Supplementary Information

Data-driven available capacity estimation of lithium-ion batteries based on fragmented charge capacity

Zhen Zhang[1], Xin Gu[1], Yuhao Zhu[1], Teng Wang[1], Yichang Gong[1]& Yunlong Shang[1, *]

[1] School of Control Science and Engineering, Shandong University, Jinan 250061, China

***Corresponding Authors*:** Yunlong Shang (yshang@sdu.edu.cn)

Supplementary Note

Supplementary Note 1. Expression form of fast charging in dataset #4

Multi-stage CC charging is employed to achieve fast charging in dataset #4, and the generalized form of the process can be expressed as follows:

$$C_1\left(Q_1\%\right),\ C_2\left(80\%\right),\ 1\mathrm{C}$$

where C_1 denotes the C-rate in the first stage of CC charging, Q_1% denotes the state of charge (SOC) at the end of the first constant-current charging, and C_2 denotes the C-rate in the second stage of CC charging. The C-rate in the third stage of CC charging is 1C. In addition, there are cases where $C_1 = C_2$ on certain cells.

Supplementary Note 2. Calculation principle of Pearson correlation coefficient

The Pearson correlation coefficient (PCC) is a statistical measure used to assess the strength and direction of the linear relationship between two continuous variables, with values ranging from -1 to 1 [1]. This tool can be effectively utilized as a reference in feature engineering. The PCC is calculated as follows:

$$r = \frac{\sum\left(\left(X_i - X_\mu\right)\cdot\left(Y_i - Y_\mu\right)\right)}{\sqrt{\sum\left(\left(X_i - X_\mu\right)^2\right)\cdot\sum\left(\left(Y_i - Y_\mu\right)^2\right)}}$$

where r denotes the PCC. X_i and Y_i denote the observed values of any two variables. X_μ and Y_μ denote the average values of any two variables.

If $r = 1$, this indicates a completely positive correlation. If $r = -1$, this means a perfectly negative correlation. If $r = 0$, this indicates that there is no linear relationship between the two variables. When $r > 0$, 0-0.2 represents an extremely weak correlation, 0.2-0.4 a weak correlation, 0.4-0.6 a moderate correlation, 0.6-0.8 robust correlation, and 0.8-1.0 extremely robust correlation. When $r < 0$, the case is similar to the above.

Supplementary Note 3. Model fine-tuning strategy

Model fine-tuning in transfer learning (TL) is an essential technique that is typically performed to apply a pre-trained basic model on another similar task [2]. By using relatively small amounts of new data, we aim to refine the machine learning (ML) model and achieve more accurate estimates. The steps of the model fine-tuning strategy applied in this paper are as follows:

(1) Rely on dataset #1 for pre-training the basic model.

(2) Partial extraction of cell data in dataset #2 and dataset #3 that are collected under the same experiment conditions is performed. Specifically, this paper selects 10% of the total data volume as samples with an equal interval extraction method.

(3) Set the extracted samples as input variables and retrain the TL model.

Supplementary Note 4. Standardization process

Standardization is a popular method in data preprocessing, which is applied to transform data into a standard normal distribution [3]. The method contributes to ensuring that data with different features have similar scales to improve the performance of ML algorithms. Standardization is generally applied to continuous numerical features. Furthermore, standardization does not change the data distribution shape, it just re-scales it. The standardization process is as follows:

(1) Iterate through each feature and available capacity in the dataset and calculate the mean X_μ and standard deviation X_σ for all samples. The process is as follows:

$$X_\mu = \frac{1}{n}\sum_{i=1}^{n} X_i$$

$$X_\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(X_i - X_\mu\right)^2}$$

(2) The process of standardization for each feature is implemented as follows:

$$X_{std} = \frac{X_i - X_\mu}{X_\sigma}$$

(3) Proper application of standardized feature data. Replace the original dataset with the standardized dataset and input it into the ML algorithm to achieve training and testing.

Supplementary Note 5. Feature combinations across cycles

In practical engineering applications, there is difficulty in extracting candidates in the same cycle or neighboring cycles to form feature combinations. We propose feature combinations across cycles and apply them to the battery available capacity estimation in this work. This note will describe the above feature extraction process in conjunction with Supplementary Fig. 11. The description is as follows:

Assume that a cell has been operated for a long period (including fragmented charging) such that its complete cycle information is not available. Label the latest cycle as *k*. Candidate feature #1 comes from the current cycle *k*, while candidate feature #2 comes from the previous cycle *k*-*n*. Candidate features #1 and #2 are collocated into a feature combination. *n* represents the spacing between the two cycles.

Supplementary Note 6. Introduction of the LASSO algorithm

The LASSO regression algorithm was first proposed by Robert Tibshirani in 1996, and its full name is least absolute shrinkage and selection operator [4]. The algorithm adds a penalty parameter *s* based on the least squares method to compress the regression coefficients. When a regression coefficient tends to 0, its corresponding feature is eliminated, which achieves the goal of eliminating irrelevant features and streamlining the model. The LASSO algorithm has advantages in dealing with datasets with multicollinearity. Suppose there are a total of n features (independent variables x_1, x_2,…, x_n) and dependent variable y in the dataset, the regression model under the LASSO algorithm can be expressed as:

$$y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \varepsilon$$

where α denotes a constant, β_1 to β_n denote the regression coefficients, and ε denotes the random perturbation term.

This paper represents the m group observations (actual values) after standardization as (x_{i1}, x_{i2},…, x_{in}, y_i), i = 1 ,2, …, m. As a result of the standardization, it is made that x_{ij} is satisfied:

$$\begin{cases} \frac{1}{n}\sum_{i=1}^{m} x_{ij} = 0,\ j = 1,2,\ldots,n \\ \frac{1}{n}\sum_{i=1}^{m} x_{ij}{}^2 = 1,\ j = 1,2,\ldots,n \end{cases}$$

Such that $\beta' = (\beta'_1, \beta'_2 \ldots, \beta'_n)^{\mathrm{T}}$, then the estimated value of the LASSO can be defined as:

$$(\alpha', \beta') = \arg\min\left\{ \sum_{i=1}^{m} \left(y_i - \alpha - \sum_j \beta_j x_{ij} \right)^2 \right\} \ \text{s.t.} \sum_{j=1}^{n} \left| \beta_j \right| \le s$$

where s denotes the penalty parameter and $s \ge 0$.

The decrease in s leads to an overall decrease in the estimated parameters. β'^0_j is the estimate obtained by the least squares method.

$$s_0 = \sum_{j=1}^{n} \left| \beta'^0_j \right|$$

When $s \geq s_0$, the optimal solution in the estimation is the least squares solution. When $s < s_0$, some of the regression coefficients will tend to be 0 or equal to 0, and their corresponding features will be eliminated. The above process is the feature selection by the LASSO algorithm.

Supplementary Note 7. Introduction of the XGBoost algorithm

The extreme gradient boosting (XGBoost) algorithm is an integrated learning algorithm based on gradient boosting decision tree (GBDT) [5]. XGBoost is similar in structure to the GBDT model, and both of them are improved based on decision trees. The strength of the XGBoost is its high predictive performance, robustness, and automatic feature selection, which makes it suitable for classification, regression, and other ML tasks. Compared to linear models, the XGBoost algorithm has an advantage in dealing with nonlinear problems.

The objective function consists of a training loss function and a regularization penalty term. The objective function can be defined as:

$$Obj(\theta) = L(\theta) + \Omega(\theta)$$

where $L(\theta)$ denotes the training loss function, which is used to measure the difference between the predicted values from the model and the actual values from the training data. $\Omega(\theta)$denotes the regularization penalty term, which is used to smooth the weights to control the model complexity.

The complexity of each tree is defined as follows.

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

where T denotes the number of leaf nodes, ω_j denotes the weight of the jth leaf node, λ denotes the coefficient of the regularization penalty term, and γ denotes the descent value of the training loss function.

Through the optimization of the training loss function and the regularization penalty term, the objective function can be finally gained for the XGBoost algorithm as follows:

$$Obj = \sum_{j=1}^{T}\left[G_j w_j + \frac{1}{2}\left(H_j + \lambda\right) w_j^2 \right] + \gamma T$$

where G_j denotes the sum of the first-order partial derivatives in the jth leaf node and H_j denotes the sum of the second-order partial derivatives in the jth leaf node.

