Machine-learning assisted identification of battery life models

Paul Gasper[1], Kandler Smith[1], Nils Collath[2], Holger Hesse[2,3], Andreas Jossen[2]

[1] National Renewable Energy Lab

[2] Technical University of Munich

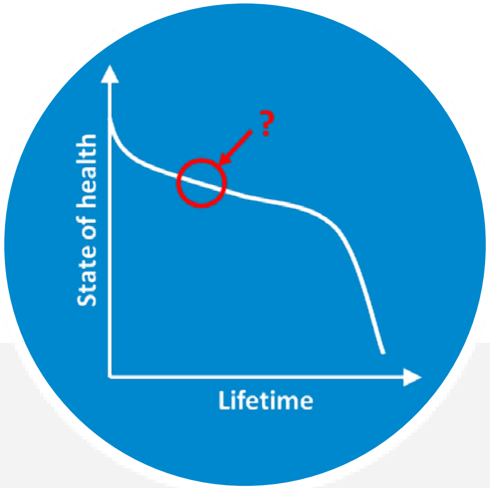
[3] Kempten University of Applied Sciences

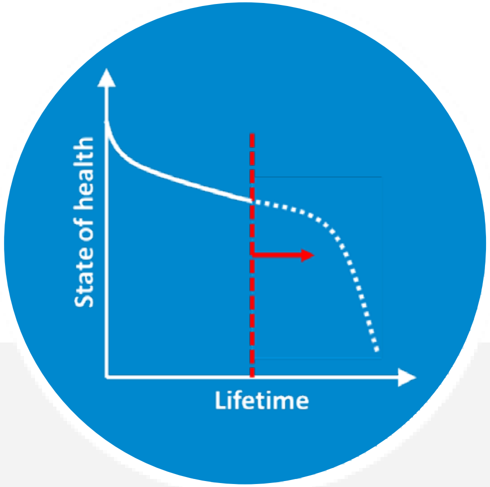
ECS Boston 2023, A01-415

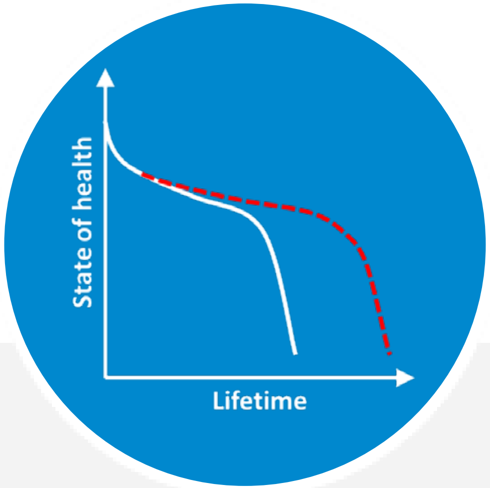
National Renewable Energy Laboratory

Image credit: Nicholas Brunhart-Lupo

Challenges for battery monitoring and lifetime







Diagnosis

Detect battery state using available information from cheap, rapid, scalable measurements.

Prediction

Anticipate future battery performance by synergizing lab data and online diagnostics.

Optimization

Extend battery lifetime or balance system utilization with degradation costs using predictive models.

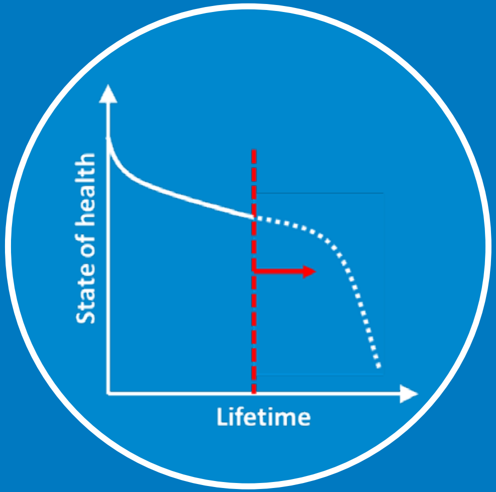
Battery health prediction

Gasper et al (2021), *JES* 168 020502

Gasper et al (2022), *JES* 169 080518

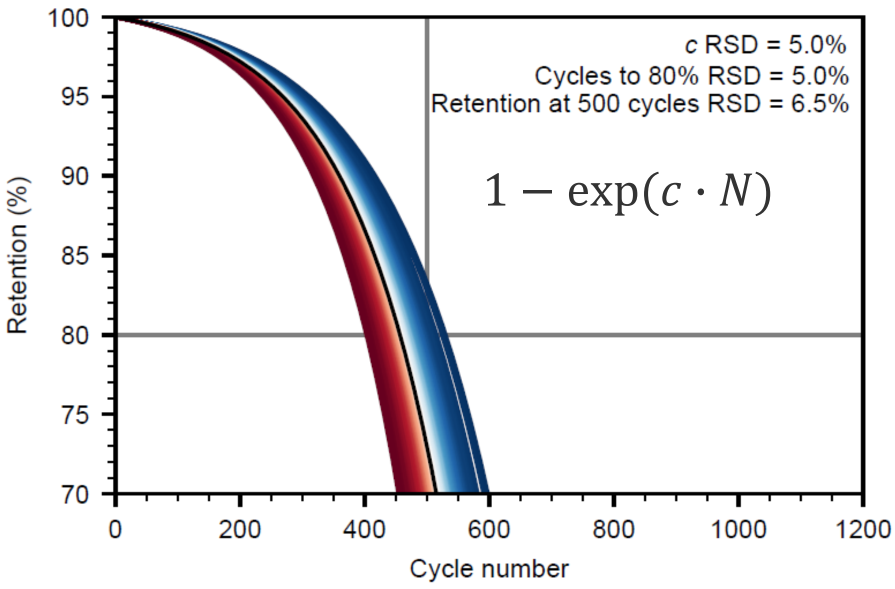
Attia et al (2022), *JES* 169 060517

Data in this section shared by TUM:
Naumann et al (2018), *J. Energy Storage* 17 153-169
Naumann et al (2020), *J. Power Sources* 451 227666



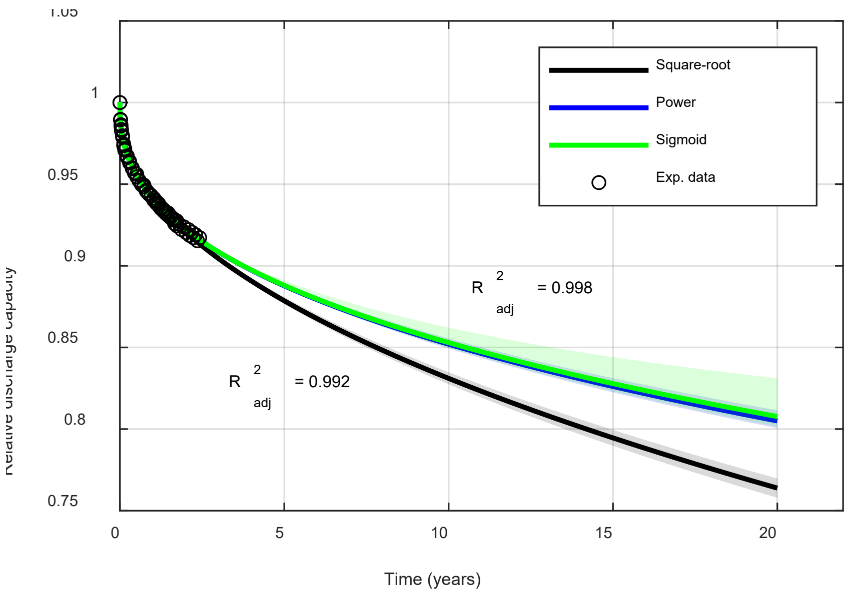
The challenge for accurate battery life prediction

