Supplementary Information

Article title

Immediate remaining capacity estimation of heterogeneous second-life lithium-ion batteries via deep generative transfer learning

Author list

Shengyu Tao[1,2], Ruohan Guo[3], Jaewoong Lee[2], Scott Moura[2], Lluc Canals Casals[4], Shida Jiang[2], Junzhe Shi[2], Stephen Harris[5], Tongda Zhang[6], Chi Yung Chung[3], Guangmin Zhou[1,*], Jinpeng Tian[3,*], Xuan Zhang[1,*]

Affiliations

1. Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, China
2. Energy, Controls, and Applications Lab (eCAL), Department of Civil and Environmental Engineering, University of California, California, USA
3. Department of Electrical and Electronic Engineering and Research Centre for Grid Modernisation, The Hong Kong Polytechnic University, Hong Kong, China
4. Environmental Engineering Research Group (ENMA), Department of Project and Construction Engineering, Universitat Politècnica de Catalunya·BarcelonaTech (UPC), Barcelona, Spain
5. Energy Storage and Distributed Resources Division, Lawrence Berkeley National Laboratory, California, USA
6. Systems Optimization Laboratory, Stanford University, California, USA

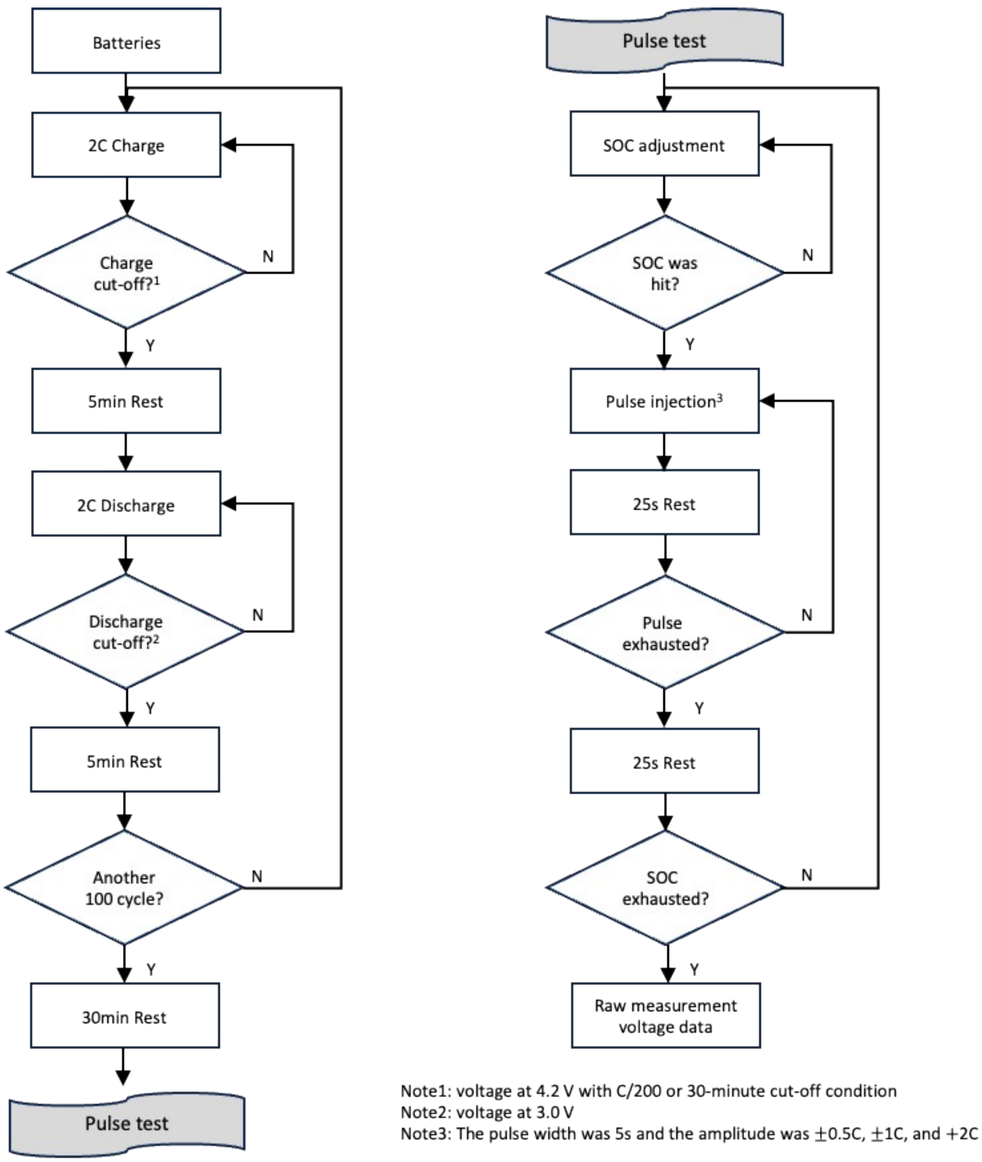
Correspondence

Guangmin Zhou: guangminzhou@sz.tsinghua.edu.cn

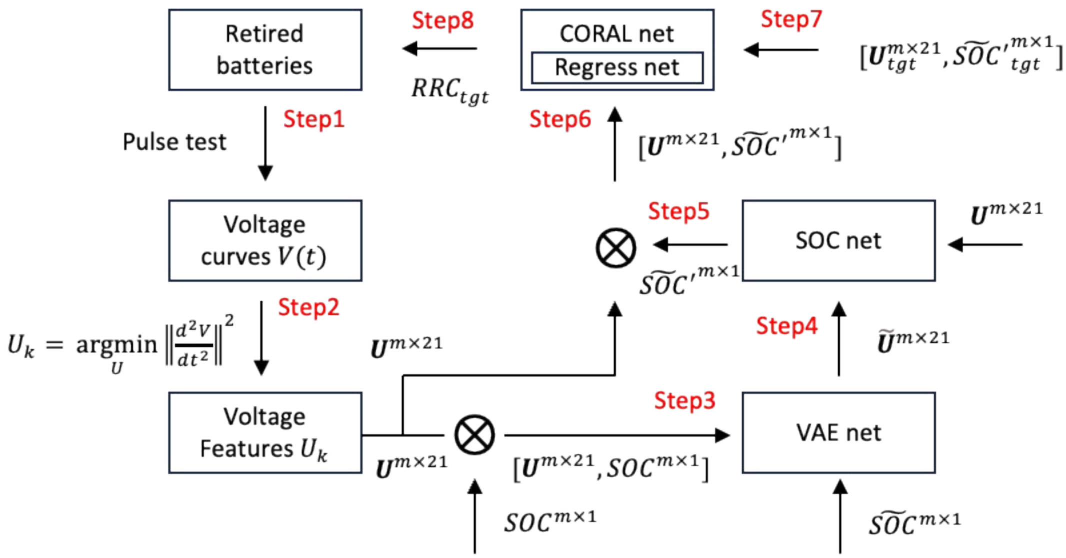
Jinpeng Tian: jinpeng.tian@polyu.edu.hk

Xuan Zhang: xuanzhang@sz.tsinghua.edu.cn

Supplementary Figure 1 The detailed experimental workflow for data generation.



Supplementary Figure 2 The detailed workflow of the proposed deep generative transfer learning method. Step1: Pulse test. Step2: Feature engineering. Step3: VAE network training. Step4: voltage feature data generation. Step5: SOC prediction. Step6: Remaining capacity prediction in one domain. Step7: Voltage feature data generation and SOC prediction in another domain. Step8: Remaining capacity prediction in another domain via CORAL transfer learning.