Supplementary Note 8. Introduction of the LightGBM algorithm

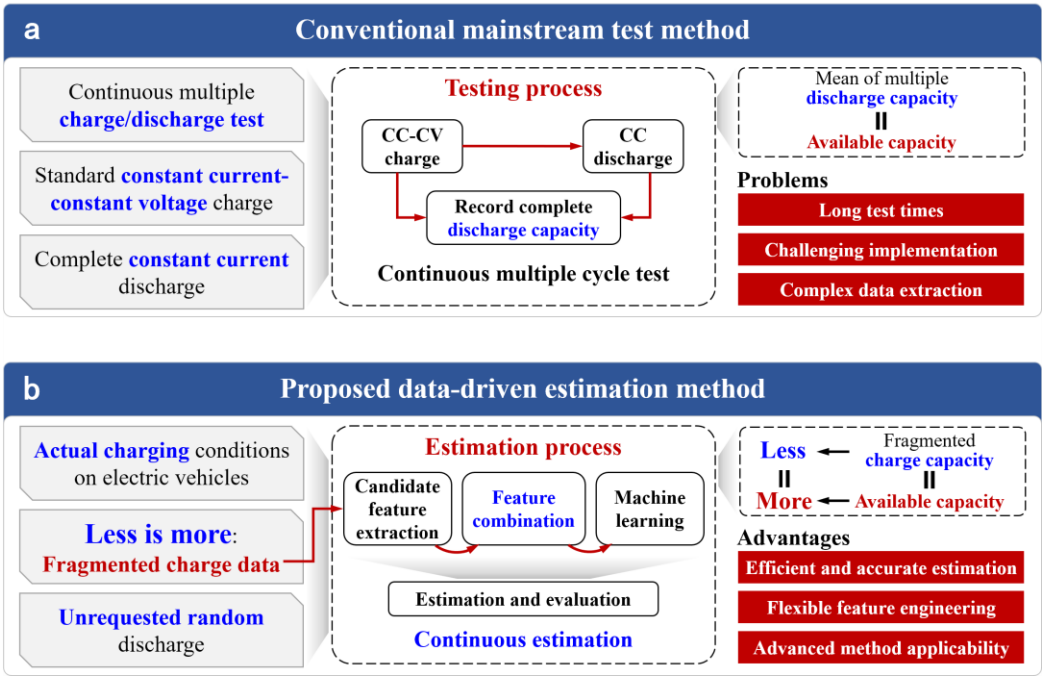
The light gradient boosting machine (LightGBM) algorithm is also based on the GBDT, which has been widely used in various fields for tasks such as prediction, data mining, etc. [6]. LightGBM has the advantages of high training efficiency, low memory occupation, and high estimation accuracy, while it supports parallel training when dealing with large-scale data. Both the XGBoost algorithm and the LightGBM algorithm are variants of the GBDT, which leads to the fact that they are very similar, but there are some crucial distinctions.

Leaf splitting strategy: The XGBoost algorithm uses the level-wise strategy, while the LightGBM algorithm uses the leaf-wise strategy. The former leads to a relatively large width of the tree, while the latter is able to reduce the depth of the tree.

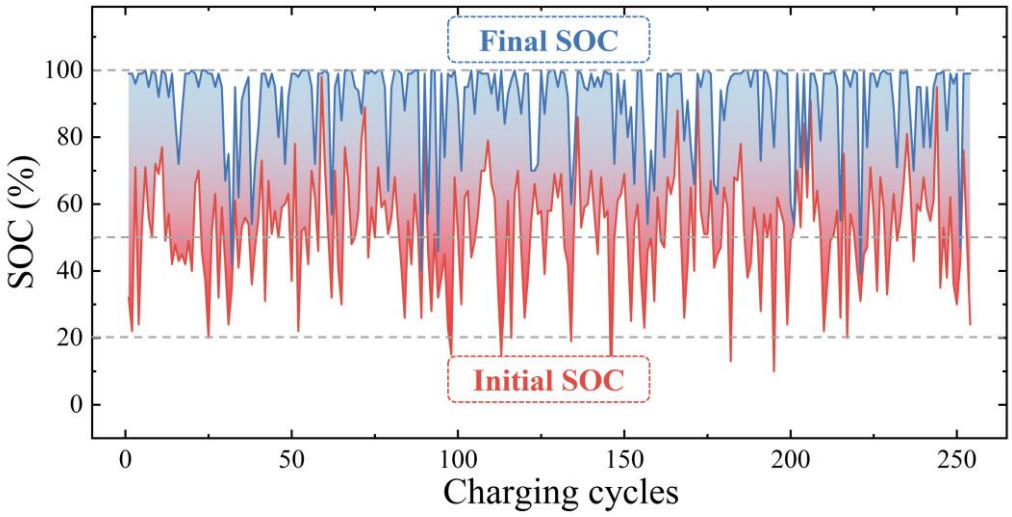
Gradient calculation method: The XGBoost algorithm uses an exact gradient calculation method. The LightGBM algorithm uses a histogram-accelerated gradient calculation method, which reduces computational complexity and memory consumption.

In addition, compared to the XGBoost, the LightGBM algorithm also has advantages in parallel training and hyperparameter optimization.

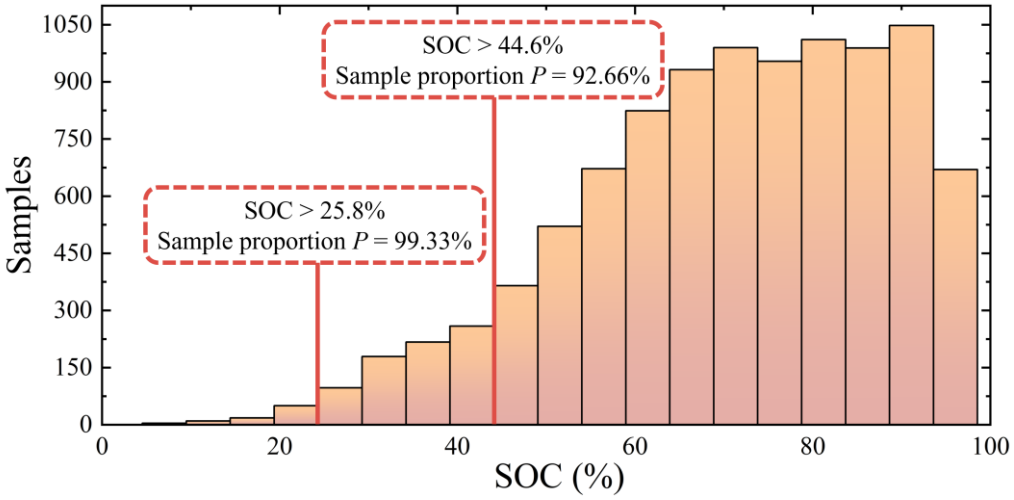
Supplementary Figure



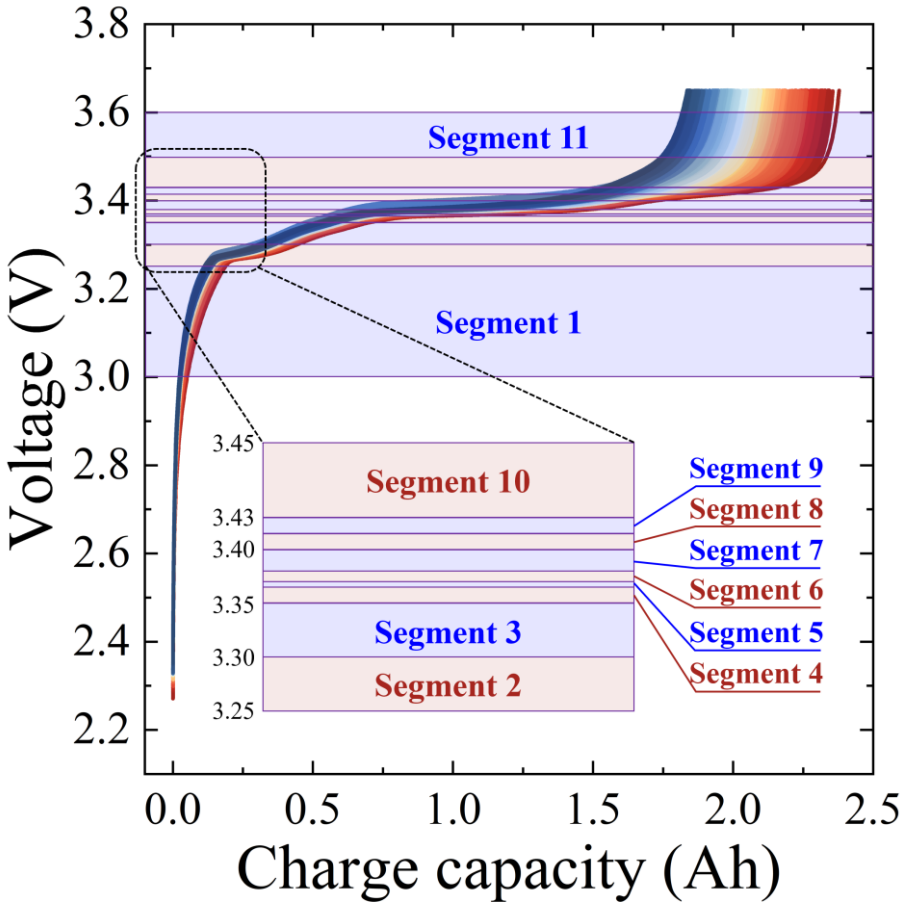
Supplementary Figure 1. Comparison between the conventional mainstream test method and the proposed data-driven estimation method. CC denotes the constant current. CV denotes the constant voltage.