Variability in fade rate → larger variability in lifetime



Uncertainty of the rate

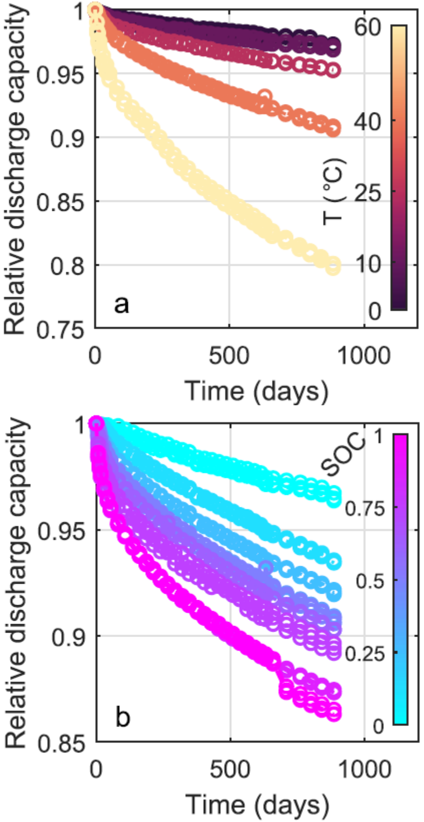
Marginal difference in fit quality → 5-year difference in predicted life

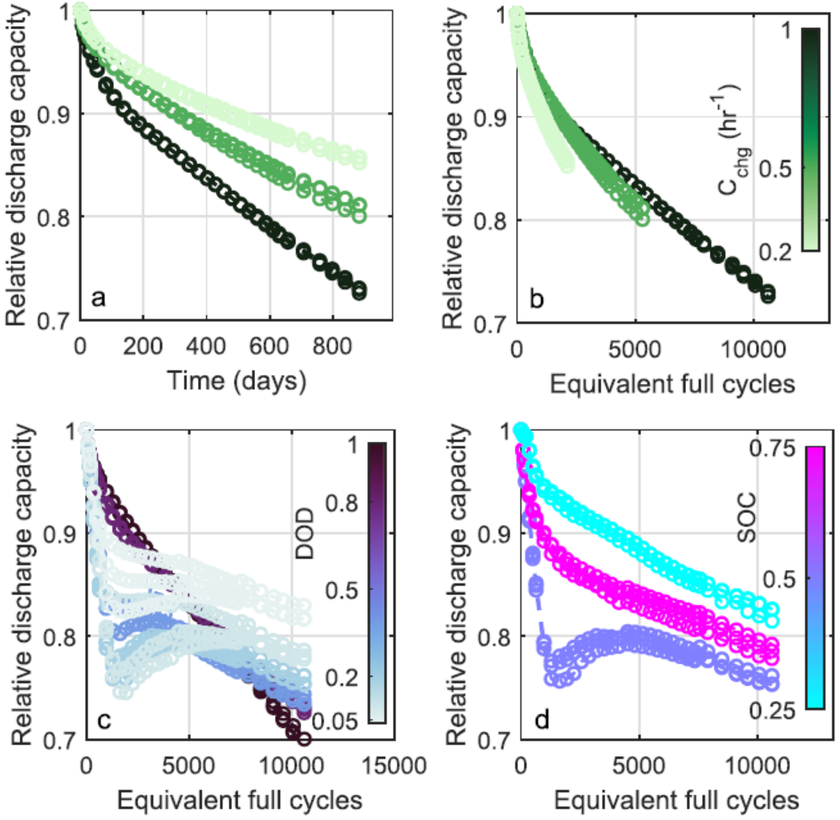


Uncertainty of the trajectory

The challenge for accurate battery life prediction

Wide variety of calendar and cycle aging trends make identification of parsimonious expressions difficult





The challenge for accurate battery life prediction

There's no clear 'best practice' from literature, i.e., each fitting problem is unique.

Reference	Description	Equation	Independent variable
Various authors[36,48–50,51]	Linear	$y = 1 - \beta_1 \cdot X$	$X = Ah$,[48,50,51] $X = Ah_{Dis}$,[49] $X = Ah_{Chg}$[36]
Takei,[52] Smith[9]	Linear	$y = \beta_0 - \beta_1 \cdot X$	$X = N$
Various authors[32,36,44]	Square root	$y = 1 - \beta_1 \cdot \sqrt{X}$	$X = Ah$,[32,36] $X = Ah_{Chg}$,[36] $X = t$[44]
Various authors[17,29,41,47,51,53–57,43,58,59]	Power law	$y = 1 - \beta_1 \cdot X^{\beta_2}$	$X = Ah$,[17,29,47,53–57,51,58,43] $X = t$,[41] $X = N$[59]
Stadler[60]	Power law	$y = \beta_0 - \beta_1 \cdot X^{\beta_2}$	$X = Ah$
Baghdadi[45]	Stretched exponential	$y = \beta_0 \cdot \exp(\beta_1 \cdot X^{\beta_2})$	$X = t$
Cuervo-Reyes[61]	Stretched exponential	$y = \beta_0 \cdot \exp\left(-\left(\frac{X}{\beta_1}\right)^{\beta_2}\right)$	$X = N$
Ecker[44]	Logarithm	$y = 1 - \beta_1 \cdot \log X$	$X = t$
Gering[62]	Sigmoidal	$y = 1 - 2 \cdot \beta_1 \cdot \left[\frac{1}{2} - \frac{1}{1 + \exp(\beta_2 \cdot X)^{\beta_3}}\right]$	$X = t$
Smith[9]	Site loss	$y = [\beta_0^2 - 2 \cdot \beta_1 \cdot \beta_0 \cdot X]^{\frac{1}{2}}$	$X = N$
de Hoog,[63] Hosen[64]	Polynomial	$y = 1 - \sum_{i=0}^{3} \beta_{1,i} \cdot X_1^i - \sum_{j=0}^{3} \beta_{2,j} \cdot X_2^j$	$X_1 = Ah$, $X_2 = DOD$

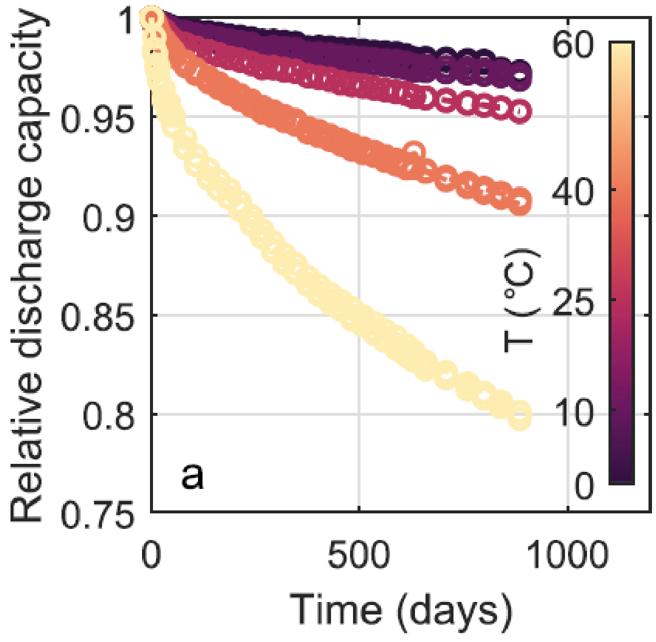
The challenge for accurate battery life prediction

There's no clear 'best practice' from literature, i.e., each fitting problem is unique.

