Supplementary Information (SI) for Energy & Environmental Science.
This journal is © The Royal Society of Chemistry 2025

Supporting Information

Uncovering the Impact of Battery Design Parameters on Health and Lifetime Using Short Charging Segments

Wendi Guo, Søren Byg Vilsen, Yaqi Li, Ashima Verma, Daniel Ioan Stroe, and Daniel Brandell**

W. G. Author 1, D. B. Author 6

Department of Chemistry, Uppsala University, Lägerhyddsvägen 1, Uppsala, 75121, Sweden E-mail: wendi.guo@kemi.uu.se, daniel.brandell@kemi.uu.se

S. B.V. Author 2

Department of Materials and Production, Aalborg University, Fibigerstræde 16, Aalborg, 9220, Denmark

Y. L. Author 3

Department of Energy, Aalborg University, Pontoppidanstræde 111, Aalborg, 9220, Denmark

A.V Author 4

Department of Chemistry, Uppsala University, Lägerhyddsvägen 1, Uppsala, 75121, Sweden

D. I. S. Author 5

Department of Energy, Aalborg University, Pontoppidanstræde 111, Aalborg, 9220, Denmark

Table S1. Definition of 29 manually engineered features.

Feature set	**Feature**	**Description**
F_t	F1	Charging time (s)
F_{EOCV}	F2	End of charge voltage (EOCV)
F_n	F3	Maximum IC peak (Ah/V) when charging voltage is between V_l and V_u
	F4	1st IC peak area (Ah) when charging voltage is between V_l and V_u
	F5	1st IC peak position (V) when charging voltage is between V_l and V_u
	F6	Lowest DV valley position (Ah) when charging voltage is between V_l and V_u
F_g	F7	Slope of the voltage curve (V) when charging voltage is between V_l and V_u
	F8	Slope of the capacity difference ΔQ (Ah) curve when charging voltage is between V_l and V_u
	F9	Slope of the current curve (A) when charging current is between 0.1A and 0.5A
$F_{v,I}$	F10	Standard deviation of the ΔQ (Ah) when charging voltage is between V_l and V_u
	F11	Median of the current (A) when the charging current is between 0.1A and 0.5A
	F12	Kurtosis of the current (A) when the charging current is between 0.1A and 0.5A

	F13	Skewness of the current (A) when the charging current is between 0.1A and 0.5A
	F14	Standard deviation of the current (A) when the charging current is between 0.1A and 0.5A
	F15	Shannon entropy of the current (A) when charging current is between 0.1A and 0.5A
	F16	Median of the voltage(V) when charging voltage is between V_l and V_u
	F17	Skewness of the voltage (V) when charging voltage is between V_l and V_u
	F18	Standard deviation of the voltage (V) when charging voltage is between V_l and V_u
	F19	Shannon entropy of the voltage (V) when charging voltage is between V_l and V_u
	F20	Kurtosis of the capacity (Ah) when charging voltage is between V_l and V_u
	F21	Skewness of the capacity (Ah) when charging voltage is between V_l and V_u
F_Q	F22	Standard deviation of the capacity (Ah) when charging voltage is between V_l and V_u
	F23	Median of ΔQ (Ah) when charging voltage is between V_l and V_u
	F24	Minimum ΔQ (Ah) when charging voltage is between V_l and V_u

	F25	Kurtosis of ΔQ (Ah) when charging voltage is between V_l and V_u
	F26	Skewness of ΔQ (Ah) when charging voltage is between V_l and V_u
	F27	Shannon Entropy of ΔQ (Ah) when charging voltage is between V_l and V_u
$F_{t,cv}$	F28	CV charging time
F_c	F29	Equivalent full cycles

Note: To ensure fair model comparison and prevent data leakage, the voltage boundaries V_l and V_u are predetermined and held constant across all splits. This design ensures that feature extraction relies solely on training data, with no influence from the test set.

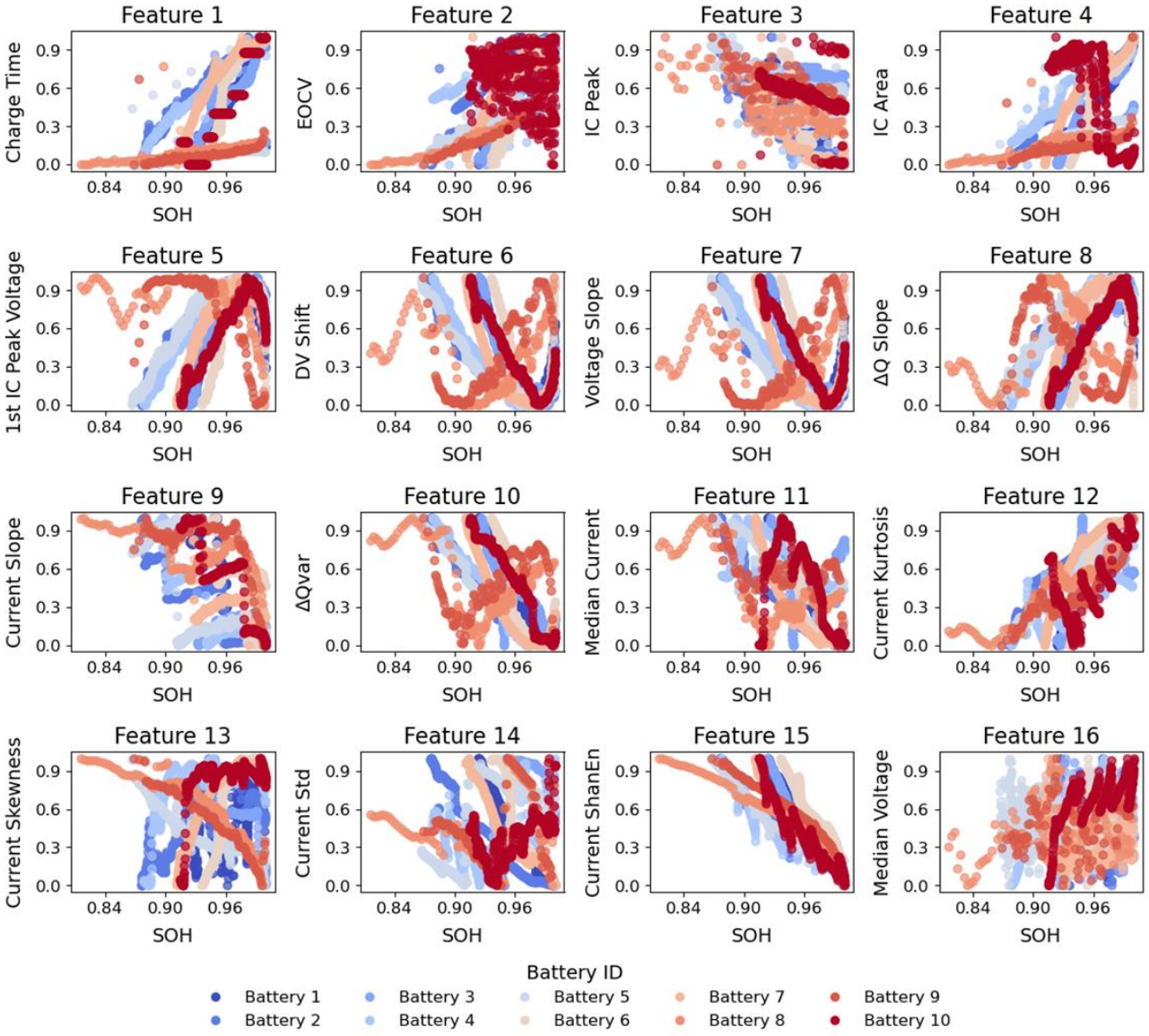

Figure S1. Visualization of the 16 manually engineered features most correlated with SOH (based on Spearman correlation), extracted from partial charging segments. Each subplot shows SOH on the x-axis and the normalized value of the corresponding feature on the y-axis. The number in each subplot title indicates the feature index. Here, the lower and upper voltage limit (V_l and V_u) are set to 3.65V and 4.1V, respectively. Gradient colors illustrate feature evolution trends across different cells, revealing notable divergences that present challenges for health status prediction under varying fast-charging conditions.

Table S2. Normal distribution of design parameters.

Feature set	Parameter	Unit	Meaning	Mean (u)	Std	Sampling
P	$C_{pos,max}$	mol/m^3	Maximum lithium concentration in the positive electrode	50073	0.01* $u_{Cpos,max}$	300
	$epss_{pos}$	1	Porosity of the positive electrode	0.69	0.01* $u_{epsspos}$	
	$r_{p,neg}$	1e-6 m	Particle radius of the negative electrode	11	0.01* $u_{rp,neg}$	
	L_{pos}	1e-6 m	Thickness of the positive electrode	84	0.01* u_{Lpos}	
	L_{neg}	1e-6 m	Thickness of the negative electrode	108	0.01* u_{Lneg}	
	D_{neg}	1e-13m^2/s	Solid phase lithium-ion diffusion coefficient	1.5	0.01* u_{Dneg}	

Note: We consider both geometric/manufacturing parameters (i.e., L_{pos}, L_{neg}, ε_{pos}) and intrinsic material properties (i.e., $C_{s,max}$, r_{neg}, r_{pos}, and D_{neg}) as design parameters, as they influence battery performance and can be varied through material selection or design optimization. A 1% relative standard deviation is typically assumed to represent the variation in design parameters under a high-precision manufacturing process.

Supplementary Note 1. Aging Experiments

Ten commercial 18650 lithium-ion cells (graphite/NMC532) were tested using a Neware battery cycler with custom four-point contact fixtures (Figure S2). The cells were maintained at constant temperatures of 0℃, 15℃, 25℃, and 35℃ inside a temperature-controlled chamber. All experiments were carried out over a two-year period at the AAU Energy battery lab. Details of the data acquisition is described in Table S3. Two cells operating at 0℃ failed during testing. The cycling protocols were designed to capture a range of realistic EV usage patterns. One set reflects continuous-use scenarios such as buses, taxis, and commercial or industrial fleets, where daily operation often exceeds 10 hours and cells are discharged continuously to the lower voltage limit, simulating high-utilization duty cycles with minimal rest. In contrast, another subset emulates consumer EV behaviour, where partial usage is more common-discharging only to around 70% depth of discharge (DOD)-to reflect typical urban driving with intermittent use. Cells were charged using various methods, including standard CCCV protocols at 0.6C, 1C, 1.3C, and 2C, as well as two multistep fast-charging protocols (2C-1.5C-1C-0.5C applied at specific SOC intervals).

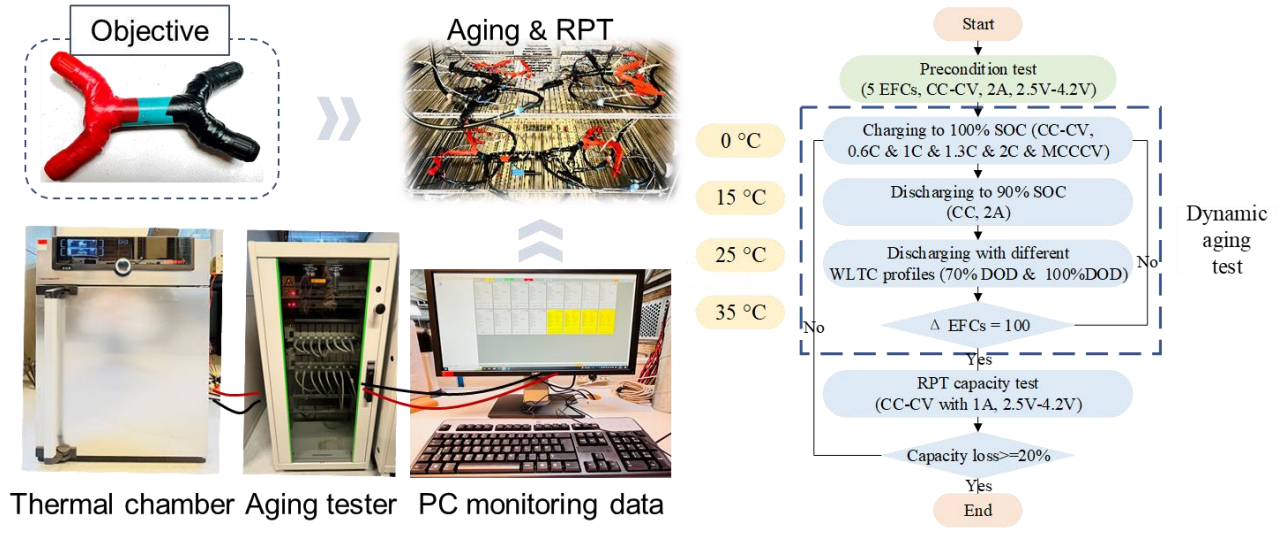

Figure S2. Equipment at AAU Energy Battery Lab and Test Procedure. The figure displays the available equipment, along with a flowchart illustrating the dynamic aging and RPT test sequence.