Supplementary Table 1 Model performance benchmarking only using target domain data (1/50 = 42 samples for Pouch31 and Pouch52, respectively)

Model	MAPE (%)		ρ	
	Pouch31	Pouch52	Pouch31	Pouch52
Linear Regression	100.72	62.23	0.19	0.01
Ridge Regression	7.13	15.95	0.17	0.72
SVM	6.64	17.94	0.11	0.55
k-NN	8.23	17.03	0.05	0.34
Random Forest	9.00	16.49	-0.02	0.33
DNN	10.51	10.31	0.07	0.69
Our work	**3.6**	**7.2**	**0.73**	**0.84**

Supplementary Table 2 Model performance benchmarking only using target domain data (1/40 = 52 samples for Pouch31 and Pouch52, respectively)

Model	MAPE (%)		ρ	
	Pouch31	Pouch52	Pouch31	Pouch52
Linear Regression	100.50	66.35	0.19	-0.04
Ridge Regression	6.69	15.85	0.14	0.71
SVM	6.31	17.98	0.10	0.55
k-NN	7.53	17.45	0.10	0.28
Random Forest	7.64	17.22	0.10	0.26
DNN	9.33	11.54	0.34	0.70
Our work	**3.7**	**8.1**	**0.70**	**0.82**

Supplementary Table 3 Model performance benchmarking only using target domain data (1/30 = 70 samples for Pouch31 and Pouch52, respectively)

Model	MAPE (%)		ρ	
	Pouch31	Pouch52	Pouch31	Pouch52
Linear Regression	63.66	67.76	0.08	-0.03
Ridge Regression	6.99	17.47	0.14	0.63
SVM	6.03	18.08	0.17	0.57
k-NN	7.74	17.50	0.05	0.21
Random Forest	7.43	17.20	0.15	0.24
DNN	6.22	11.82	0.51	0.69
Our work	**3.7**	**6.0**	**0.75**	**0.88**

Supplementary Table 4 Model performance benchmarking only using target domain data (1/20 = 105 samples for Pouch31 and Pouch52, respectively)

Model	MAPE (%)		ρ	
	Pouch31	Pouch52	Pouch31	Pouch52
Linear Regression	48.46	86.03	0.13	0.02
Ridge Regression	7.61	16.96	0.17	0.65
SVM	6.10	17.64	0.19	0.67
k-NN	7.71	17.14	0.01	0.25
Random Forest	8.00	16.26	0.08	0.34
DNN	5.15	9.64	0.64	0.74
Our work	**2.8**	**5.9**	**0.85**	**0.89**

Supplementary Table 5 Equivalent circuit model-based parameter identification settings and results.

The BOL testing case	The MOL testing case	The EOL testing case
--- Selected Inputs --- Selected SOC value: 25% Selected SOH column index: 1 Selected SOH value to test: 0.9121 OCV-SOC curve comes from BOL SOH: 0.923 --- Optimization Results --- Best $\lambda0$: 0.80268 Best $\lambda1$: 2.2408 R1: 0.018114 Ohms R2: 0.084128 Ohms C: 286.4023 Farads Tau: 24.0944 s Q_true: 1.9154 Ah Q_nominal: 2.1 Ah Estimated Q: 2.0099 Ah MAPE to Q_nominal: 4.29 % MAPE to Q_true: 4.9337 %	--- Selected Inputs --- Selected SOC value: 25% Selected SOH column index: 8 Selected SOH value to test: 0.8785 OCV-SOC curve comes from BOL SOH: 0.923 --- Optimization Results --- Best $\lambda0$: 1.204 Best $\lambda1$: 2.4749 R1: 0.017815 Ohms R2: 0.11348 Ohms C: 262.7274 Farads Tau: 29.8145 s Q_true: 1.8448 Ah Q_nominal: 2.1 Ah Estimated Q: 2.0653 Ah MAPE to Q_nominal: 1.6542 % MAPE to Q_true: 11.9474 %	--- Selected Inputs --- Selected SOC value: 25% Selected SOH column index: 65 Selected SOH value to test: 0.6598 OCV-SOC curve comes from BOL SOH: 0.923 --- Optimization Results --- Best $\lambda0$: 0.70234 Best $\lambda1$: 2.8763 R1: 0.018151 Ohms R2: 0.070114 Ohms C: 301.2463 Farads Tau: 21.1215 s Q_true: 1.3856 Ah Q_nominal: 2.1 Ah Estimated Q: 2.1048 Ah MAPE to Q_nominal: 0.22842 % MAPE to Q_true: 51.9073 %