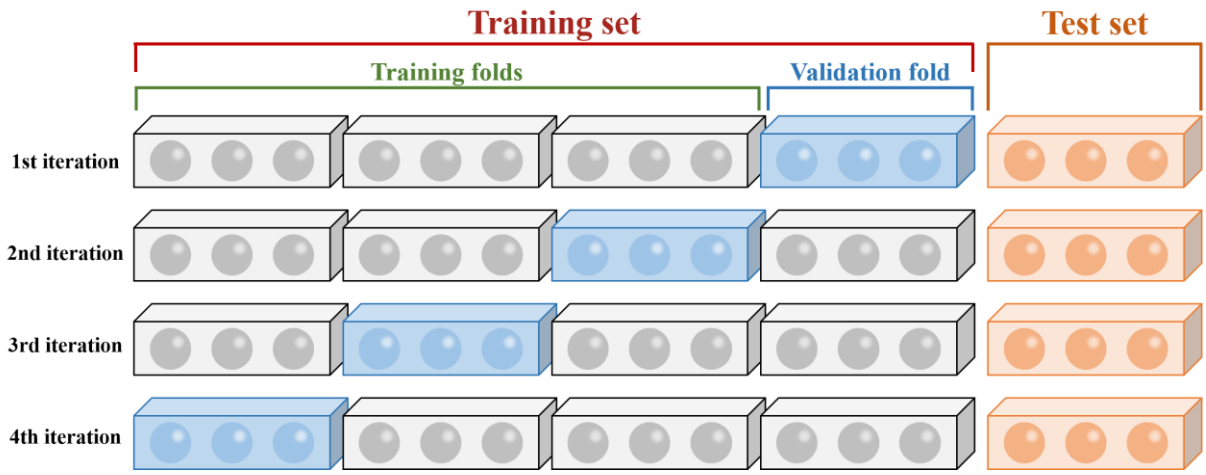
Supplementary Figure 2. Battery state of charge (SOC) distribution for a commercial electric vehicle (EV) with over 250 charges. The gradient from red to blue corresponds to the change in battery SOC during EV charging.



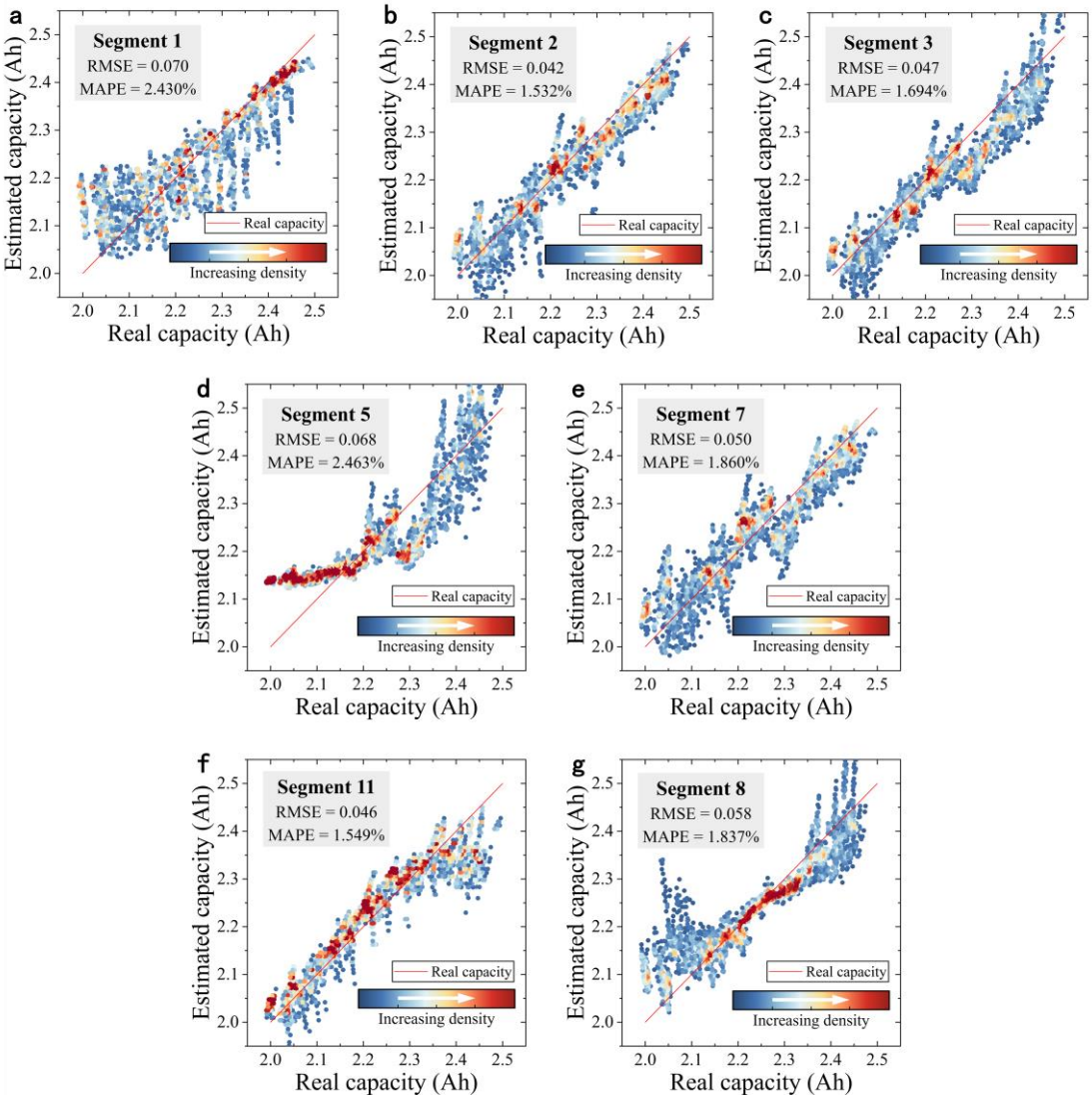
Supplementary Figure 3. Sample data distribution for a commercial EV (electric vehicle) with over 250 charges. SOC denotes the state of charge.



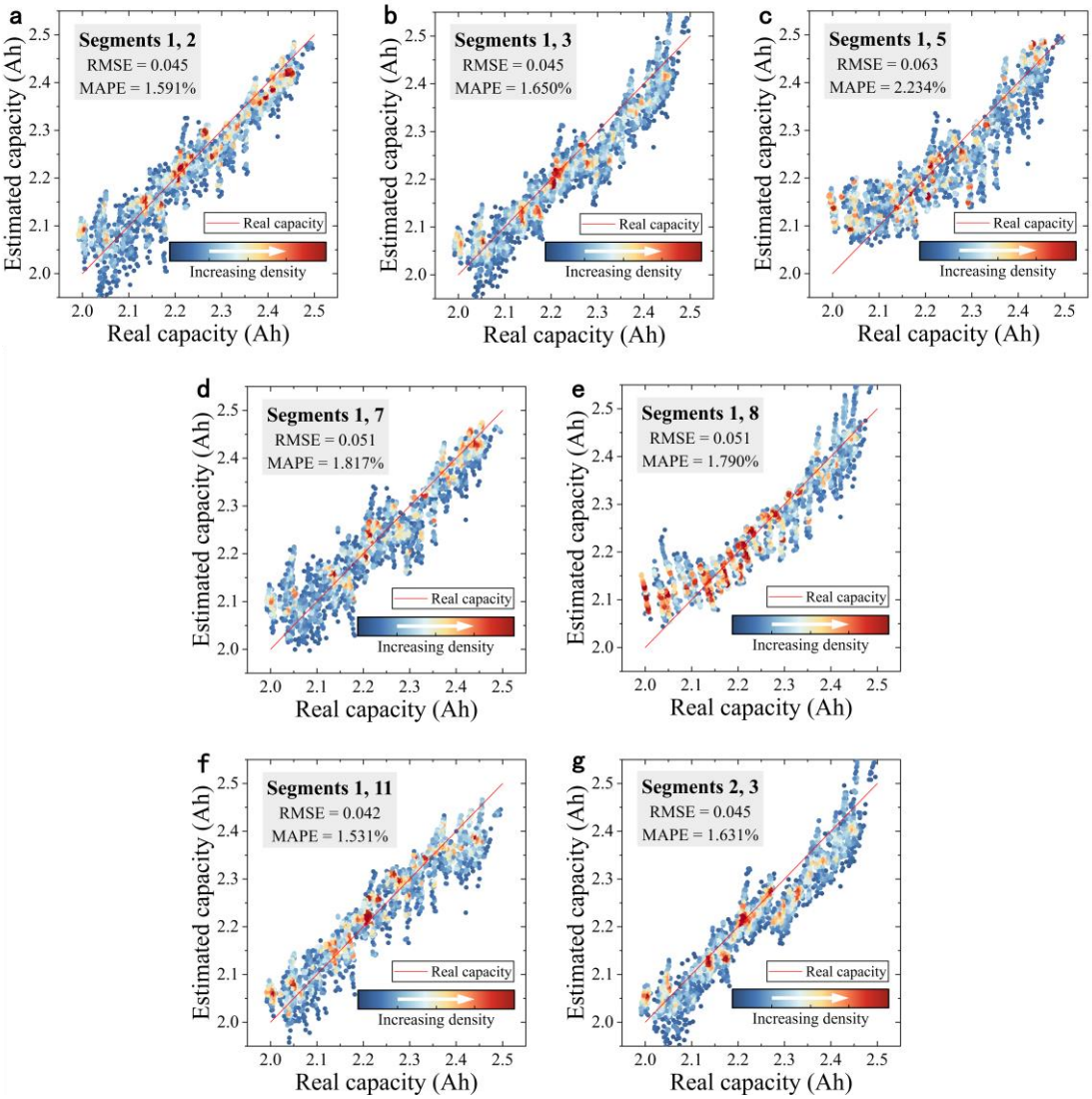
Supplementary Figure 4. 11 segments as candidate features. The gradient correspond to changes in battery SOH (state of health).