Table S3. Comparison of fast charging datasets: test matrix and charging protocols.

Crate-Vmax	**Discharging Cut-off condition**	**Charging protocols CC1(Q1%)-CC2(Q2%)-CC3(Q3%)-CC4(Q4%)**	**Temperature [°C]**	**# cells**	**Avg. charge time [min]**
2C-4.2V	Depth of Discharge=70%	[2, 2, 2, 2] [2, 2, 2, 2]	0 0	1 1	51 46
1.3C-4.2V	2.5V	[1.3,1.3,1.3,1.3]	15	2	63
1C-4.2V	2.5V	[1, 1, 1, 1]	25	1	75
1.3C-4.2V	2.5V	[1.3, 1.3, 1.3, 1.3]	25	1	59
MC1-4.2V	2.5V	[2, 1.5, 1, 0.5] [40%,60%,80%,100%]	25	1	64
MC2-4.2V	2.5V	[2, 1.5, 1, 0.5] [20%,60%,80%,100%]	25	1	68
2C-4.2V	2.5V	[2, 2, 2, 2]	25	1	46
2C-4.2V	Depth of Discharge=70%	[2, 2, 2, 2]	35	1	45

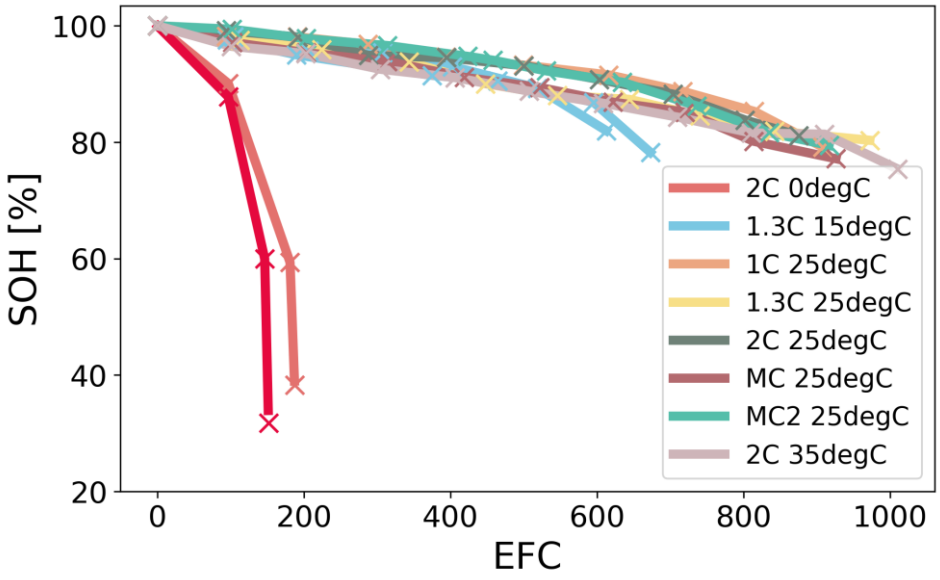

Figure S3. Degradation trajectories of graphite/NMC532 lithium-ion batteries under multiple fast charging protocols. Even under the same C-rates (i.e., 2C 0°C) distinct aging behaviors are observed.

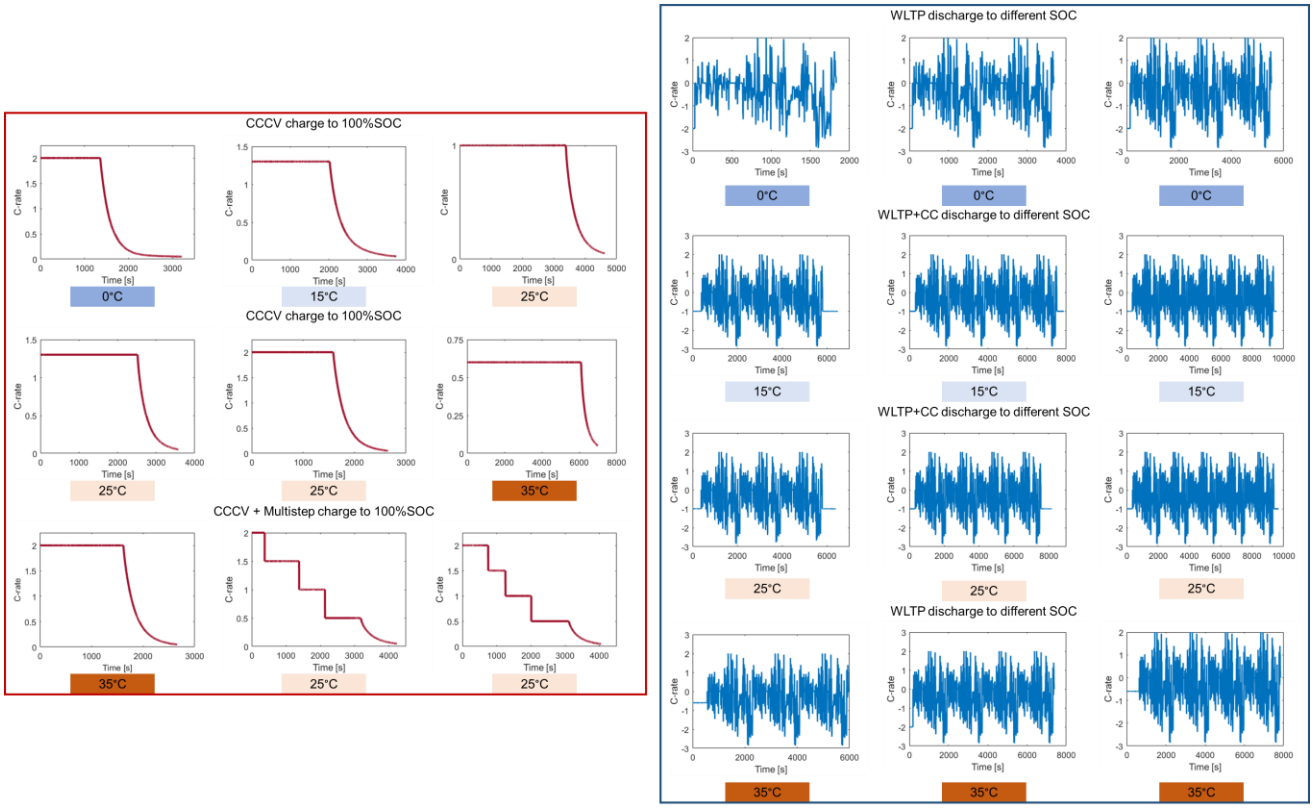

Figure S4. Aging tests designs for cells 1-10. (a) Charging under multiple C-rates using either CCCV or multistep protocols to 100% SOC. (b) Discharging under various dynamic profiles and temperatures, including WLTP, WLTP+CC, and different depth of discharge (DOD).

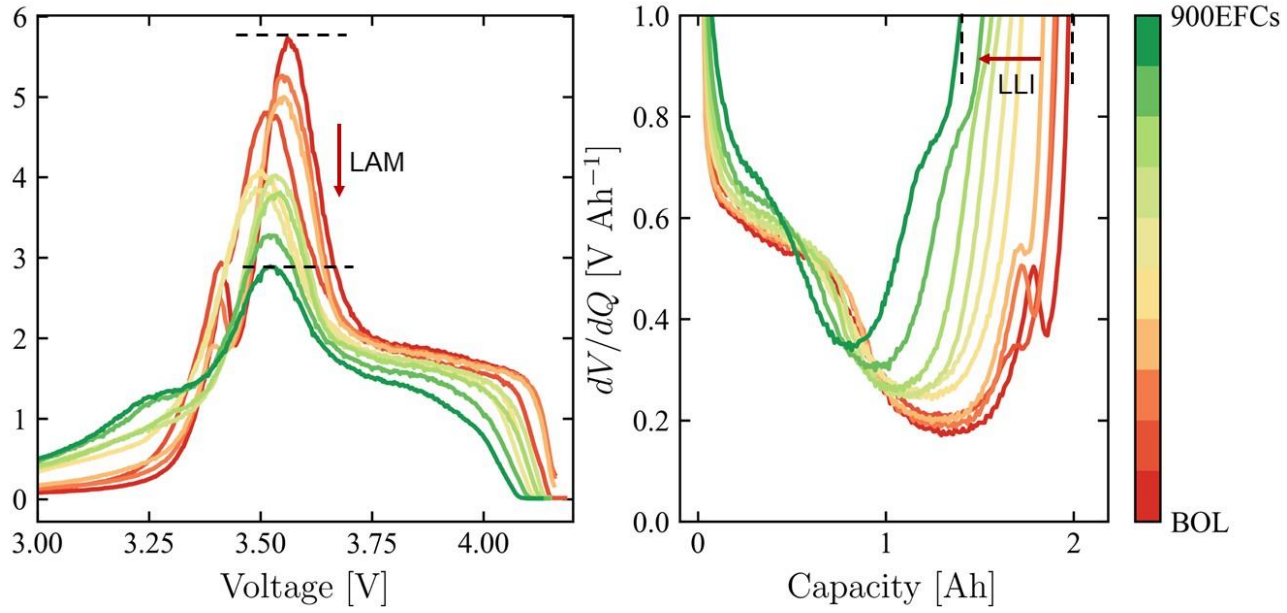

Figure S5. Representative illustration of the relationship between non-destructive IC/DV analysis and aging mechanisms, including loss of active material (LAM) and loss of lithium inventory (LLI). IC and DV curves are shown across aging cycles at 1C 25 ºC. LAM and LLI are evaluated using IC/DV features from measured charging data.

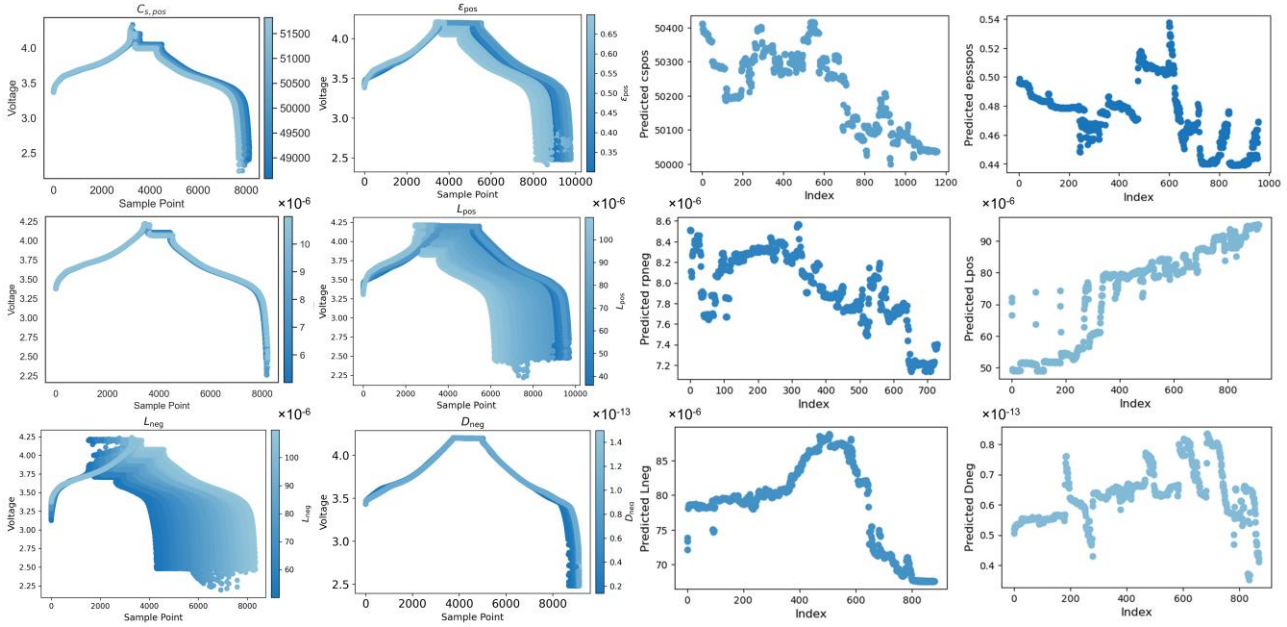

Figure S6. Simulated CCCV profiles generated by sweeping six design parameters using a digital twin model[1]. The corresponding trends of the inferred effective design parameters are shown based on random forest mapping, under an aging case at 1C and 25ºC.