Supplementary Note 1 Parameter Identification of the Equivalent Circuit Model

1 Model

Consider the equivalent circuit model (ECM) given by the following differential equations:

$$\dot{z}(t) = \frac{1}{Q}I(t), \qquad z(0) = z_0 \tag{1}$$

$$\dot{V}_c(t) = -\frac{1}{R_2 C}V_c(t) + \frac{1}{C}I(t), \qquad V_c(0) = V_{c0} \tag{2}$$

$$V(t) = OCV(z(t)) + V_c(t) + R_1 I(t) \tag{3}$$

where the parameters R_1, R_2, C, Q are unknown. We seek to identify these parameters from measurements of $V(t), I(t)$. To achieve this goal, we follow two steps:

1. Formulate a parametric model.
2. Formulate a parameter identification algorithm.

Let's begin.

2 Parametric Model

Ultimately, we seek to obtain a linear-in-the-parameters model in the form:

$$z(t) = \theta^T \phi(t) \tag{4}$$

where θ is a vector containing the unknown parameters, and scalar signal $z(t)$ and vector signal $\phi(t)$ are measured or processed from measurements. To arrive at this linear-in-the-parameters form, we follow these steps:

1. Derive the transfer function from $I(t)$ to $V(t)$
2. Apply filters to process $z(t)$ and $\phi(t)$ from measured signals (will be clear later)

Let's go.

First, note the model is nonlinear due to the open circuit voltage term: $OCV(z(t))$. Let's approximate this term by it's first-order Taylor series approximation w.r.t. $z(t)$ around $z(t) = z_0$,

$$OCV(z(t)) \approx \underbrace{OCV(z_0)}_{=\beta_0} + \underbrace{\frac{dOCV}{dz}\Big|_{z=z_0}}_{=\beta_1}\underbrace{(z - z_0)}_{=\tilde{z}(t)} + H.O.T \tag{5}$$

$$OCV(z(t)) \approx \beta_1 \tilde{z}(t) + \beta_0 \tag{6}$$

where $\tilde{z}(t) = z(t) - z_0$, $\beta_0 = OCV(z_0)$, $\beta_1 = OCV'(z_0)$. This allows us to re-write the model into proper linear form:

$$\dot{\tilde{z}}(t) = -\frac{1}{Q}I(t), \qquad \tilde{z}(0) = 0 \tag{7}$$

$$\dot{V}_c(t) = -\frac{1}{R_2 C} V_c(t) + \frac{1}{C} I(t), \qquad V_c(0) = V_{c0} = 0 \tag{8}$$

$$\tilde{V}(t) = \beta_1 \tilde{z}(t) + V_c(t) + R_1 I(t) \tag{9}$$

where $\tilde{V}(t) = V(t) - \beta_0 = V(t) - OCV(z_0)$. That is, we consider the measured output to be the voltage deviation from the initial voltage. Note also that I have assumed $V_c(0) = 0$. This is true when the battery is at initially at equilibrium, but also convenient because we don't need to worry about initial conditions when taking the Laplace transform.

