



Can a day of CAN data say anything about a battery's future?

Yes, within stated limits — because the method is Kennedy–O'Hagan model calibration, not curve fitting. And how the MW Battery Engine (v11c) does it today.

In brief

- Battery degradation is non-linear. A 12–24 hour log cannot *learn* that curve, and the engine does not ask it to.
- The non-linear path comes from **physics and a synthetic fleet** (PyBaMM-style models): the prior. The log **locates** this cell on that family of paths: its duty cycle, resistance, imbalance, capacity spread and present health.
- That is **Kennedy–O'Hagan (KOH) calibration** (2001): an imperfect physics model, unknown physical constants, a discrepancy term for what the physics misses, and few field observations.
- The engine reports a **range**, not a number, and says when the data cannot resolve something.
- **The 11c app runs this end to end today**: physics checks on the raw log, per-cell true degradation, an anchored and monotone manifold walk, posterior RUL intervals, lever ranking against posterior σ , and fleet processing.

1 • The question a battery scientist asks

Show a battery scientist a remaining-life forecast made from a day of BMS data and the reply is predictable:

“Batteries are non-linear electrochemical systems. Most battery scientists would not accept that a short CAN log can predict state of health and remaining useful life. Reviewers will not either.”

The objection is correct, and it would be fatal if the engine fitted a degradation curve to a few hours of data. Twelve hours contain almost no fade: at one cycle a day, a cell loses a few thousandths of a percentage point of capacity. No method can read a knee from that. The engine does something different, and statisticians have a well-studied name for it.

2 • The answer in one equation

Kennedy and O'Hagan (2001) asked how to combine an expensive, non-linear computer model with a handful of real-world measurements. Their answer writes each observation as

$$y(x) = \eta(x, \theta) + \delta(x) + \epsilon$$

Term	In KOH	In the 11c engine
x	inputs, conditions	the cell's 16-parameter state from CAN: current, voltages, temperature, SoC, DoD, C-rate, resistance, cycle count, calendar age
$\eta(x, \theta)$	the physics simulator	the MW manifold: a Pyro VAE trained on a large synthetic fleet under degradation priors — $\sqrt{t}$ SEI growth, cycling stress, Arrhenius, plating, swelling, cracking
θ	physical constants	posteriors carried in the checkpoint (activation energy, rate and stress coefficients), shown with their uncertainty in the Physics Lab
$\delta(x)$	model discrepancy	per-cell true degradation , estimated from the log: capacity ratio and fade-rate multiplier, each with uncertainty
ϵ	measurement noise	per-feature observation noise; BMS resolution and IR floor reported per log

The non-linearity lives in η . It is fixed by physics and by thousands of simulated cells before this battery is seen. The log answers an easier question: *where on this family of physically allowed curves does this cell sit, and how does it differ from the typical member?* A map and one position fix will place you on a winding road. Here the physics is the map and the CAN log is the fix.

3 • What a short log tells you — and what 11c does with it

Quantity	From one 12–24 h log?	How 11c handles it
Present SoH	Yes	every forecast path is anchored to the measured SoH; nothing starts from a guess
Per-cell DCIR and dR/dT	Yes, from current steps	temperature-normalised to 25 °C, 300-sample bootstrap CI per cell
Duty cycle	Yes	temperature, DoD swings, C-rate, rest share and cycles/day become the forecast’s default duty
Capacity spread vs SoC imbalance	If the log reaches both ends of the SoC swing	separated by a weighted fit: imbalance shows the same sign at both ends, capacity loss opposite signs. Otherwise the app says <i>not resolvable</i> and does not invent a difference
This cell’s fade rate	No — almost no fade in a day	prior from the manifold, scaled per cell by its DCIR, with a double-count guard against what the manifold already encodes
Knee onset	No	prior conditioned on duty and resistance; a repeat log adds resistance growth, an early knee signal

So one log gives **today’s state tightly and the future loosely**, and the posterior says so. A repeat log adds the cell’s own rate once enough has changed to stand above sensor noise — typically weeks to months, depending on throughput. The most useful addition is **resistance growth**: a resistance “elbow” is reported to precede the capacity knee (Fermín-Cueto et al., 2020).