Supplementary Note 2. Monte Carlo experiment

The Monte Carlo experiment is performed when splitting the training set such that the models are unbiased to specific long-lived or short-lived cells. In this way, different primary training-testing pairs can be evaluated to guarantee statistical confidence. The random seeds for the Monte Carlo experiments are from 0 to 9999 in Spyder (Python 3.12). In this study, we employed a fixed random seed generator using NumPy's default_rng to ensure consistent Monte Carlo sampling across different experiment conditions, thereby maintaining identical train-test splits and guaranteeing the robustness and statistical reliability of the results. Such 100 Monte Carlo experiments generate 100 testing situations for assessing the performance without underlying physical features.

MMD Regularization in Random Forest Regression

To investigate the best way to enhance the generalization capability of the multi-task Random Forest (RF) model across batteries with different operating conditions, we incorporate a domain adaptation strategy based on Maximum Mean Discrepancy (MMD). We add a regularization term that penalizes the feature distributions discrepancy between training and testing datasets. The total training objective is:

$$L_{total} = w_{SOH} \cdot L_{SOH} + w_{RUL} \cdot L_{RUL} + \lambda_{SOH} \cdot MMD(X_{train}, X_{test}) + \lambda_{RUL} \cdot MMD(X_{train}, X_{test})$$

Where L_{SOH} and L_{RUL} are the task-specific losses, and w_{SOH} and w_{RUL} are their respective weights. In our experiments, we set w_{SOH}=1.0, w_{RUL}=0.001. The terms λ_{SOH} and λ_{RUL} are MMD regularization weights for each task. $MMD(X_{train}, X_{test})$ measures the feature distribution gap;

We adopt a Gaussian kernel-based formulation of MMD. The squared MMD between the training set X_{train} and the test set X_{test} is computed as:

$$MMD^2(X_{train}, X_{test}) = E_{x,x'}[k(x,x')] + E_{y,y'}[k(y,y')] - 2\,E_{x,y}[k(x,y)]$$

where $k(x,y) = exp(-\gamma\|x-y\|^2)$ is the Gaussian kernel function, and γ is the kernel bandwidth hyperparameter, which we set to 1. Here, $x, x' \in X_{train}$ are two independently sampled feature vectors from the training set; $y, y' \in X_{test}$ are two independently sampled feature vectors from the test set.

To enforce sample-level alignment, we compute the MMD distance for each training sample based on its similarity to the test distribution. This per-sample MMD value is then used to adjust the training targets:

$$y'_{SOH,i} = y_{SOH,i} + \lambda_{SOH} \cdot MMD_i$$

$$y'_{RUL,i} = y_{RUL,i} + \lambda_{RUL} \cdot MMD_i$$

The adjusted targets y' are then used to train the multi-task RF model. We conduct a grid search over λ_{SOH} and λ_{RUL}, aiming to minimize the average weighted RMSE over 50 random splits, while

keeping the RF hyperparameters fixed. The optimal MMD weights are obtained as those that yield the best average generalization performance.

CORAL-Based Regularization in Multi-task Random Forest

To further improve model generalization across batteries with different feature distributions, we also incorporate a domain alignment technique known as CORrelation Alignment (CORAL). The Coral loss between the source (training) features X_s and target (testing) X_t features is defined as:

$$L_{CORAL}(X_S, X_t) = \frac{1}{4d^2}\|C_S - C_t\|_F^2$$

Where $C_s=Cov(X_s)$ and $C_t=Cov(X_t)$ are the empirical covariance matrices of the source and target domains, respectively, and $\| \quad \|_F$ denotes the Frobenius norm. A small value is added to the diagonal of the covariance matrices to ensure numerical stability.

To integrate CORAL into the training process of a multi-task random forest model, we treat the CORAL loss as a regularization term and apply it to the label space through target modification. Specifically, given task-specific weights w_{SOH} and w_{RUL} and CORAL regularization strengths λ_{SOH} and λ_{RUL}, we adjust the training labels as:

$$y'_{SOH,i} = w_{SOH} \cdot y_{SOH,i} + \lambda_{SOH} \cdot L_{CORAL}(X_{train}, X_{test})$$

$$y'_{RUL,i} = w_{RUL} \cdot y_{RUL,i} + \lambda_{RUL} \cdot L_{CORAL}(X_{train}, X_{test})$$

In our experiments, we set w_{SOH}=1.0, w_{RUL}=0.001. The adjusted training targets *y'* are then used to fit the random forest model. Similar to the MMD-based strategy, we perform a grid search over CORAL regularization weights to identify the optimal values $(\lambda_{SOH}, \lambda_{RUL})$ that minimize the average weighted RMSE across 50 random train/test splits, while keeping the RF hyperparameters fixed.

Mixed-inputs aging indicators

To explore health status prediction across varying operating conditions, we use a multistage charging format, denoted as C1-C2-C3-C4. Due to potential sensor errors and missing temperature data, temperature is assumed to follow the preset value (T). These conditions are incorporated as aging health indicators for analysis. Details can be seen from our previous study[2].

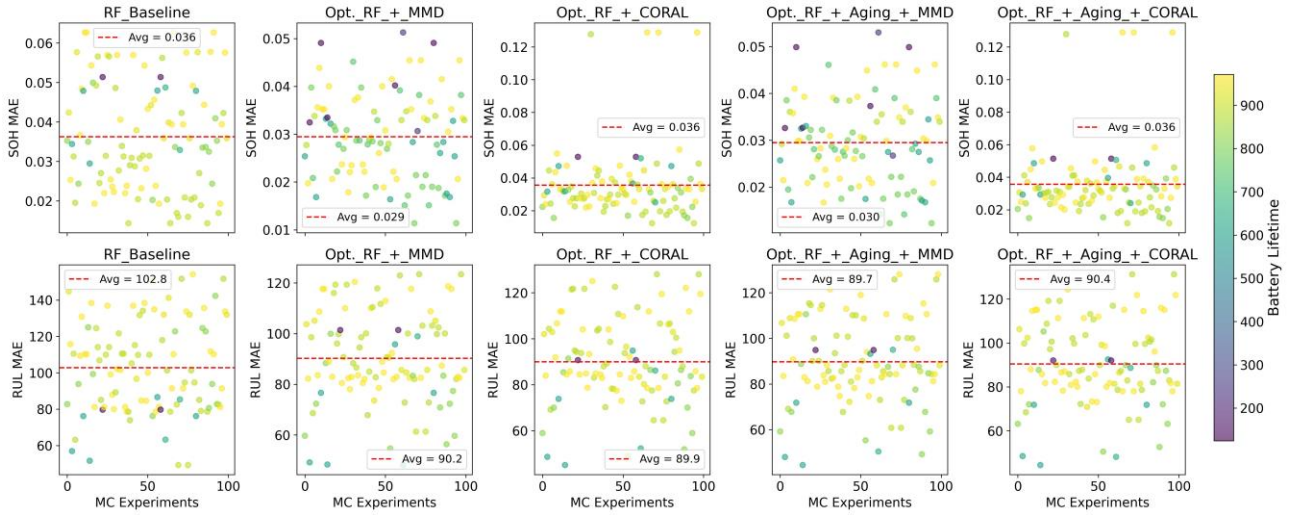

Figure S7. Performance comparison of (a) RF baseline (with optimized hyperparameters), (b) MMD-aided RF, (c) CORAL-aided RF, (d) MMD-aided RF with mixed inputs aging indicators, (e) CORAL-aided RF with mixed inputs aging indicators using the 16 correlated manual engineered features (without design parameters). The training-to-testing ratio is 0.8:0.2. Prediction performance across 100 Monte Carlo experiments.

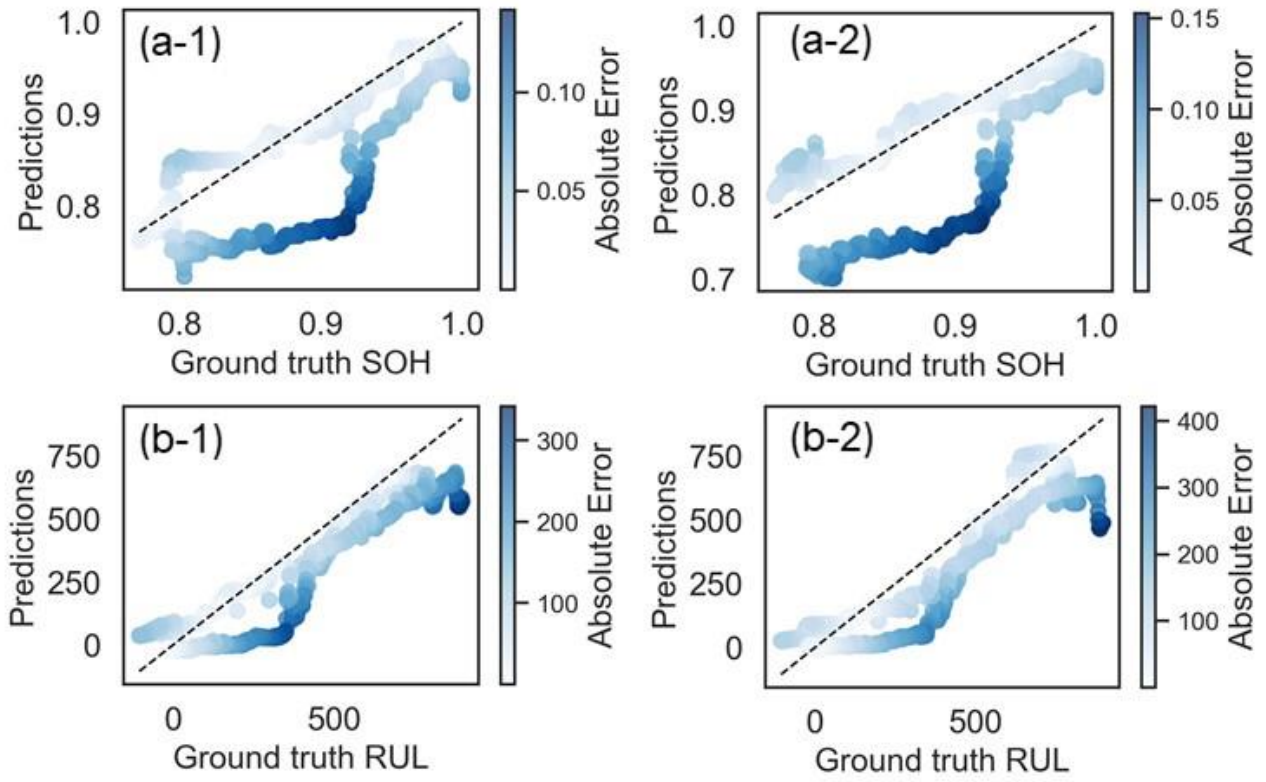

Figure S8: The worst-case prediction results with (a-1, b-1) or without physical features (a-2, b-2) during the 100 times Monte Carlo train-test splits, where train ratio is 0.8. (a-1) the measured SOH and (b-1) RUL against ground truth with design parameters. (a-2) the measured SOH and (b-2) RUL against ground truth without design parameters. The results indicate the SOH errors can exceed 0.15 and RUL can have an error larger than 400EFCs. While including design parameters can improve the performance, especially in higher SOH and RUL ranges. The outlier makes the performance results less representative.

