depicted in Fig. 6, a shallow network is utilized to establish a mapping of the 100 points on the early-life experimental and late-life simulation curves of $dQ/dV(V)$ to the four half-cell model parameters (δ_p , δ_n , m_p , m_n). The standard data-driven loss is calculated using the predicted half-cell model parameters and their corresponding true values obtained from early-life experimental and late-life simulated data. Let $\mathbf{y} = [m_p, m_n, \delta_p, \delta_n]^T$ be the vector of true values and $\hat{\mathbf{y}} = [\hat{m}_p, \hat{m}_n, \hat{\delta}_p, \hat{\delta}_n]^T$ be the vector of predicted parameters. The MSE over both the early-life experimental points and late-life simulated points can be calculated as:

$$\mathcal{L}_1 = \mathcal{L}_{\text{MSE}}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{N_{\text{exp}} + N_{\text{sim}}} \left(\sum_{i=1}^{N_{\text{exp}}} \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|^2 + \sum_{j=1}^{N_{\text{sim}}} \|\mathbf{y}_j - \hat{\mathbf{y}}_j\|^2 \right) \quad (1)$$

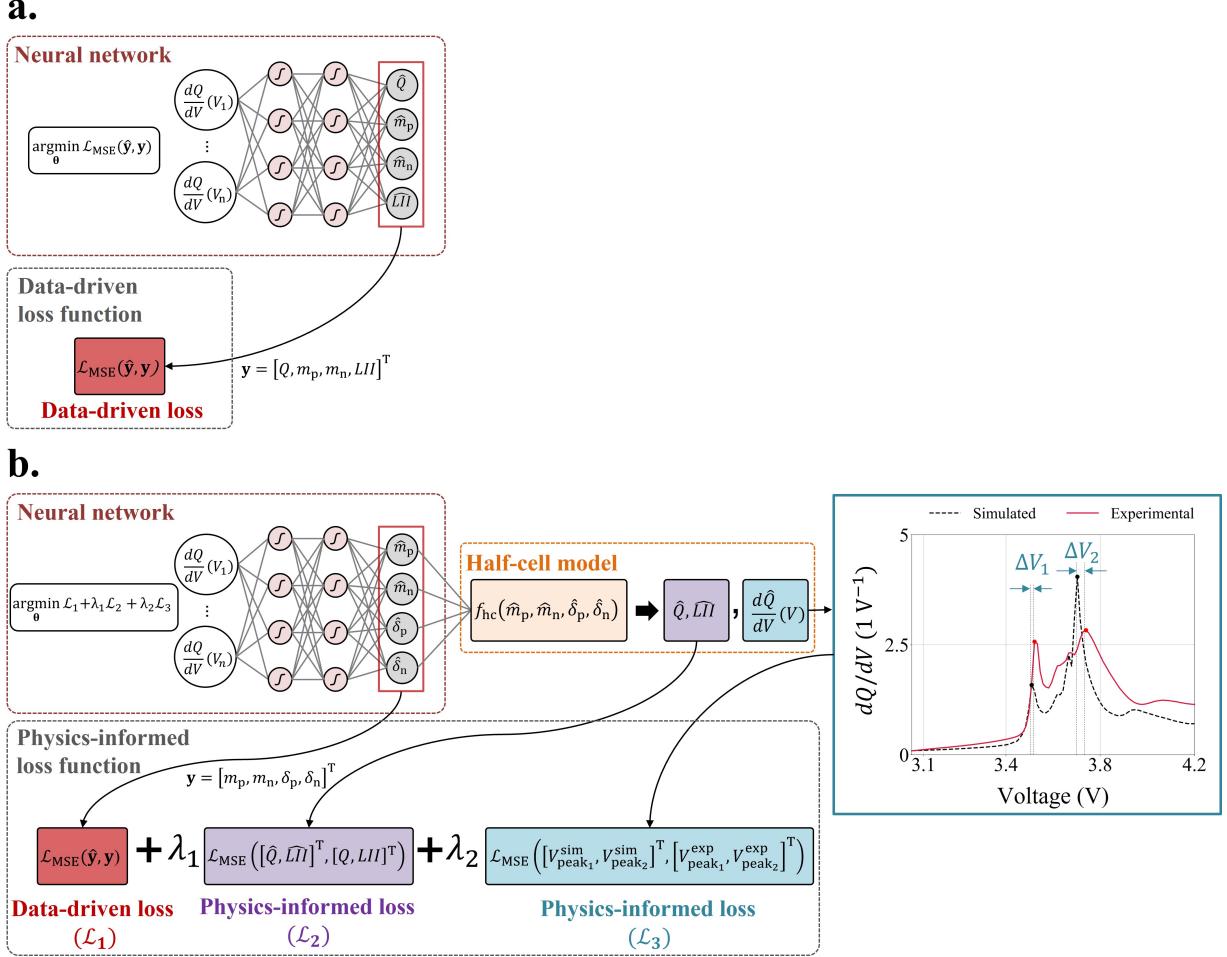


Figure 6: An overview of the architectures of a baseline neural network (a) and PINN (b) for battery degradation diagnostics depicting data-driven and physics-informed loss terms.

where N_{exp} and N_{sim} denote the numbers of data points for the early-life experimental points and late-life simulated points, respectively; y_i and \hat{y}_i represent the true and predicted parameter vectors for the i -th early-life point; and y_j and \hat{y}_j represent the j -th element of the true health parameters vector and predicted vector for the late-life points.

Within the loss function, a half-cell surrogate model, denoted as f_{hc} , is integrated to establish a mapping between the predicted half-cell model parameters and the battery capacity (Q) and the lithium inventory indicator (LII). This integration ensures that the values of battery capacity and lithium inventory remain consistent with the physics-based half-cell model by verifying that the network-predicted half-cell model parameters lead to simulated curves with endpoints that closely match those of the true fitted curves. Consequently, a second loss term is generated to measure the difference between the capacity(\hat{Q}) and LII degradation parameter obtained by passing the predicted half-cell model parameters from the network into the half-cell surrogate model and the true ones. Let $\mathbf{U} = [Q, LII]^T$ be the vector of true values and $\hat{\mathbf{U}} = [\hat{Q}, \hat{LII}]^T$ be the vector of outputs from the half-cell surrogate. The \mathcal{L}_2 loss can be calculated as:

$$\mathcal{L}_2 = \mathcal{L}_{\text{MSE}}(\hat{\mathbf{U}}, \mathbf{U}) = \frac{1}{N} \sum_{k=1}^N \|\hat{\mathbf{U}}_k - \mathbf{U}_k\|^2 \quad (2)$$

where $N = N_{\text{exp}} + N_{\text{sim}}$ represents the total number of data points corresponding to the early-life experimental and late-life simulated measurements. Here, $\hat{\mathbf{U}}_k$ and \mathbf{U}_k represent the k -th element of the predicted and true parameters vector, respectively. The use of a half-cell surrogate model prevents any disruption in the gradient computation process that would occur if the half-cell model were directly used within the loss function. This ensures that the predicted parameters remain connected to the computational graph in each epoch, enabling the computation of the gradient of the network output with respect to the inputs.

Another valuable piece of information that can be integrated into the physics-based loss function is the difference between the peak positions (voltages) observed in the simulated and experimental $dQ/dV(V)$ curves. Typically, two main peaks can be observed on the $dQ/dV(V)$ curve. The first peak at the lower voltage corresponds to the intercalation of lithium ions into the electrode material during discharge. As lithium ions are released from the anode during discharge, they intercalate into the cathode's active material, leading to an increase in capacity. The second peak corresponds to the electrochemical reactions occurring at the cathode, such as the deintercalation of lithium ions from the cathode material. The shape, intensity, and position of these peaks provide valuable insights into the electrochemical behavior and performance of the battery [61]. As we have constrained two of the parameters in the loss function (Q and LII), we also aim to constrain the other two predicted parameters, which are the positive and negative active masses (m_p and m_n). Since the phase transition peaks in the $dQ/dV(V)$ curve are sensitive to the values of m_p and m_n [56], by constraining the position of these peaks in the loss function, we can ensure that the resulting active mass parameters are consistent with the half-cell model constraints. To incorporate this information, we introduce an additional loss term that calculates the difference between the positions of the two major peaks in the experimental $dQ/dV(V)$ curves and their corresponding simulated counterparts. To achieve this, we extend the half-cell surrogate model to map the predicted half-cell model parameters ($\delta_p, \delta_n, m_p, m_n$) by the network to the positions of the two major peaks in the $dQ/dV(V)$ curves. A peak detection algorithm is employed, utilizing criteria such as prominence, distance, height, and width, to identify the major peaks on the curves. The identified peaks are then sorted in descending order based on their corresponding y-values, and the first two peaks are selected for comparison. Let $\hat{\mathbf{V}}_{\text{peak}}^{\text{sim}} = [\hat{V}_{\text{peak}_1}^{\text{sim}}, \hat{V}_{\text{peak}_2}^{\text{sim}}]^T$ represent the vector of peak positions on the simulated $dQ/dV(V)$ curve, and $\mathbf{V}_{\text{peak}}^{\text{exp}} = [V_{\text{peak}_1}^{\text{exp}}, V_{\text{peak}_2}^{\text{exp}}]^T$ represent the vector of actual peak positions corresponding to the training set. The loss function \mathcal{L}_{MSE} is utilized to measure the mean squared error between these vectors, yielding the loss term \mathcal{L}_3 :

$$\mathcal{L}_3 = \mathcal{L}_{\text{MSE}}(\hat{\mathbf{V}}_{\text{peak}}^{\text{sim}}, \mathbf{V}_{\text{peak}}^{\text{exp}}) = \frac{1}{2} \left(\|\hat{\mathbf{V}}_{\text{peak}_1}^{\text{sim}} - \mathbf{V}_{\text{peak}_1}^{\text{exp}}\|^2 + \|\hat{\mathbf{V}}_{\text{peak}_2}^{\text{sim}} - \mathbf{V}_{\text{peak}_2}^{\text{exp}}\|^2 \right) \quad (3)$$

Finally, to ensure that the resulting loss terms are on the same scale, we compute a weighted summation of these terms, as depicted in Fig. 6.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_1 + \lambda_1 \mathcal{L}_2 + \lambda_2 \mathcal{L}_3 \quad (4)$$

where λ_1 , and λ_2 are positive scalar values that control the relative weight of each physics-informed loss term, and they are empirically determined. The weighted summation ensures that the different loss terms contribute proportionally to the overall loss function, allowing for a balanced optimization process. The custom total loss term directly reflects the higher-level objective we want the model to minimize, rather than relying solely on the loss between the true and predicted values of the network. Essentially, the training process involves updating the network weights in a manner that minimizes the customized total loss term value computed in each iteration.

5.2 Data Augmentation

Data augmentation is a technique used to enhance the accuracy of machine learning models by expanding the training data. One approach involves leveraging a simple and efficient physics-based simulation model to generate additional input/output samples that closely resemble the original dataset. Augmenting samples into the training data increases the dataset's size and diversity, which enables the model to learn from a wider range of examples [62].

In this study, we enhance the training dataset by incorporating simulation data from the half-cell model, initially comprising only early-life battery degradation data from the cycling experiment. The simulation data is sampled from the entire design space of health parameters, including data from the late-life stage. This augmented dataset provides more information about the future degradation path of the experimental cells than what can be learned solely from the early-life experimental data. Consequently, it facilitates a more accurate estimation of late-life health parameters. For a visual representation of this approach, refer to Fig. 5 c. It is worth noting that the augmentation process can be implemented in different ways, depending on the specific selection of simulated degradation data to be combined with the early-life experimental data. One approach is to add simulation data points that are filtered to include only the highest degradation values for each of the health parameters: Q , m_p , m_n , and LII . By focusing on these higher degradation levels, the augmented dataset may offer additional insights into the extreme trends and behavior exhibited by the battery cells.

5.3 Delta Learning

Delta learning is a technique where a secondary model is trained to correct the errors or biases of a primary model. The secondary model, known as the delta model or correction model, is specifically designed to account for discrepancies between the predictions of the primary model and the observed experimental data. By incorporating the learned corrections from the delta model, the accuracy and performance of the primary model can be improved [36]. In this study, as depicted in Fig. 5 b and Fig. 5c, we implement delta learning using two different machine learning methods: GPR, also known as kriging, which leads to the formulation of multi-fidelity co-kriging, and the elastic net, which corresponds to a conventional form of delta learning. The key difference between these two delta learning approaches lies in the unique probabilistic formulation of co-kriging for computing the posterior distribution or final prediction, in contrast to the deterministic formulation of the elastic net.

5.3.1 Delta Learning with Kriging (a.k.a Multi-Fidelity Co-Kriging)

GPR, also known as kriging, is a probabilistic machine learning method used to estimate an unknown mapping function $f(\cdot)$. It samples a random realization of this function from a Gaussian process, representing a distribution over a collection of functions. A Gaussian process is defined by its mean function, $\mu(\cdot)$, and covariance function (kernel), $k(\cdot, \cdot)$. The mean function captures the expected value or general trend of the data being modeled, while the kernel describes the covariance between the function outputs at two different inputs. In GPR, a Gaussian process prior is assumed on the underlying mapping function, $f(\cdot)$, using a prior covariance function. This covariance function incorporates hyperparameters and a mean function, which can be chosen based on existing knowledge or assumptions. Various types of kernels, such as linear, quadratic, or general basis functions, can be considered, depending on the prior understanding of the problem. An example of a common kernel function is provided in Eq. (5) which is used in this study as well. It incorporates both distance-based and smoothness-based components to capture the underlying patterns in the data.

$$k_{\text{Matern}}(\mathbf{x}, \mathbf{x}') = \sigma^2 \left(1 + \frac{\sqrt{2\nu}}{l} \|\mathbf{x} - \mathbf{x}'\| \right) \exp \left(-\frac{\sqrt{2\nu}}{l} \|\mathbf{x} - \mathbf{x}'\| \right) \quad (5)$$

where σ^2 represents the variance, controlling the overall variability of the Gaussian process. The parameter ν determines the smoothness of the function, with half-integer values (e.g., 1/2, 3/2, 5/2) providing a flexible and smooth representation of the data. The length scale parameter l determines the range over which the correlation between data points decreases. Smaller values of l result in a more localized correlation, while larger values yield a smoother and more global correlation. The term $\|\mathbf{x} - \mathbf{x}'\|$ denotes the Euclidean distance between data points x and x' . The exponential function $\exp(\cdot)$ attenuates the correlation as the distance between data points increases.