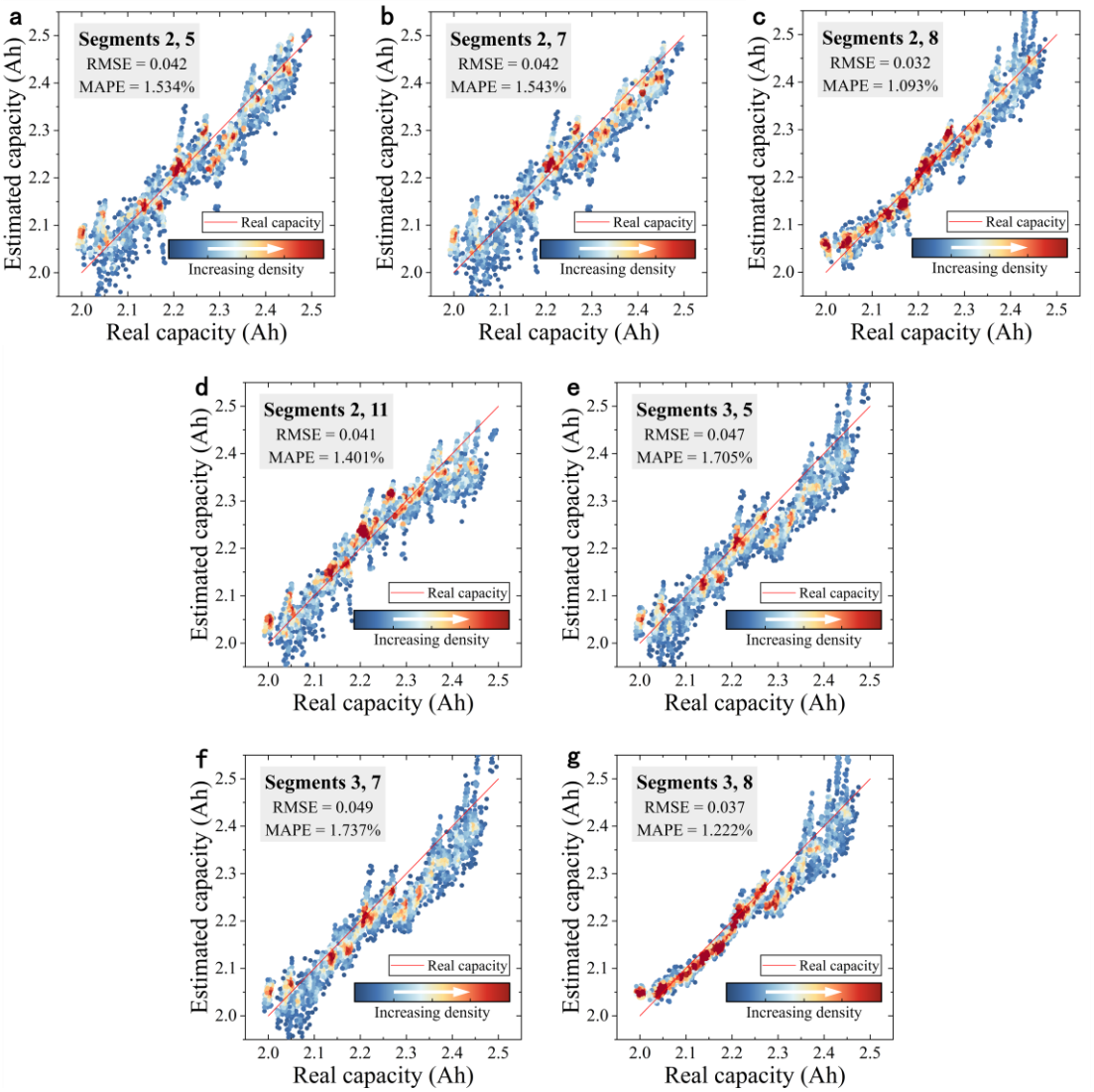
Supplementary Figure 5. Principle of K-fold cross-validation (*K*=4).



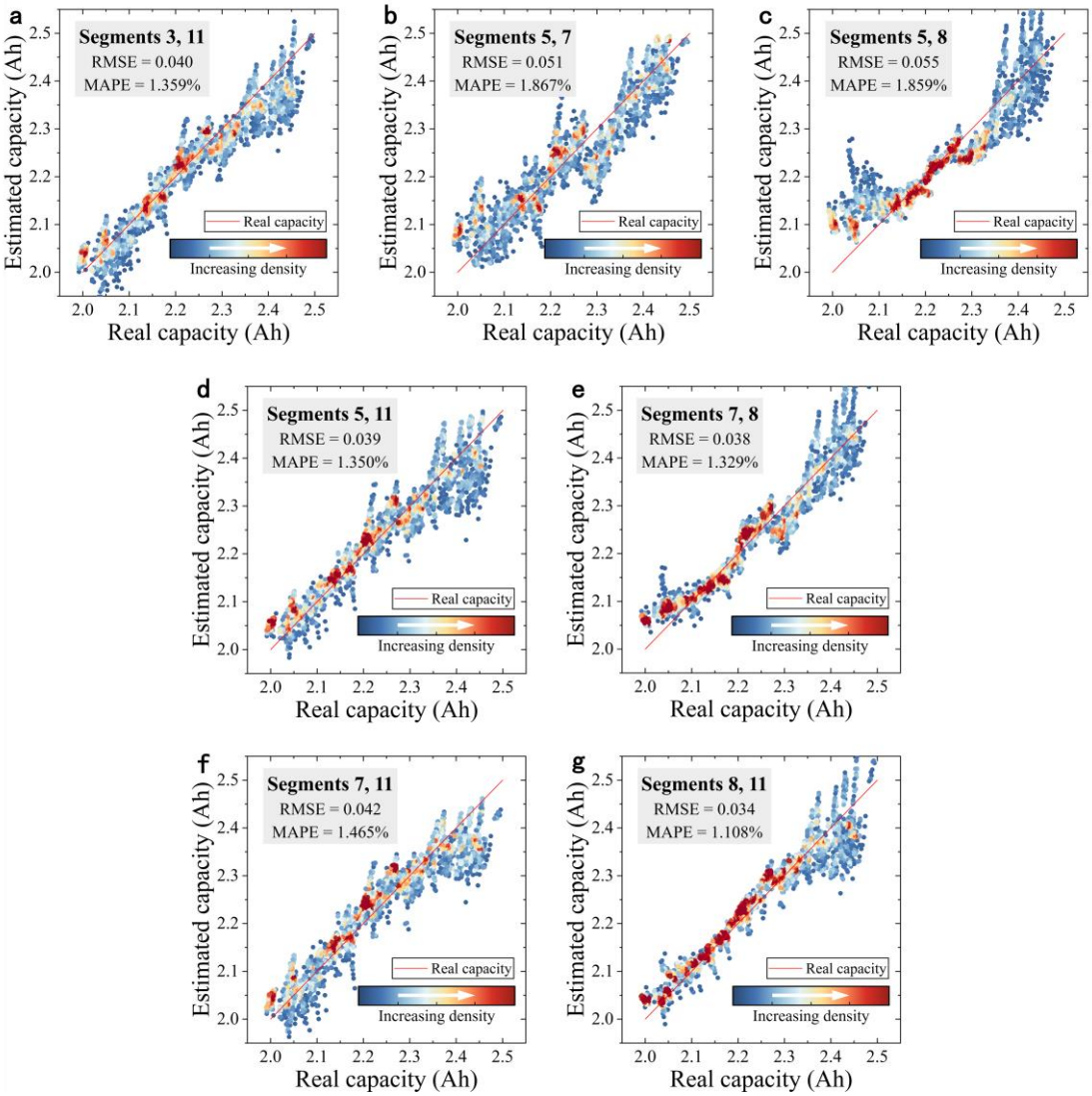
Supplementary Figure 6. Visualization results of the LASSO algorithm for different feature combinations. The available capacity estimation performance is plotted for 7 feature combinations, which include (segment 1) (**a**), (segment 2) (**b**), (segment 3) (**c**), (segment 5) (**d**), (segment 7) (**e**), (segment 11) (**f**), and (segment 8) (**g**). The gradients correspond to varying point-aggregation densities. RMSE denotes root-mean-square error and MAPE denotes mean absolute percentage error.