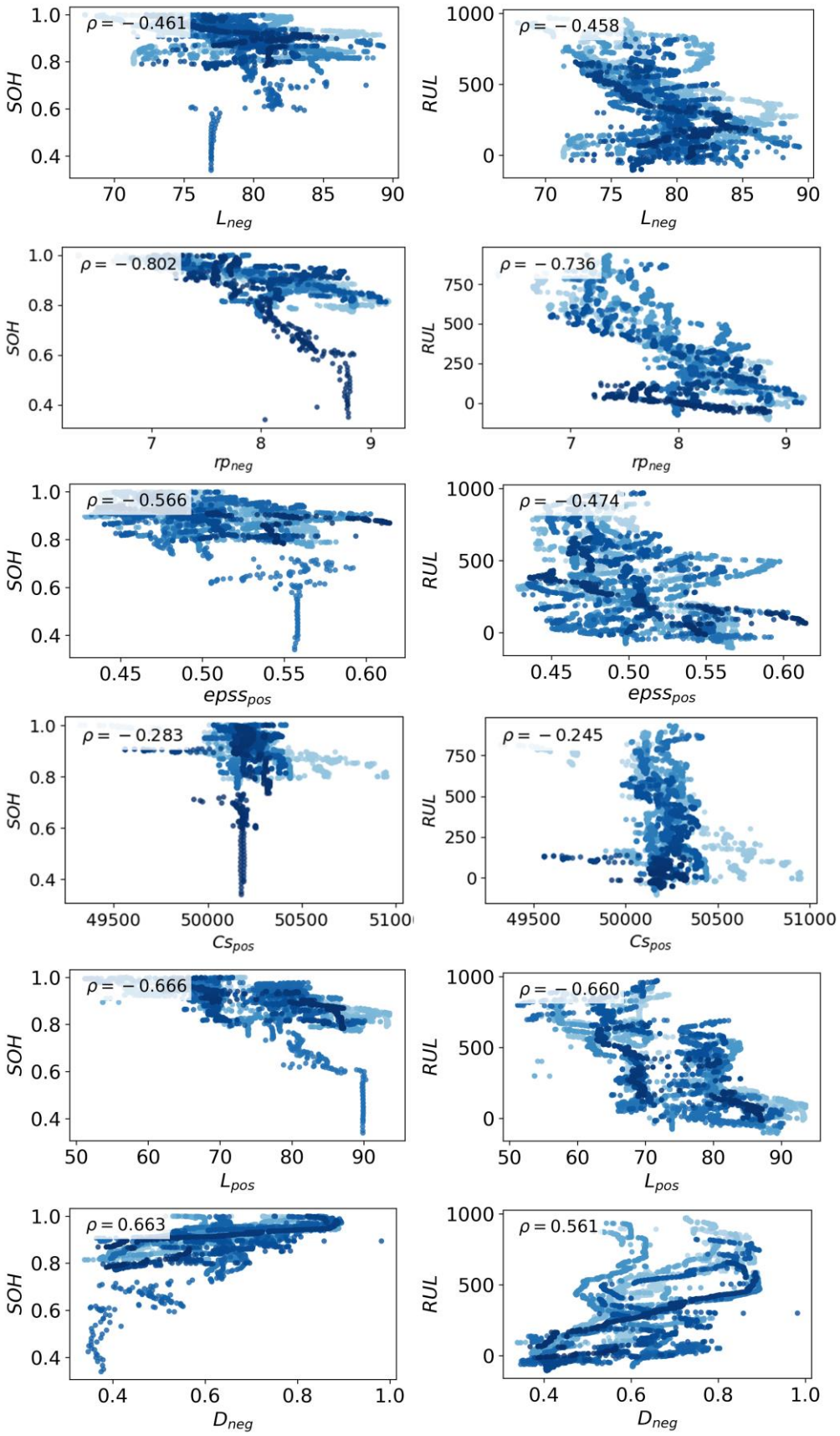

Figure S9. Spearman correlation analysis of sensitive design parameters in the digital twin model.

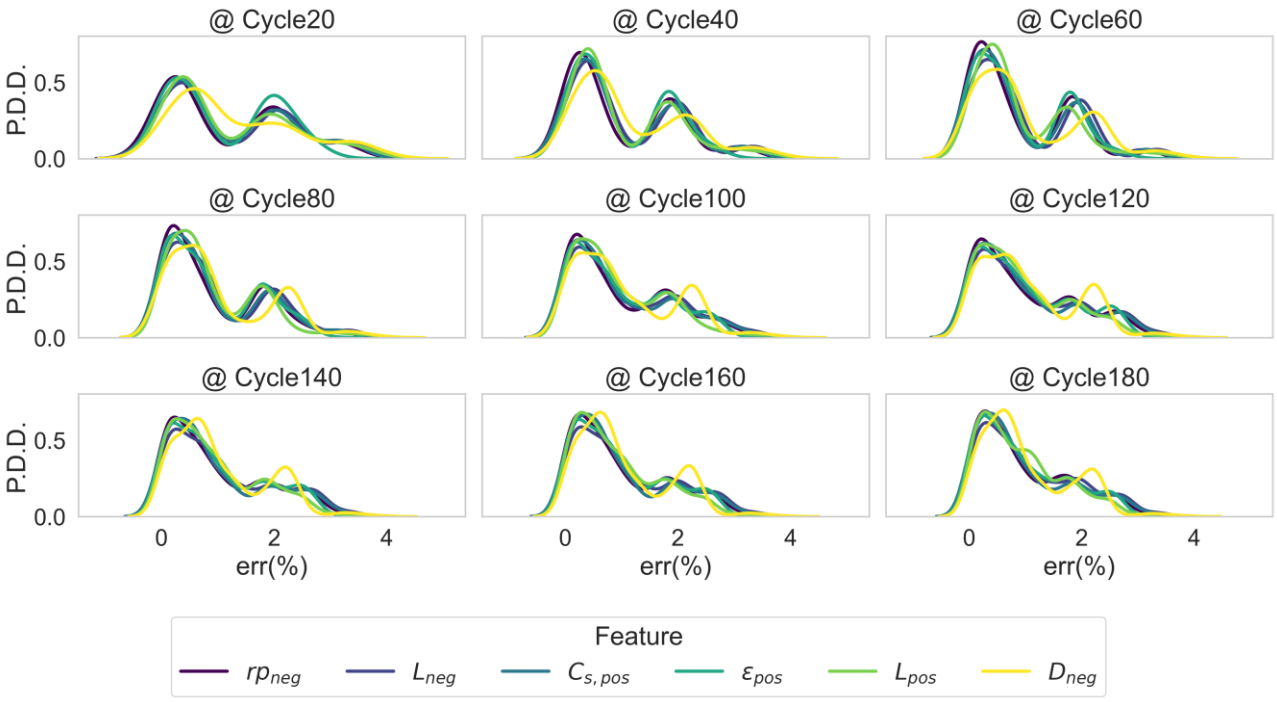

Figure S10. The feature wise comparison of the probability density distribution (PDD) for SOH estimation error rate using each feature when the time of cycle changes.

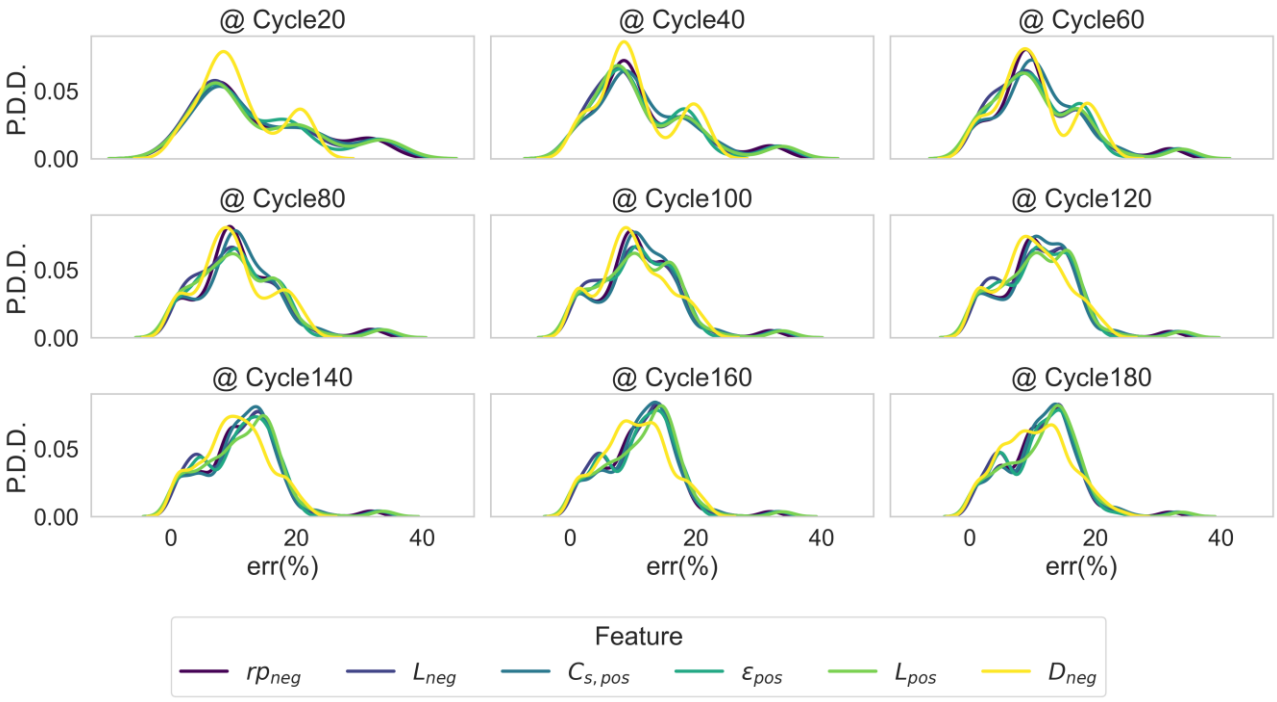

Figure S11. The feature wise comparison of the probability density distribution (PDD) for RUL prediction error rate using each feature when the time of EFCs changes.

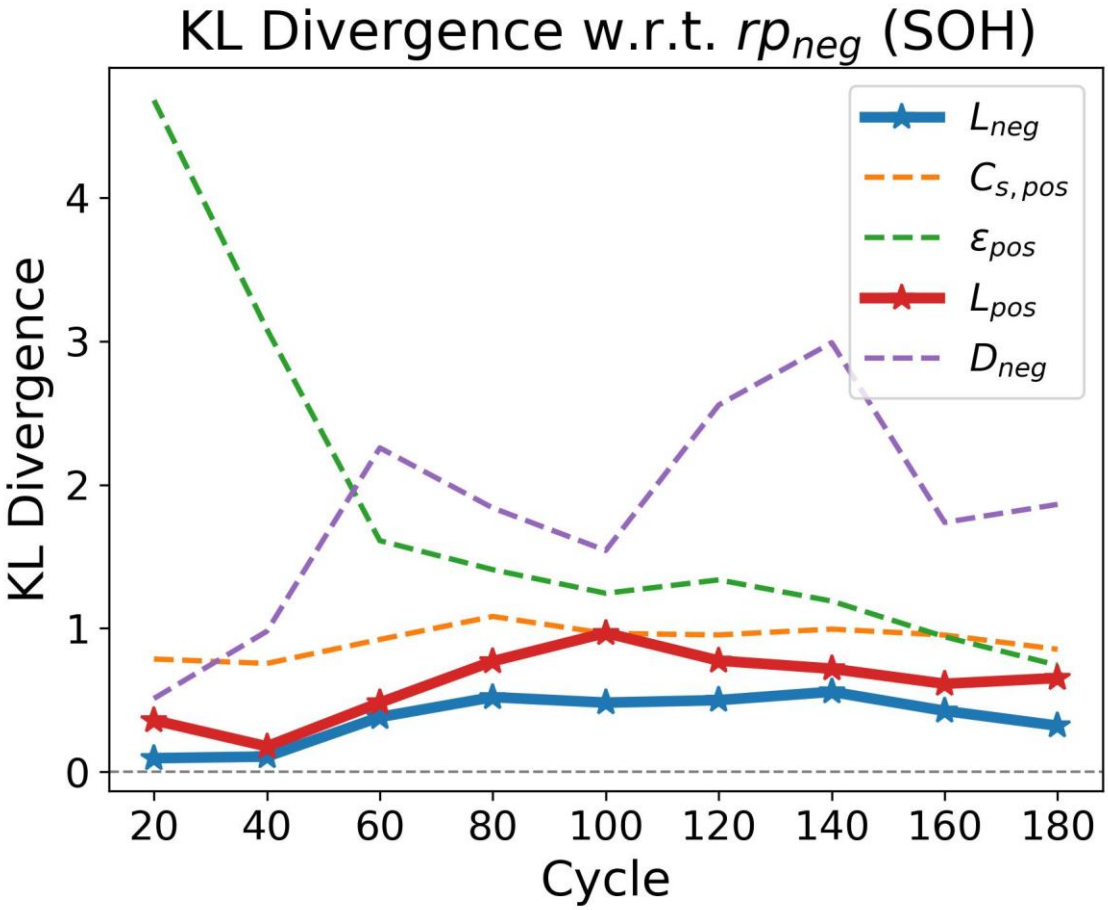

Figure S12. KL divergence between prediction error distributions of each feature and the reference feature rp_{neg} across cycles (SOH estimation task). Lower KL divergence indicates greater substitutability. In the early stage (before 60 cycles), features L_{neg} and L_{pos} show the lowest divergence, suggesting moderate similarity to rp_{neg} in supporting SOH estimation. However, as cycling progresses, their KL divergence increases steadily--indicating that these features become less effective substitutes for rp_{neg} over time. Conversely, features such as ε_{pos}, D_{neg}, and $C_{s,pos}$, maintain higher divergence values throughout, reinforcing the unique and irreplaceable role of rp_{neg} in long-term SOH estimation.