We take the Laplace transform next to derive transfer function from $I(t)$ to $\tilde{V}(t)$. A few lines of work results in:

$$\tilde{V}(s) = \frac{R_1 Q s^2 + \left(\frac{R_1 Q}{R_2 C} + \frac{Q}{C} - \beta_1\right) s - \frac{\beta_1}{R_2 C}}{Q s^2 + \frac{Q}{R_2 C} s + 0} I(s) \tag{10}$$

Note that I've abused notation by indicating the frequency-domain signals $\tilde{V}(s)$, $I(s)$ by the same capital letters as the time-domain signals. The meaning should be clear from context. At this stage let's represent the transfer function by the following generalized notation:

$$\tilde{V}(s) = \frac{b_2 s^2 + b_1 s + b_0}{a_2 s^2 + a_1 s + a_0} I(s) \tag{11}$$

where the coefficients are defined in the table below.

Transfer Function Coefficients

Now we take the inverse Laplace transform to arrive at a higher-order differential equation model:

$$a_2 \ddot{\tilde{V}}(t) + a_1 \dot{\tilde{V}}(t) + a_0 \tilde{V}(t) = b_2 \ddot{I}(t) + b_1 \dot{I}(t) + b_0 I(t) \tag{12}$$

Pause for a moment and observe the equation above. This equation is linear-in-the-parameters, and contains measured signals $I(t)$, $V(t)$. The only problem is that it requires derivatives of measured signals. To solve this problem, we filter both sides with a second order filter:

$$\Lambda(s) = \frac{\lambda_0}{s^2 + \lambda_1 s + \lambda_0} \tag{13}$$

with stable poles (verified by setting $\lambda_0,\lambda_1 > 0$). Applying this filter and re-organizing yields:

$$\underbrace{\left\{\frac{\lambda_0}{s^2+\lambda_1 s+\lambda_0}\right\}\ddot{\tilde{V}}(t)}_{=z(t)} = \underbrace{\begin{bmatrix}\frac{b_2}{a_2} & \frac{b_1}{a_2} & \frac{b_0}{a_2} & \frac{a_1}{a_2} & \frac{a_0}{a_2}\end{bmatrix}}_{=\theta^T} \underbrace{\begin{bmatrix}\ddot{I}(t)\\ \dot{I}(t)\\ I(t)\\ -\dot{\tilde{V}}(t)\\ -\tilde{V}(t)\end{bmatrix}\cdot\left\{\frac{\lambda_0}{s^2+\lambda_1 s+\lambda_0}\right\}}_{=\phi(t)} \tag{14}$$

which yields the linear-in-the-parameters model we seek. However, we need to sort out how to implement the filters. Let us focus on implementing

$$\frac{\lambda_0}{s^2+\lambda_1 s+\lambda_0}y(t) \tag{15}$$

where $y(t)$ represents $\ddot{\tilde{V}}(t),\ddot{I}(t),\dot{I}(t),I(t),-\dot{\tilde{V}}(t),-\tilde{V}(t)$. To illustrate, let's focus on $y(t)=\ddot{\tilde{V}}(t)$, i.e.

$$z(t) = \frac{\lambda_0 s^2}{s^2+\lambda_1 s+\lambda_0}\tilde{V}(t) \tag{16}$$

To implement, we will convert this transfer function into state-space form. Any state-space realization will do. Let's pick the controllable canonical form. The corresponding A,B,C,D matrices are:

$$A = \begin{bmatrix}0 & 1\\ -\lambda_0 & -\lambda_1\end{bmatrix}, \quad B = \begin{bmatrix}0\\1\end{bmatrix} \tag{17}$$

$$C = \begin{bmatrix}-\lambda_0^2 & -\lambda_1\lambda_0\end{bmatrix}, \quad D = \lambda_0 \tag{18}$$

Following the same approach, we list the state-space matrices for the remaining signals.