References	Equation
Alhaider[48]	$(\gamma_1 \cdot DOD + \lvert SOC - 0.5 \rvert \cdot \gamma_2) \cdot \exp(\rho_{Ah} \cdot C_{rate})$
Baghdadi[45]	$\exp\left(\exp\left(\frac{\gamma_1}{T} + \gamma_2\right) \cdot C_{rate}\right)$
Cordoba-Arenas[51]	$[\gamma_1 + \gamma_2 \cdot Ratio^{\gamma_3} + \gamma_4 \cdot (SOC_{min} - SOC_0)^{\gamma_5}] \cdot \exp\left(\frac{\gamma_6}{T}\right)$ $[\gamma_1 + \gamma_2 \cdot (SOC_{min} - SOC_0)^{\gamma_3} + \gamma_4$ $\cdot \exp(\gamma_5 \cdot (C_{Chg,0} - C_{Chg})) + \gamma_6 \cdot (SOC_{min} - SOC_0)] \cdot \exp\left(\frac{\gamma_7}{T}\right)$
de Hoog,[63] Hosen[64]	$Q = 1 - \sum_{i=0}^{3} \beta_{1,i} \cdot Ah^i - \sum_{j=0}^{3} \beta_{2,j} \cdot DOD^j$
Diao[46]	$\exp(\gamma_1 \cdot T + \gamma_2)$ $\gamma_1 \cdot T + \gamma_2$
Ebbesen,[47] Schimpe[36]	$\gamma_1 \cdot \exp\left(\frac{\gamma_2}{T}\right)$
Mathieu[41]	$exp\left(\gamma_1 + \frac{\gamma_2}{T} + \gamma_3 \cdot SOC + \gamma_4 \cdot I + \gamma_5 \cdot \frac{I}{T} + \frac{\gamma_6}{T^2} + \gamma_7 \cdot SOC^2\right)$
Naumann[29]	$(\gamma_1 \cdot C_{rate} + \gamma_2) \cdot (\gamma_3 \cdot (DOD - 0.6)^3 + \gamma_4)$
Petit[58]	$\gamma_1 \cdot exp\left(\frac{\gamma_2 + \gamma_3 \lvert I \rvert}{T}\right)$
Sarasketa-Zabala[43]	$\gamma_1 \cdot DOD^2 + \gamma_2 \cdot DOD + \gamma_3$ $\gamma_1 \cdot \exp(\gamma_2 \cdot DOD) + \gamma_3 \cdot \exp(\gamma_4 \cdot DOD)$
Saxena[53]	$\gamma_1 \cdot SOC \cdot (1 + \gamma_2 \cdot DOD + \gamma_3 \cdot DOD^2)$
Schimpe[36]	$\gamma_1 \cdot \exp\left(\frac{\gamma_2}{T} + \gamma_3 \cdot C_{Chg}\right)$
Schmalstieg[32]	$\gamma_1 + \gamma_2 \cdot (V - \gamma_3)^2 + \gamma_4 \cdot DOD$ $exp(\gamma_2 \cdot DOD^{\gamma_3})$
Smith[9]	$1 + \gamma_1 \cdot DOD$ $\gamma_1 \cdot \exp\left(\frac{\gamma_2}{T}\right) \cdot DOD^{\gamma_3}$
Stadler[60]	$Q_{Loss@Ah} = \gamma_1 + \gamma_2 \cdot Ratio + \gamma_3 \cdot T^2 + \gamma_4 \cdot T + \gamma_5 \cdot Ratio^2 + \gamma_6 \cdot SOC_{max}^2 + \gamma_7 \cdot SOC_{min} + \gamma_8 \cdot SOC_{min}^2 + \gamma_9 \cdot P_{Chg}^2 + \gamma_{10} \cdot SOC_{max} \cdot Ratio$ $+ \gamma_{11} \cdot Ratio \cdot P_{Chg} + \gamma_{12} \cdot T \cdot SOC_{max} + \gamma_{13} \cdot P_{Chg} + \gamma_{14} \cdot SOC_{max} + \gamma_{15} \cdot T \cdot Ratio + \gamma_{16} \cdot SOC_{max} \cdot SOC_{min} + \gamma_{17} \cdot T \cdot P_{Chg}$
Suri[54]	$(\gamma_1 \cdot SOC + \gamma_2) \cdot \exp\left(\frac{\gamma_3 + \gamma_4 \cdot C_{rate}}{T}\right)$
Todeschini[55]	$\gamma_1 + \gamma_2 \cdot DOD + \gamma_3 \cdot \exp(C_{rate})$
Uddin[17]	Linear interpolation by SOC_{max}, DOD, C_{Chg}, and C_{Dis} between test points
Wang 2011[56]	$\gamma_1 \cdot \exp\left(\frac{\gamma_2 + \gamma_3 \cdot C_{rate}}{T}\right)$
Wang 2014[50]	$(\gamma_1 \cdot T^2 + \gamma_2 \cdot T + \gamma_3) \cdot \exp((\gamma_4 \cdot T + \gamma_5) \cdot C_{rate})$

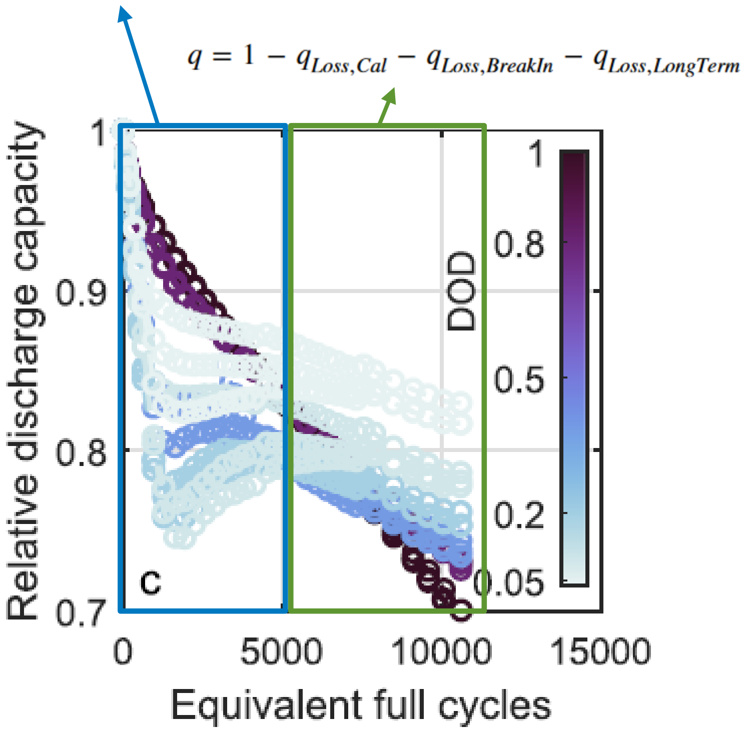
Split the data into additive or competitive states