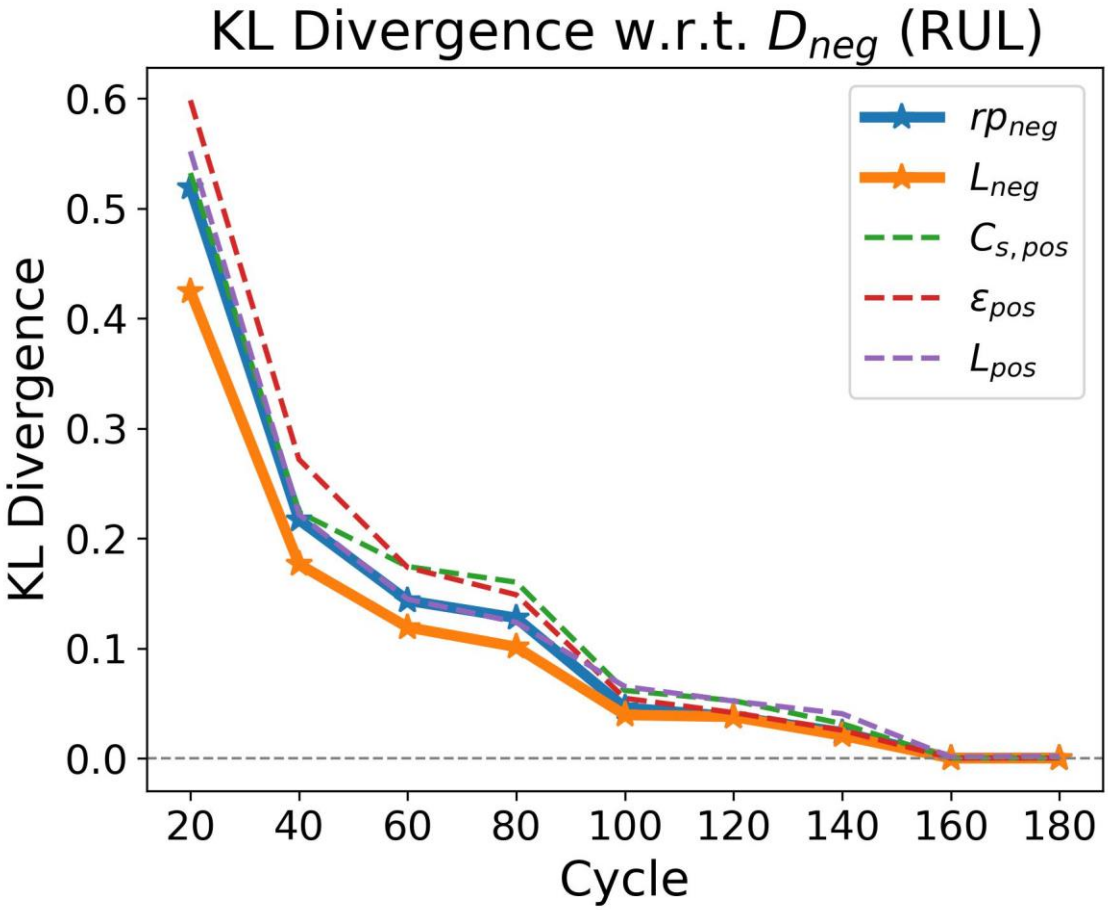

Figure S13. KL divergence between prediction error distributions of each feature and the reference feature D_{neg} across cycles (RUL prediction task). A higher KL divergence indicates less similarity in predictive behavior. In the early life stage (before 100 cycles), D_{neg} remains distinct, suggesting limited substitutability by other features. Over time, KL divergence decreases, indicating growing similarity. Notably, the features rp_{neg} and L_{neg} (marked with stars) exhibit the most consistent similarity to D_{neg}, implying they are the most substantiable design features for RUL prediction.

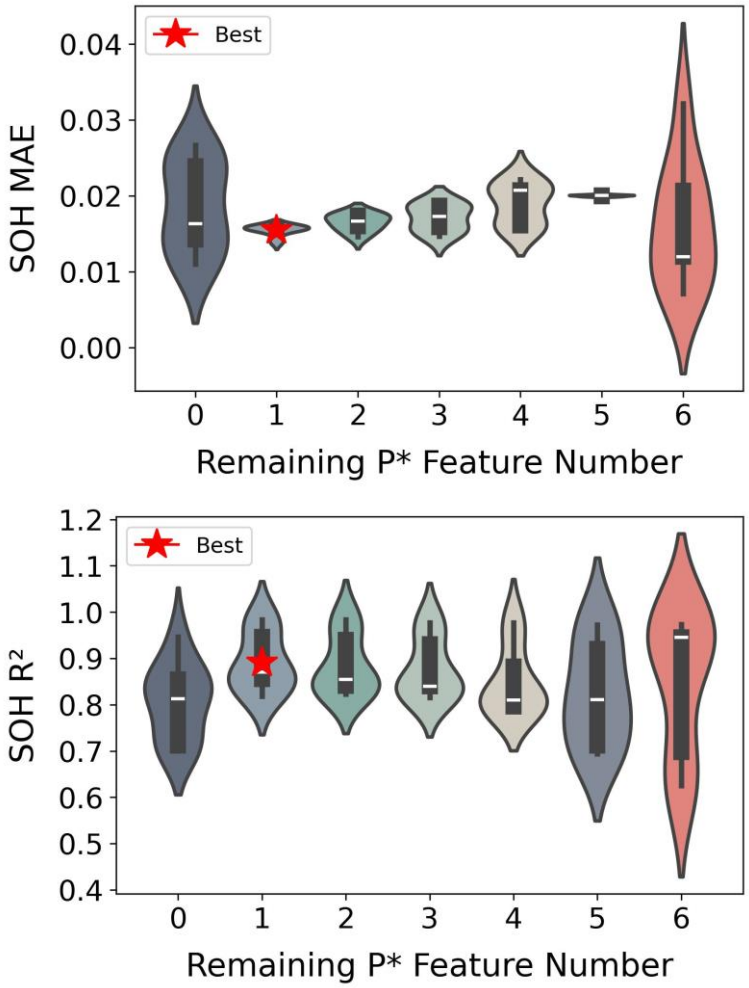

Figure S14. SOH prediction performance (R^2 and MAE) as a function of the number of remaining P* features. Each violin plot shows the distribution of SOH R^2 and MAE scores from multiple feature combinations, corresponding to different counts of retained P* features (from 0 to 6). The red star indicates the best-performing combination. Results highlight that removing certain redundant P* features (at 1 retained features) can yield higher prediction accuracy.

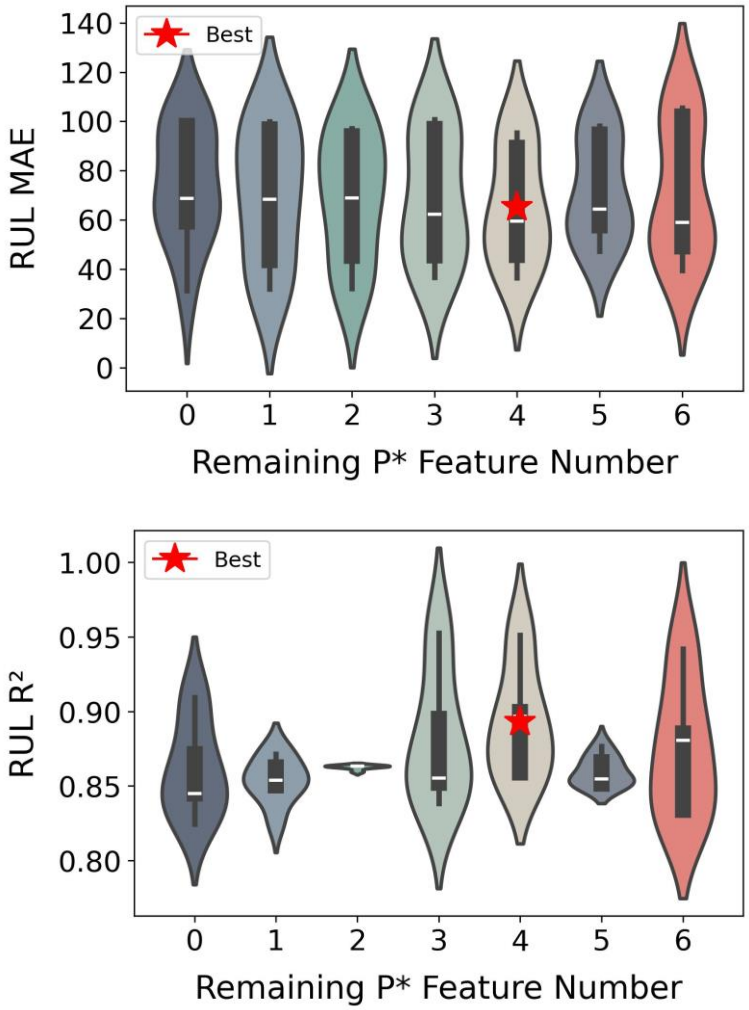

Figure S15. RUL prediction performance (R^2 and MAE) as a function of the number of remaining P* features. Each violin plot shows the distribution of RUL R^2 scores and MAE from multiple feature combinations, corresponding to different counts of retained P* features (from 0 to 6). The red star indicates the best-performing combination. Results highlight that removing certain redundant P* features (at 4 retained features) can yield higher prediction accuracy.