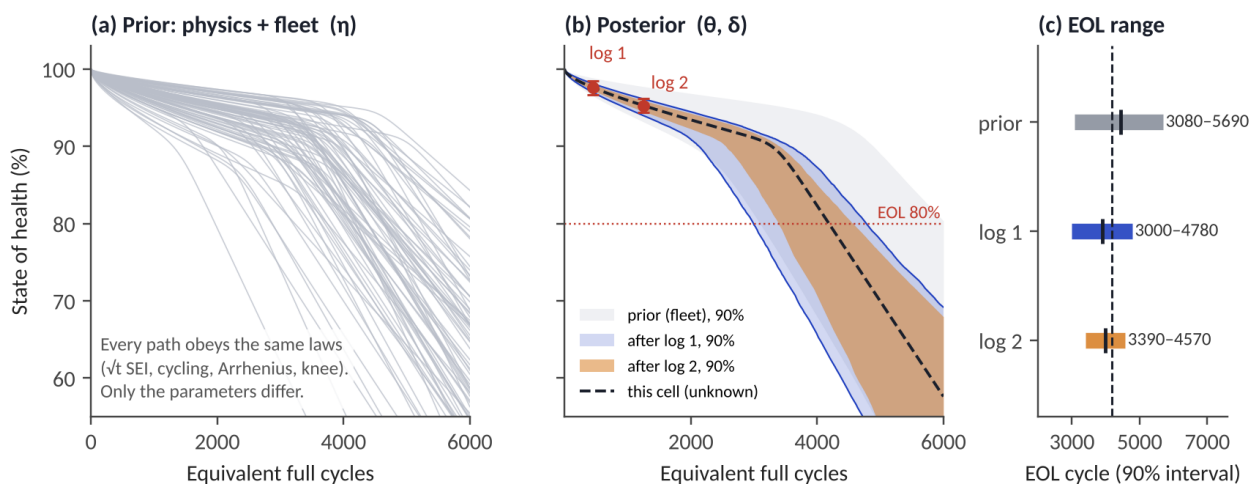


Figure 1 — Schematic of the logic, not a result. A toy model with 40,000 synthetic cells. (a) The prior η : every path obeys the same laws; parameters differ. (b) Conditioning on log 1 (duty, DCIR, present SoH) keeps only paths consistent with this cell; log 2 adds SoH change and DCIR growth. (c) The 90% end-of-life interval narrows from 3,080–5,690 cycles (prior) to 3,000–4,780 (one log) to 3,390–4,570 (two logs); the dashed line is the true value.

4 • How the 11c app answers each objection

Every row below is a feature in the current app, not a plan.

A battery scientist asks	What 11c does today
Is the log itself trustworthy?	Before any model runs, nine physics checks on the raw log, per cell and per pack: Ohmic response (MP1), Arrhenius resistance (MP2), OCV $\leftrightarrow$ SoC at rest (MP3), coulomb closure $\int I dt$ vs ΔSoC (MP4), $\text{SoH} \leq 1$ (MP5), chemistry voltage window (MP6), balance at rest (MP7), Joule heating vs thermal gradient (MP8), relaxation to OCV (MP9). An admissibility score summarises them.
Batteries are non-linear.	The curve comes from the manifold, not the log. The life forecast walks each cell forward in cycles with calendar age advancing at the log’s own pace; decoded SoH change is added to the measured SoH.
Capacity cannot recover.	Every path is constrained to be non-increasing. The unclamped decoder is kept and the number of steps where it would have risen is reported (MP10, L7), so the constraint is visible, not hidden.

A battery scientist asks	What 11c does today
Every cell is different.	Per-cell true degradation: each cell's path is its present SoH A minus its fade multiplier m times the manifold's fade. A comes from end-of-swing capacity, m from DCIR, both shrunk across cells so noise is not mistaken for signal. The pack limit is the weakest cell, which can differ from the one the manifold alone picks. On a synthetic 16-cell log with planted differences, 11c recovered each capacity offset to within 0.13 percentage points and kept a 3% SoC imbalance out of the capacity estimate.
How sure are you?	Coherent posterior sample paths per forecast (48 by default, up to 128). EOL is reported as a mean, a 90% credible window, $P(\text{EOL within 1,000 cycles})$, and the share of paths extrapolated past the horizon.
Is the cell-level difference real?	The true line is compared with the manifold at the horizon. If the gap is inside posterior σ , the app says the two are not distinguishable. If the log has no per-cell evidence, it says so and why.
Which operating lever matters?	Virtual Lab: sweep temperature, SoC, DoD, C-rate, pace or calendar age, one at a time, linked or as a grid, on paired posterior paths. A lever counts as resolved only if its effect exceeds posterior σ . Values outside the checkpoint's training range are flagged as extrapolation. A separate attribution chart shows how many cycles each driver adds or removes.
Is this state even inside what the model knows?	MW distance — the standardised off-manifold residual — per cell and through the log, split by feature tier; a training-range check on every input.
Does it scale beyond one pack?	Quick mode runs the physics checks and the per-cell estimate on the CPU, with no neural network. Fleet upload takes up to 100 logs at once. A converter maps any CAN layout to the expected one: an LLM writes a JSON mapping, which is applied deterministically, so no generated code ever runs.