Several methods based on Gaussian processes have been developed to combine information from models with different levels of fidelity, depending on the complexity of the relationships between these fidelities. In the field of engineering design, a commonly used approach is co-kriging, which combines multiple models providing different levels of accuracy or fidelity. It assumes a linear relationship between these models, allowing for the integration of their predictions. Building upon the two-level multi-fidelity modeling proposed in [63], we consider high-fidelity data (i.e., early-life experimental data) denoted as $\mathbf{Y}_H = (\mathbf{y}_H^{(1)}, \dots, \mathbf{y}_H^{(N_H)})^T$ at locations $\mathbf{X}_H = (\mathbf{x}_H^{(1)}, \dots, \mathbf{x}_H^{(N_H)})^T$, and low-fidelity data (i.e., simulation data spanning the entire lifetime of the battery cells) denoted as $\mathbf{Y}_L = (\mathbf{y}_L^{(1)}, \dots, \mathbf{y}_L^{(N_L)})^T$ at locations $\mathbf{X}_L = (\mathbf{x}_L^{(1)}, \dots, \mathbf{x}_L^{(N_L)})^T$. Here, $\mathbf{y}_H^{(i)}, \mathbf{y}_L^{(i)} \in \mathbb{R}^4$ (since we are predicting cell's capacity and three degradation parameters), and $\mathbf{x}_H^{(i)}, \mathbf{x}_L^{(i)} \in \mathbb{R}^d$, where d represents the number of points on the input $dQ/dV(V)$ curve. By concatenating the data from both fidelities, i.e., $\mathbf{X} = \{\mathbf{X}_H, \mathbf{X}_L\}$ and $\mathbf{Y} = \{\mathbf{Y}_H, \mathbf{Y}_L\}$, we can construct a multivariate Gaussian process using co-kriging based on the auto-regressive model [64].

$$f_H(\mathbf{x}) = \rho f_L(\mathbf{x}) + f_\Delta(\mathbf{x}), \quad (6)$$

where \mathbf{x} represents the input vector at which the prediction is made, $f_L(\mathbf{x})$ and $f_H(\mathbf{x})$ are the low-fidelity and high-fidelity Gaussian processes that model the simulation and experimental data, respectively. $f_\Delta(\mathbf{x})$ is a Gaussian process that captures the difference between $f_H(\mathbf{x})$ and $\rho f_L(\mathbf{x})$, and ρ is a regression parameter. In this approach, it is assumed that:

$$\text{Cov}\{f_H(\mathbf{x}), f_L(\mathbf{x}') | f_L(\mathbf{x})\} = 0, \quad \forall \mathbf{x} \neq \mathbf{x}', \mathbf{x}, \mathbf{x}' \in \mathbb{R}^d, \quad (7)$$

To implement this formulation, as illustrated in Fig. 5b, we first train a GPR model to construct the $f_L(\cdot)$ using the dataset $\{\mathbf{X}_L, \mathbf{Y}_L\}$. Next, we calculate the discrepancy matrix $\mathbf{Y}_\Delta = \mathbf{Y}_H - \rho f_L(\mathbf{X}_H)$, where $f_L(\mathbf{X}_H)$ represents the mean values of $f_L(\cdot)$ at the corresponding locations common to those of \mathbf{X}_H . Finally, we train another GPR model to obtain $f_\Delta(\cdot)$ by training it on the dataset $\{\mathbf{X}_H, \mathbf{Y}_\Delta\}$. The final high-fidelity prediction at a new location \mathbf{x}_* , i.e., a vector of input points on a $dQ/dV(V)$ curve, consists of a posterior distribution $\hat{\mathbf{y}}(\mathbf{x}_*) \sim \mathcal{N}(\hat{\mu}(\mathbf{x}_*), \hat{s}^2(\mathbf{x}_*))$, with posterior mean and variance given by the following equations [65].

$$\hat{\mu}(\mathbf{x}_*) = \rho \hat{\mu}_L(\mathbf{x}_*) + \hat{\mu}_\Delta(\mathbf{x}_*) + \mathbf{c}_\Delta(\mathbf{x}_*) \mathbf{C}_\Delta(\mathbf{X}_H, \mathbf{X}_H)^{-1} (\mathbf{Y}_\Delta - \hat{\mu}_\Delta(\mathbf{X}_H)), \quad (8)$$

$$\hat{s}^2(\mathbf{x}_*) = \rho^2 \sigma_L^2(\mathbf{x}_*) + \sigma_\Delta^2(\mathbf{x}_*) - \mathbf{c}(\mathbf{x}_*)^T \mathbf{C}^{-1} \mathbf{c}(\mathbf{x}_*), \quad (9)$$

where $\hat{\mu}_L(\mathbf{x}_*)$ is obtained based on $f_L(\cdot)$, and $\hat{\mu}_\Delta(\mathbf{X}_H) = (\hat{\mu}_\Delta^{(1)}, \dots, \hat{\mu}_\Delta^{(N_H)})^T$ and $\mathbf{C}_\Delta(\mathbf{X}_H, \mathbf{X}_H)$ is obtained based on constructed $f_\Delta(\cdot)$. Also, $\sigma_L^2(\mathbf{x}_*) = k_L(\mathbf{x}_*, \mathbf{x}_*)$, and $\sigma_\Delta^2(\mathbf{x}_*) = k_\Delta(\mathbf{x}_*, \mathbf{x}_*)$, and the vector of covariances (\mathbf{c}) for a given input \mathbf{x}_* , is expressed by Eq. (10).

$$\mathbf{c}(\mathbf{x}_*) = \begin{bmatrix} \rho \mathbf{c}_L(\mathbf{x}_*) \\ \mathbf{c}_H(\mathbf{x}_*) \end{bmatrix} = \begin{bmatrix} [\rho k_L(\mathbf{x}_*, \mathbf{x}_L^{(1)}), \dots, \rho k_L(\mathbf{x}_*, \mathbf{x}_L^{(N_L)})]^T \\ [k_H(\mathbf{x}_*, \mathbf{x}_H^{(1)}), \dots, k_H(\mathbf{x}_*, \mathbf{x}_H^{(N_H)})]^T \end{bmatrix}, \quad (10)$$

where $k_H(\mathbf{x}, \mathbf{x}') = \rho^2 k_L(\mathbf{x}, \mathbf{x}') + k_\Delta(\mathbf{x}, \mathbf{x}')$ [66] and \mathbf{c}_Δ is obtained as $\mathbf{c}_\Delta(\mathbf{x}_*) = \mathbf{c}_H(\mathbf{x}_*) - \rho^2 \mathbf{c}_L(\mathbf{x}_*)$. Additionally, the covariance matrix representing the covariances across all observations is denoted as \mathbf{C} and is defined by Eq. (11).

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_L(\mathbf{X}_L, \mathbf{X}_L) & \rho \mathbf{C}_L(\mathbf{X}_L, \mathbf{X}_H) \\ \rho \mathbf{C}_L(\mathbf{X}_H, \mathbf{X}_L) & \rho^2 \mathbf{C}_L(\mathbf{X}_H, \mathbf{X}_H) + \mathbf{C}_\Delta(\mathbf{X}_H, \mathbf{X}_H) \end{bmatrix}, \quad (11)$$

where \mathbf{C}_L represents the covariance matrix based on the kernel of $f_L(\cdot)$, and \mathbf{C}_Δ represents the covariance matrix based on the kernel of $f_\Delta(\cdot)$, we can identify their hyperparameters, along with ρ , by assuming parameterized forms for these kernels. This identification process is achieved through the maximization of the log-marginal likelihood term shown in Eq. (12).

$$\ln L = -\frac{1}{2}(\mathbf{Y} - \mu)^T \mathbf{C}^{-1} (\mathbf{Y} - \mu) - \frac{1}{2} \ln |\mathbf{C}| - \frac{N_H + N_L}{2} \ln(2\pi). \quad (12)$$

The steps of the implemented algorithm are summarized in Algorithm 1.

Algorithm 1 Two-fidelity co-kriging algorithm

Step 1: Use a GPR model trained on simulation data generated from the half-cell model across the entire design space of health parameters to construct $f_L(\cdot)$ and compute $\hat{\mu}_L(\mathbf{X}_L) = (\hat{\mu}_L^{(1)}, \dots, \hat{\mu}_L^{(N_L)})^T$ and $\mathbf{C}_L(\mathbf{X}_L, \mathbf{X}_L)$.

Step 2: Use $f_L(\cdot)$ to estimate $\hat{\mathbf{y}}_H$ at locations corresponding to \mathbf{X}_H : $f_L(\mathbf{X}_H)$.

Step 3: Calculate the discrepancy matrix as follows: $\mathbf{Y}_\Delta = \mathbf{Y}_H - \hat{\mu}_L(\mathbf{X}_H)$. This is based on the reasoning that $\hat{\mu}_L(\mathbf{X}_H)$ is the most probable observation of the Gaussian process $f_L(\cdot)$.

Step 4: Choose a specific kernel function $k_\Delta(\cdot, \cdot)$ (Matern kernel in this work), and train a GPR model using the dataset $\{\mathbf{X}_H, \mathbf{Y}_\Delta\}$ to construct $f_\Delta(\cdot)$. Then compute $\hat{\mu}_\Delta(\mathbf{X}_H) = (\hat{\mu}_\Delta^{(1)}, \dots, \hat{\mu}_\Delta^{(N_H)})^T$ and $\mathbf{C}_\Delta(\mathbf{X}_H, \mathbf{X}_H)$.

Step 5: The posterior mean and variance for any given input, \mathbf{x}_* , are computed by Eq. (8) and Eq. (9), respectively.

5.3.2 Delta Learning with Elastic Net

In another implementation of delta learning (see Fig. 5d), we employ a simple elastic net model trained on simulation data to estimate the health parameters of cells in the late-life stage. However, since the simulation data does not precisely reflect the heavily aged $dQ/dV(V)$ curves observed in the experimental tests, the estimation model alone may produce inaccurate predictions. To overcome this limitation, we introduce a corrector elastic net model which is trained using early-life degradation data, and its purpose is to capture the "delta" or prediction bias of the first model for the light-degradation region. This learned bias can be extended to effectively correct the outputs in the late-life stage. During the training phase, the corrector model leverages the available early-life experimental data to learn the prediction bias of the estimation model. In the testing phase, both the estimation model and the corrector model work in conjunction. The final prediction is generated by combining the output of the estimation model with the correction provided by the corrector model. It is worth noting that the implementation structures of delta learning and co-kriging are quite similar in this study. The main difference lies in the choice of estimation and corrector models. In co-kriging, we utilized two GPR models, whereas in delta learning, we employed two elastic net models. Broadly speaking, we can consider the co-kriging implementation in this study as a special case of delta learning.

6 Results

In this section, we present a detailed comparative analysis of four PIML methods for battery cell degradation diagnostics. Our goal is to assess each model's performance in comparison to its data-driven baseline and explore specific analyses for the novel approaches, PINN and co-kriging.

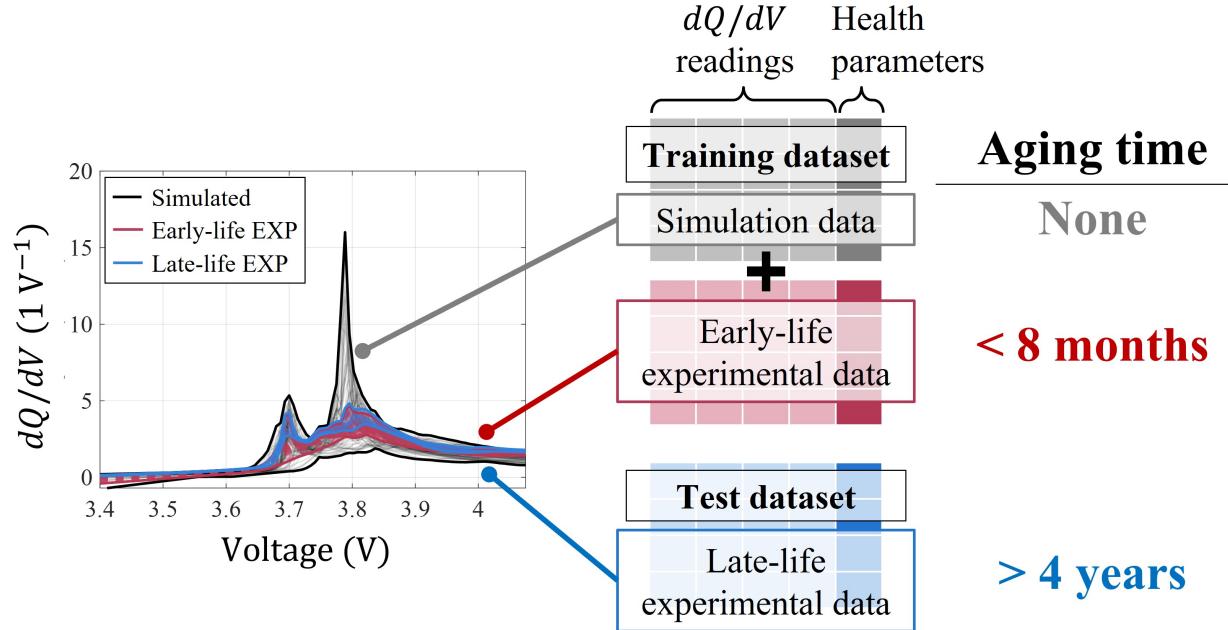


Figure 7: Training and test setup for PIML approaches.

We start by evaluating the overall performance of each PIML model relative to its data-driven counterpart, providing insights into their predictive accuracy. We also investigate the impact of varying training sample sizes on performance and assess the extrapolation capabilities of PINN and co-kriging. Finally, we conduct sensitivity analysis by investigating the effects of changing model parameters and structure. Through this analysis, we offer a comprehensive view of the strengths and limitations of each PIML method in battery cell degradation diagnostics.

6.1 Training and Test Setup

To assess the effectiveness of the PIML techniques, we trained each model using the initial 10 characterization points, representing approximately seven months of experimental aging data. Subsequently, we compared the performance of each PIML model with its respective data-driven baseline counterpart, which was solely trained on early-life experimental data points. The key distinction between the PIML methods and their baseline counterparts lies in how they integrate the late-life simulation data into the training process. These methods use different approaches to incorporate this data, aiming to achieve improved extrapolation performance in the late-life stage. The general setup for train/test split is shown in the Fig. 7.

To comprehensively evaluate the performance of the PIML models, we conducted a four-fold cross-validation study. This approach ensures that the entire experimental dataset, comprising 24 cells, is thoroughly assessed while considering potential variations across different subsets of the data. In our study, we divided the dataset into four mutually exclusive folds, with each fold including one battery cell from each group in the test set. Table 2 summarizes the numbers of training and test points for each fold.

In the following subsections, we provide details about the parameter settings implemented in the training process for PINN and co-kriging, two newly proposed approaches.

Table 2: Numbers of training points in 4-fold cross-validation

Method	Fold	Training Points	Testing Points
PINN	1,2	212	200
	3,4	212	192
Data Augmentation	1,2	212	200
	3,4	212	192
Delta Learning	1,2	964	200
	3,4	964	192
Baseline Methods	1,2	180	200
	3,4	180	192

6.1.1 PINN

We opted for a simplified neural network architecture in our PINN approach, comprising two hidden layers. This choice reflects our emphasis on simplicity and considers the size limitation of the available data, which makes a complex neural network susceptible to overfitting. By employing a simpler network architecture with fewer parameters, we intentionally limit the model's ability to memorize noise present in the data, thereby promoting improved generalization. To further mitigate the risk of overfitting, we implemented an early stopping mechanism. This mechanism involved monitoring the validation loss during training and terminating the training process when the validation loss started to increase. Additionally, we incorporated cross-validation techniques and integrated simulated data from the late-life stage into the training process. These strategies collectively contribute to enhancing the generalization capability of the PINN model and serve as effective safeguards against overfitting, differentiating it from a conventional neural network model. During model training, we utilized the Adam optimization algorithm to iteratively update network weights. The rectified linear unit (ReLU) function was used as the activation function throughout the training process, defined as:

$$\text{ReLU}(x) = \max(0, x) \quad (13)$$

The hyperparameters of the network are summarized in Table 3.

Table 3: Neural network hyperparameters

Hyperparameter	Value
Number hidden of layers	2
Max epoch	1000-2000
Number of hidden neurons	30
Learning rate	0.005-0.01
Batch size	200
Trainable parameters	2274

In the testing phase, the trained network operates independently, removing the need for the physics-informed loss function. When combined with the half-cell model, the trained network assesses degradation levels at various stages of cell life. Essentially, the trained network takes over the manual half-cell fitting process by predicting the half-cell model parameters.

6.1.2 Co-Kriging

Training a kriging (GPR) model with a specific kernel involves determining several parameters to effectively capture the underlying patterns in the data. The Matern kernel, which is used in this study, is a versatile kernel function that can be adjusted with a few hyperparameters to fit the characteristics of the data. The key parameters that should be determined when training a GPR model with a Matern kernel are summarized in the Table 4.

Table 4: Key parameters for training GPR models with Matern kernel for each fold

Parameter	Estimation Model	Correction Model
Length scale (l)	5258.770	1.000
Signal variance (σ^2)	0.015	0.006-0.008
Smoothness parameter (ν)	3/2	3/2

6.2 Error Metrics

For each fold, we calculated an average test error of each individual PIML model for each of the four health parameters (Q , m_p , m_n , and LII). These errors measured the average differences between the predicted parameter values and the corresponding ground truth values, which were the degradation parameters obtained through fitting the half-cell model to experimental full-cell curves (see Sec. 3.1). We computed a normalized average of the individual health parameter test errors across the four folds using the following root mean squared percentage error (RMSPE) formula:

$$RMSPE_t = \sqrt{\frac{1}{\sum_{k=1}^4 N_k} \sum_{k=1}^4 \sum_{i=1}^{N_k} \left(\frac{\hat{y}_{t,i} - y_{t,i}}{y_{t,i}} \right)^2} \times 100\% \quad (14)$$

Here, N_k represents the number of test samples, and the subscript t denotes the t -th degradation parameter. $\hat{y}_{t,i}$ and $y_{t,i}$ correspond to the predicted value and true value, respectively, for the t -th health parameter at the i -th test point. Note that our earlier study on degradation diagnostics relevant to this work used “RMSE%” in place of “RMSPE” [30].