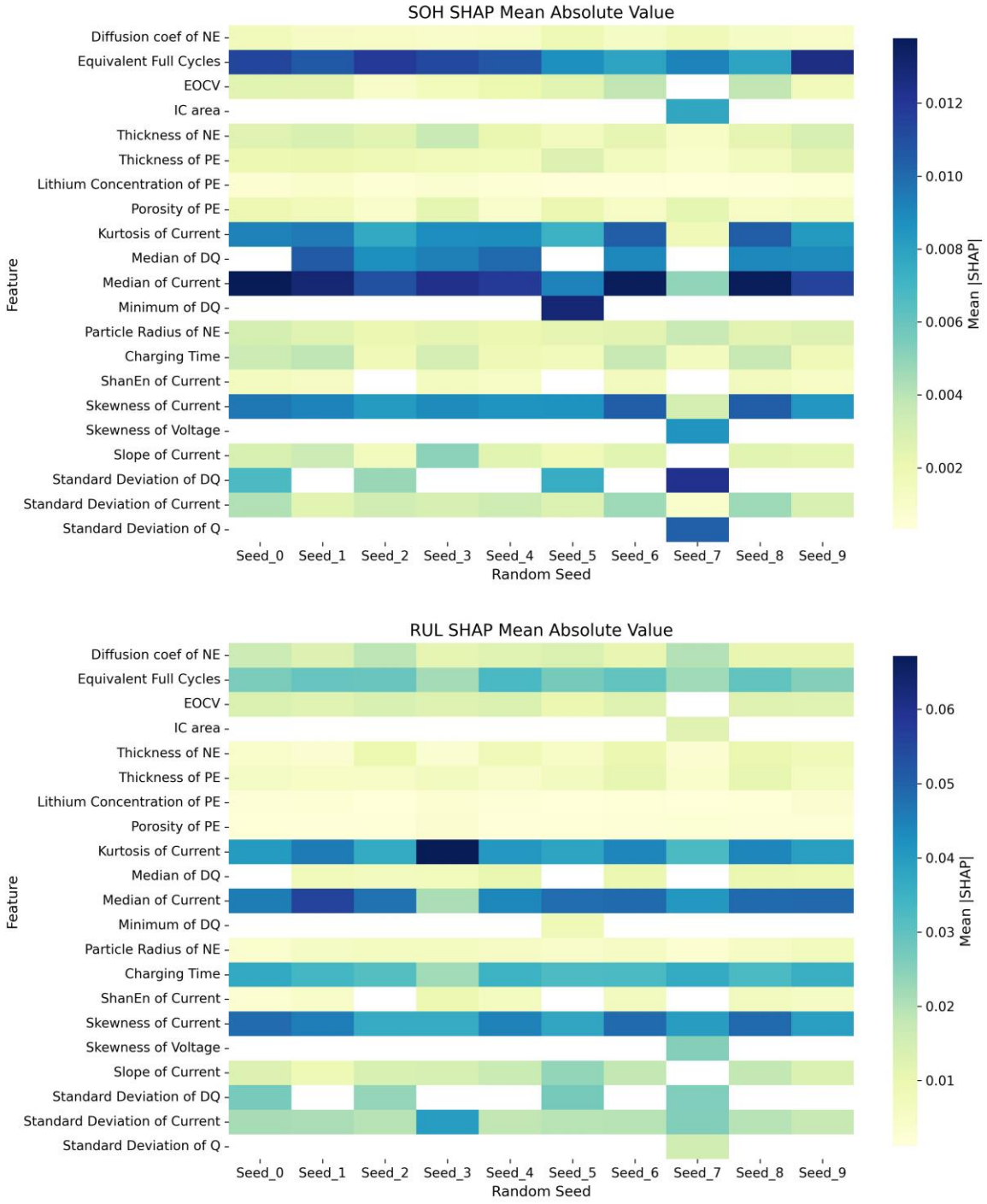

Figure S16. Each feature's impact is aggregated by averaging the absolute SHAP values across all data points over different train-test splits. Feature importance is summarized as one row per model (SOH and RUL). Darker blue indicates higher feature impact. Notably, among the design parameters, the particle radius of the negative electrode and its diffusion coefficient emerge as key indicators for SOH estimation and RUL prediction, respectively.

Supplementary Note 3. Machine learning models

Our goal of this machine learning pipeline is not to develop an accurate model but rather to build a model that fits all the data well enough such that the importance of underlying design parameters integration can be analyzed. We selected a regularized linear model along with four advanced models - MLP, SVR and ensemble models such as Random Forest and XGBoost – due to their ability to capture nonlinear relationships, ease of implementation, and short training time. To prevent overfitting and to optimize model hyperparameters (Table S4), a 5-fold cross-validation was carried out systematically using GridSearchCV from Python package Sklearn. An 80-20 split was used to find the best hyperparameters. Because we observe that the prediction accuracy depends on train-test splits, all of the analyses conducted here are repeated over 10 random train-test splits (Figure S17). Most of the discussion is based on the results from the random forest model, which provides better performance.

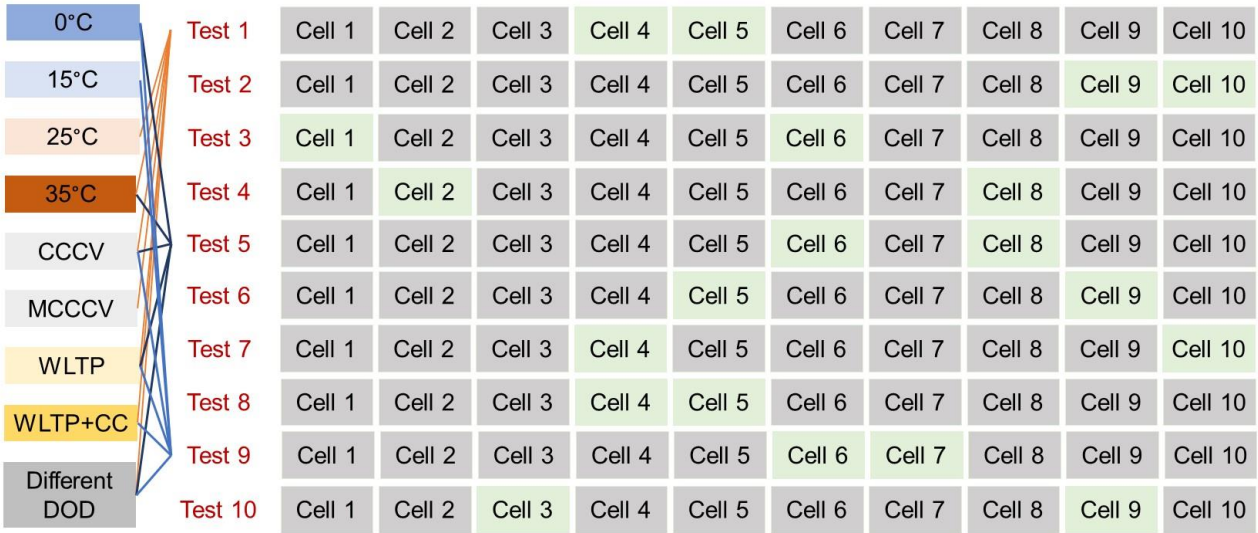

Figure S17. Example of data splitting for model training and testing. Training-to-test ratios range from 0.3 to 0.8; the figure illustrates the test case setup when the training ratio is set to 0.8.

Table S4. Hyperparameters of ML algorithms.

ML algorithm	Hyperparameters	#
RF	Number of trees Maximum depth of each tree Minimum number of samples required to split an internal node Minimum number of samples required to be a leaf node Number of features to consider when looking for the best split Whether bootstrap samples are used when building trees	6
SVR	Regularization parameter Epsilon in the epsilon-insensitive loss function Kernel type Kernel coefficient Degree of the polynomial kernel	5
XGBoost	Number of boosting rounds Maximum tree depth Fraction of training data randomly sampled for each tree L1 regularization term L2 regularization term Minimum loss reduction required to make a further partition Learning rate Feature subsample ratio Minimum sum of instance weight in a child	9
MLP	Number of neurons in hidden layers Activation function Weight optimization algorithm L2 regularization parameter Learning rate schedule for weight updates	5
Linear Regression	Whether to calculate the intercept for the model	1

We next investigated the (Spearman) correlation (ρ) between the features that we used in the ML model development, shown in Figure S18. It can be found some manual engineered features are highly correlated (Figure S18(a)), which means they influence the prediction output simultaneously.

If considering adding 6 design parameters from the digital twin model, in this way, to remove the redundancy highly correlated features while inject physical insights.

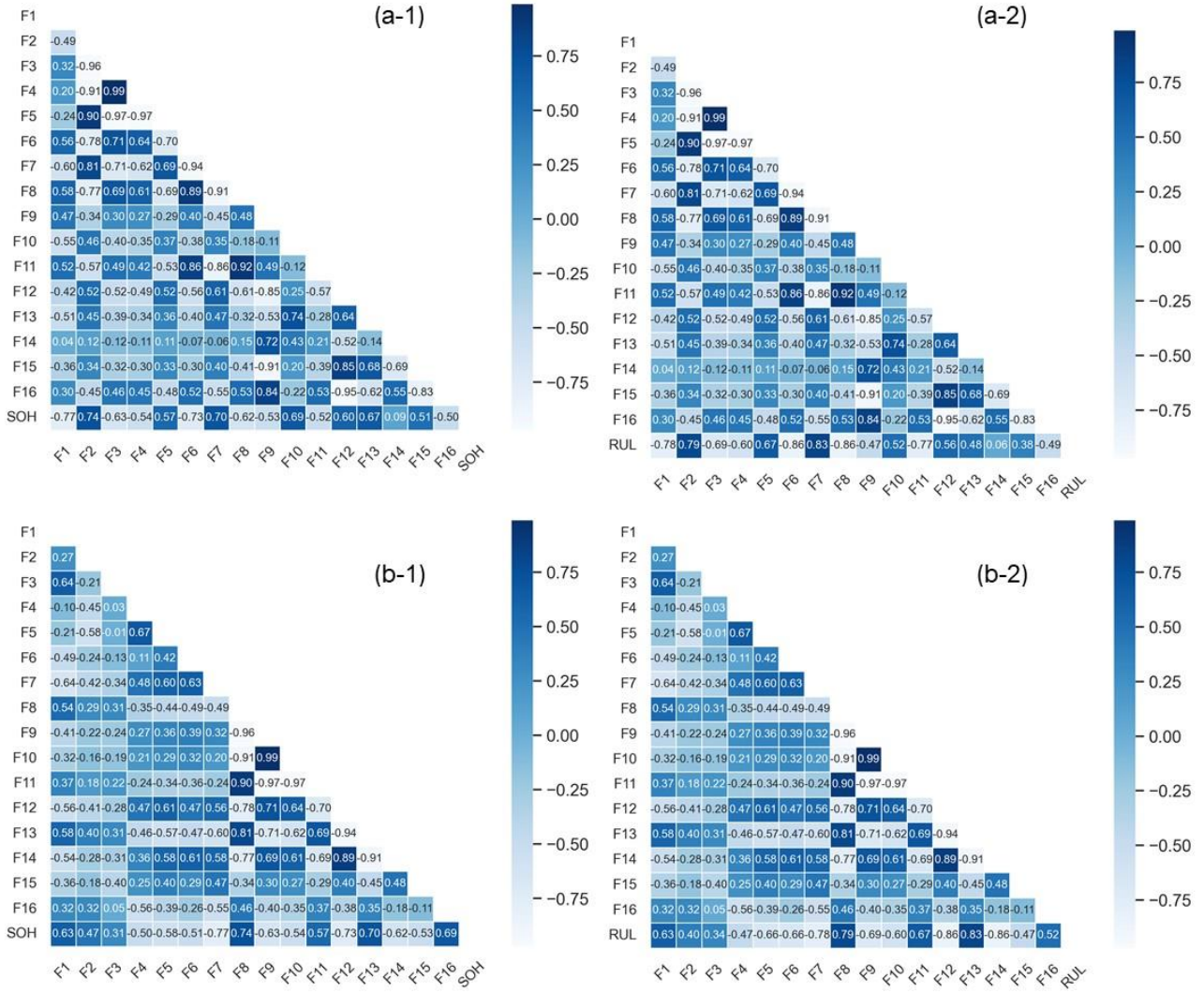

Figure S18. Feature–SOH&RUL Correlation Heatmaps. (a) Correlation with manually extracted features (excluding underlying design parameters). (b) Correlation between SOH&RUL and the full feature set, including six learned design parameters from the digital twin model and ten manually extracted features.

To select features most relevant to the prediction tasks, we adopt a global feature selection strategy based on the sum of absolute Spearman correlation coefficients. Let x_i denote the ith feature, and y_{SOH}, y_{RUL} denote the two target variables. The relevance score for each feature is defined as:

$$Score(x_i) = |\rho(x_i, y_{SOH})| + |\rho(x_i, y_{RUL})|$$

where $\rho(\bullet)$ denotes the Spearman rank correlation coefficient. The top 16 features with the highest scores are selected for model training.

Figure S19. presents example parity plots from the random forest model using one random train-test split under three conditions: (1) Without P*, using hyperparameters optimized without design parameters; (2) With P*, but still using the same hyperparameters from the non-physical setup; (3) With P*, using newly optimized hyperparameters based on the updated feature set. It can be concluded that simply integrating P*, even without further hyperparameter optimization, improves lifetime prediction.