$$q = 1 - q_{Loss,Cal}$$



$$q = 1 - q_{Loss,Cal} - q_{Loss,BreakIn}$$

$$q = 1 - q_{Loss,Cal} - q_{Loss,BreakIn} - q_{Loss,LongTerm}$$



Fitting calendar fade – Bilevel optimization

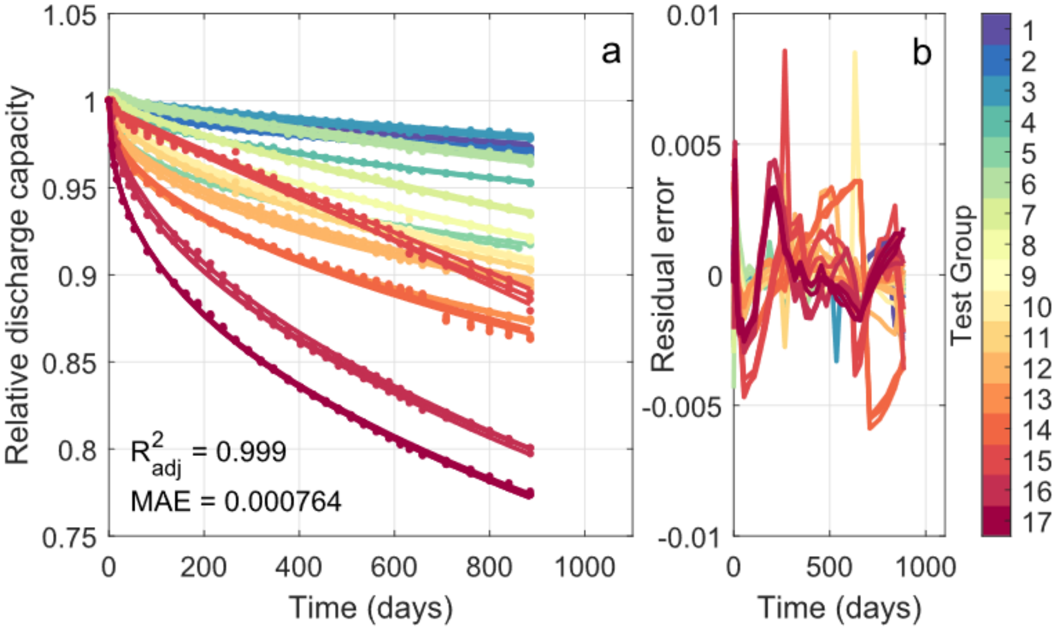
$$q_{Loss,Cal} = 2 \cdot q_1 \cdot \left[\frac{1}{2} - \frac{1}{1 + exp\left((q_2 \cdot t)^{q_3}\right)}\right]$$

Local parameter (stress dependent)

Inner optimization loop

Global parameter (cell dependent)

Outer optimization loop

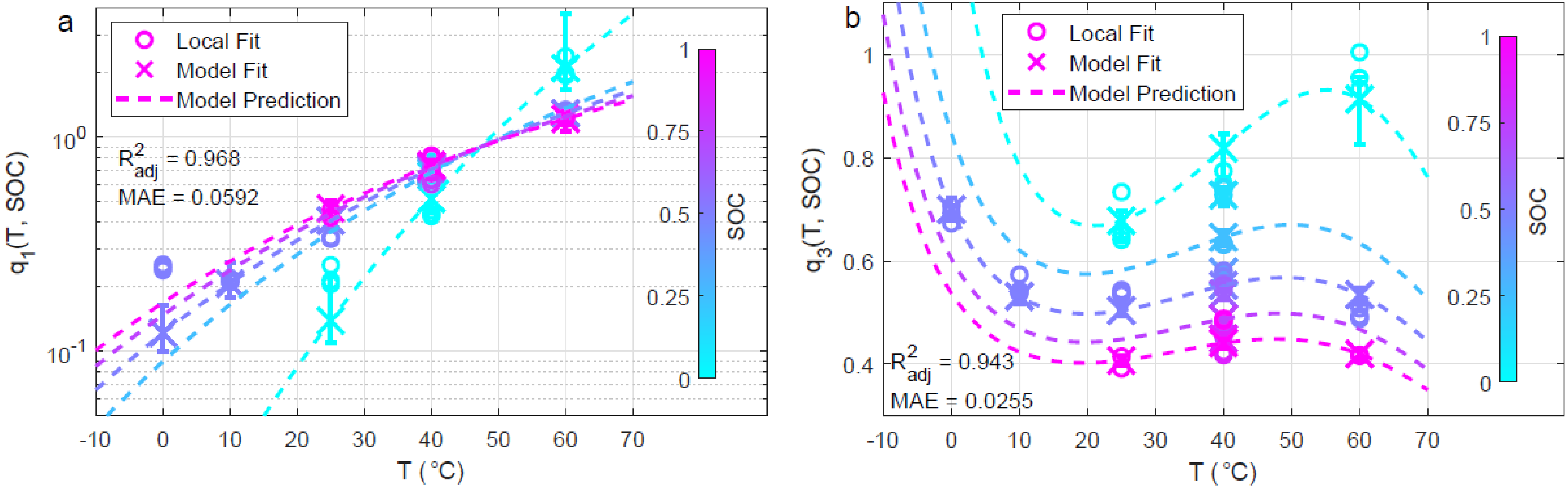


Fitting calendar fade – Symbolic regression

$$q_{Loss,Cal} = 2 \cdot q_1 \cdot \left[\frac{1}{2} - \frac{1}{1 + exp\left((q_2 \cdot t)^{q_3}\right)}\right]$$

$$q_1 = q_{1,a} \cdot exp\left(q_{1,b} \cdot \frac{U_a^{\ 0.5}}{T^2}\right) \cdot exp\left(q_{1,c} \cdot \frac{U_a^{\ 0.5}}{T}\right)$$