6.3 Overall Performance Comparison

The results, illustrating the average test errors for individual health parameters, can be found in Fig. 8. As depicted in Fig. 8, all the PIML methods exhibited improved error rates compared to their data-driven counterparts. It is noteworthy that the degradation trends of the negative and positive active mass parameters (m_p and m_n) were generally more complex and challenging to predict compared to the capacity and lithium inventory. The degradation of electrode active materials involves intricate physical and chemical processes, such as solid-state diffusion, phase transformations, and mechanical stresses. These processes interact in a highly nonlinear and complex manner, making it challenging to model and predict their combined effects accurately. Despite these inherent difficulties, the PIML methods demonstrated a high level of accuracy in predicting these parameters. This highlights the effectiveness of the PIML approaches in capturing the underlying degradation patterns and enhancing the prediction capabilities for the more intricate parameters.

Specifically, the average test error of the PINN demonstrated a remarkable improvement compared to its baseline neural network model with the same network structure. The baseline neural network displayed a broader range of errors, implying a higher degree of variability in its predictions’ behavior across different runs. It’s important to note that the increased width of the error bars reflects greater variability in performance across different training runs, rather than solely indicating elevated predictive uncertainty within the trained models. In contrast, the PINN consistently

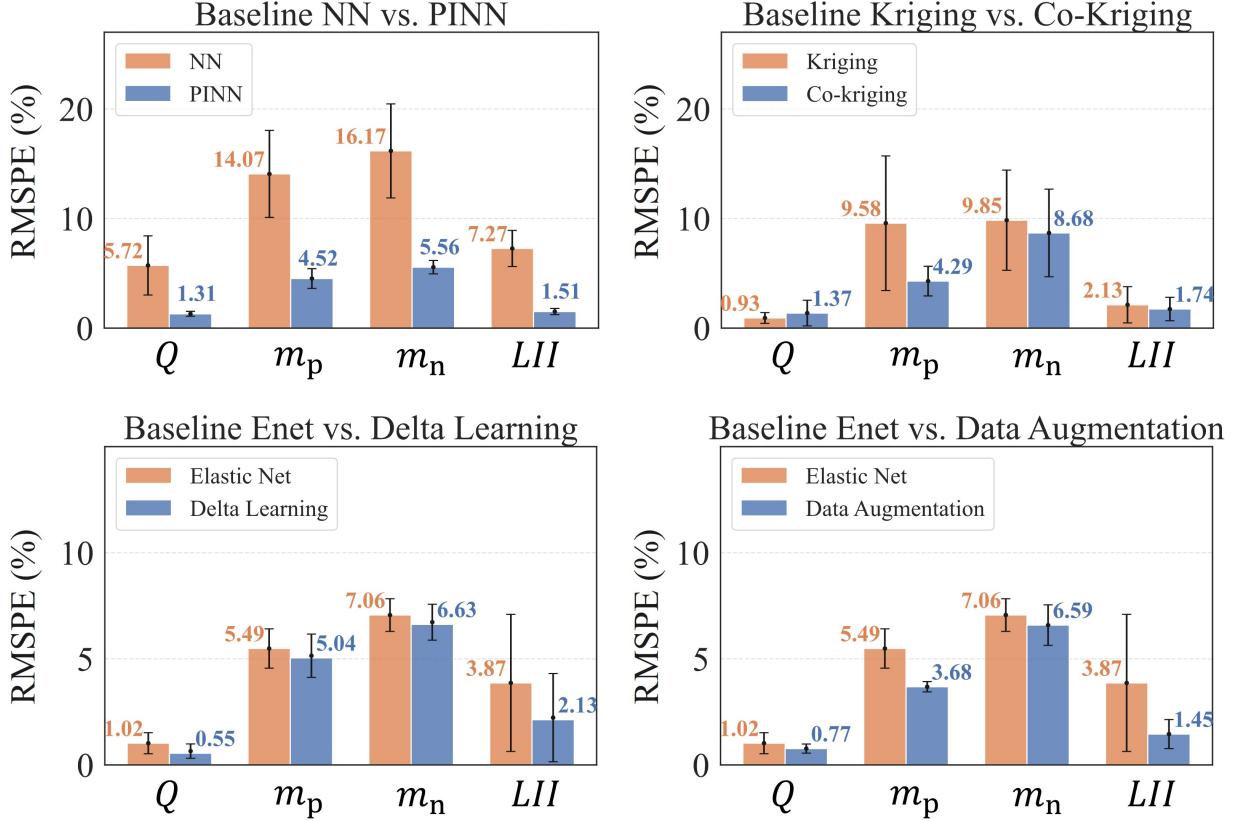


Figure 8: Overall performance comparisons between physics-informed ML models and purely data-driven baselines.

delivered improved performance in each fold with a smaller range of errors across multiple runs. The superior performance of the PINN compared to the baseline neural network can be attributed to the PINN’s enhanced extrapolation capability, as showcased in Fig. 10, stemming from two key factors. First, the PINN leveraged a concept known as observational bias, as discussed in the research by Pateras et al. [67]. This means that during the learning (model training) process, a deliberate physical bias was introduced. This bias can be implemented through the use of physics-informed data structures, in the context of either simulation or observation. This was achieved by incorporating simulated late-life stage degradation data alongside early-life experimental data during training. This approach minimized a sampling error, providing a more comprehensive representation of the underlying data distribution. In contrast, the baseline neural network relied solely on early-life experimental data. Second, the PINN adopted an inductive bias [67] approach by integrating the half-cell model within its loss function (Fig. 6b). This integration allowed the PINN to predict the half-cell model parameters rather than directly predicting the health parameters. This intermediate prediction step ensured the alignment of the final predicted health parameters with the constraints imposed by the half-cell model. These constraints were incorporated by passing the network-predicted parameters to the half-cell model and introducing two physics-informed loss terms to the loss function. These terms were minimized through the training process. In contrast, the baseline neural network predicted health parameters directly from early-life experimental data without imposing the constraints informed by the physics-based half-cell model, as shown in Fig. 6a. To ensure a fair comparison, we performed ten cross-validation runs for each model and computed the mean and standard deviation of the RMSPE across these runs. Additionally, while training these models, we regularly assessed their performance using a validation dataset, which is a distinct subset of the training dataset, including data that the model hasn’t encountered during its training

phase. An early stopping mechanism was also integrated into the training process. Particularly, if the validation loss fails to exhibit improvement over a specified number of epochs (50 in this case), we halt the training prematurely. This strategy aims to prevent the model from memorizing noise inherent in the training data. By doing so, we ensure that the model version used for evaluation possesses stronger generalization capabilities.

Regarding co-kriging, we observed significant improvements in the error rates for the positive active mass (m_p) and capacity (Q) parameters. Additionally, the error rates for the negative active mass and lithium inventory parameters were also lower compared to the purely data-driven GPR (kriging) baseline model. It is important to note that the GPR baseline model was trained exclusively on the early-life experimental data and did not incorporate the simulation data from the half-cell model.

We benchmarked data augmentation and delta learning techniques against a simple elastic net model (baseline), trained using only the early-life experimental data. Notably, data augmentation exhibited superior performance to delta learning when trained on data encompassing the entire simulated parameter range, which includes both medium and high-degradation regions. Specifically, the data augmentation technique showed substantial improvement, particularly for the positive active mass parameter (m_p), compared to the baseline elastic net model. On the other hand, the error rates for the other three parameters were relatively close for both delta learning and data augmentation.

In the evaluation of baseline models, the elastic net outperformed both the regular neural network and GPR (kriging) model in terms of overall accuracy. The baseline neural network exhibited the lowest overall accuracy, which was somewhat expected given the limited amount of training data available for network training. Notably, the baseline kriging model demonstrated excellent predictive capabilities, particularly in estimating capacity (Q) and lithium inventory (LII). However, when it came to predicting the active mass parameters (m_p and m_n), the elastic net emerged as the most accurate baseline model.

6.4 Performance Under Different Training Sample Sizes

Here, we examined the effect of including a different number of early-life experimental data points (n_{exp}) in the training process on the performance and error rates of PIML models. To carry out this analysis, we kept the test set fixed, following the same methodology as before. In each fold, the test set consisted of the cell from each group corresponding to that fold. However, we varied the number of experimental characterization points in the training set; specifically, we explored training scenarios involving 5, 10, and 15 data points per training cell. This range of scenarios resulted in a total training point count varying from 90 to 270 across different iterations. To provide context, the number of data points per cell corresponds to different durations of cycle aging data. With 5 data points per training cell, we consider 3 months of cycle aging data. With 10 data points per training cell, we extend the duration to 8 months, and with 15 data points per training cell, we utilize 12 months of cycling data. These variations allow us to investigate the impact of including different amounts of early-life data on the model's performance. The comparative results of this analysis are summarized in Fig. 9a for PIML methods considered. By examining the relationship between training sample size and model performance, we can determine the optimal amount of early-life data necessary for achieving accurate predictions and minimizing errors in PIML models.

As depicted in Fig. 9a, the performance of the PINN model shows a relatively consistent performance with slight improvements as more early-life experimental data points are included in the training process. This could be attributed to the utilization of late-stage simulation data in the training process, as well as the inclusion of constraints on the predicted half-cell model parameters within the loss function. In contrast, co-kriging exhibits a noticeable improvement in performance when transitioning from $n_{exp} = 90$ to $n_{exp} = 180$ training points. However, the performance gains become less prominent when further increasing the number of points from $n_{exp} = 180$ to $n_{exp} = 270$. This suggests that the additional training points beyond $n_{exp} = 180$ benefit less on enhancing the performance of the co-kriging model. We also observe similar trends for delta learning and data augmentation techniques, where utilizing $n_{exp} = 180$ characterization points from early-life

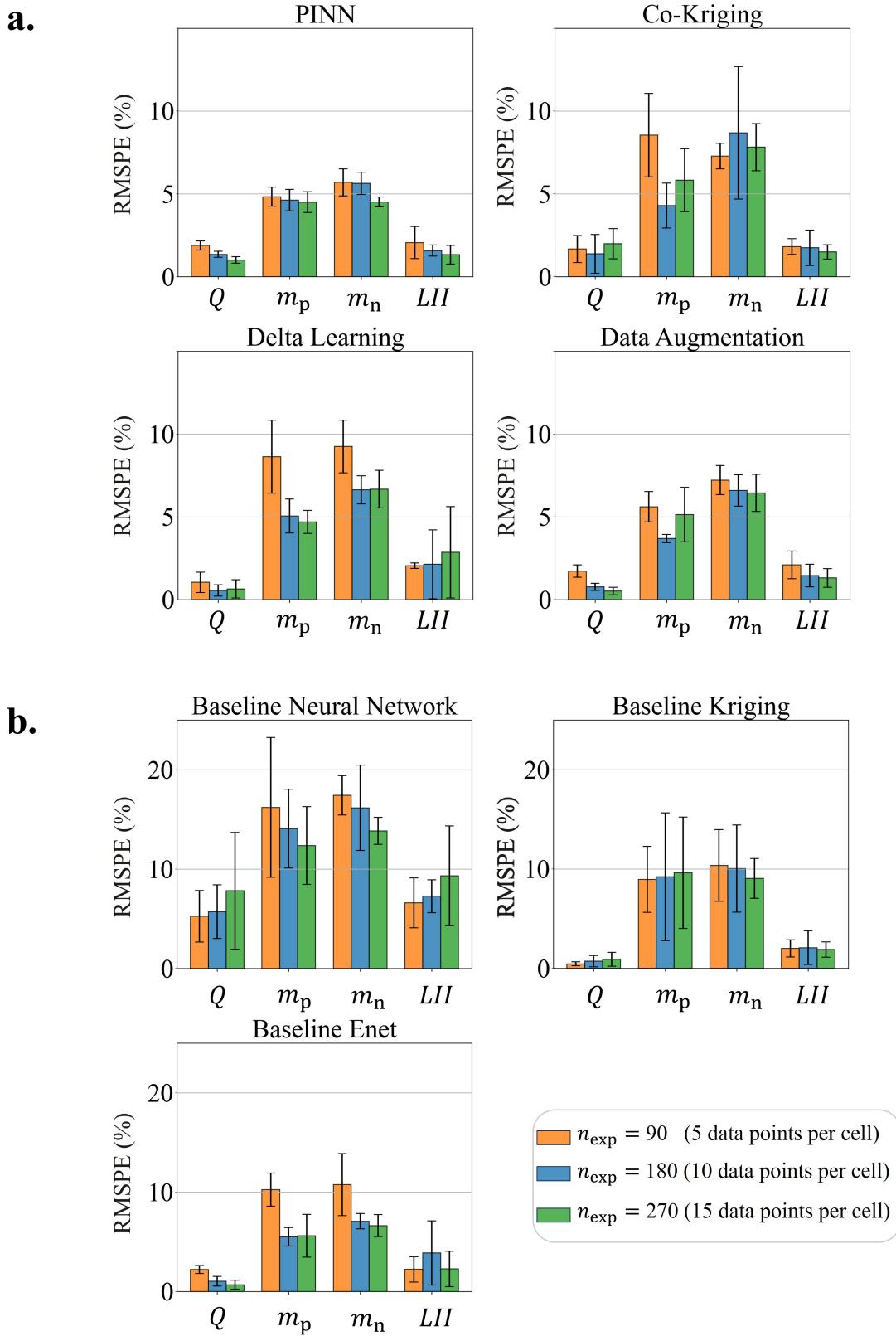


Figure 9: Impact of training sample size on the performance of **a.** PIML models **b.** baseline models (n_{exp} represents the total number of experimental data points).

experiments leads to decreased error for health parameters compared to using only 5 characterization points from each cell ($n_{\text{exp}} = 90$). Notably, the average results obtained with $n_{\text{exp}} = 180$ training points are comparable to those achieved when considering a greater number of training points. These findings align with the patterns observed in the PINN and co-kriging models, reinforcing the notion that an initial increase in training sample size can yield substantial improvements in model performance. This suggests that the additional information provided by $n_{\text{exp}} = 180$ early-life experimental data points (8 months of cycle aging data) is sufficient to capture key degradation patterns and reduce the prediction error for health parameters. However, adding more high-fidelity experimental data beyond a certain point does not significantly impact the model's performance. This can be attributed to the fact that the models have already been informed with the physics of late-life stages through the half-cell model. As a result, the incorporation of additional training points may not provide substantial new insights or improve the models' predictions.

The baseline model results are shown in Fig. 9b. When the number of experimental training points increased from 5 to 15 per cell, we observed a consistent trend of accuracy improvement. The neural network and elastic net baseline models both show significant improvement with this training size increase from $n_{\text{exp}} = 90$ to $n_{\text{exp}} = 180$ data points. However, further increasing to a total of $n_{\text{exp}} = 270$ experimental training points did not significantly reduce the error rates. In contrast, the baseline kriging model maintained consistent accuracy, regardless of the number of experimental training data points.

6.5 Extrapolation Performance

In this section, we delve into the extrapolation performance of the proposed PIML models and highlight the enhanced accuracy achieved by incorporating a physics-based half-cell model during the training process, specifically for the late-life stage. We conduct an analysis focusing on two novel approaches, namely PINN and co-kriging, for degradation diagnostics. To accomplish this, we compare the extrapolation performance of these methods to the purely data-driven counterpart models for a given test cell, considering two different conditions.

As mentioned in Sec. 5.1, we reinforce the underlying mapping between the $dQ/dV(V)$ curves and true health parameters into the PINN by adding the input data from the high-degradation region. Additionally, we constrain the predicted half-cell model parameters using a half-cell surrogate model. Furthermore, in the loss function, we perform peak difference minimization between the resulting $dQ/dV(V)$ curves obtained from the predicted half-cell model parameters and the experimental ones from the training set. To showcase the effect of incorporating these enhancements into the baseline model, we compare the extrapolation error of a specific test cell (G2C2) with a C/24 discharge rate and a temperature of 55°C. We analyze the extrapolation error in the degradation parameters space for both the physics-informed and regular neural networks. The results are visualized in Fig. 10, where the bubble size represents the overall error, given by:

$$C_1 + C_2 \times RMSE([\hat{m}_p, \hat{m}_n, \hat{LII}]^T, [m_p, m_n, LII]^T) \quad (15)$$

Here, C_1 and C_2 are constants, and the root mean squared error (RMSE), denoted as $RMSE$, is computed based on the three degradation parameters: m_p , m_n , and LII . Based on the analysis, it is evident that as the degree of cell degradation increases, the extrapolation error of the regular neural network increases significantly compared to the physics-informed approach, particularly in the medium and late-life stages. This observation demonstrates the effectiveness of incorporating late-life stage degradation trends by augmenting the neural network with high degradation simulation data and imposing constraints to ensure consistency with the physics-based half-cell model during the training process. This integration enables the physics-informed network to capture and generalize the degradation patterns more accurately, leading to improved predictions even in the presence of severe degradation.