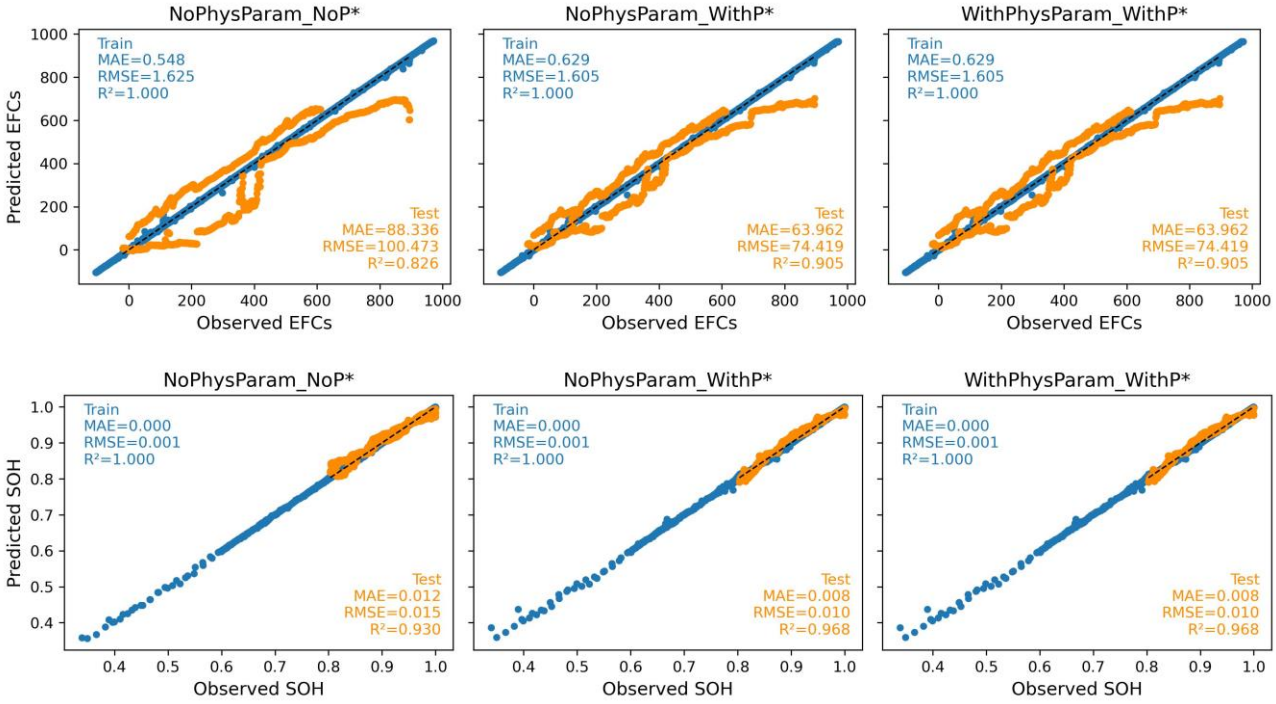

Figure S19. Parity plots for the multitask Random Forest models using features extracted from the [3.65V, 4.10V] window. Results demonstrate that incorporating design parameters—without re-optimizing hyperparameters—already enhances the accuracy of health status prediction.

Table S5. Optimal hyperparameters for multitask random forest model.

Hyperparameter	Value
Number of trees	100
Maximum depth of each tree	20
Minimum number of samples required to split an internal node	15
Minimum number of samples required to be a leaf node	10
Number of features to consider when looking for the best split	sqrt
Whether bootstrap samples are used when building trees	False

To assess the performance of the proposed pipeline, we compare it against different machine learning models, including linear regression, support vector regression (SVR), XGboost, and multilayer Perceptron (MLP). In Figure S20, prediction errors are shown on the left y-axis and computational cost on the right. Results indicate that ensemble methods, such as XGBoost and Random Forest, outperform linear regression, SVR, and MLP in prediction accuracy. Among these models, Random Forest performs better than XGBoost, likely due to its inherent ability to handle multi-output prediction tasks more effectively. Interestingly, incorporating design parameters does not uniformly increase training time. For models like XGBoost and MLP, training time is slightly reduced (XGBoost: 1.37s → 0.61s; MLP: 1.54s → 1.39s), while minor increases are observed for linear regression (0.001s → 0.003s), SVR (1.24s → 1.58s), and Random Forest (1.09s → 1.11s), suggesting the overall computational burden remains minimal.

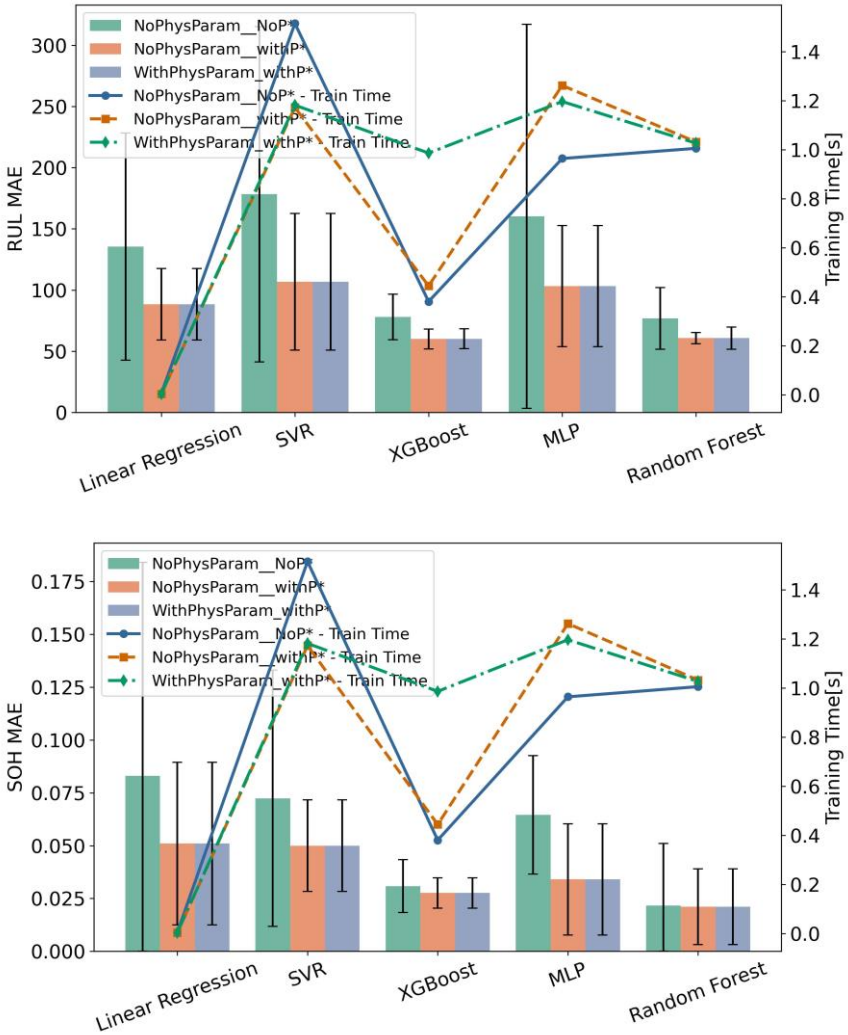

Figure S20. Comparative results of accuracy and computational burden for different ML models, both SOH and RUL prediction.

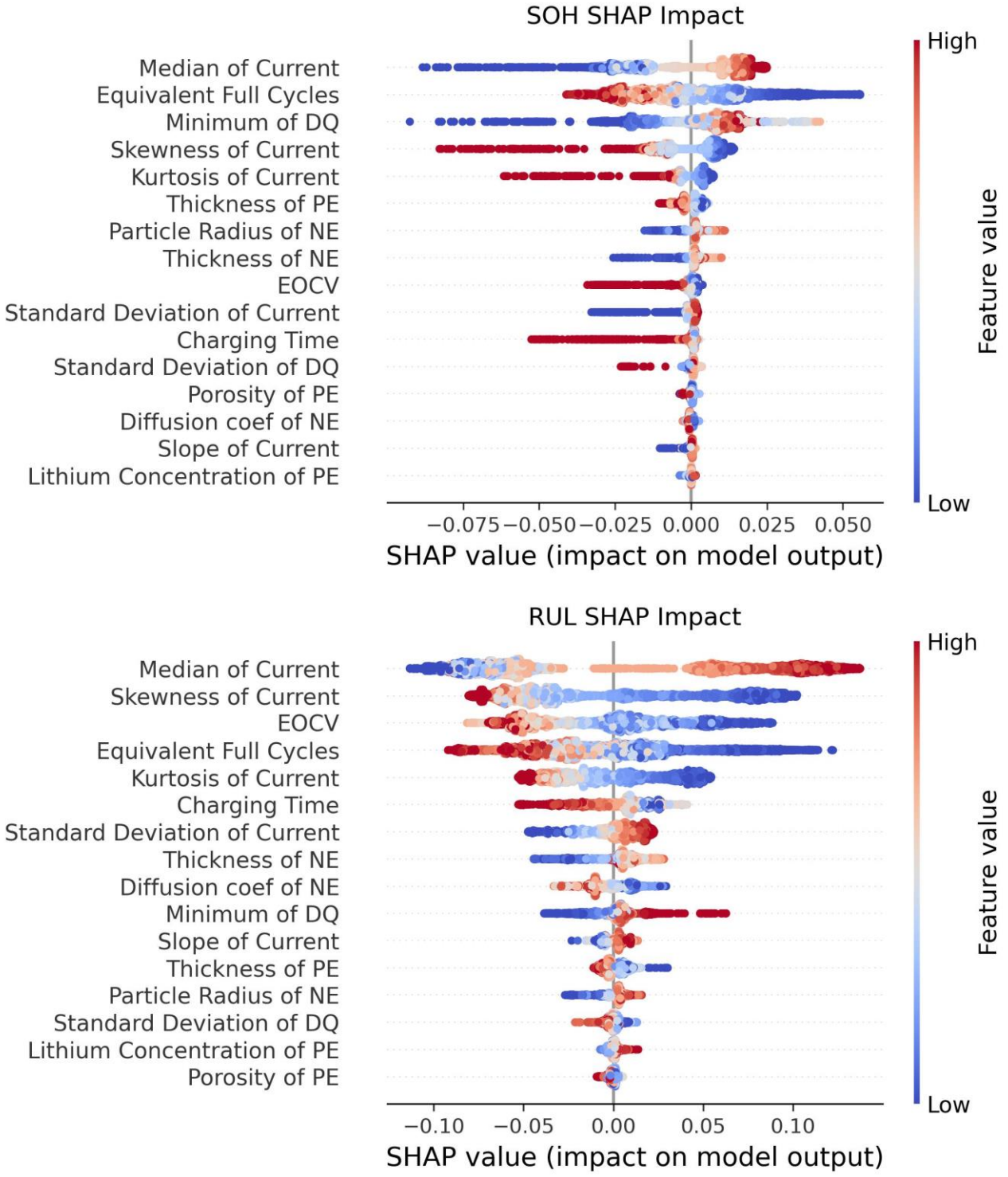

Figure S21. SHAP analysis illustrating feature importance and model interpretability for both SOH and RUL prediction using the multitask Random Forest model on the final outputs. Panels (a) and (b) show the impact of each input feature on the model outputs for SOH and RUL, respectively, highlighting the contributions of both usage-derived indicators and inferred design parameters.