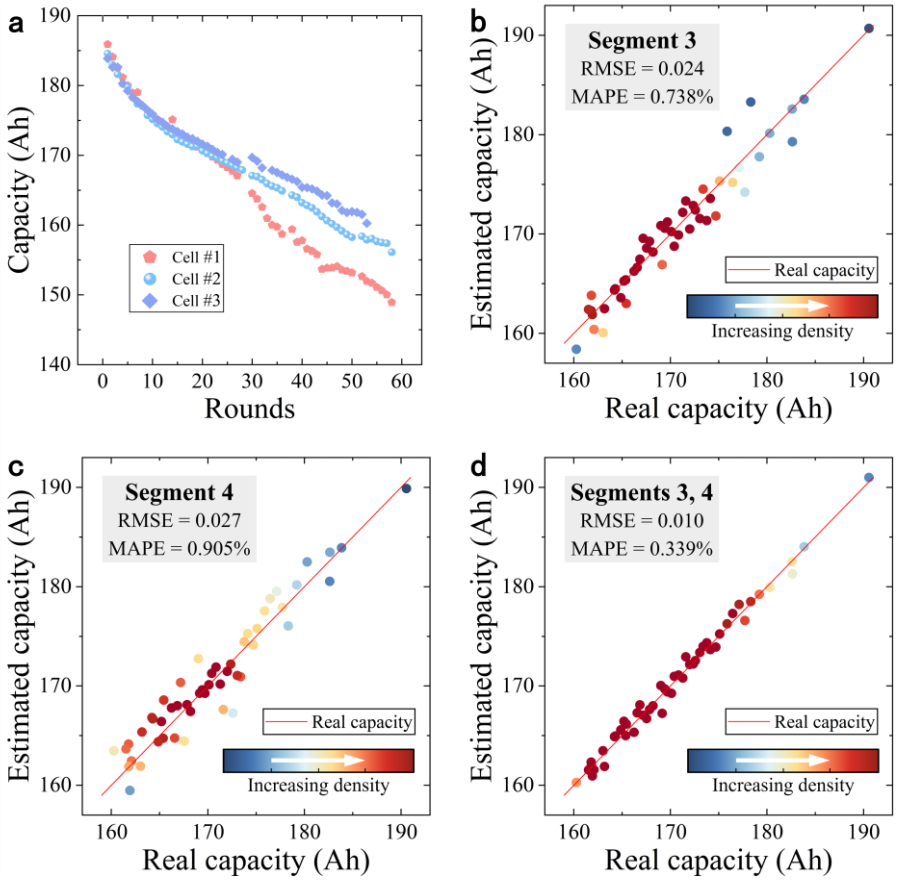
Supplementary Figure 7. Visualization results of the LASSO algorithm for different feature combinations. The available capacity estimation performance is plotted for 7 feature combinations, which include (segments 1, 2) (**a**), (segments 1, 3) (**b**), (segments 1, 5) (**c**), (segments 1, 7) (**d**), (segments 1, 8) (**e**), (segments 1, 11) (**f**), and (segments 2, 3) (**g**). The gradients correspond to varying point-aggregation densities. RMSE denotes root-mean-square error and MAPE denotes mean absolute percentage error.



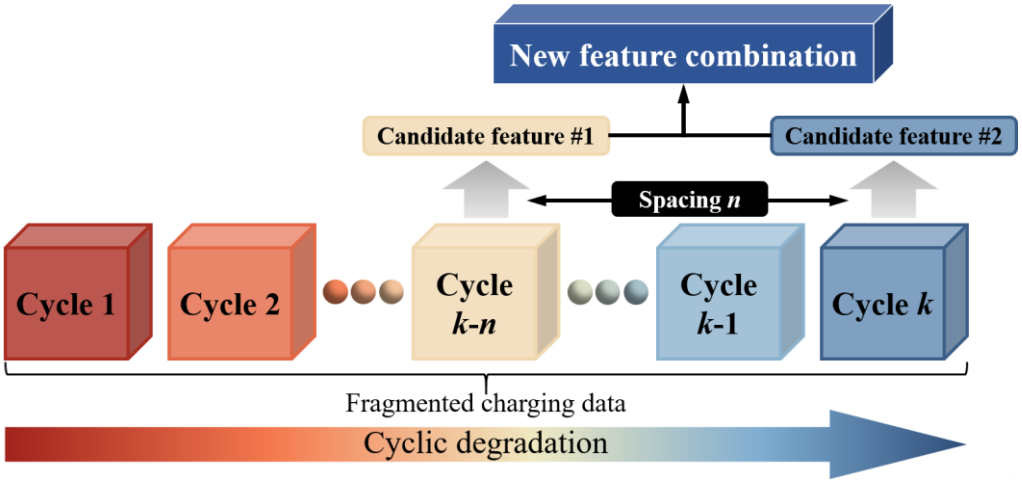
Supplementary Figure 8. Visualization results of the LASSO algorithm for different feature combinations. The available capacity estimation performance is plotted for 7 feature combinations, which include (segments 2, 5) (**a**), (segments 2, 7) (**b**), (segments 2, 8) (**c**), (segments 2, 11) (**d**), (segments 3, 5) (**e**), (segments 3, 7) (**f**), and (segments 3, 8) (**g**). The gradients correspond to varying point-aggregation densities. RMSE denotes root-mean-square error and MAPE denotes mean absolute percentage error.



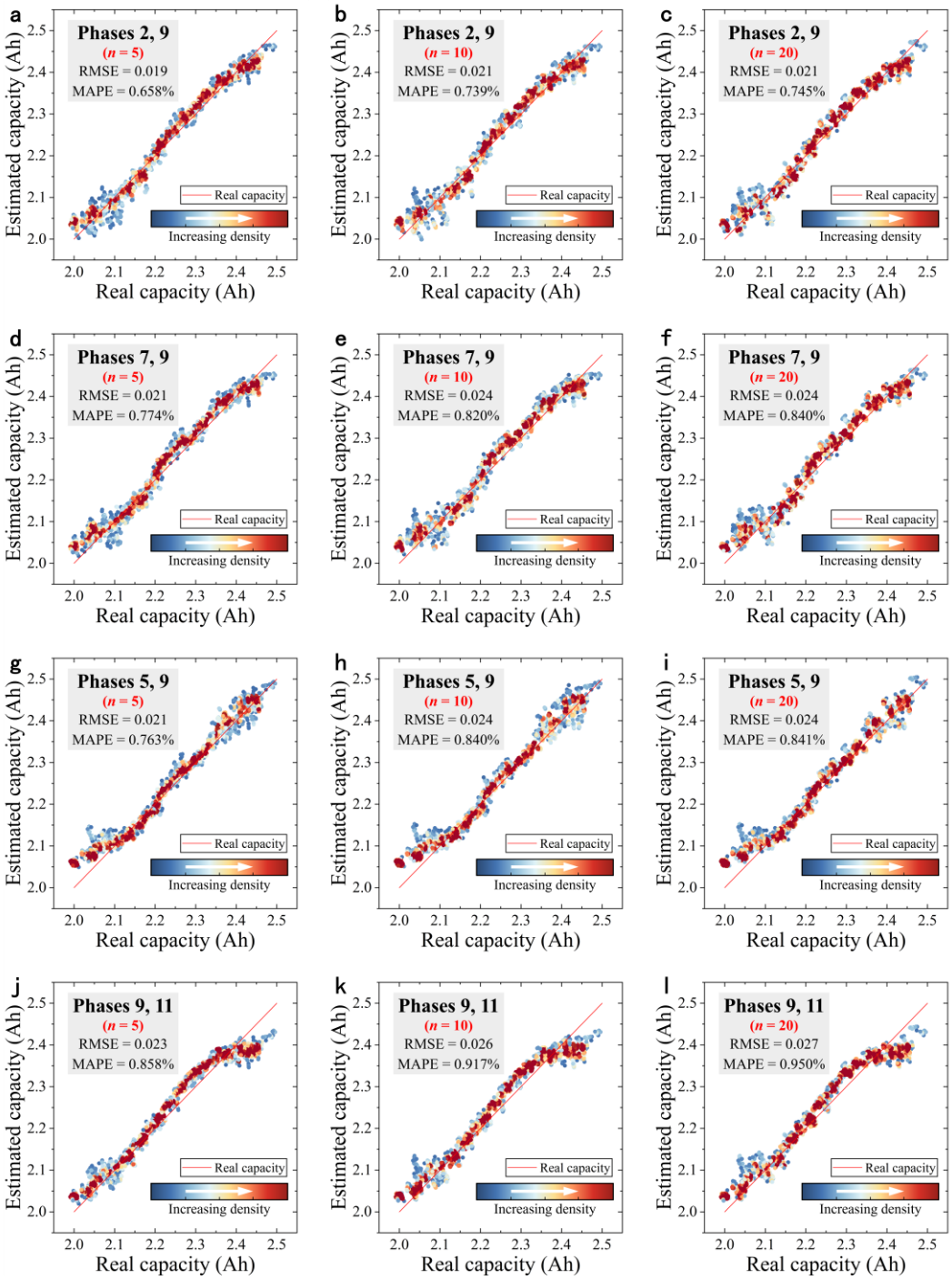
Supplementary Figure 9. Visualization results of the LASSO algorithm for different feature combinations. The available capacity estimation performance is plotted for 7 feature combinations, which include (segments 3, 11) (**a**), (segments 5, 7) (**b**), (segments 5, 8) (**c**), (segments 5, 11) (**d**), (segments 7, 8) (**e**), (segments 7, 11) (**f**), and (segments 8, 11) (**g**). The gradients correspond to varying point-aggregation densities. RMSE denotes root-mean-square error and MAPE denotes mean absolute percentage error.