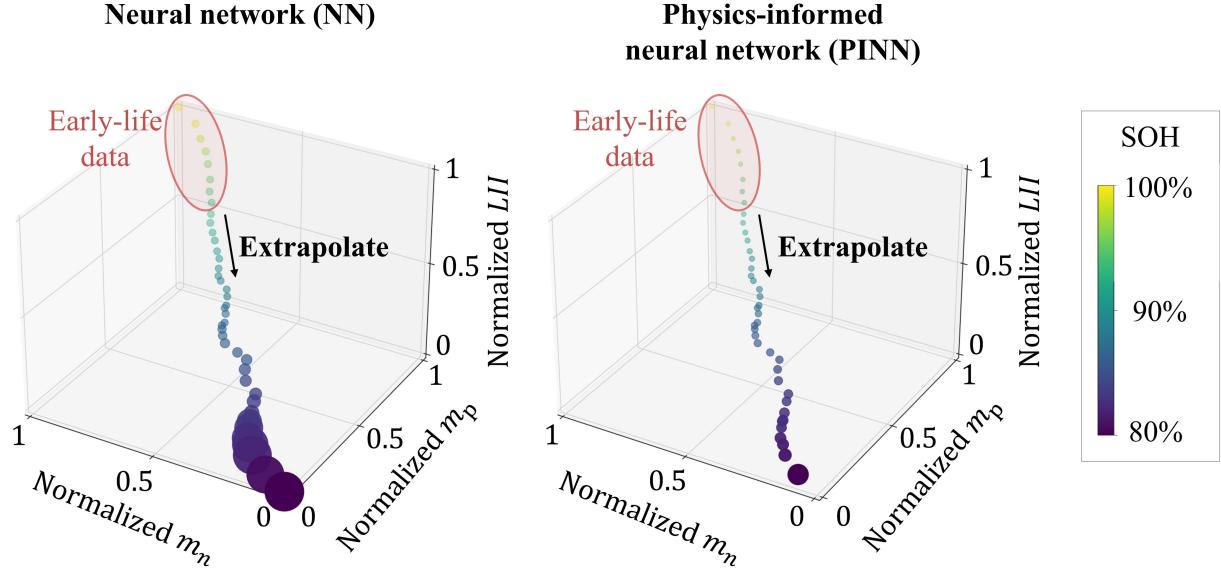


Figure 10: Extrapolation error comparison for purely data-driven and physics-informed neural networks. The bubble size represents the RMSE computed based on the three degradation parameters.

To gain a better understanding of the extrapolation error, we visualize the predicted trajectory of each health parameter against its true trajectory obtained from the half-cell fitting process. The detailed results of trajectory predictions by the considered PIML methods for all cells can be found in the Appendix. Here, we focus on presenting the results for G3C3, a test cell with a higher discharge rate of C/3 and a temperature of 37°C. Fig. 11 illustrates the predicted trajectory of the four health parameters using both the PINN and co-kriging models compared to their baseline data-driven models. The predicted trajectory of the health parameters by these models exhibits significant improvement in tracking the true fitted values, particularly in the case of the positive and negative active mass parameters (m_p and m_n), when compared to the baseline models. While the trajectory prediction for the parameters Q and LII also shows improvement, it is relatively less pronounced compared to the other two parameters. This indicates that incorporating trends from the high-degradation region and constraining machine learning models with our physics-based half-cell model is likely to improve the extrapolation of diagnosed degradation trends in the late-life stage, especially when dealing with more intricate patterns, such as those observed in m_p and m_n .

6.6 Sensitivity Analysis

In this section, we perform an analysis on the sensitivity of two novel approaches, PINN and co-kriging, proposed in this study for degradation diagnostics. We aim to examine the impact of modifying their structure or parameters on the diagnostics error rate specifically during the late-life stage. For the PINN model, we focus on adjusting the weights assigned to each of the three loss terms incorporated in the physics-informed loss function. As for the co-kriging model, we assess the consequences of altering the kernel functions utilized for estimating and correcting within each kriging model.

6.6.1 PINN

Effect of Loss Terms

As described in Sec. 5.1, our implementation of the PINN incorporates three distinct loss terms. The first term, denoted as (\mathcal{L}_1) , quantifies the disparity between the network's predictions

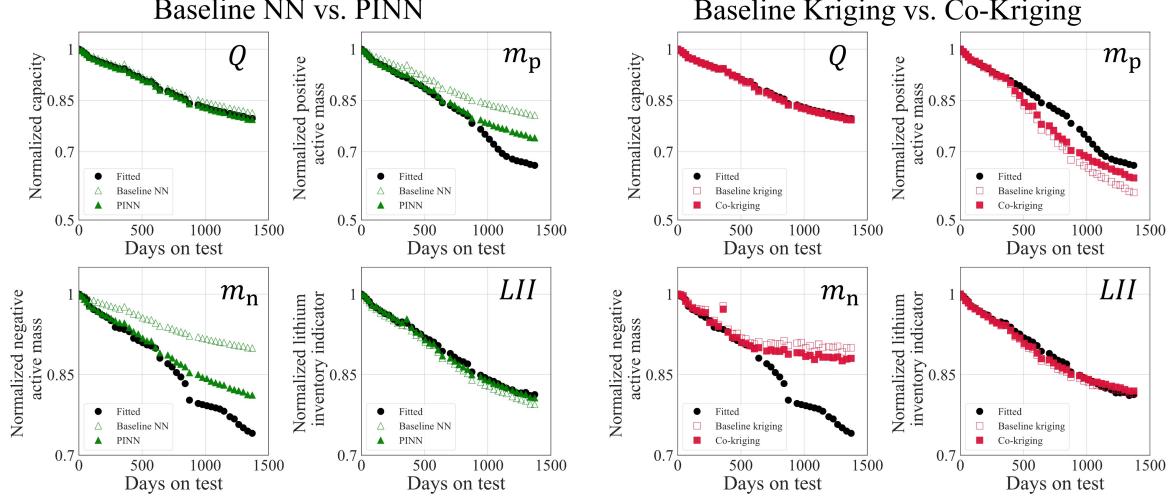


Figure 11: Trajectory prediction comparison between purely data-driven and physics-informed machine learning models.

of the half-cell model parameters and the actual fitted data. Additionally, we utilize two physics-informed loss terms. The second term (\mathcal{L}_2) arises from inputting the network’s predictions into the half-cell model and ensuring that the obtained health parameters conform to the model’s constraints. The third term (\mathcal{L}_3) considers the constraints on the peak positions of $dQ/dV(V)$ curves resulting from the half-cell model. These loss terms collectively contribute to the comprehensive training of our PINN model.

To evaluate the sensitivity of the PINN model’s performance to each of these terms, we assess the ratio of their weights (w_i) in computing the total loss relative to the summation of all weights (r_i). This can be expressed as:

$$r_i = \frac{w_i}{\sum_{k=1}^3 w_k}, \quad (i = 1, 2, 3) \quad (16)$$

This ratio provides insight into the contribution and importance of each loss term in the overall training process of the PINN model. Initially, to ensure uniform scaling and equal impact of each loss term on the total loss, we assume a ratio of $r_i = 1/3 \approx 0.33$ for each of the weights. In Fig. 12, we plot the average prediction error rates for different values of this ratio corresponding to each loss term.

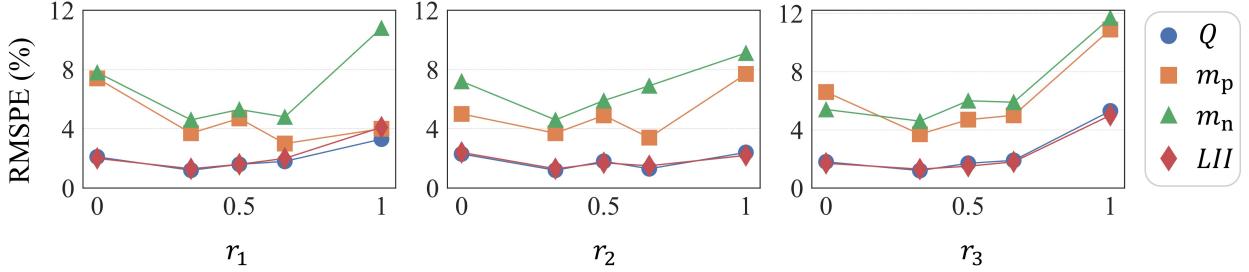


Figure 12: Sensitivity of prediction error rate of PINN model to each loss term.

Five different ratios ranging from 0 to 1 are examined, as shown in Fig. 12. For each value of each ratio (r_i) considered, we modify one weight (w_i) while keeping the other two weights constant,

contributing equally to the total loss. In cases where each loss term is considered alone ($r_i = 1$ for a specific loss term), the error rates tend to be higher. Notably, the first point in each subplot corresponds to the scenario where the specified loss term is not considered at all. The second point represents an equal weight assignment ($r_i = 1/3 \approx 0.33$) to all loss terms in the loss function. The third and fourth points correspond to $r_i = 1/2 = 0.5$ and $r_i = 2/3 \approx 0.66$, respectively. Remarkably, combining the three loss terms (represented by the three middle points in the subplots) results in lower overall error rates. For diagnosing Q , m_n , and LII , assigning equal weights to the loss terms yields the minimum error rates. However, in the case of m_p diagnosis, increasing the ratio of the first and second loss terms between 0.5 and 1 can lead to more accurate results. These findings highlight the sensitivity of the PINN model's performance to the selection and weighting of different loss terms, emphasizing the importance of careful consideration when designing the loss function for specific degradation diagnostics tasks.

Effect of Loss Function vs. Augmentation

We conduct a sensitivity analysis to understand the specific impacts of two factors in enhancing the performance of the PINN model; the customized loss function and the augmentation of the training dataset. As mentioned in Sec. 5.1, the augmentation of the training dataset for the PINN model aimed to maintain consistency with other PINN approaches, ensuring a fair comparison. Additionally, this augmentation was intended to inform the PINN model about degradation trends in the late-life stage of the cells. While the simulation training data is generated from the same physical model, the primary enhancement in prediction accuracy stems from incorporating a customized loss function. To better demonstrate the contribution of each physics-informed technique, both the baseline neural network model and the PINN model were trained on only early-life experimental data, as well as on the augmented dataset that included the highest 20% degradation from the simulation data, consistent with the settings for PINN model training. Ten repeated training runs were conducted under each training setup, and the RMSPE results averaged over these two runs are presented in Fig. 13.

Two observations can be made from this figure. First, training the baseline model with the augmented dataset showed only marginal accuracy improvement in predicting health parameters, suggesting that the augmentation alone did not significantly enhance accuracy (Baseline NN vs. Baseline NN + Augmentation). Second, utilizing only early-life experimental data to train a PINN model (i.e., excluding the physics-based simulation data from the training set) resulted in only marginal increases in error rates (PINN vs. PINN + Augmentation).

These comparison results suggest that augmenting a training dataset with simulation data from the half-cell model and incorporating this model in the loss function do not yield the same effect in improving the ability of a neural network model to extrapolate. We can say, at least empirically from Fig. 13, that improved extrapolation capability can be mainly attributed to customizing the loss function with known physics.

6.6.2 Co-Kriging

As outlined in Sec. 5.3.1, the co-kriging approach allows for the consideration of various types of kernel functions in each kriging model, depending on prior knowledge and understanding of the problem. The kernel function plays a crucial role in determining the correlation between data points and influences the accuracy of interpolation and extrapolation at unobserved locations. Different kernel functions possess distinct properties and assumptions. Here, we examine the impact of selecting different kernel functions on the prediction accuracy of the co-kriging approach in the context of degradation diagnostics. By evaluating the use of various kernels, we gain insights into the performance and effectiveness of the co-kriging method. The results of employing different kernel functions are presented and summarized in Fig. 14. This analysis allows us to assess the influence of kernel selection on the overall predictive capabilities of the co-kriging approach, aiding

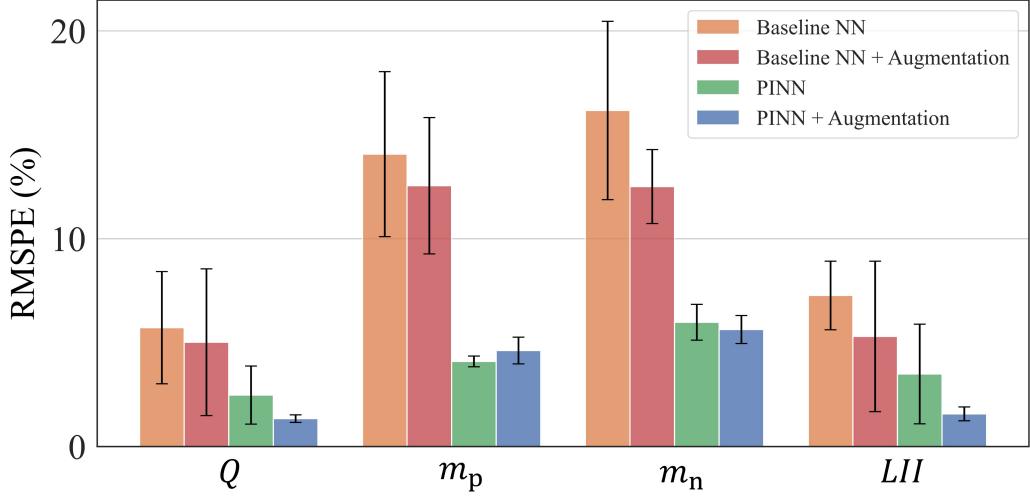


Figure 13: Sensitivity of the PINN model to augmentation and the physics-informed loss function and sensitivity of the baseline NN model to augmentation.

in the determination of the most appropriate kernel function for our specific degradation diagnostics problem. A detailed description of the tested kernel functions is provided in the 8.

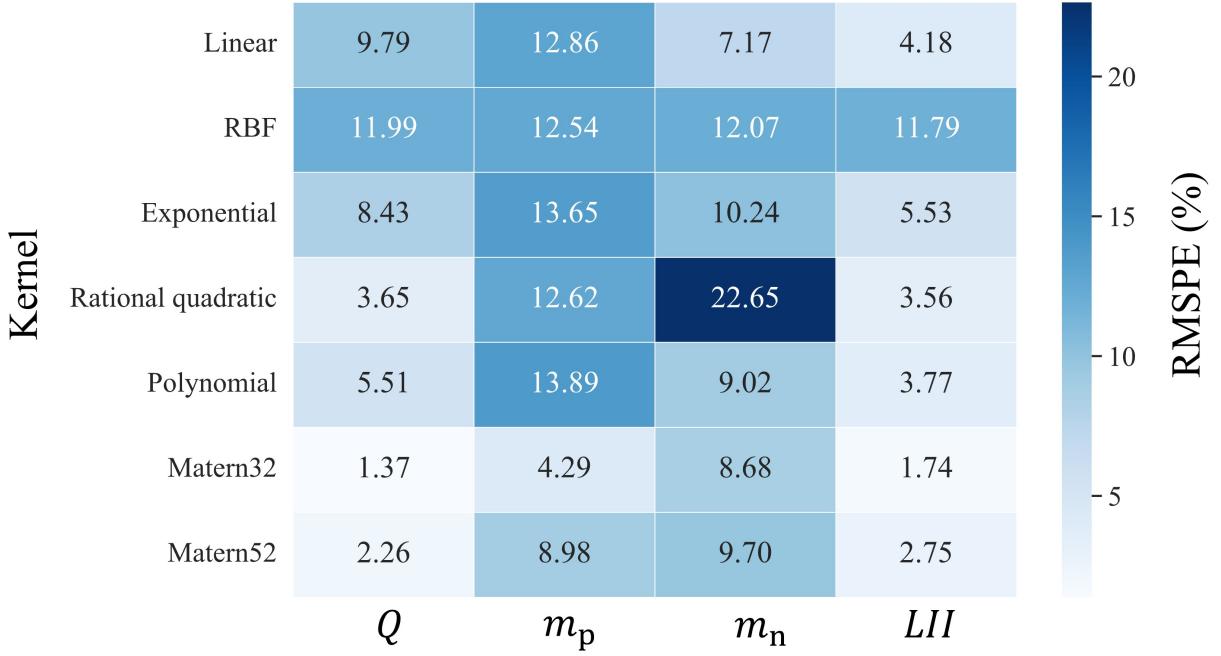


Figure 14: Sensitivity of prediction error rate of co-kriging model to different kernel functions.

As shown in Fig. 14, the co-kriging approach using the Matern32 kernel function for each kriging model exhibits the lowest average error rates across all predicted health parameters. The Matern32 kernel is a well-suited option for certain data types due to its moderate smoothness and flexibility. It effectively models functions with a moderate level of smoothness while accommodating variability and irregular patterns commonly found in real-world data.