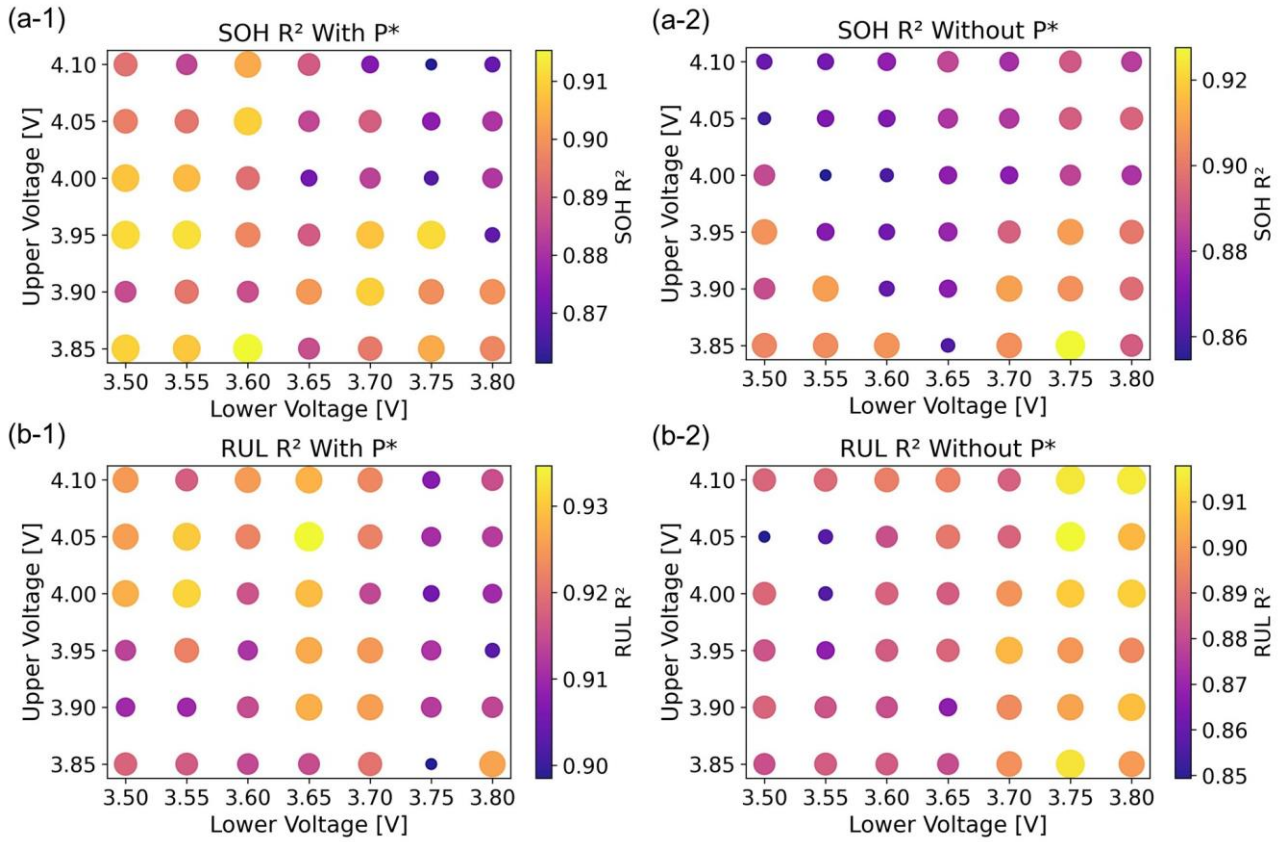

Figure S22. Validation of the proposed machine learning pipeline across different charging voltage windows, using a random train-test split (80% training, 20% testing). (a) SOH estimation R^2: (a-1) with all P* and (a-2) without P*. (b) RUL prediction R^2: (b-1) with all P* and (b-2) without P*.

Supplementary Note 3. Digital twin assistance in parameter uncertainty quantification

Figure S23 presents the uncertainty quantification of the predicted capacity for six design parameters under 2C CCCV fast charging conditions, based on the integrated model in COMSOL 6.3. The particle radius (rp_{neg}), solid-state diffusion coefficient (D_{neg}), and thickness (L_{neg}) of the negative electrode exhibit broad yet smooth distributions in predicted capacity, indicating a strong and relatively stable influence on model outputs. Parameters such as the solid-phase lithium concentration ($C_{s,pos}$) and porosity ($epss_{pos}$) of the positive electrode display narrower distributions, indicating a relatively lower influence on capacity variability. In contrast, the positive electrode thickness (L_{pos}) shows a multimodal distribution, reflecting potential nonlinearities or regime-dependent effects that may increase variability in long-term degradation modelling (i.e., RUL prediction).

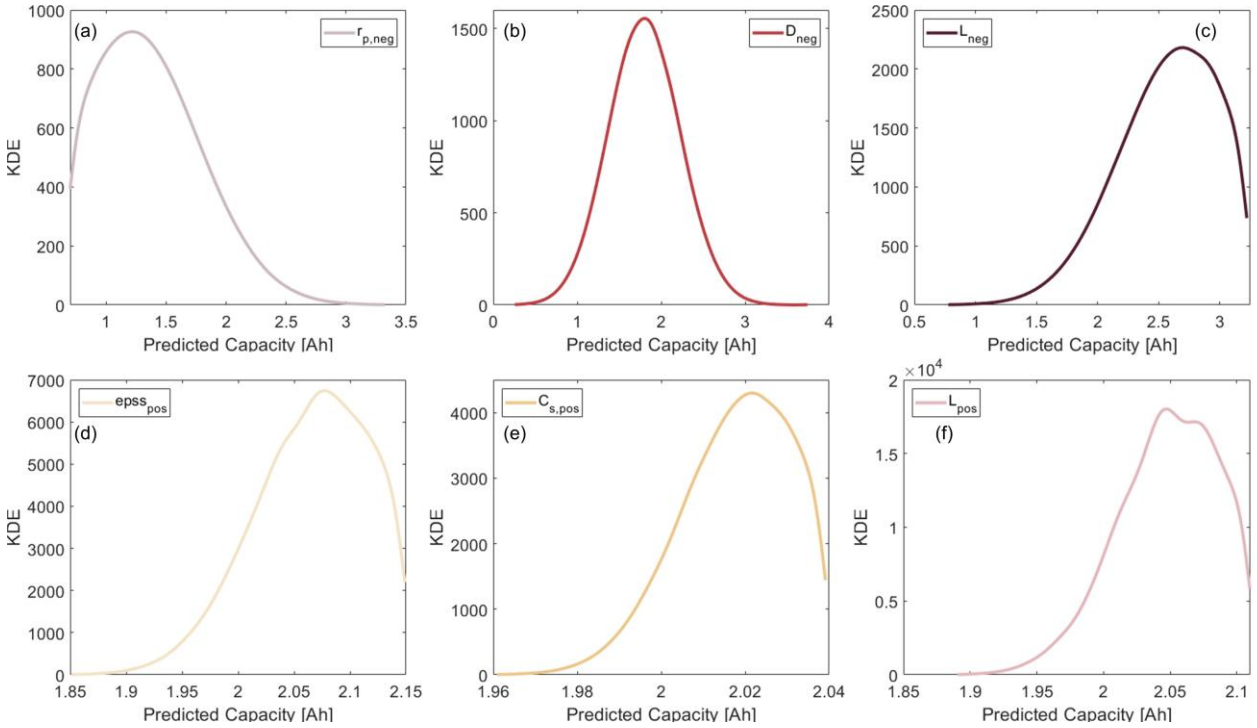

Figure S23. Uncertainty quantification of capacity variability under 2C charge/discharge conditions using a Multiphysics digital twin model[1] , evaluated across six key cell design parameters.

Table S6. Recent works using partial charging/discharging data to predict battery health.

Ref.	Feature numbers	Metrics	Dataset
Ref[3]	Depends on sampling Δt (charging)	RMSE≈2%-3%	D1,D2
Ref[4]	6 (discharging)	MAE≤1.5%	D3,D4,D5,D6
Ref[5]	17(discharging)	RMSE≈1%	D3
Ref[6]	16(charging)	MAPE=0.87%	D3,D7,D8,D9
Ref[7]	15(charging)	RMSE≤2.99%	D10
Ref[8]	16(charging/discharging)	MAE≤1.17%	D10,D11
Ref[9]	2(feature sequences) (charging)	MAPE≤2%	D1,D2,D12,D13
Ref[10]	21(pulse injection) (charging/discharging)	MAPE≤7.2%	D14

D1. https://doi.org/10.5287/bodleian:KO2kdmYGg

D2. https://phm-datasets.s3.amazonaws.com/NASA/5.+Battery+Data+Set.zip

D3. https://data.matr.io/1/projects/5c48dd2bc625d700019f3204

D4. https:// data.matr.io/1/projects/5d80e633f405260001c0b60a

D5,D6. Custom-designed test on LFP prismatic cells (none open source)

D7. https://zenodo.org/record/6405084

D8. https://doi.org/10.5281/zenodo.10963339

D9. https://data.mendeley.com/datasets/nsc7hnsg4s/2

D10. https://data.mendeley.com/datasets/yz4pttm73n/2

D11. Custom-designed test (none open source)

D12. https://doi.org/10.5281/zenodo.6379165

D13. Custom-designed test (none open source)

D14. https://doi.org/10.5281/zenodo.10715209

Supplementary Discussions

Beyond the role of the proposed physics-informed ML framework in linking early battery design and manufacturing with real-world applications—such as fast charging and EV operation—it also holds strong potential for use in recycling management.

Manufacturing stage: By analyzing early testing data (e.g., formation cycles or short charging profiles), the framework can infer the initial RUL (equivalent to lifetime) and detect trends in design parameter shifts. This enables early identification of potential design flaws or manufacturing inconsistencies, allowing defective batteries to be screened and recycled early, thereby enhancing manufacturing quality control.

Retirement stage: The inferred design and health parameters can guide the sorting strategy for retired batteries. For example, cells with well-preserved active materials—indicated by parameters such as particle radius, solid-state diffusion coefficient, and porosity—can be prioritized for second-life applications, while severely degraded cells can be directed to material recycling. This improves both the efficiency and economic value of resource recovery.

Compared to existing methods such as Tao et al.[11,12], this framework similarly does not rely on complete historical data and requires only charging segments for inference. Its key advantage is that it avoids costly and time-consuming investigations into design parameters and underlying electrochemical properties, such as electrode structure, solid-state diffusion coefficient, and lithium-ion concentration, while still capturing meaningful electrochemical trends. Explainable ML enables the extraction of physical insights from synthetic data, providing both high data efficiency and strong interpretability. With retraining and parameter updates across different battery systems, the framework shows potential as a closed-loop, intelligent tool for battery lifecycle management.

Reference

1 W. Guo, Z. Sun, J. Guo, Y. Li, S. B. Vilsen and D. I. Stroe, *Adv Energy Mater*, 2024, **14**, 2401644.

2 W. Guo, Z. Sun, Y. Li, S. Jin, S. B. Vilsen and D. I. Stroe, *IEEE*, 2024, preprint, DOI: 10.1109/IPEMC-ECCEAsia60879.2024.10567562.

3 R. R. Richardson, C. R. Birkl, M. A. Osborne and D. A. Howey, *IEEE Trans Industr Inform*, 2019, **15**, 127–138.

4 Y. Che, Y. Zheng, Y. Wu, X. Sui, P. Bharadwaj, D. I. Stroe, Y. Yang, X. Hu and R. Teodorescu, *Appl Energy*, DOI:10.1016/j.apenergy.2022.119663.

5 C. Zhao, P. B. Andersen, C. Træholt and S. Hashemi, *J Energy Storage*, 2023, **65**, 107151.

6 F. Wang, Z. Zhai, Z. Zhao, Y. Di and X. Chen, *Nat Commun*, DOI:10.1038/s41467-024-48779-z.

7 X. Li, D. Yu, S. B. Vilsen and D. I. Stroe, *Journal of Energy Chemistry*, 2024, **92**, 591–604.

8 S. B. Vilsen and D. I. Stroe, *Cell Rep Phys Sci*, DOI:10.1016/j.xcrp.2025.102646.

9 C. Lin, J. Xu, D. Jiang, J. Hou, Y. Liang, X. Zhang, E. Li and X. Mei, *Journal of Energy Chemistry*, 2024, **88**, 409–420.

10 S. Tao, R. Guo, J. Lee, S. Moura, L. Canals Casals, S. Jiang, J. Shi, S. J. Harris, T. Zhang, C.-Y. Chung, G. Zhou, J. Tian and X. Zhang, *Energy Environ Sci*, DOI:10.1039/d5ee02217g.

11 S. Tao, M. Zhang, Z. Zhao, H. Li, R. Ma, Y. Che, X. Sun, L. Su, C. Sun, X. Chen, H. Chang, S. Zhou, Z. Li, H. Lin, Y. Liu, W. Yu, Z. Xu, H. Hao, S. Moura, X. Zhang, Y. Li, X. Hu and G. Zhou, *Energy Environ Sci*, DOI:10.1039/d4ee03839h.

12 S. Tao, R. Ma, Z. Zhao, G. Ma, L. Su, H. Chang, Y. Chen, H. Liu, Z. Liang, T. Cao, H. Ji, Z. Han, M. Lu, H. Yang, Z. Wen, J. Yao, R. Yu, G. Wei, Y. Li, X. Zhang, T. Xu and G. Zhou, *Nature Communications*, DOI:10.1038/s41467-024-54454-0.