$$\phi_1(t) = \frac{\lambda_0 s^2}{s^2+\lambda_1 s+\lambda_0}I(t) \tag{19}$$

$$A = \begin{bmatrix}0 & 1\\ -\lambda_0 & -\lambda_1\end{bmatrix}, \quad B = \begin{bmatrix}0\\1\end{bmatrix} \tag{20}$$

$$C = \begin{bmatrix}-\lambda_0^2 & -\lambda_1\lambda_0\end{bmatrix}, \quad D = \lambda_0 \tag{21}$$

$$\phi_2(t) = \frac{\lambda_0 s}{s^2+\lambda_1 s+\lambda_0}I(t) \tag{22}$$

$$A = \begin{bmatrix}0 & 1\\ -\lambda_0 & -\lambda_1\end{bmatrix}, \quad B = \begin{bmatrix}0\\1\end{bmatrix} \tag{23}$$

$$C = \begin{bmatrix}0 & \lambda_0\end{bmatrix}, \quad D = 0 \tag{24}$$

$$\phi_3(t) = \frac{\lambda_0}{s^2+\lambda_1 s+\lambda_0}I(t) \tag{25}$$

$$A = \begin{bmatrix} 0 & 1 \\ -\lambda_0 & -\lambda_1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{26}$$

$$C = \begin{bmatrix} \lambda_0 & 0 \end{bmatrix}, \quad D = 0 \tag{27}$$

$$\phi_4(t) = \frac{-\lambda_0 s}{s^2 + \lambda_1 s + \lambda_0} \tilde{V}(t) \tag{28}$$

$$A = \begin{bmatrix} 0 & 1 \\ -\lambda_0 & -\lambda_1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{29}$$

$$C = \begin{bmatrix} 0 & -\lambda_0 \end{bmatrix}, \quad D = 0 \tag{30}$$

$$\phi_5(t) = \frac{-\lambda_0}{s^2 + \lambda_1 s + \lambda_0} \tilde{V}(t) \tag{31}$$

$$A = \begin{bmatrix} 0 & 1 \\ -\lambda_0 & -\lambda_1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{32}$$

$$C = \begin{bmatrix} -\lambda_0 & 0 \end{bmatrix}, \quad D = 0 \tag{33}$$

Finally, for convenience, we document in the table below the forward and inverse transformations from $(R_1,R_2,C,Q) \leftrightarrow (\theta_1,\theta_2,\theta_3,\theta_4)$.

Parameter Transformations

We have completed the most difficult work – formulating the parametric model. Next, we formulate a parameter identification algorithm.

3 Parameter Identification Algorithm

In the previous section, we have formulated a parametric model in the form:

$$z(t) = \theta^T \phi(t) \tag{34}$$

Consider the true parameter vector $\theta^\star$. Then the model perfectly satisfies

$$z(t) = (\theta^\star)^T \phi(t) \tag{35}$$

If $\hat{\theta}$ is an estimate of $\theta^\star$, then the quality of the estimation is measured by the residual

$$\epsilon = z - \hat{\theta}^T \phi \tag{36}$$

Our parameter identification algorithm is based on minimizing ϵ in some sense. Here, we will specifically consider the Least Squares algorithm with normalization. Namely, we seek to minimize the following cost function

$$J(\hat{\theta}) = \frac{1}{2}\int_0^t \frac{\left[z(\tau) - \hat{\theta}^T(t)\phi(\tau)\right]^2}{m^2(\tau)} d\tau \tag{37}$$

The value of $\hat{\theta}(t)$ that minimizes this cost function is generated by:

$$\dot{\hat{\theta}}(t) = P(t) \cdot \epsilon(t) \cdot \phi(t), \qquad \hat{\theta} = \hat{\theta}_0 \tag{38}$$

$$\dot{P}(t) = -P\frac{\phi\phi^T}{m^2}P, \qquad P(0) = P_0 \tag{39}$$

$$m^2(t) = 1 + \alpha\phi^T\phi, \qquad \alpha > 0, \quad P_0 = P_0^T \succ 0 \tag{40}$$