Belonging to the Matern family of kernels, the Matern32 kernel offers a range of smoothness options. With a smoothness parameter of $3/2$, it strikes a balance between the extremely smooth RBF kernel (with a smoothness parameter of ∞) and the more oscillatory Matern52 kernel (with a smoothness parameter of $5/2$). This moderate level of smoothness allows the model to capture variations and fluctuations in the data without overly constraining it. Consequently, the Matern32 kernel provides the necessary flexibility to accommodate different patterns and irregularities present in the data, making it a suitable choice for achieving accurate predictions. However, it is important to note that the polynomial kernel outperforms the Matern32 kernel in predicting m_n . The Matern32 kernel's moderate smoothness assumption may not be ideal for data exhibiting stronger non-linearities, as observed in the degradation patterns of the negative electrode. On the other hand, the polynomial kernel offers the flexibility to represent more complex functions with higher-order polynomials and does not impose any smoothness constraint. This characteristic makes it well-suited for capturing more intricate variations and non-linear relationships in the data, making it a better choice for accurately modeling the m_n parameter.

Overall, choosing the right kernel for kriging and co-kriging depends on understanding the data's smoothness, trends, and noise levels. For smoother data, the RBF or Matern32 kernels are suitable, while non-linear patterns are better captured by the Matern52 or polynomial kernels. Evaluating different kernels, considering data complexity, incorporating domain knowledge, and performing cross-validation are vital for achieving accurate predictions while maintaining model interpretability and alignment with the underlying phenomenon.

7 Discussion

7.1 Qualitative Analysis of PIML Techniques

Selecting the appropriate PIML method for diagnosing battery degradation and general predictive purposes depends on several critical factors. These include the availability of data, the complexity of the specified process, and the desired level of accuracy and interpretability of the obtained results [43, 68, 69]. Each PIML approach discussed in this study has its strengths and limitations, arising from their distinct approaches to integrating physics-based knowledge and data-driven techniques. A qualitative comparison of PIML methods is presented in Table 5. In the subsequent paragraphs, we delve into the specific factors that have been evaluated in Table 5 for each PIML method.

Table 5: A qualitative comparison of state-of-the-art PIML approaches covered in this study

Quantity of Interest	PINN	Co-Kriging	Delta Learning with Enet	Data Augmentation
Model Flexibility	High	Medium	Medium	Medium
Data Requirements	Low	Medium-low	Medium-low	Medium
Ease of Implementation	Low	Medium	High	High
Computational Cost	Low	Low	Low	Low
Generalization Capability	High	Medium	Medium	Medium
Interpretability	High	Medium-low	Medium-low	Low
Prediction Accuracy	High	High	High	High
Uncertainty Quantification Capability	Medium-low	High	Low	Low
Scalability to Large Applications	High	High	High	High
Applicability to Different Problem Types	Medium-low	Medium-low	Medium	High

7.1.1 Model Flexibility

Model flexibility is a key consideration in PIML methods, and it refers to the capability of each method to adapt to various types of physical models. PINNs, in this context, demonstrate a high degree of flexibility due to their ability to directly incorporate complex physics-based mappings and equations, including PDEs, into their loss function during the training process. A multitude of extensions for PINNs have been explored in the literature, encompassing areas such as energy conservation laws [70], finite element method [71] and stochastic and fractional PDEs for multi-scale

problems [72, 73]. This distinctive capability empowers PINNs to adeptly capture intricate physical relationships, thus reducing the reliance on solely simulated physics-based data. Conversely, other PIML techniques considered here, may require synthetic simulation data as an integral part of their training process. This dependence on simulated data can potentially impose limitations, particularly when confronted with highly complex equations characteristic of inverse and ill-posed problems [43].

7.1.2 Data Requirements

The data requirements for each PIML method are vital considerations, including factors such as the quantity of data needed, data quality, and the availability of high-fidelity data for calibration or validation. These requirements vary among the methods, necessitating a comprehensive understanding of their effective implementation in practical applications. For PINNs, the amount of data required depends on the depth of the neural network used. Shallow neural networks, as employed in this study, may not demand an excessively large dataset for training. However, a notable strength of PINNs is their ability to achieve high accuracies even with limited amounts of data [44]. This characteristic makes PINNs particularly valuable in situations where data availability is limited, providing a significant advantage over other methods. In contrast, delta learning, as a hybrid multi-fidelity model, relies heavily on high-fidelity data, which might be scarce compared to the more easily accessible low-fidelity data points [45]. To address this challenge, the delta learning approach introduces a corrector model to refine estimations based on low-fidelity data, making it more feasible to achieve accurate results even with limited high-fidelity data. Careful selection and filtering of the high-fidelity data are crucial in ensuring its effectiveness in enhancing the overall model accuracy [65]. Data augmentation, on the other hand, relies on low-fidelity synthetic data generated from a physics-based model. The success of data augmentation largely hinges on the relevance and diversity of the synthetic data points used [68, 74]. Therefore, meticulous consideration in generating synthetic data is vital to achieve meaningful and reliable results.

7.1.3 Ease of Implementation

Ease of implementation is another factor when selecting a PIML method for a specific problem [68, 69]. Some PIML approaches offer straightforward algorithms or readily available libraries, which significantly simplifies their implementation process. In this context, both delta learning and data augmentation can be easily applied using the readily available machine learning packages in Python and MATLAB [45]. However, PINNs present a relatively more challenging implementation compared to delta learning and data augmentation. Customizing the loss function in a way that incorporates the physics-based constraints on the network outputs and ensuring consistency for gradient computation requires careful attention. Furthermore, co-kriging demands more consideration towards hyperparameter tuning and kernel selection [65], compared to delta learning (when using simpler models like the elastic net), and data augmentation.

7.1.4 Computational Cost

The main challenges of using PIML for online prediction are the cost and time required for training and inference processes [69]. Computational cost is inherently intertwined with complexity of the physics-based model and the machine learning model [45]. In the context of the study, where online estimation applications are targeted, all the considered methods have been specifically designed and implemented with computational efficiency in mind.

7.1.5 Scalability to Large Applications

Scalability assesses the ability of each method to scale with increasing data sizes and problem complexities. The ability to scale effectively is particularly important for real-world applications, where datasets can be vast, and the problems may involve intricate physical processes. The PIML methods demonstrated the capability to effectively handle larger dataset sizes and more complex

problem scenarios within the context of this study, as well as for other problem domains [75]. However, when dealing with larger-sized problems, careful considerations and customizations are required to ensure optimal performance. Scaling to larger datasets and more complex problems may involve adjustments in computational resources, algorithmic optimizations, or fine-tuning of hyperparameters. It should be noted that among the considered methods here, data augmentation emerges as a relatively more straightforward approach to scale from small-scale problems to larger ones [76]. The inherent advantage of data augmentation lies in its ability to utilize expanded dataset, allowing for seamless extensions to larger datasets without significant constraints on data availability.

7.1.6 Generalization Capability

Generalization capability is another crucial factor that investigates how well each method can generalize to unseen data or different operating conditions while considering issues such as overfitting. In this regard, PINNs exhibit superior performance compared to other PIML methods. Unlike other methods that rely solely on generated physics-based data for training, PINNs can directly incorporate physics-based constraints into the loss function, enhancing their ability to generalize to unseen data and different operating conditions [69, 77]. In contrast, delta learning and data augmentation depend on specific data trends during training, leading to relatively lower generalization capabilities.

7.1.7 Interpretability

Interpretability examines how well each method allows for interpretability and physical insights. Some methods provide explicit representations of the underlying physical processes, while others may have more black-box nature [78]. PINNs excel in this regard, as the physics-based constraints are explicitly incorporated into the loss function, enabling clear mathematical descriptions of their influence on the model outputs [69]. On the other hand, data augmentation may lack the same level of interpretability since its reliance on synthetic data may not directly reveal the underlying relationships between inputs and outputs.

7.1.8 Prediction Accuracy

Traditional machine learning methods often lack the ability to extrapolate, which can be solved by the PIML methods [79]. Prediction accuracy evaluates the predictive performance of each PIML method on both training and test datasets, which, as discussed extensively in this study, all the PIML methods showed promising results.

7.1.9 Applicability to Different Problem Types

Applicability to different problem types assesses whether each method is well-suited to specific problem types or domains. In this regard, PINNs can be more challenging to apply in some cases, as the explicit definition of physics-based mapping functions between inputs and outputs may not be readily available [43]. However, PINNs become a viable option when generating training data from physics-based models is more difficult than incorporating equations into the loss function and training process. Similarly, co-kriging faces challenges when defining correlations between different fidelity levels of data is not straightforward [80]. Conversely, delta learning, in general, exhibits adaptability to different problem structures by selecting appropriate regression models, making it applicable to various scenarios. On the other hand, data augmentation stands out as the most versatile approach across various fields and applications. Its ability to use synthetic simulation data from physics-based models for training purposes makes it a straightforward and effective choice [68].

7.1.10 Uncertainty Quantification Capability

Lastly, uncertainty quantification examines whether each method provides a means to quantify uncertainties in predictions. In the case of PINNs, as they are not inherently probabilistic, methods like ensembles [81], quantile regression [82], and Bayesian neural networks [83, 84] should be considered to quantify uncertainty. Each of these techniques offers a unique perspective on capturing and expressing uncertainties inherent in predictive modeling, particularly in scenarios involving short-term data. Ensembles involve training multiple models independently and aggregating their predictions. Quantile regression is a technique that enables the estimation of different quantiles of the predictive distribution. In the context of PINNs, quantile regression allows for modeling a range of possible outcomes and their associated uncertainties. Bayesian neural networks present another avenue for incorporating uncertainty into PINNs. Unlike conventional neural networks, Bayesian neural networks treat model parameters as probability distributions. This probabilistic approach allows for the modeling of parameter uncertainty, offering a more nuanced representation of the model's predictions. On the other hand, co-kriging is a powerful approach to uncertainty quantification. It naturally accounts for uncertainty by modeling correlations and providing prediction variances [85]. Co-kriging explicitly models the covariance structure between different variables, capturing the correlation between their variations. The incorporation of a well-defined covariance structure enables co-kriging to effectively propagate uncertainties across variables [84]. For delta learning and data augmentation with elastic net model, while they may not inherently provide uncertainty estimates, it is possible to introduce uncertainty quantification methods such as Bayesian [86] and bootstrap methods [87] to enhance the prediction reliability and quantify uncertainties.

In summary, making informed decisions based on these factors and considerations is required to effectively apply PIML methods in different scenarios. In addition, ongoing research and advancements in various PIML methods continue to expand their applicability and improve their performance, making it important to stay updated with the latest developments in the field.

7.2 Method Selection Guidelines

PIML techniques are primarily applied to address and solve physics or engineering problems in situations where acquiring data and formulating tasks can be particularly challenging for researchers who lack domain knowledge and experience. Moreover, current research heavily depends on domain-specific datasets, complicating the identification of best practices for specific problems. Here, we will succinctly summarize the conclusions drawn from our comprehensive study and the reviewed literature, offering insights into when to choose each PIML technique.

7.2.1 PINN

PINNs excel in situations where the underlying physics of the problem can be represented by a set of differentiable mapping functions. Although PINNs can achieve high accuracy with a limited amount of high-fidelity data, it is important to note that implementing this method requires more intricacy compared to other PIML approaches in terms of both mathematics and programming resources [69]. Creating a loss function that incorporates physics-informed terms with a closely matched convergence rate, while also ensuring that the neural network's computational graph is interconnected to facilitate gradient calculations for outputs with respect to inputs, can be a challenging task [88]. Therefore, careful consideration and expertise are needed to effectively employ PINNs for a given problem. However, once implemented, PINNs tend to generalize well to new, unseen data and are less prone to overfitting. In the specific case of degradation diagnostics for lithium-ion batteries, the prediction results of the PINN model for health parameters at a late-life stage show a significant improvement compared to the purely data-driven counterpart. When compared to other PIML approaches presented, the PINN methodology demonstrated superior accuracy and consistency in predicting more intricate degradation trends, specifically in the loss of positive and negative active mass (m_p and m_n).

7.2.2 Delta Learning

Delta learning offers ease of implementation and flexibility by allowing the selection of any regression machine learning model as the estimator and corrector models. This adaptability makes it an attractive option, especially when the physical model is not explicitly known, but limited high-fidelity data is available through experiments. It's crucial to select and evaluate machine learning models carefully to ensure accuracy and reliability in the delta learning approach. To achieve optimal results, diverse sampling of high-fidelity points across the entire design space is important, particularly for accurate extrapolation.

Delta Learning with Elastic Net Regression

For cases where predicted trends are not overly intricate, opting for simpler models like elastic net can be advantageous. Elastic net effectively captures underlying patterns and provides accurate predictions. This is particularly valuable when prioritizing ease of implementation and time efficiency. In the specific study's case of predicting capacity (Q) and lithium inventory indicator (LII), the elastic net model showed good performance. The choice of elastic net is also suitable due to the small size of the training data, which reduces the risk of overfitting associated with more complex models [30].

Delta Learning with Kriging (Co-Kriging)

Incorporating kriging (GPR) models introduces a powerful approach known as co-kriging, which is effective in modeling multi-fidelity data problems. Co-kriging leverages correlated fidelity data to fill in missing values and estimate uncertainties in predicted values. It facilitates the quantification of result reliability. The co-kriging method is particularly useful for predicting capacity (Q) and lithium inventory indicator (LII), as observed in this study. The capacity prediction using co-kriging exhibited a near-zero error rate, showcasing its high accuracy. Additionally, co-kriging showed a significant improvement in predicting the positive active mass parameter (m_p).

7.2.3 Data Augmentation

In situations where the underlying physical model or correlations between different fidelity sources of data are not well understood or difficult to formulate explicitly, data augmentation offers a data-driven framework to improve model performance. By generating additional training samples, data augmentation caters to various cost considerations based on available resources. This approach is commonly used alongside deep learning models, enabling them to effectively capture complex non-linear relationships, as extensively documented in the literature. However, even with a simple elastic net model, data augmentation has demonstrated effectiveness, as evidenced in the case of this study. The method's ease of implementation and computational efficiency makes it particularly advantageous for rapid prototyping and large-scale applications.

8 Conclusion

In this study, we introduced two PIML approaches: PINN and co-kriging. These approaches were designed for diagnosing cell degradation without depending on late-life aging data. Through a comparative analysis, we evaluated these methods alongside delta learning and data augmentation. We used battery aging data from a long-term cycling experiment for this evaluation. The comparative results highlighted the superior performance of the PIML approaches compared to their purely data-driven baseline models. This performance improvement was especially notable in terms of extrapolation during the late-life stage of degradation. Additionally, we conducted a sensitivity analysis to gain deeper insights into implementing the two new approaches: PINN and co-kriging. We also engaged in a detailed discussion of the advantages and limitations of each method, grounded

in the context of battery degradation diagnostics and existing literature. Furthermore, we provided guidance on when to choose a specific PIML method based on the insights gathered. This study contributes valuable knowledge to the realm of battery degradation diagnostics and PIML method selection.