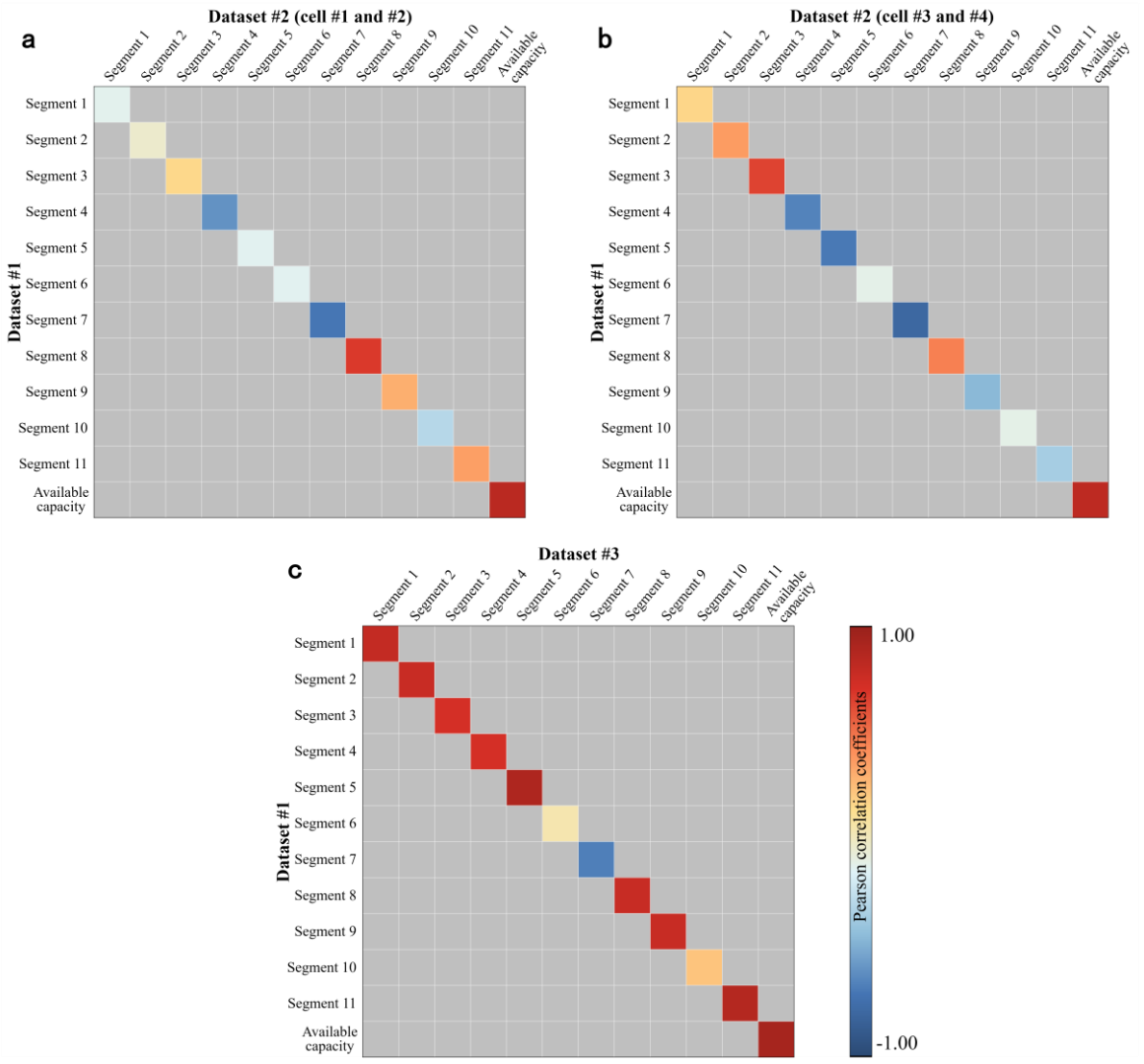
Supplementary Figure 10. Visualization results under a realistic forklift load profile. Available capacity degradation curves for the three cells **(a)**. The available capacity estimation performance is plotted for 3 feature combinations, which include (segment 3) **(b)**, (segment 4) **(c)**, and (segments 3, 4) **(d)**. The gradients correspond to varying point-aggregation densities. RMSE denotes root-mean-square error and MAPE denotes mean absolute percentage error.



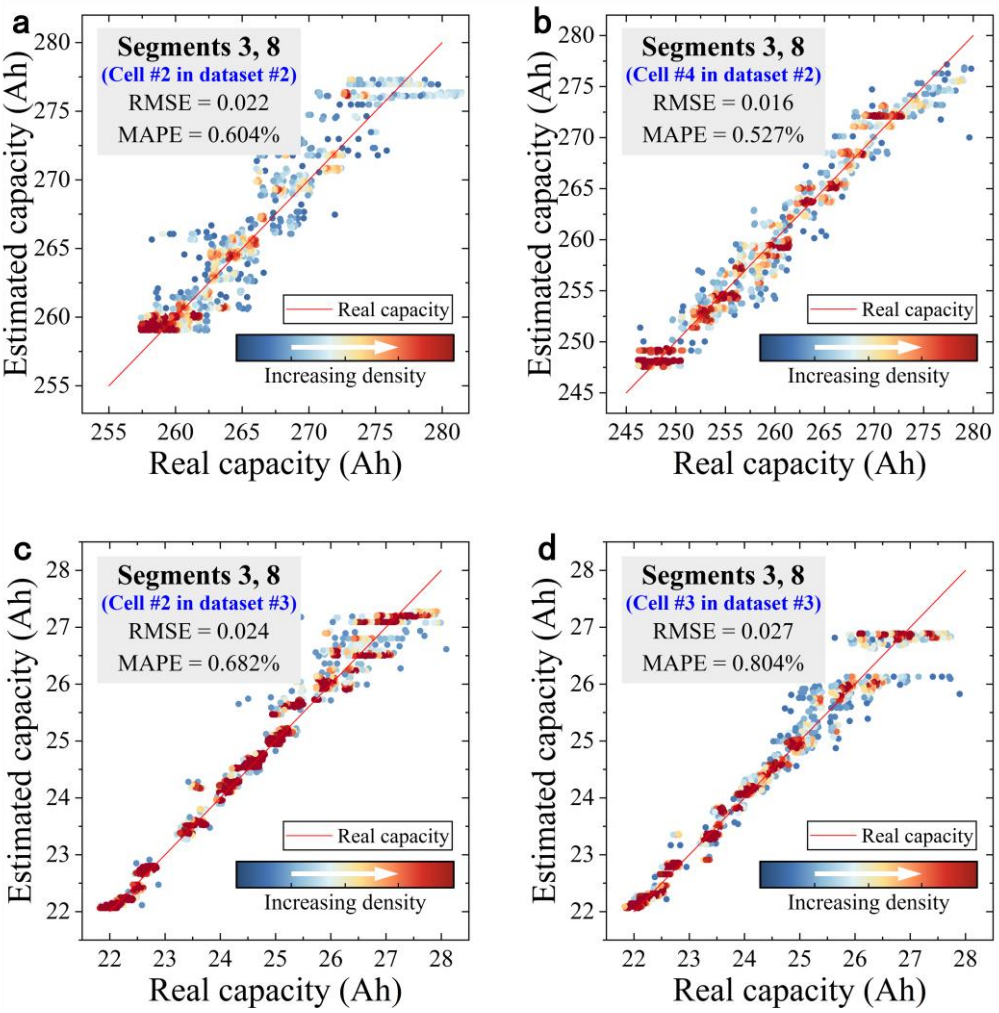
Supplementary Figure 11. Process of feature combinations across cycles. *n* is the spacing between two candidate features. The gradient is used to describe the data in different cycles as the battery cyclic degradation occurs.