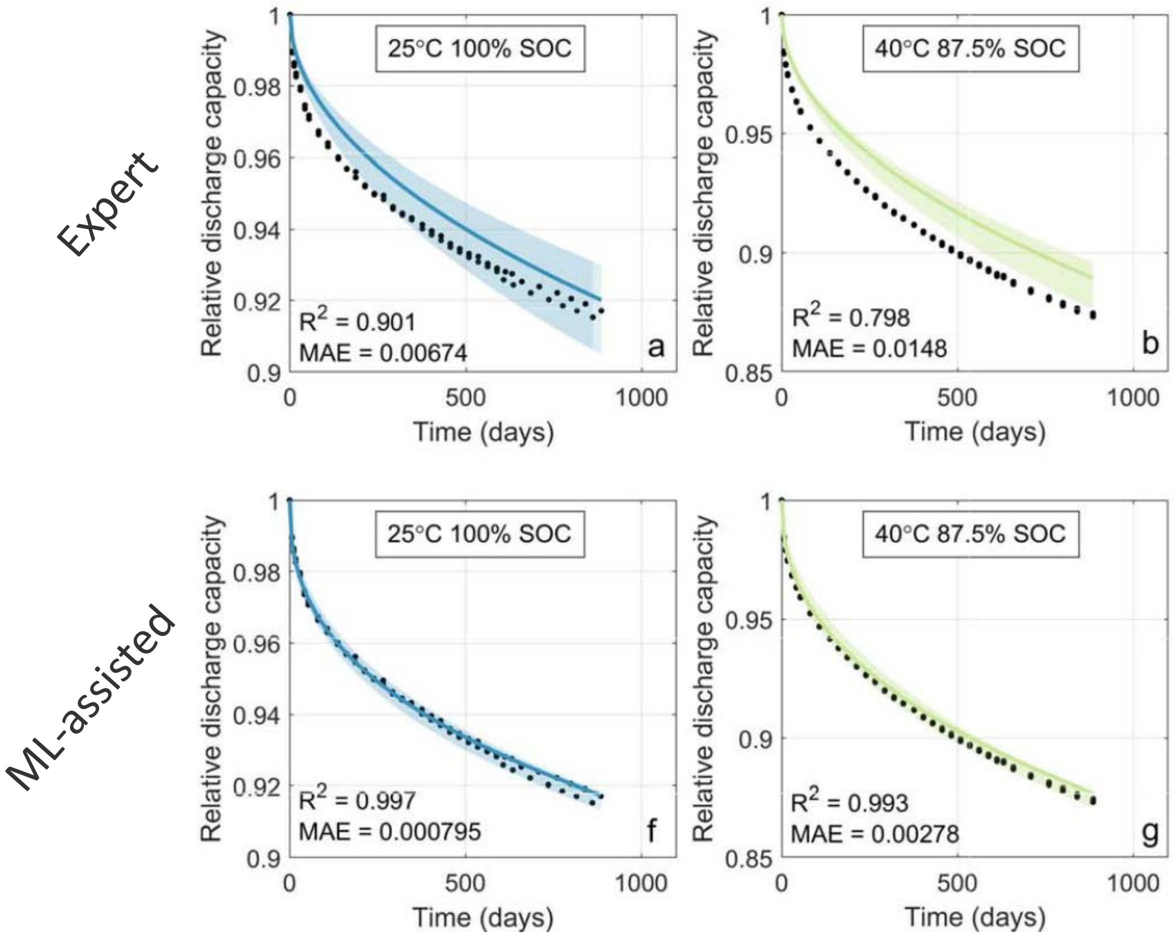
$$q_3 = q_{3,a} \cdot exp\left(q_{3,b} \cdot \frac{U_a^{\ 1/3}}{T^4}\right) \cdot exp\left(q_{3,c} \cdot T^3 \cdot U_a^{\ 1/4}\right) \cdot exp\left(q_{3,d} \cdot \frac{U_a^{\ 1/3}}{T^3}\right) \cdot exp\left(q_{3,e} \cdot T^2 \cdot U_a^{\ 1/4}\right)$$



Results - Calendar

ML-assisted model identification fits all test cases more accurately than the expert model.

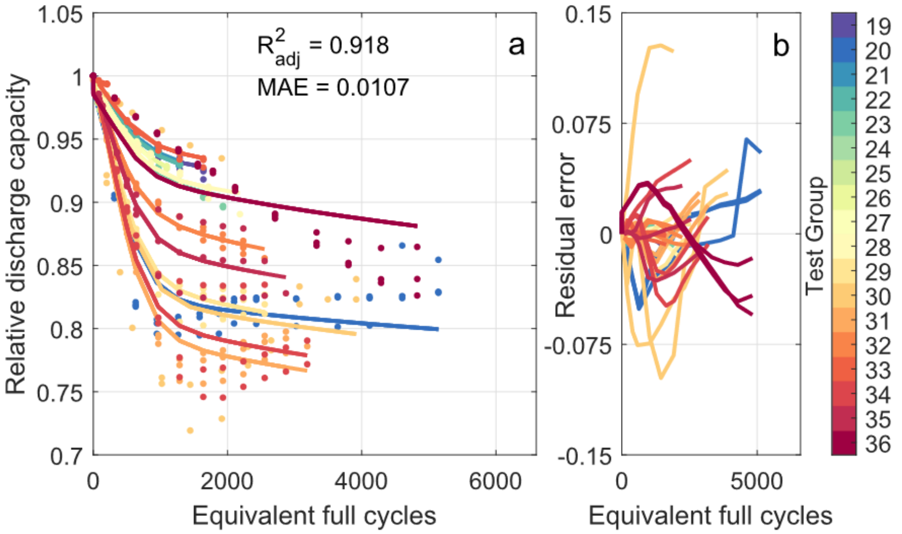
Fit at extreme values of temperature and SOC is much improved.

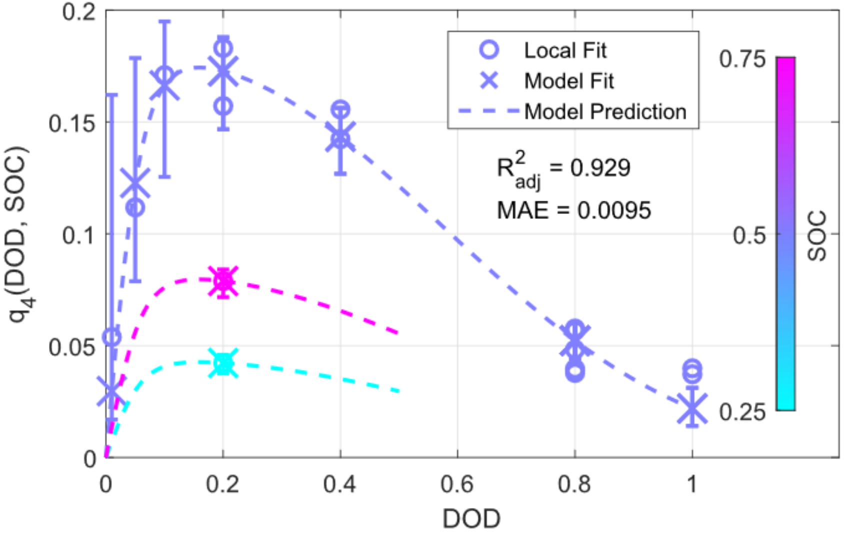


Fitting cycling break-in fade

$$q = 1 - q_{Loss,Cal} - q_{Loss,BreakIn}$$

$$q_{Loss,BreakIn} = 2 \cdot q_4 \cdot \left[\frac{1}{2} - \frac{1}{1 + exp((q_5 \cdot EFC)^{q_6})} \right]$$