References

- [1] Anthony Barré, Benjamin Deguilhem, Sébastien Grolleau, Mathias Gérard, Frédéric Suard, and Delphine Riu. A review on lithium-ion battery ageing mechanisms and estimations for automotive applications. *Journal of Power Sources*, 241:680–689, 2013.
- [2] Kandler Smith, Paul Gasper, Andrew M Colclasure, Yuta Shimonishi, and Shuhei Yoshida. Lithium-ion battery life model with electrode cracking and early-life break-in processes. *Journal of The Electrochemical Society*, 168(10):100530, 2021.
- [3] Michael Schimpe, Markus Edler von Kuepach, Maik Naumann, Holger C Hesse, Kandler Smith, and Andreas Jossen. Comprehensive modeling of temperature-dependent degradation mechanisms in lithium iron phosphate batteries. *Journal of The Electrochemical Society*, 165(2):A181, 2018.
- [4] Paul Gasper, Nils Collath, Holger C Hesse, Andreas Jossen, and Kandler Smith. Machine-learning assisted identification of accurate battery lifetime models with uncertainty. *Journal of The Electrochemical Society*, 169(8):080518, 2022.
- [5] Gregory L Plett. Extended kalman filtering for battery management systems of lipb-based hev battery packs: Part 3. state and parameter estimation. *Journal of Power sources*, 134(2):277–292, 2004.
- [6] Chao Hu, Byeng D Youn, and Jaesik Chung. A multiscale framework with extended kalman filter for lithium-ion battery soc and capacity estimation. *Applied Energy*, 92:694–704, 2012.
- [7] B. Saha, K. Goebel, S. Poll, and J. Christophersen. Prognostics methods for battery health monitoring using a bayesian framework. *IEEE Trans. Instrum. Meas*, 58(2):291–296,, 2009. doi: 10.1109/TIM.2008.2005965.
- [8] C. Hu, H. Ye, G. Jain, and C. Schmidt. Remaining useful life assessment of lithium-ion batteries in implantable medical devices. *J. Power Sources*, 2018. doi: 10.1010/j.jpowsour.2017.11.056.
- [9] Tianheng Feng, Lin Yang, Xiaowei Zhao, Huidong Zhang, and Jiaxi Qiang. Online identification of lithium-ion battery parameters based on an improved equivalent-circuit model and its implementation on battery state-of-power prediction. *Journal of Power Sources*, 281:192–203, 2015.
- [10] D Andre, M Meiler, K Steiner, H Walz, T Soczka-Guth, and DU Sauer. Characterization of high-power lithium-ion batteries by electrochemical impedance spectroscopy. ii: Modelling. *Journal of Power Sources*, 196(12):5349–5356, 2011.
- [11] Xaosong Hu, Shengbo Li, and Huei Peng. A comparative study of equivalent circuit models for li-ion batteries. *Journal of Power Sources*, 198:359–367, 2012.
- [12] Scott J Moura, Nalin A Chaturvedi, and Miroslav Krstić. Adaptive partial differential equation observer for battery state-of-charge/state-of-health estimation via an electrochemical model. *Journal of Dynamic Systems, Measurement, and Control*, 136(1), 2014.
- [13] Rui Xiong, Linlin Li, Zhirun Li, Quanqing Yu, and Hao Mu. An electrochemical model based degradation state identification method of lithium-ion battery for all-climate electric vehicles application. *Applied energy*, 219:264–275, 2018.
- [14] Brian Bole, Chetan S Kulkarni, and Matthew Daigle. Adaptation of an electrochemistry-based li-ion battery model to account for deterioration observed under randomized use. In *Annual Conference of the PHM Society*, volume 6, 2014.

- [15] Maitane Berecibar, Floris Devriendt, Matthieu Dubarry, Igor Villarreal, Noshin Omar, Wouter Verbeke, and Joeri Van Mierlo. Online state of health estimation on nmc cells based on predictive analytics. *Journal of Power Sources*, 320:239–250, 2016.
- [16] C. Hu, G. Jain, C. Schmidt, C. Strief, and M. Sullivan. Online estimation of lithium-ion battery capacity using sparse bayesian learning. *J. Power Sources*, 289:105–113,, 2015. doi: 10.1016/j.jpowsour.2015.04.166.
- [17] Duo Yang, Xu Zhang, Rui Pan, Yujie Wang, and Zonghai Chen. A novel gaussian process regression model for state-of-health estimation of lithium-ion battery using charging curve. *Journal of Power Sources*, 384:387–395, 2018.
- [18] Robert R Richardson, Christoph R Birk, Michael A Osborne, and David A Howey. Gaussian process regression for in situ capacity estimation of lithium-ion batteries. *IEEE Transactions on Industrial Informatics*, 15(1):127–138, 2018.
- [19] Sheng Shen, Mohammadkazem Sadoughi, Meng Li, Zhengdao Wang, and Chao Hu. Deep convolutional neural networks with ensemble learning and transfer learning for capacity estimation of lithium-ion batteries. *Applied Energy*, 260:114296, 2020.
- [20] S. Shen, M. Sadoughi, X. Chen, M. Hong, and C. Hu. A deep learning method for online capacity estimation of lithium-ion batteries. *J. Energy Storage*, 25:100817,, 2019.
- [21] D. Roman, S. Saxena, V. Robu, M. Pecht, and D. Flynn. Machine learning pipeline for battery state-of-health estimation. *Nat. Mach. Intell.*, 3(5):447–456,, 2021. doi: 10.1038/s42256-021-00312-3.
- [22] Xiaoyu Li, Changgui Yuan, Zhenpo Wang, and Jiale Xie. A data-fusion framework for lithium battery health condition estimation based on differential thermal voltammetry. *Energy*, 239: 122206, 2022.
- [23] Yuanyuan Li, Daniel-Ioan Stroe, Yuhua Cheng, Hanmin Sheng, Xin Sui, and Remus Teodorescu. On the feature selection for battery state of health estimation based on charging-discharging profiles. *Journal of Energy Storage*, 33:102122, 2021.
- [24] Brian Ospina Agudelo, Walter Zamboni, Fabio Postiglione, and Eric Monmasson. Battery state-of-health estimation based on multiple charge and discharge features. *Energy*, 263: 125637, 2023.
- [25] Zhongwei Deng, Xiaosong Hu, Penghua Li, Xianke Lin, and Xiaolei Bian. Data-driven battery state of health estimation based on random partial charging data. *IEEE Transactions on Power Electronics*, 37(5):5021–5031, 2021.
- [26] Jiachi Yao and Te Han. Data-driven lithium-ion batteries capacity estimation based on deep transfer learning using partial segment of charging/discharging data. *Energy*, 271:127033, 2023.
- [27] Yi Li, Changfu Zou, Maitane Berecibar, Elise Nanini-Maury, Jonathan C-W Chan, Peter Van den Bossche, Joeri Van Mierlo, and Noshin Omar. Random forest regression for online capacity estimation of lithium-ion batteries. *Applied energy*, 232:197–210, 2018.
- [28] Paul Gasper, Andrew Schiek, Kandler Smith, Yuta Shimonishi, and Shuhei Yoshida. Predicting battery capacity from impedance at varying temperature and state of charge using machine learning. *Cell Reports Physical Science*, 3(12):101184, 2022.
- [29] Jialong Liu, Qiangling Duan, Mina Ma, Chunpeng Zhao, Jinhua Sun, and Qingsong Wang. Aging mechanisms and thermal stability of aged commercial 18650 lithium ion battery induced by slight overcharging cycling. *Journal of power sources*, 445:227263, 2020.
- [30] Adam Thelen, Yu Hui Lui, Sheng Shen, Simon Laflamme, Shan Hu, Hui Ye, and Chao Hu. Integrating physics-based modeling and machine learning for degradation diagnostics of lithium-ion batteries. *Energy Storage Materials*, 50:668–695, 2022.
- [31] Christoph R Birk, Matthew R Roberts, Euan McTurk, Peter G Bruce, and David A Howey. Degradation diagnostics for lithium ion cells. *Journal of Power Sources*, 341:373–386, 2017.

- [32] Adam Thelen, Yu Hui Lui, Sheng Shen, Simon Laflamme, Shan Hu, and Chao Hu. Physics-informed machine learning for degradation diagnostics of lithium-ion batteries. In *International design engineering technical conferences and computers and information in engineering conference*, volume 85383, page V03AT03A041. American Society of Mechanical Engineers, 2021.
- [33] Xuebing Han, Minggao Ouyang, Languang Lu, Jianqiu Li, Yuejiu Zheng, and Zhe Li. A comparative study of commercial lithium ion battery cycle life in electrical vehicle: Aging mechanism identification. *Journal of Power Sources*, 251:38–54, 2014.
- [34] Matthieu Dubarry, Maitane Berecibar, A Devie, D Anseán, Noshin Omar, and Igor Villarreal. State of health battery estimator enabling degradation diagnosis: Model and algorithm description. *Journal of Power Sources*, 360:59–69, 2017.
- [35] Jinpeng Tian, Rui Xiong, Weixiang Shen, and Fengchun Sun. Electrode ageing estimation and open circuit voltage reconstruction for lithium ion batteries. *Energy Storage Materials*, 37: 283–295, 2021.
- [36] Peter M Attia, William C Chueh, and Stephen J Harris. Revisiting the t0. 5 dependence of sei growth. *Journal of the Electrochemical Society*, 167(9):090535, 2020.
- [37] Hendrix Demers, Andrea Paolella, and Karim Zaghib. Characterization technique for advanced materials for lithium batteries in an sem. *Microscopy and Microanalysis*, 26(S2):2790–2792, 2020.
- [38] Seong Beom Lee and Simona Onori. A robust and sleek electrochemical battery model implementation: a matlab® framework. *Journal of The Electrochemical Society*, 168(9): 090527, 2021.
- [39] Jie Li, K Adewuyi, Nima Lotfi, Robert G Landers, and Jonghyun Park. A single particle model with chemical/mechanical degradation physics for lithium ion battery state of health (soh) estimation. *Applied energy*, 212:1178–1190, 2018.
- [40] Jorn M Reniers, Grietje Mulder, and David A Howey. Review and performance comparison of mechanical-chemical degradation models for lithium-ion batteries. *Journal of The Electrochemical Society*, 166(14):A3189–A3200, 2019.
- [41] Guodong Fan, Dongliang Lu, M Scott Trimble, Gregory L Plett, Chong Zhu, and Xi Zhang. Nondestructive diagnostics and quantification of battery aging under different degradation paths. *Journal of Power Sources*, 557:232555, 2023.
- [42] Jonas Keil and Andreas Jossen. Electrochemical modeling of linear and nonlinear aging of lithium-ion cells. *Journal of The Electrochemical Society*, 167(11):110535, 2020.
- [43] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.
- [44] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.
- [45] Muratahan Aykol, Chirranjeevi Balaji Gopal, Abraham Anapolksky, Patrick K Herring, Bruis van Vlijmen, Marc D Berliner, Martin Z Bazant, Richard D Braatz, William C Chueh, and Brian D Storey. Perspective—combining physics and machine learning to predict battery lifetime. *Journal of The Electrochemical Society*, 168(3):030525, 2021.
- [46] Yicun Huang, Changfu Zou, Yang Li, and Torsten Wik. Minn: Learning the dynamics of differential-algebraic equations and application to battery modeling. *arXiv preprint arXiv:2304.14422*, 2023.
- [47] Pengfei Wen, Zhi-Sheng Ye, Yong Li, Shaowei Chen, Pu Xie, and Shuai Zhao. Physics-informed neural networks for prognostics and health management of lithium-ion batteries. *IEEE Transactions on Intelligent Vehicles*, 2023.

- [48] Weiming Xian, Bing Long, Min Li, and Houjun Wang. Prognostics of lithium-ion batteries based on the verhulst model, particle swarm optimization and particle filter. *IEEE Transactions on Instrumentation and Measurement*, 63(1):2–17, 2013.
- [49] Chenyu Xue, Bo Jiang, Jiangong Zhu, Xuezhe Wei, and Haifeng Dai. An enhanced single-particle model using a physics-informed neural network considering electrolyte dynamics for lithium-ion batteries. *Batteries*, 9(10):511, 2023.
- [50] Tobias Hofmann, Jacob Hamar, Marcel Rogge, Christoph Zoerr, Simon Erhard, and Jan Philipp Schmidt. Physics-informed neural networks for state of health estimation in lithium-ion batteries. *Journal of The Electrochemical Society*, 170(9):090524, 2023.
- [51] Sina Navidi, Adam Thelen, Tingkai Li, and Chao Hu. Physics-informed neural networks for degradation diagnostics of lithium-ion batteries. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 87301, page V03AT03A049. American Society of Mechanical Engineers, 2023.
- [52] Weihan Li, Jiawei Zhang, Florian Ringbeck, Dominik Jöst, Lei Zhang, Zhongbao Wei, and Dirk Uwe Sauer. Physics-informed neural networks for electrode-level state estimation in lithium-ion batteries. *Journal of Power Sources*, 506:230034, 2021.
- [53] Hao Tu, Scott Moura, Yebin Wang, and Huazhen Fang. Integrating physics-based modeling with machine learning for lithium-ion batteries. *Applied Energy*, 329:120289, 2023.
- [54] Fei Feng, Sangli Teng, Kailong Liu, Jiale Xie, Yi Xie, Bo Liu, and Kexin Li. Co-estimation of lithium-ion battery state of charge and state of temperature based on a hybrid electrochemical-thermal-neural-network model. *Journal of Power Sources*, 455:227935, 2020.
- [55] Y.H. Lui. Physics-based prognostics of implantable-grade lithium-ion battery for remaining useful life prediction. *J. Power Sources*, 2020. doi: 10.1016/j.jpowsour.2020.229372.
- [56] Yu Hui Lui, Meng Li, Austin Downey, Sheng Shen, Venkat Pavan Nemani, Hui Ye, Collette VanElzen, Gaurav Jain, Shan Hu, Simon Laflamme, et al. Physics-based prognostics of implantable-grade lithium-ion battery for remaining useful life prediction. *Journal of Power Sources*, 485:229327, 2021.
- [57] Matthieu Dubarry, Cyril Truchot, and Bor Yann Liaw. Synthesize battery degradation modes via a diagnostic and prognostic model. *Journal of power sources*, 219:204–216, 2012.
- [58] Kotub Uddin, Surak Perera, W Dhammadika Widanage, Limhi Somerville, and James Marco. Characterising lithium-ion battery degradation through the identification and tracking of electrochemical battery model parameters. *Batteries*, 2(2):13, 2016.
- [59] Matthieu Dubarry and David Beck. Big data training data for artificial intelligence-based li-ion diagnosis and prognosis. *Journal of Power Sources*, 479:228806, 2020.
- [60] Zeyu Ma, Zhenpo Wang, Rui Xiong, and Jiuchun Jiang. A mechanism identification model based state-of-health diagnosis of lithium-ion batteries for energy storage applications. *Journal of Cleaner Production*, 193:379–390, 2018.
- [61] Caihao Weng, Yujia Cui, Jing Sun, and Huei Peng. On-board state of health monitoring of lithium-ion batteries using incremental capacity analysis with support vector regression. *Journal of Power Sources*, 235:36–44, 2013.
- [62] David A Van Dyk and Xiao-Li Meng. The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1):1–50, 2001.
- [63] Marc C Kennedy and Anthony O'Hagan. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000.
- [64] David A Barajas-Solano and Alexandre M Tartakovsky. Multivariate gaussian process regression for multiscale data assimilation and uncertainty reduction. *arXiv preprint arXiv:1804.06490*, 2018.

- [65] Xiu Yang, David Barajas-Solano, Guzel Tartakovsky, and Alexandre M Tartakovsky. Physics-informed cokriging: A gaussian-process-regression-based multifidelity method for data-model convergence. *Journal of Computational Physics*, 395:410–431, 2019.
- [66] Alexander Forrester, Andras Sobester, and Andy Keane. *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 2008.
- [67] Joseph Pateras, Pratip Rana, and Preetam Ghosh. A taxonomic survey of physics-informed machine learning. *Applied Sciences*, 13(12):6892, 2023.
- [68] Chuizheng Meng, Sungyong Seo, Defu Cao, Sam Griesemer, and Yan Liu. When physics meets machine learning: A survey of physics-informed machine learning. *arXiv preprint arXiv:2203.16797*, 2022.
- [69] Yanwen Xu, Sara Kohtz, Jessica Boakye, Paolo Gardoni, and Pingfeng Wang. Physics-informed machine learning for reliability and systems safety applications: State of the art and challenges. *Reliability Engineering & System Safety*, page 108900, 2022.
- [70] Ameya D Jagtap, Ehsan Kharazmi, and George Em Karniadakis. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 365:113028, 2020.
- [71] Tianju Xue, Alex Beatson, Sigrid Adriaenssens, and Ryan Adams. Amortized finite element analysis for fast pde-constrained optimization. In *International Conference on Machine Learning*, pages 10638–10647. PMLR, 2020.
- [72] Liu Yang, Dongkun Zhang, and George Em Karniadakis. Physics-informed generative adversarial networks for stochastic differential equations. *SIAM Journal on Scientific Computing*, 42(1):A292–A317, 2020.
- [73] Guofei Pang, Lu Lu, and George Em Karniadakis. fpinns: Fractional physics-informed neural networks. *SIAM Journal on Scientific Computing*, 41(4):A2603–A2626, 2019.
- [74] Mritula Chandrasekaran, Jarek Francik, and Dimitrios Makris. Gait data augmentation using physics-based biomechanical simulation. *arXiv preprint arXiv:2307.08092*, 2023.
- [75] Karthik Kashinath, M Mustafa, Adrian Albert, JL Wu, C Jiang, Soheil Esmaeilzadeh, Kamyar Azizzadenesheli, R Wang, A Chattopadhyay, A Singh, et al. Physics-informed machine learning: case studies for weather and climate modelling. *Philosophical Transactions of the Royal Society A*, 379(2194):20200093, 2021.
- [76] Daniel Vázquez Pombo, Peder Bacher, Charalampos Ziras, Henrik W Bindner, Sergiu V Spataru, and Poul E Sørensen. Benchmarking physics-informed machine learning-based short term pv-power forecasting tools. *Energy Reports*, 8:6512–6520, 2022.
- [77] Ján Drgoňa, Aaron R Tuor, Vikas Chandan, and Draguna L Vrabie. Physics-constrained deep learning of multi-zone building thermal dynamics. *Energy and Buildings*, 243:110992, 2021.
- [78] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.
- [79] Wen Li Zhao, Pierre Gentine, Markus Reichstein, Yao Zhang, Sha Zhou, Yeqiang Wen, Changjie Lin, Xi Li, and Guo Yu Qiu. Physics-constrained machine learning of evapotranspiration. *Geophysical Research Letters*, 46(24):14496–14507, 2019.
- [80] Loïc Brevault, Mathieu Balesdent, and Ali Hebbal. Overview of gaussian process based multi-fidelity techniques with variable relationship between fidelities, application to aerospace systems. *Aerospace Science and Technology*, 107:106339, 2020.
- [81] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Adv. Neural Inf. Process. Syst.*, m(Nips):6403–6414,, 2017.
- [82] Alex J Cannon. Quantile regression neural networks: Implementation in r and application to precipitation downscaling. *Computers & geosciences*, 37(9):1277–1284, 2011.