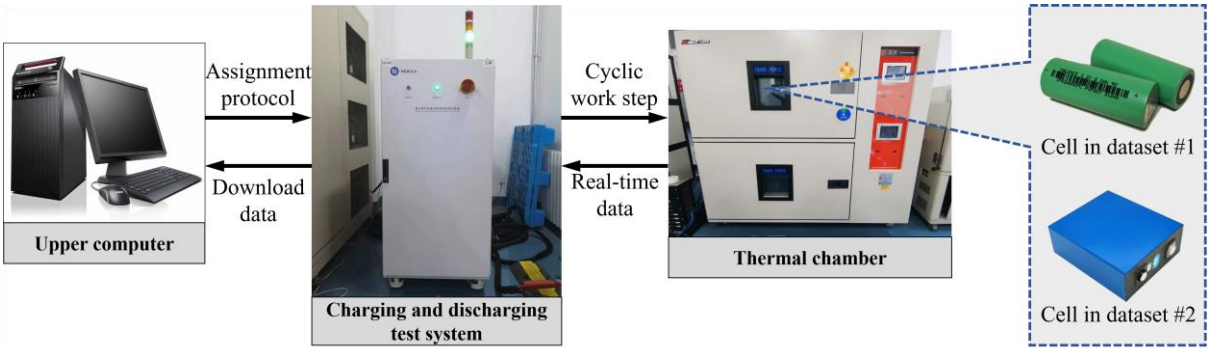
Supplementary Figure 12. Performance of feature combinations across cycles. The results when the feature combination is (segments 2, 9) and *n*=5, 10, and 20 are exhibited as (**a**), (**b**), and (**c**). The results when the feature combination is (segments 7, 9) and *n*=5, 10, and 20 are exhibited as (**d**), (**e**), and (**f**). The results when the feature combination is (segments 5, 9) and *n*=5, 10, and 20 are exhibited as (**g**), (**h**), and (**i**). The results when the feature combination is (segments 9, 11) and *n*=5, 10, and 20 are exhibited as (**j**), (**k**), and (**l**). The gradients correspond to varying point-aggregation densities. RMSE denotes root-mean-square error and MAPE denotes mean absolute percentage error.



Supplementary Figure 13. Correlations between the same candidate features or between available capacities. Various color blocks correspond to differentiated correlations. The gradient from blue to red denotes to the correlation coefficient from small to large.



Supplementary Figure 14. Test results of estimated capacity versus real capacity by transfer learning under (segments 3, 8). Tests are performed on 4 cells in datasets #2 and #3. Cell #2 in dataset #2 (**a**), cell #4 in dataset #2 (**b**), cell #2 in dataset #3 (**c**), and cell #3 in dataset #3 (**d**). The gradients correspond to varying point-aggregation densities. RMSE denotes root-mean-square error and MAPE denotes mean absolute percentage error.



Supplementary Figure 15. The equipment connection principle schematic.

Supplementary Table

Supplementary Table 1. Detailed charge/discharge rate information for dataset #4 (cell #1 - #25)

Cell label	Charge current rate	Discharge current rate
#1	1C (4%), 6C (80%), 1C	4C
#2	2C (10%), 6C (80%), 1C	4C
#3	2C (2%), 5C (80%), 1C	4C
#4	3.6C (2%), 4.85C (80%), 1C	4C
#5	3.6C (9%), 5C (80%), 1C	4C
#6	3.6C (22%), 5.5C (80%), 1C	4C
#7	3.6C (80%), 1C	4C
#8	4C (13%), 5C (80%), 1C	4C
#9	4C (40%), 6C (80%), 1C	4C
#10	4C (4%), 4.85C (80%), 1C	4C
#11	4C (80%), 1C	4C
#12	4.4C (8%), 4.85C (80%), 1C	4C
#13	4.4C (24%), 5C (80%), 1C	4C
#14	4.4C (47%), 5.5C (80%), 1C	4C
#15	4.4C (55%), 6C (80%), 1C	4C
#16	4.4C (80%), 1C	4C
#17	4.65C (19%), 4.85C (80%), 1C	4C
#18	4.65C (44%), 5C (80%), 1C	4C
#19	4.65C (69%), 6C (80%), 1C	4C
#20	4.8C (80%), 1C	4C
#21	4.9C (61%), 4.5C (80%), 1C	4C
#22	4.9C (69%), 4.25C (80%), 1C	4C
#23	5C (67%), 4C (80%), 1C	4C
#24	5.2C (10%), 4.75C (80%), 1C	4C
#25	5.2C (37%), 4.5C (80%), 1C	4C

Supplementary Table 2. Detailed charge/discharge rate information for dataset #4 (cell #26 - #50)

Cell label	Charge current rate	Discharge current rate
#26	5.3C (54%), 4C (80%), 1C	4C
#27	5.4C (40%), 3.6C (80%), 1C	4C
#28	5.4C (50%), 3C (80%), 1C	4C
#29	5.4C (50%), 3.6C (80%), 1C	4C
#30	5.4C (60%), 3C (80%), 1C	4C
#31	5.4C (60%), 3.6C (80%), 1C	4C
#32	5.4C (80%), 1C	4C
#33	5.6C (19%), 4.6C (80%), 1C	4C
#34	5.6C (25%), 4.5C (80%), 1C	4C
#35	5.6C (36%), 4.3C (80%), 1C	4C
#36	5.6C (38%), 4.25C (80%), 1C	4C
#37	5.6C (47%), 4C (80%), 1C	4C
#38	5.6C (58%), 3.5C (80%), 1C	4C
#39	5.6C (58%), 3.5C (80%), 1C	4C
#40	6C (30%), 3.6C (80%), 1C	4C
#41	6C (31%), 4.25C (80%), 1C	4C
#42	6C (40%), 3C (80%), 1C	4C
#43	6C (40%), 3.6C (80%), 1C	4C
#44	6C (40%), 4C (80%), 1C	4C
#45	6C (50%), 3C (80%), 1C	4C
#46	6C (50%), 3.6C (80%), 1C	4C
#47	7C (30%), 3.6C (80%), 1C	4C
#48	7C (40%), 3C (80%), 1C	4C
#49	7C (40%), 3.6C (80%), 1C	4C
#50	8C (15%), 3.6C (80%), 1C	4C

Supplementary Table 3. Detailed experimental information on datasets

Dataset	Manufacturer	Nominal capacity (Ah)	Type	Voltage range (V)	Experimental information
Dataset #1	VALENCE	2.5	Cylinder cell	2.0-3.65	1. All four cells were charged with the CC-CV charging protocol. Charged at a constant current rate of 1C until the voltage reached 3.65V and then 3.65V was sustained until the charging current dropped to 0.15A. 2. Discharged at a constant current rate of 4C. 3. Ambient temperature is 25°C.
Dataset #2	HUAWEI	280	Prismatic cell	2.5-3.65	1. All four cells were charged with the CC charging protocol. Charged at a constant current rate of 0.5C until the voltage reached 3.65V. 2. Discharged at a constant current rate of 0.5C. 3. The ambient temperatures of cells #1 and #2 is 35°C, while that of cells #3 and #4 is 45°C.
Dataset #3 [7]	GOTION	27	Prismatic cell	2.0-3.65	1. All three cells were charged with the CC-CV charging protocol. Charged at a constant current rate of 1C until the voltage reached 3.65V and then 3.65V was sustained until the charging current dropped to 1.35A. 2. Discharged at a constant current rate of 1C. 3. Ambient temperature is 45°C.
Dataset #4 [8]	A123 SYSTEMS	1.1	Cylinder cell	2.0-3.65	1. All fifty cells were charged with the fast charging protocol. 2. All fifty cells discharged with the CC-CV discharge at 4C to 2.0V with a current cutoff of C/50. 3. Ambient temperature is 30°C.

Supplementary Table 4. Detailed charging time information of the candidate features

Segment	Average charging time (s)	Average proportion	Median charging time (s)	Median proportion
CC charging	2974.8	-	2955.7	-
Segment 1	157.1	5.3%	153.8	5.2%
Segment 2	324.5	10.9%	323.8	10.9%
Segment 3	326.9	11.0%	325.5	10.9%
Segment 4	161.9	5.4%	146.3	4.9%
Segment 5	139.2	4.7%	103.7	3.5%
Segment 6	489.9	16.5%	514.3	17.3%
Segment 7	538.8	18.1%	547.0	18.4%
Segment 8	234.6	7.9%	233.8	7.9%
Segment 9	192.7	6.5%	202.5	6.8%
Segment 10	273.3	9.2%	270.4	9.1%
Segment 11	74.7	2.5%	73.4	2.5%

Supplementary Table 5. Pearson correlation coefficients for the 8 segments

Segment	1	2	3	5	7	8	9	11	Available capacity
1	-	0.909	0.882	0.880	-0.889	0.798	0.745	-0.823	0.856
2	0.909	-	0.959	0.880	-0.962	0.766	0.741	-0.903	0.880
3	0.882	0.959	-	0.926	-0.974	0.835	0.758	-0.896	0.903
5	0.880	0.880	0.926	-	-0.916	0.827	0.669	-0.756	0.826
7	-0.889	-0.962	-0.974	-0.916	-	-0.759	-0.729	0.881	-0.875
8	0.798	0.766	0.835	0.827	-0.759	-	0.886	-0.830	0.919
9	0.745	0.741	0.758	0.669	-0.729	0.886	-	-0.891	0.957
11	-0.823	-0.903	-0.896	-0.756	0.881	-0.830	-0.891	-	-0.936

Supplementary Table 6. Feature combinations

Number of candidate features	One feature	Two features
Feature combinations	Segment 1	Segments 1, 2
	Segment 2	Segments 1, 3
	Segment 3	Segments 1, 5
	Segment 5	Segments 1, 7
	Segment 7	Segments 1, 8
	Segment 8	Segments 1, 9
	Segment 9	Segments 1, 11
	Segment 11	Segments 2, 3
		Segments 2, 5
		Segments 2, 7
		Segments 2, 8
		Segments 2, 9
		Segments 2, 11
		Segments 3, 5
		Segments 3, 7
		Segments 3, 8
		Segments 3, 9
		Segments 3, 11
		Segments 5, 7
		Segments 5, 8
		Segments 5, 9
		Segments 5, 11
		Segments 7, 8
		Segments 7, 9
		Segments 7, 11
		Segments 8, 9
		Segments 8, 11
		Segments 9, 11