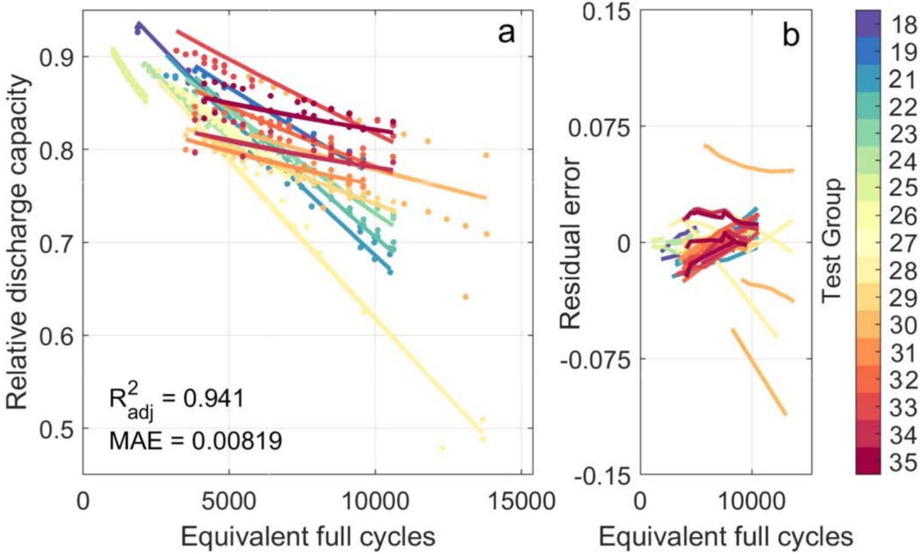


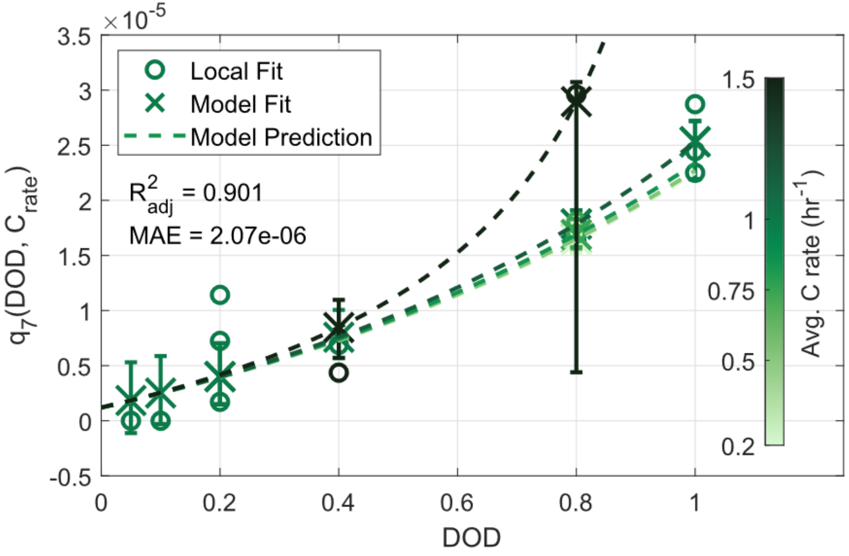


Fitting long-term cycling fade

$$q = 1 - q_{Loss,Cal} - q_{Loss,BreakIn} - q_{Loss,LongTerm}$$

$$q_{Loss,LongTerm} = (q_7 \cdot EFC)^{q_8}$$



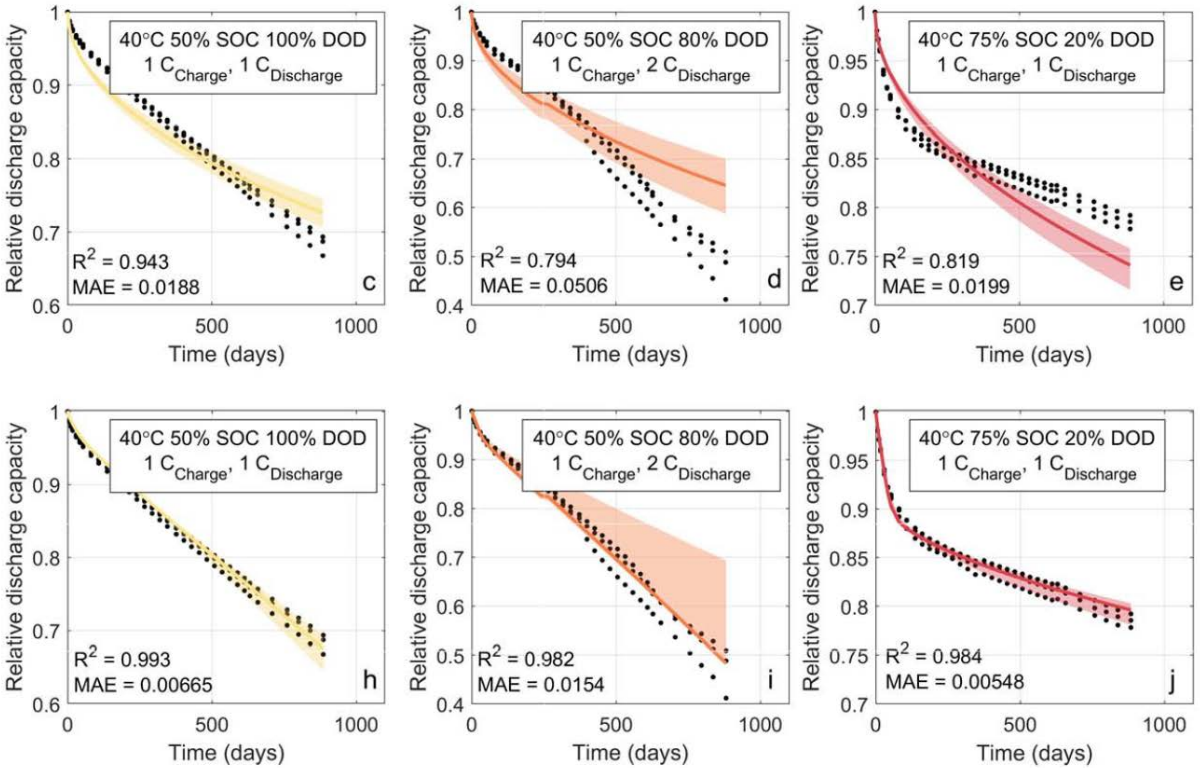


Results - Cycling

ML-assisted model identification fits all test cases more accurately than the expert model.

Fit at extreme values of DOD and C_{Rate} is much improved.

Expert



ML-assisted

Predicting degradation during dynamic use

Invert

Linearize

Accumulate

$$x^* = f^{-1}\left(y_{t-1}, \mathbf{S}\right)$$

$$\delta y_t = \frac{\mathrm{d}f\left(f^{-1}\left(y_{t-1}, \mathbf{S}\right), \mathbf{S}\right)}{\mathrm{d}x} \cdot \delta x$$