- [83] Vikram Mullachery, Aniruddh Khera, and Amir Husain. Bayesian neural networks. *arXiv preprint arXiv:1801.07710*, 2018.
- [84] Venkat Nemanic, Luca Biggio, Xun Huan, Zhen Hu, Olga Fink, Anh Tran, Yan Wang, Xiaoge Zhang, and Chao Hu. Uncertainty quantification in machine learning for engineering design and health prognostics: A tutorial. *Mechanical Systems and Signal Processing*, 205:110796, 2023.
- [85] Ilias Bilionis and Nicholas Zabaras. Multi-output local gaussian process regression: Applications to uncertainty quantification. *Journal of Computational Physics*, 231(17):5718–5746, 2012.
- [86] Jamie L Crandell, Corrine I Voils, YunKyung Chang, and Margarete Sandelowski. Bayesian data augmentation methods for the synthesis of qualitative and quantitative research findings. *Quality & quantity*, 45:653–669, 2011.
- [87] Tomohiro Endo, Tomoaki Watanabe, and Akio Yamamoto. Confidence interval estimation by bootstrap method for uncertainty quantification using random sampling method. *Journal of Nuclear Science and Technology*, 52(7-8):993–999, 2015.
- [88] Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and mitigating gradient pathologies in physics-informed neural networks. *arXiv preprint arXiv:2001.04536*, 2020.

Appendix

A1. Degradation quantification using a half-cell model

In a half-cell model, the pseudo-full-cell OCV curve can be constructed and mathematically expressed as follows,

$$V_c(Q)|_{Q=Q_c} = V_p(q_p)|_{q_p=\frac{Q-\delta_p}{m_p}} - V_n(q_n)|_{q_n=\frac{Q-\delta_n}{m_n}}$$

where $V_c(Q)$ is the pseudo-full-cell OCV curve, Q_c refers to the cell capacity at different states of charge, $V(q)$ is a half-cell curve with specific capacity q (mAh/g), m is the active mass (g), and δ is the half-cell curve slippage (mAh). The subscripts p and n correspond to the positive and negative electrodes (PE and NE). The two slippage parameters, δ_p and δ_n , quantify the horizontal distance of the left endpoint of the positive and negative half-cell curves to $Q_c = 0$.

Then, the differential voltage curve ($dV/dQ(Q)$) of a half-cell model can be derived as follows,

$$\frac{dV_c}{dQ}(Q)|_{Q=Q_c} = \frac{1}{m_p} \frac{dV_p(q_p)}{dq_p}|_{q_p=\frac{Q-\delta_p}{m_p}} - \frac{1}{m_n} \frac{dV_n(q_n)}{dq_n}|_{q_n=\frac{Q-\delta_n}{m_n}}$$

In this work, we follow a manual approach, presented in [30], to fit a half-cell model to cells by adjusting four half-cell parameters (m_p , m_n , δ_p , and δ_n). This manual process focuses on: 1. aligning peaks from the reconstructed $dV/dQ(Q)$ curves to the peaks from experimental $dV/dQ(Q)$ curves; 2. matching the reconstructed OCV curves to experimental QV curves, which considers both the endpoint locations and the overall shape. First, m_n and δ_n are adjusted to match the peaks on $dV/dQ(Q)$ curves since the negative electrodes contribute major peaks in the OCV. After tuning the half-cell curve of the negative electrode, m_p and δ_p are used to adjust peak magnitudes on $dV/dQ(Q)$ curves and line up the endpoints of QV curves. This fitting process is repeated to fine-tune the fitting results to match the fitted curves as much as possible.

A2. Automatic Fitting vs. Manual Fitting

Here, we conducted a comparative analysis between manual and automatic approaches for fitting the half-cell model to justify the necessity of the development of the PIML methods using the manual fitting approach. We compared the positive and negative active mass estimates by manual fitting, as outlined in Sec. 3, with half-cell parameter estimates by nonlinear optimization using the original half-cell potential data (i.e., automatic fitting). This comparison additionally included the PINN model, trained on parameter estimates by the manual fitting approach. It was performed on two cells destructively analyzed after certain aging periods as an experimental validation of positive and negative active mass estimations. Automatic fitting employed a bi-objective loss function to match the endpoints of the simulated and experimental voltage vs. capacity (QV) curves and improve the overall agreement between these two curves by minimizing an RMSE loss. Experimental validation initially involved recovering and treating two samples from each working electrode of the two selected cells. Then, each sample was coupled with lithium metal to build coin half-cells. These coin half-cells were cycled at a slow rate to obtain their capacity. Finally, the full cell's remaining positive and negative active mass was calculated based on the nominal specific capacity and area ratio [56]. The comparison results are depicted in Fig. 15, revealing certain limitations of nonlinear optimization (automatic fitting).

- First, the optimization problem for automatic fitting has multiple local minima, leading to run-to-run variability in optimal active mass parameters depending on the initial guess. We illustrated this variability by presenting the mean and error bars (spread) derived from five optimization runs, each starting at a different initial guess.
- Second, manual fitting considers phase transitions by matching characteristic features (e.g., the position and height of a peak) of simulated and experimental $dQ/dV(V)$ curves.

However, incorporating feature matching into nonlinear optimization could be challenging due to the need to automatically extract features and track the trend of how each half-cell model parameter evolves.

- Third, even if such optimization were feasible, its implementation on a battery management system would be impractical due to the extensive tuning required. The use of an optimization strategy reliant on iterative tuning for feature matching is likely to surpass the time constraints imposed by real-time battery management requirements. Moreover, the challenge of tuning the system to generate reliable initial parameter estimates for diverse operating conditions introduces an additional layer of complexity.

In contrast, building machine learning models for degradation diagnostics reduces the time cost associated with manual fitting and, mostly importantly, automates health parameter estimation by taking a measured $dQ/dV(V)$ curve as the model input, running a forward pass of the model, and producing an estimate of the health parameters as the output. This automated process is particularly suitable for on-board battery management system applications where extensive tuning is infeasible.

Additionally, the comparison results in Fig. 15 show that active mass estimates by the machine learning method (i.e., PINN) align well with those by manual fitting; the former offers the additional benefits of reduced time cost and automation. Both methods generally agreed better with experimental validation than automatic fitting. In conclusion, our machine learning approach strikes a practical balance between accuracy and feasibility for real-world applications.

A3. Kernel functions in GPR (kriging)

Linear kernel

The linear kernel models a linear relationship between input data points. Let \mathbf{x} and \mathbf{x}' be two input feature vectors. The linear kernel function, denoted as $k_{\text{linear}}(\mathbf{x}, \mathbf{x}')$, is defined as the dot product of the two feature vectors:

$$k_{\text{linear}}(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$$

Radial basis function (RBF) kernel

The RBF kernel, also known as the squared exponential kernel, captures smooth, non-linear relationships between data points. It measures the similarity between two data points based on the distance between them in the feature space. It assigns higher similarity (covariance) to data points that are closer to each other and lower similarity to data points that are farther apart. The RBF kernel is commonly used when the assumption of smoothness and stationarity is reasonable in the data.

$$k_{\text{RBF}}(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2l^2}\right)$$

where σ is the kernel bandwidth or length scale parameter. It determines the "width" of the kernel and controls the smoothness of the resulting function.

Exponential kernel

The exponential kernel is similar to the RBF kernel and also captures smooth, non-linear relationships. However, it tends to have a faster decay, making it more suitable for cases where data points are expected to be less correlated as the distance between them increases.

$$k_{\text{exp}}(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|}{l}\right)$$

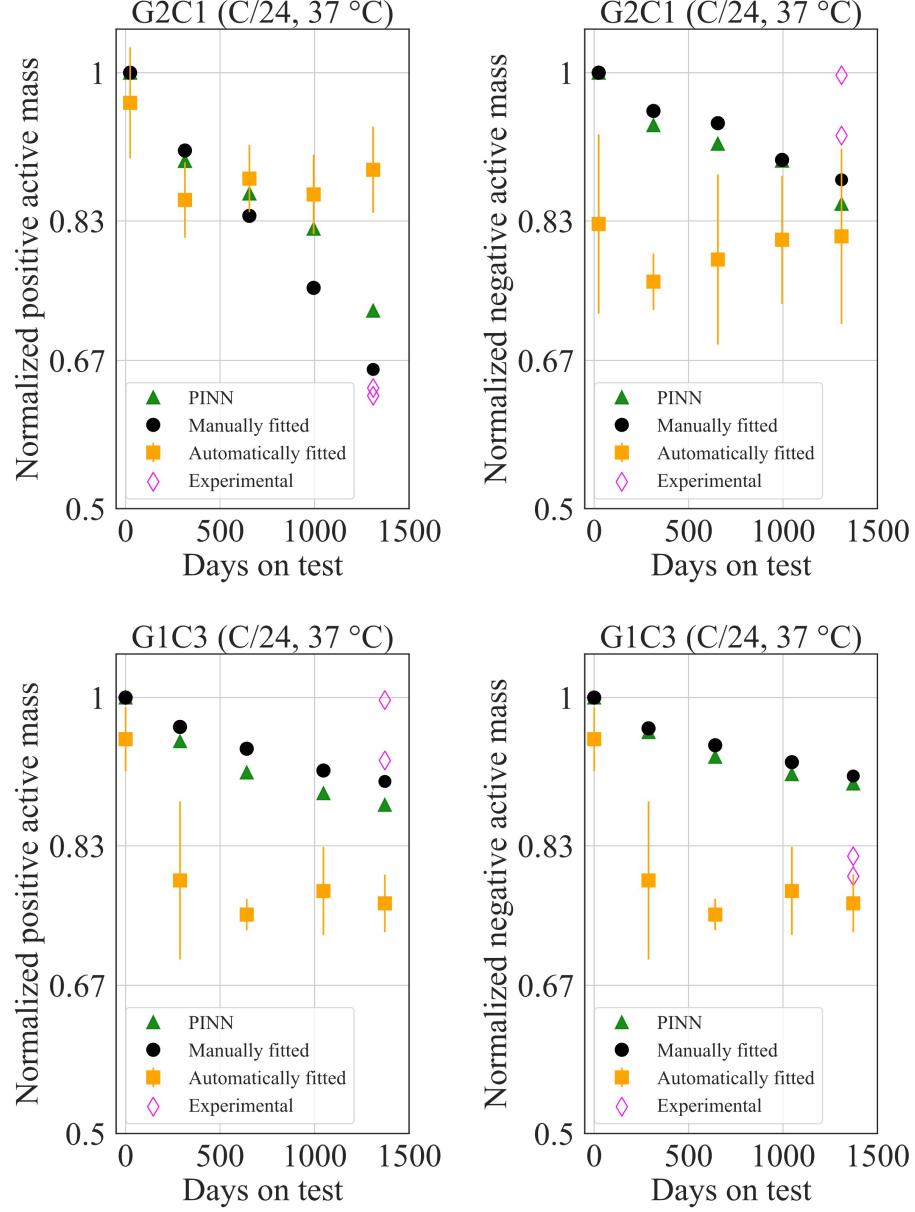


Figure 15: Comparison of active mass estimates by PINN, manual and automatic fitting, and experimental validation for two selected cells.

Rational quadratic kernel

The rational quadratic (RQ) kernel is a smooth non-linear kernel with a tunable parameter α . It can capture a wide range of smoothness levels in the data and can be useful when the underlying relationship is not strictly linear or purely smooth.

$$k_{\text{RQ}}(\mathbf{x}, \mathbf{x}') = \sigma^2 \left(1 + \frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\alpha l^2} \right)^{-\alpha}$$

The value of α controls the balance between the linear and non-linear components of the kernel. When α is large, the RQ kernel behaves more like the RBF kernel, emphasizing non-linear patterns.

On the other hand, when α is close to zero, the RQ kernel behaves more like the linear kernel, capturing linear relationships in the data.

Polynomial kernel

The polynomial kernel models polynomial relationships between data points. It can be useful when the relationship between the inputs is known to be polynomial in nature. It is a generalization of the linear kernel and can capture various degrees of polynomial relationships between data points.

$$k_{\text{poly}}(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}' + c)^d$$

where d is the degree of the polynomial, and c is an optional constant term that can be included to control the bias of the polynomial expansion.

Matérn 3/2 kernel

The Matérn kernel family in GPR modeling includes two specific variants known as Matern32 and Matern52 kernels. Both kernels are characterized by their ability to model different levels of smoothness in the data, making them useful in various applications. The Matérn 3/2 kernel is used to model data with sharp changes and smoothness. It is commonly used when the data is expected to have abrupt variations, but some level of smoothness is also present, making it suitable for modeling real-world data with moderate roughness. It is also computationally efficient and suitable for datasets with a moderate number of data points.

$$k_{\text{Mat32}}(\mathbf{x}, \mathbf{x}') = \sigma^2 \left(1 + \frac{\sqrt{3}\|\mathbf{x} - \mathbf{x}'\|}{l} \right) \exp \left(-\frac{\sqrt{3}\|\mathbf{x} - \mathbf{x}'\|}{l} \right)$$

where ρ is the length scale parameter.

Matérn 5/2 kernel

The Matérn52 kernel is another member of the Matérn family, with a smoothness parameter of $\nu = 5/2$ (differentiable twice). It is smoother than the Matérn32 kernel and can effectively model data with higher roughness or irregularity.

$$k_{\text{Mat52}}(\mathbf{x}, \mathbf{x}') = \sigma^2 \left(1 + \frac{\sqrt{5}\|\mathbf{x} - \mathbf{x}'\|}{l} + \frac{5\|\mathbf{x} - \mathbf{x}'\|^2}{3l^2} \right) \exp \left(-\frac{\sqrt{5}\|\mathbf{x} - \mathbf{x}'\|}{l} \right)$$

where ρ is the length scale parameter.

A4. Trajectory prediction results

Here, we present the trajectory prediction results for all the test cells across six groups. The plots display the predicted health parameters generated by each of the four PIML methods considered, alongside the fitted true values.