5 • The knee, and where MW goes beyond classic KOH

Capacity fade has two regimes: slow, roughly $\sqrt{t}$ SEI growth, then acceleration once plating or particle cracking switches on (Attia et al., 2022). Classic KOH places Gaussian processes on η and δ ; a GP is smooth by construction and blurs an on/off switch. MW puts the switch in the prior as a regime the inference can represent directly. The decoder of the trained manifold induces a Riemannian metric $G(z) = J^T W J$ (Arvanitidis et al., 2018). Where the regime switches, the same step in time produces a much larger step in SoH: the metric stretches and the manifold folds. The walk follows the fold, and its projection onto the (cycles, SoH) plane is the knee. The app reports a knee only when a two-segment fit clearly beats a straight line ($F > 10$) and the second segment is steeper.

	Classic KOH	MW (11c)
Priors	Gaussian processes	physical priors: LogNormal rate constants, Beta fractions, Bernoulli regimes
Cost	$O(N^3)$ in data points	amortised: one encoder–decoder pass per state, independent of fleet size
Discrepancy	one $\delta(x)$	per cell (capacity, rate) and per physics law (MP1–MP10), each readable
Geometry	none	learned manifold; knee as curvature; off-manifold states flagged

For statisticians: Brynjarsdóttir and O'Hagan (2014) showed that θ and δ are confounded when priors are weak. Physically grounded priors are the standard remedy. In MW they are what makes the calibration identifiable, and 11c adds an explicit guard so the per-cell term does not re-count what the manifold already explains.

6 • What the engine claims, and what it does not

It delivers

- Present SoH and per-cell state with credible intervals, from one log.
- RUL as a posterior interval at a stated duty cycle.
- Intervals that narrow as repeat logs arrive, and an explicit flag when they cannot.
- Physics checks on the data before any model runs.

It does not claim

- An exact knee cycle from a single day of data.
- Electrochemical mechanism identification — that needs RPTs, dQ/dV and post-mortem.
- To replace laboratory testing; it works between tests.
- To be a safety system. It cannot detect thermal runaway or internal shorts.

7 • What it takes to run 11c on your cells

You provide	11c returns
• One BMS / CAN log, 12–24 h, in any column layout (the converter maps it) • Label data: calendar age, rated cycles, capacity, chemistry • Ideally a charge reaching SoC ≥ 0.8 and a discharge to ≤ 0.3, with rests at both ends — this is what lets per-cell capacity resolve • Optionally, a repeat log weeks to months later	• Nine physics checks on the log, with an admissibility score • Per-cell DCIR at 25 °C with confidence intervals, relaxation, divergence, thermal and usage profiles • Per-cell SoH, capacity spread and SoC imbalance, or a plain statement that they are not resolvable • Pack life forecast: EOL and RUL with credible intervals, the limiting cell, and a knee when one is identified • Lever ranking against posterior σ, and a written verdict

On the theory. The method is ready to use. The one open question is empirical calibration on a partner’s cells: blind predictions from 11c, filed before the cells reach end of life, checked for whether the 90% intervals contain the outcome about 90% of the time. The app already produces everything that study needs.

References

Kennedy, M. C. & O’Hagan, A. (2001). Bayesian calibration of computer models. *J. R. Stat. Soc. B*, 63(3), 425–464.

Brynjarsdóttir, J. & O’Hagan, A. (2014). Learning about physical parameters: the importance of model discrepancy. *Inverse Problems*, 30, 114007.

Arvanitidis, G., Hansen, L. K. & Hauberg, S. (2018). Latent space oddity: on the curvature of deep generative models. *ICLR*.

Attia, P. M. et al. (2022). Review — “Knees” in lithium-ion battery aging trajectories. *J. Electrochem. Soc.*, 169, 060517.

Fermín-Cueto, P. et al. (2020). Identification and machine learning prediction of knee-point and knee-onset in capacity degradation curves of lithium-ion cells. *Energy and AI*, 1, 100006.

Sulzer, V. et al. (2021). Python Battery Mathematical Modelling (PyBaMM). *J. Open Research Software*, 9(1), 14.

Modak, R. & Walawalkar, R. *Formal Theory of the Modak–Walawalkar Framework as Geometric Intelligence*. doi:10.5281/zenodo.19304813.