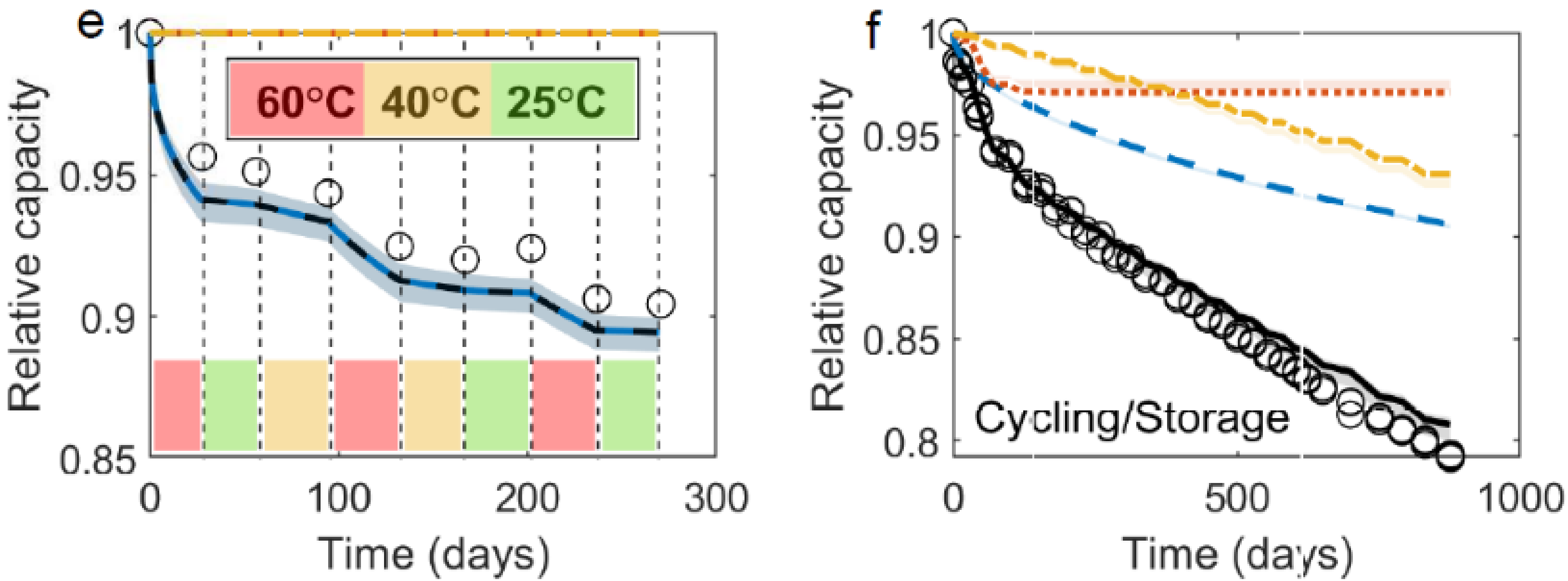
SEI growth rate is not dependent on time passed, but rather on current SEI thickness

t_0, t_1, ……

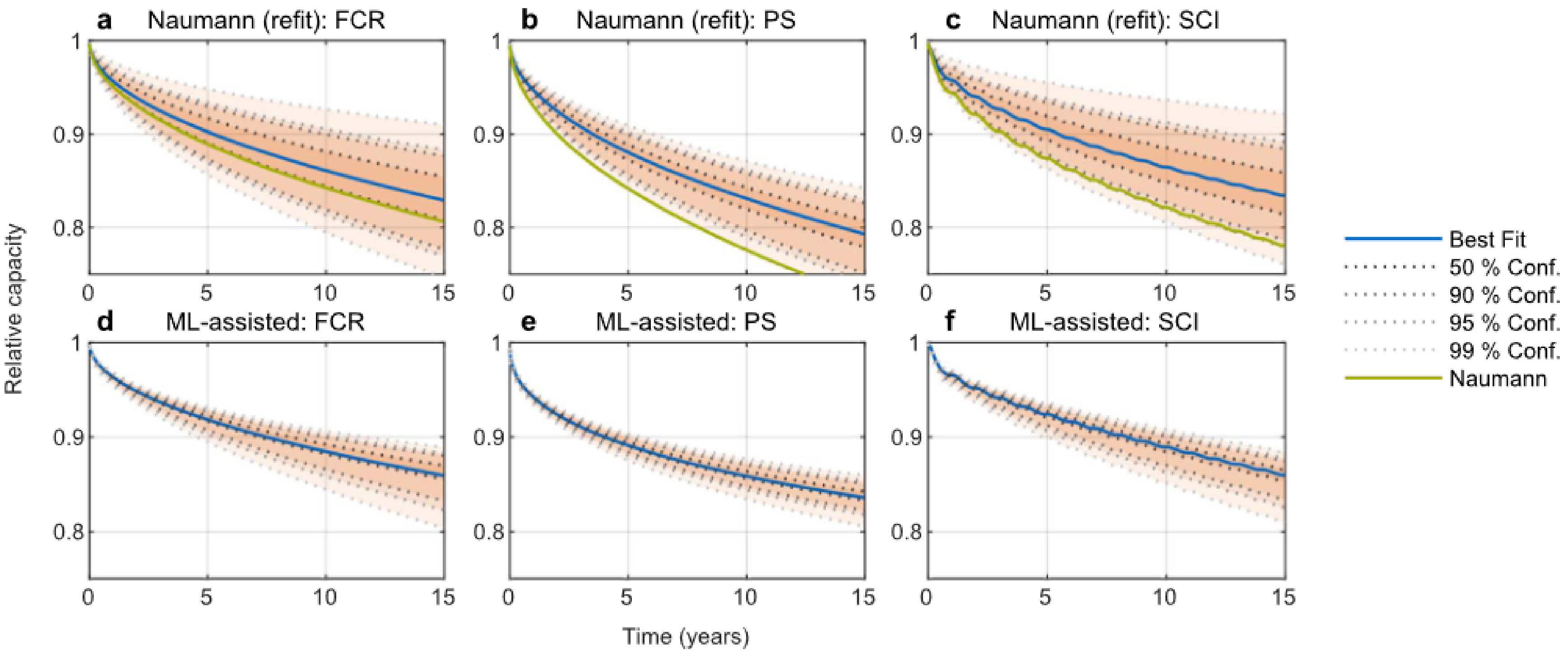
……. t_{n-1}, t_n

Degradation per day or per cycle can be linearized

Predicting degradation during dynamic use



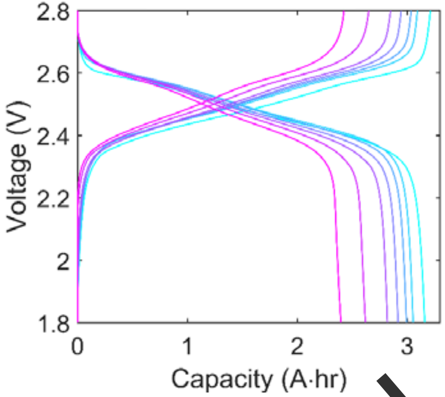
Incorporation into techno-economic simulation (SimSES)



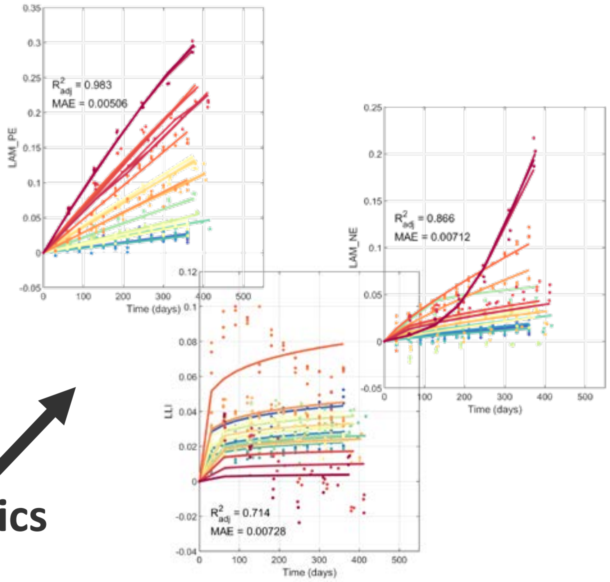
Credit: Nils Collath, Holger Hesse, Andreas Jossen