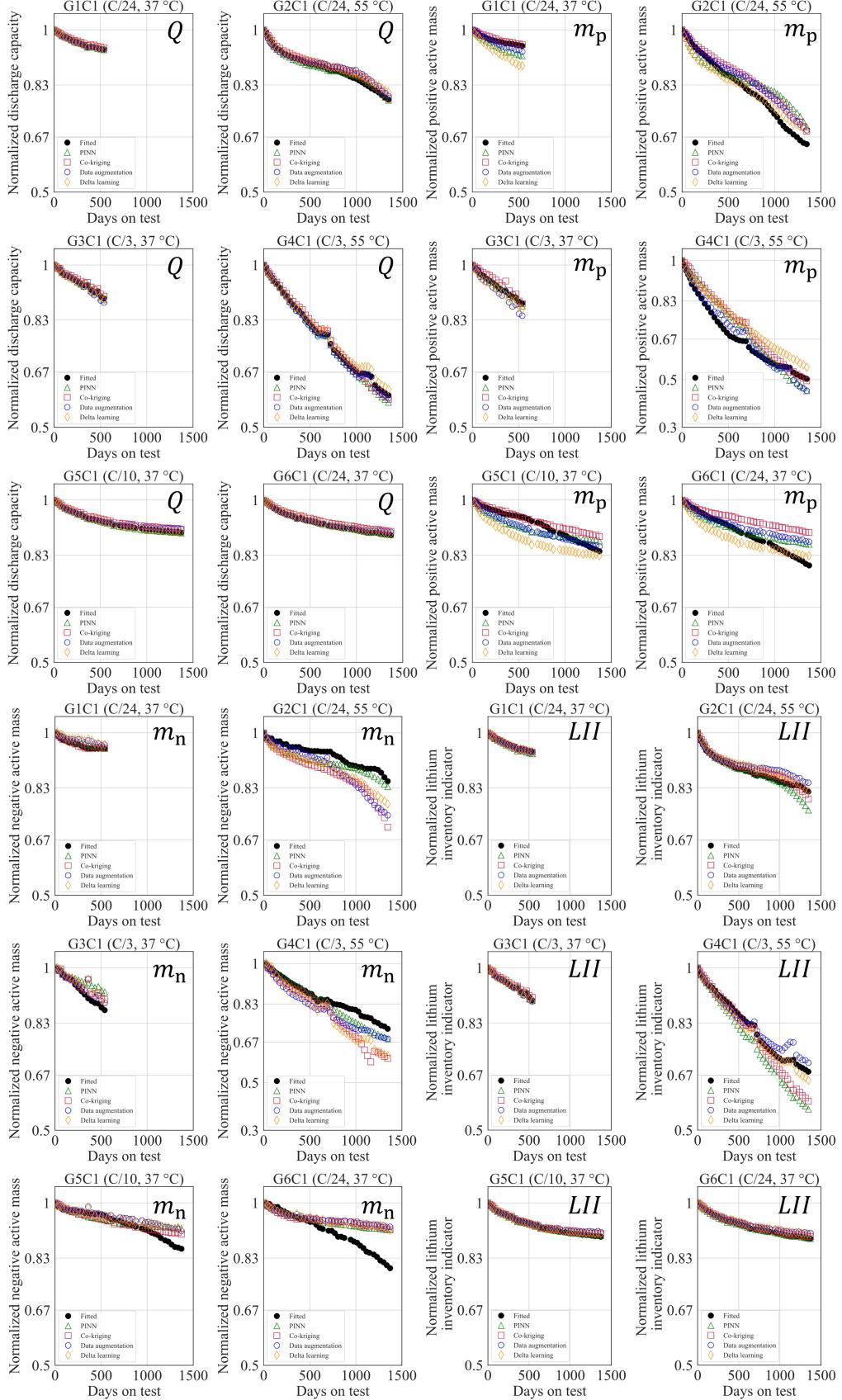


Figure 16: Trajectory prediction results for the first fold (first cell of each group).

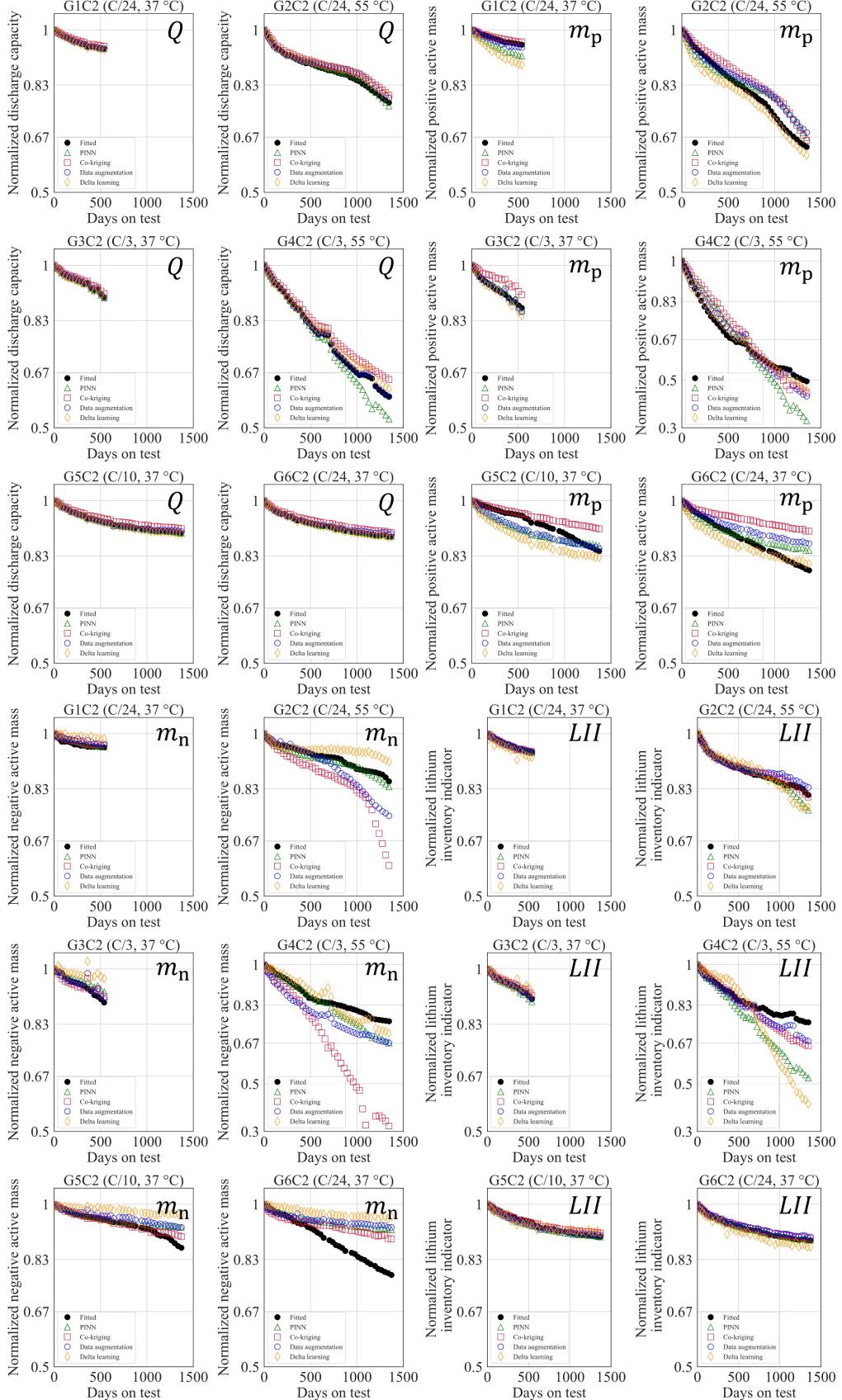


Figure 17: Trajectory prediction results for the second fold (second cell of each group).

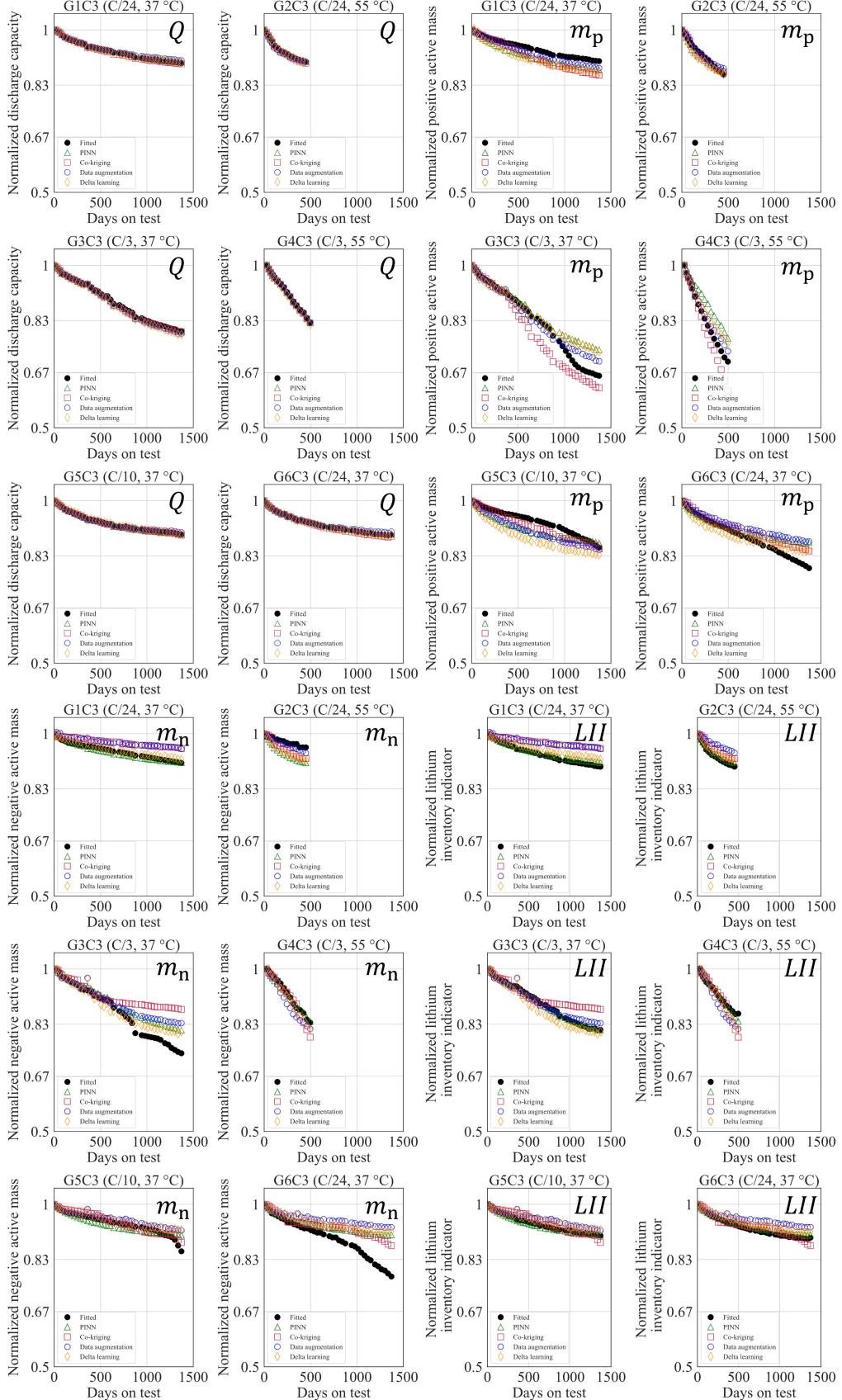


Figure 18: Trajectory prediction results for the third fold (third cell of each group).

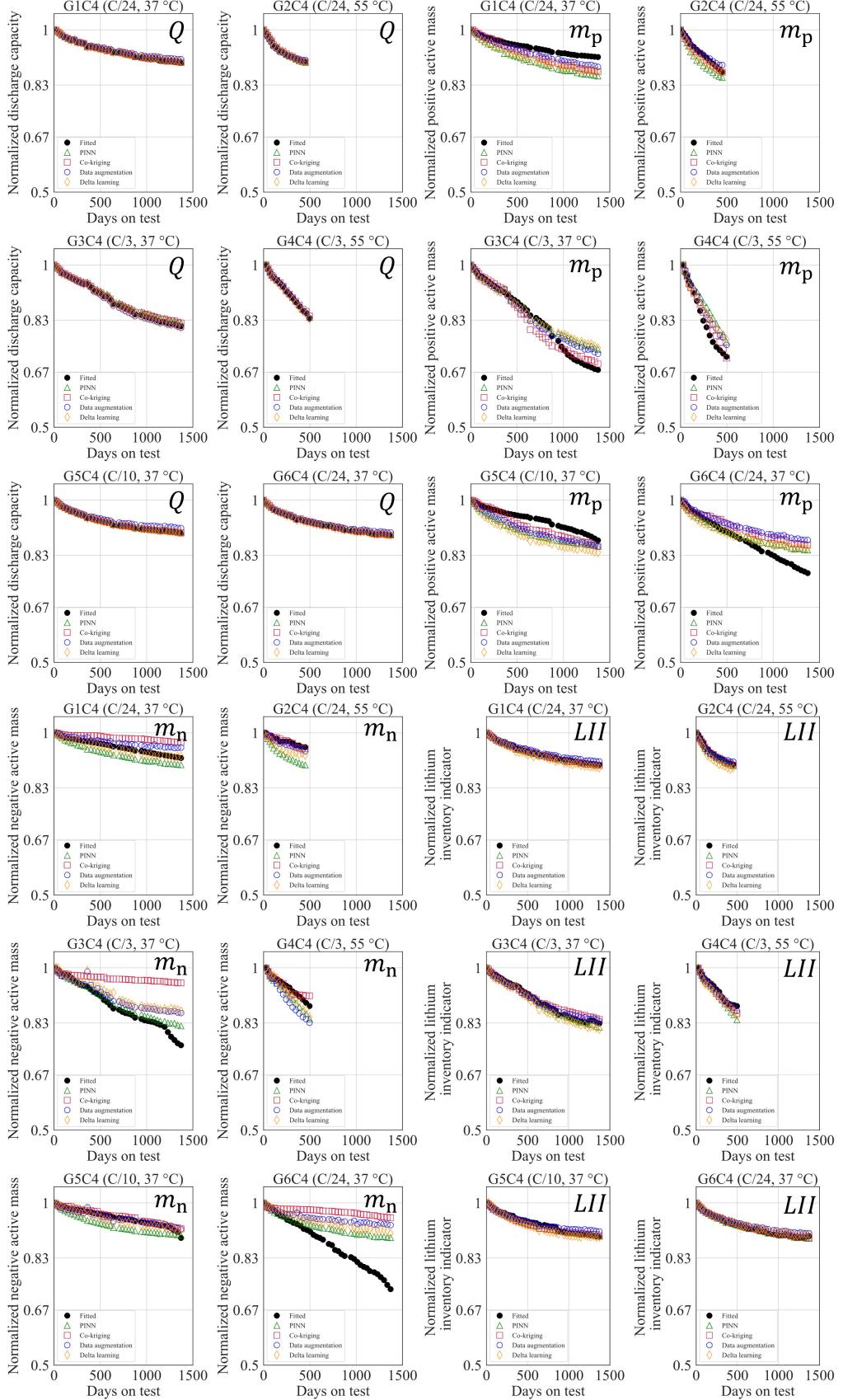


Figure 19: Trajectory prediction results for the forth fold (forth cell of each group).

A5. Half-cell surrogate model

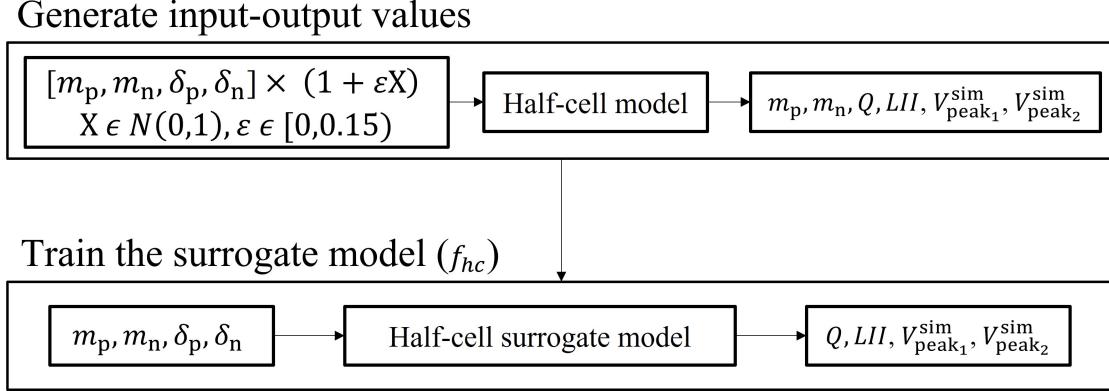


Figure 20: Training half-cell surrogate model.

As described in Sec. 5.1, we integrate a half-cell surrogate model into the loss function of the PINN model to ensure both differentiability and consistency of the computation graph between the inputs and outputs. Here, we provide detailed information about how we implemented the half-cell surrogate model for this purpose.

As depicted in Fig. 20, we initially utilize the half-cell model to obtain health parameters and two simulated $dQ/dV(V)$ peak positions(voltage values) by inputting perturbed values of the half-cell model parameters ($\delta_p, \delta_n, m_p, m_n$). The objective is to encompass a broader range of possible parameters for inputting into the half-cell model. This is necessary because in the loss function, we need to be able to map the deviated values of half-cell parameters, which are predicted by the network, to health parameters.

The perturbed values are obtained through standard normal sampling from a 15% range of perturbation for each parameter. Subsequently, we use the true values, perturbed values, and corresponding degradation and peak position values as input and output values, respectively. Using these values, we establish a straightforward mapping function between the half-cell model parameters and capacity, lithium inventory indicator, and two $dQ/dV(V)$ curves peak voltage values ($Q, LII, V_{\text{peak}_1}^{\text{sim}}, V_{\text{peak}_2}^{\text{sim}}$) to train a surrogate neural network (f_{hc}) with perfect accuracy in mapping these values.