Supplementary Table 7. Hyperparameters

LASSO		XGBoost		LightGBM	
Alpha	[0:0.001:1]	Learning_rate	0.1	Objective	regression
Lambda	[0:0.001:1]	N_estimators	100	Metric	rmse
K-fold	4	Max_depth	3	Boosting_type	gbdt
Monte Carlo *N*	1	Objective	reg:squarederror	Num_leaves	31
		Min_child_weight	1	Learning_rate	0.05
		Subsample	0.8	Feature_fraction	0.9
		Alpha	0		
		Lambda	1		

Supplementary Table 8. Basic model validation results under the Lasso algorithm

Input feature combinations	MAPE	RMSE
Segments 3, 9	0.564%	0.015
Segments 2, 9	0.710%	0.018
Segments 7, 9	0.793%	0.020
Segments 5, 9	0.830%	0.024
Segments 9, 11	0.977%	0.027
Segments 1, 9	0.999%	0.028
Segments 8, 9	1.098%	0.031
Segment 9	1.190%	0.033

Supplementary Table 9. Basic model validation results under the XGboost algorithm

Input feature combinations	MAPE	RMSE
Segments 3, 9	0.411%	0.012
Segments 2, 9	0.596%	0.017
Segments 7, 9	0.727%	0.020
Segments 5, 9	0.661%	0.018
Segments 9, 11	0.853%	0.024
Segments 1, 9	0.785%	0.024
Segments 8, 9	0.967%	0.026
Segment 9	1.167%	0.034

Supplementary Table 10. Basic model validation results under the LightGBM algorithm

Input feature combinations	MAPE	RMSE
Segments 3, 9	0.417%	0.012
Segments 2, 9	0.581%	0.017
Segments 7, 9	0.735%	0.020
Segments 5, 9	0.688%	0.019
Segments 9, 11	0.830%	0.023
Segments 1, 9	0.772%	0.024
Segments 8, 9	0.953%	0.026
Segment 9	1.155%	0.033

Supplementary Table 11. RMSEs and MAPEs when the feature combination contains 3 to 8 candidates

Feature combinations	Number of candidate features	RMSE	MAPE
Segments 2, 3, 9	3	0.015	0.564%
Segments 2, 3, 7, 9	4	0.016	0.560%
Segments 2, 3, 5, 7, 9	5	0.015	0.532%
Segments 2, 3, 5, 7, 9, 11	6	0.015	0.544%
Segments 1, 2, 3, 5, 7, 9, 11	7	0.015	0.543%
Segments 1, 2, 3, 5, 7, 8, 9, 11	8	0.015	0.545%

Supplementary Table 12. Validation results under spacing of 5,10 and 20 RMSEs and MAPEs when the feature combination contains 3 to 8 candidates

Input feature combinations	Spacing (n)	MAPE	RMSE
Segments 3, 9	$n = 5$	0.571%	0.017
	$n = 10$	0.666%	0.019
	$n = 20$	0.674%	0.019
Segments 2, 9	$n = 5$	0.658%	0.019
	$n = 10$	0.739%	0.021
	$n = 20$	0.745%	0.021
Segments 7, 9	$n = 5$	0.774%	0.021
	$n = 10$	0.820%	0.024
	$n = 20$	0.840%	0.024
Segments 5, 9	$n = 5$	0.763%	0.021
	$n = 10$	0.840%	0.024
	$n = 20$	0.841%	0.024
Segments 9, 11	$n = 5$	0.858%	0.023
	$n = 10$	0.917%	0.026
	$n = 20$	0.950%	0.027

Supplementary Table 13. Detailed partitioning of training and test sets in dataset #4

Training set	Test set
Cell #1	Cell #3
Cell #2	Cell #7
Cell #4	Cell #11
Cell #5	Cell #16
Cell #6	Cell #19
Cell #8	Cell #20
Cell #9	Cell #22
Cell #10	Cell #25
Cell #12	Cell #31
Cell #13	Cell #32
Cell #14	Cell #37
Cell #15	Cell #38
Cell #17	Cell #45
Cell #18	Cell #46
Cell #21	Cell #49
Cell #23	
Cell #24	
Cell #26	
Cell #27	
Cell #28	
Cell #29	
Cell #30	
Cell #33	
Cell #34	
Cell #35	
Cell #36	
Cell #39	
Cell #40	
Cell #41	
Cell #42	
Cell #43	
Cell #44	
Cell #47	
Cell #48	
Cell #50	

Supplementary Table 14. Estimation performance of feature combinations across cycles on dataset #4

Cell label	RMSE			
	n = 0/1	n = 5	n = 10	n = 20
#3	0.011	0.012	0.014	0.015
#7	0.017	0.018	0.018	0.018
#11	0.013	0.013	0.014	0.015
#16	0.020	0.021	0.022	0.023
#19	0.013	0.014	0.014	0.015
#20	0.020	0.021	0.023	0.023
#22	0.013	0.014	0.015	0.016
#25	0.014	0.014	0.016	0.018
#31	0.016	0.016	0.017	0.019
#32	0.018	0.018	0.019	0.020
#37	0.017	0.017	0.018	0.019
#38	0.019	0.019	0.020	0.021
#45	0.014	0.014	0.016	0.016
#46	0.013	0.013	0.014	0.015
#49	0.016	0.016	0.018	0.019

Supplementary Table 15. Constant current (CC) charging-based methods

Methods	References	Input features	Disadvantages
Direct extraction method	Zheng et al. [9]	Charge capacity in the optimal voltage interval	1. The battery available capacity cannot be continuously estimated. 2. The fixed voltage interval leads to insufficient flexibility in estimation. 3 To improve the estimation accuracy, the voltage interval range is usually large and difficult to obtain. 4. The optimal voltage interval for different datasets needs to be re-selected.
	Shu et al. [10]	Charging time in the optimal voltage interval	1. Since the charging time is the feature, the requirements for sampling frequency and sampling accuracy are higher. 2. The battery available capacity cannot be continuously estimated. 3. Unable to adapt to fragmented charging data. 4. The optimal voltage interval for different datasets needs to be re-selected.
Indirect extraction method	Weng et al. [11]	Incremental capacity peak	1. The peak is not necessarily extracted in every cycle. 2. The battery available capacity cannot be continuously estimated. 3. High requirements for sampling accuracy and collection equipment for sample data. 4. Data interpolation and filtering processes cut into accuracy.

Supplementary Table 16. Correlation analysis between datasets

		Dataset #2 (cell #1 and #2)	Dataset #2 (cell #3 and #4)	Dataset #3
Dataset #1	Segment 1	-0.146	0.227	0.827
	Segment 2	-0.041	0.402	0.796
	Segment 3	0.219	0.684	0.750
	Segment 4	-0.594	-0.684	0.761
	Segment 5	-0.163	-0.709	0.891
	Segment 6	-0.165	-0.114	0.128
	Segment 7	-0.748	-0.809	-0.708
	Segment 8	0.694	0.527	0.888
	Segment 9	0.359	-0.461	0.888
	Segment 10	-0.335	-0.124	0.245
	Segment 11	0.451	0.342	0.849
	Available capacity	0.831	0.853	0.937

Supplementary Reference

[1] Schober, P., Boer, C., & Schwarte, L. A. Correlation coefficients: appropriate use and interpretation. *Anesth. Analg.* **126**, 1763-1768 (2018).

[2] Shin, H. C. et al. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Trans. Med. Imaging* **35**, 1285-1298 (2016).

[3] Gal, M. S., & Rubinfeld, D. L. Data standardization. *NYUL Rev.* **94**, 737 (2019).

[4] Tibshirani, R. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B* **58**, 267-288 (1996).

[5] Chen, T. & Guestrin, C. in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 785–794 (2016).

[6] Ke, G. et al. Lightgbm: A highly efficient gradient boosting decision tree. *Adv. Neural Inf. Process. Syst.* **30** (2017).

[7] Lu, J., Xiong, R., Tian, J., Wang, C., & Sun, F. Deep learning to estimate lithium-ion battery state of health without additional degradation experiments. *Nat. Commun.* **14**, 2760 (2023).

[8] Severson, K. A. et al. Data-driven prediction of battery cycle life before capacity degradation. *Nat. Energy* **4**, 383-391 (2019).

[9] Zheng, Y. et al. A novel capacity estimation method based on charging curve sections for lithium-ion batteries in electric vehicles. *Energy* **185**, 361-371 (2019).

[10] Shu, X. et al. A flexible state-of-health prediction scheme for lithium-ion battery packs with long short-term memory network and transfer learning. *IEEE Trans. Transp. Electrification* **7**, 2238-2248 (2021).

[11] Weng, C., Feng, X., Sun, J., & Peng, H. State-of-health monitoring of lithium-ion battery modules and packs via incremental capacity peak tracking. *Appl. Energy* **180**, 360-368 (2016).