Incorporation into electrochemical models

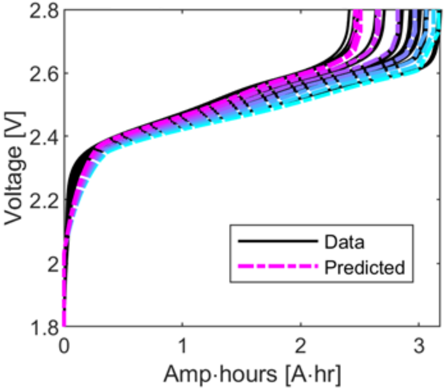
Aging data



Machine-learn degradation trajectory models



Predict cell voltage response throughout lifetime



Diagnostics

LLI

LAM_{PE}

LAM_{NE}

Dual-tank model

Thank you!

www.nrel.gov

NREL/PR-5700-86369

This work was authored [in part] by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by the U.S. Department of Energy Office of Energy Efficiency and Renewable Energy Vehicle Technologies Office. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

NREL Transforming ENERGY

Optimization algorithms

Local   Global (assumed constant)

$$q = 1 - \boxed{\beta_1} t^{\boxed{0.5}}$$

1. Bi-level (nested) optimization
   - Local parameters correspond to unique behaviors of each cell
   - Global parameters correspond to behaviors shared by all cells
2. Symbolic regression [2,3]
   - Algorithmically generate descriptors from input features
   - Find optimal subset of descriptors using LASSO regularization
   - Both linear and multiplicative models are searched

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots$$

$$\exp(\log(Y)) = exp(\beta_0 + \beta_1 X_1 + \beta_2 \log(X_2) + \cdots)$$

$$Y = exp(\beta_0) exp(\beta_1 X_1) \cdot X_2^{\beta_2} \cdot \cdots$$

Symbolic regression overview

SISSO: Ouyang et. al.:
https://doi.org/10.1103/PhysRevMaterials.2.083802

Fortran, Matlab, Python [1, sklearn: 2]

1: Input features

A: **T**
B: **SOC**

2: Apply operators to generate new features

A: {**T**, $\mathbf{T}^2$}
B: {**SOC**, $\mathbf{SOC}^2$}

A: {**T**, $\mathbf{T}^2$}
B: {**SOC**, $\mathbf{SOC}^2$}
C: {$\mathbf{SOC/T}^2$,…}

Feature matrix is very wide, with many highly correlated features

3: Search for the subset of features that model the data (regularization)

$$Y = \beta_0 + \beta_1 \mathbf{SOC} + \beta_2 \mathbf{SOC/T}^2 + \cdots$$

This search has combinatorial complexity:
(1000 choose 5) = **$8\cdot10^{12}$**

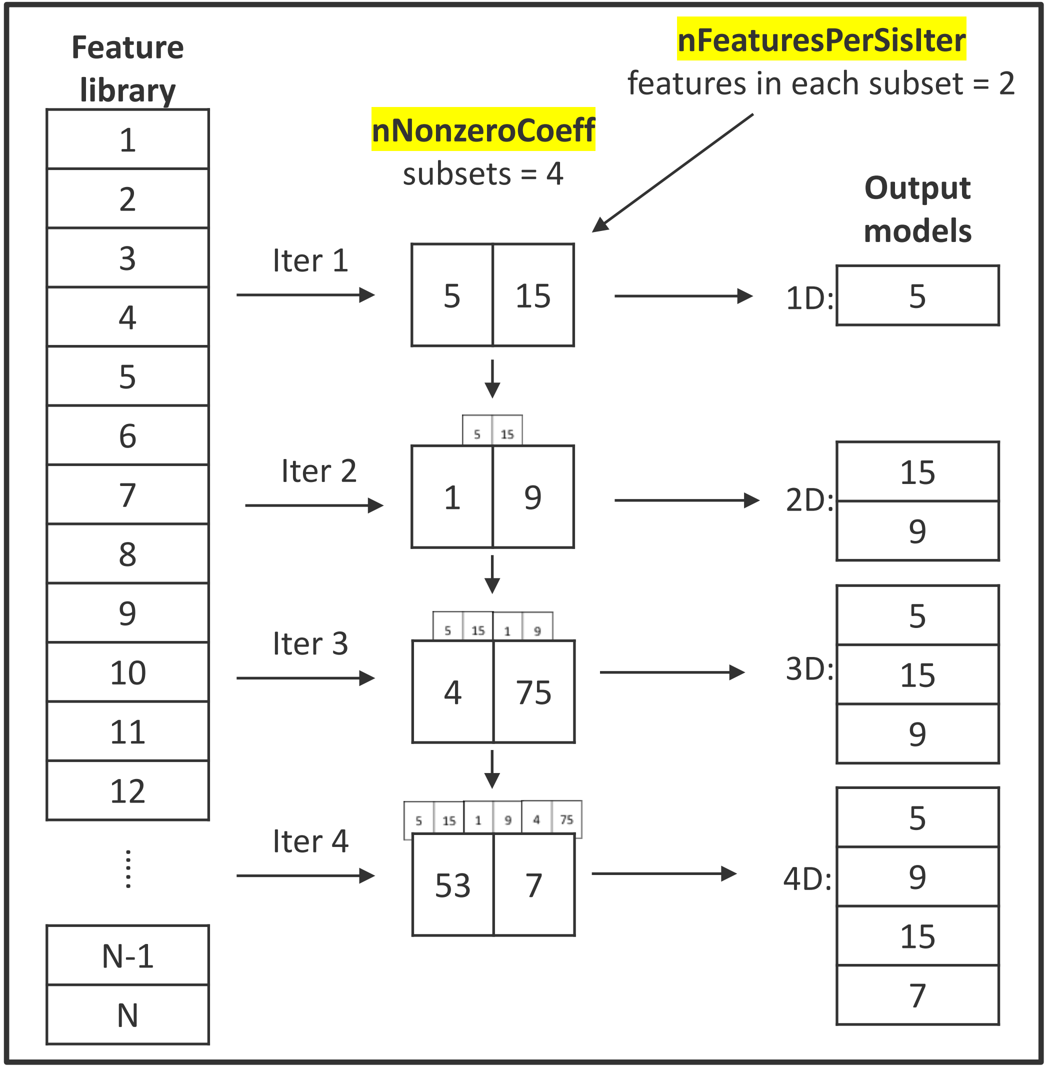
Linear

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots$$

Multiplicative

$$\exp(\log(Y)) = exp(\beta_0 + \beta_1 X_1 + \beta_2 \log(X_2) + \cdots)$$

$$Y = exp(\beta_0) exp(\beta_1 X_1) \cdot X_2^{\beta_2} \cdot \cdots$$



Iteration 1 feature selection is done based on correlation ranking to **target**

Iteration >1 feature selection is done based on correlation ranking to **residuals from prior iteration**