batteries

MDPI

Article

What Limits Lithium-Ion Battery State of Charge Estimation with a Hybrid Large Language Model? Layer Type Aware Adaptation and the Initial Condition Bottleneck

Alexey A. Druzhinin

ReCharge Lab, Moscow Center for Advanced Studies, 123592 Moscow, Russia; drusalexey@gmail.com

Abstract

State of charge (SoC) is the unobserved quantity that a battery management system must reconstruct from terminal voltage, current and temperature. Recent estimators adapt frozen large language models by prompt-tuning the key–value cache of a dense decoder, an interface that does not exist in the linear-attention majority of current hybrid models. We adapt a frozen four-billion-parameter hybrid decoder by matching the injection to each layer type: learned key–value prefixes in the full-attention minority, and a trainable initial recurrent state (S_0) in the gated-delta majority. Training only 22.3 million parameters (0.55%), this layer-type-aware adaptation reduces mean absolute error (MAE) by a factor of 2.3 relative to prefixes alone on a preliminary single-cell diagnostic. On the final four-billion-parameter configuration, the same comparison yields a factor of 1.86: pooled mean absolute error decreases from 2.18% for key–value prefixes alone to 1.17% for the full adapter. The nickel-based drive-cycle pool is 1.17% MAE on 18,712 windows. That pool does not include LFP. The LFP value on shared files is a fit of 1.22%. Held-out files of the same cell reach 2.53% in a separate run, and the nickel-based pool of that run is 1.24%. This represents parity with dedicated dense-LLM estimators without claiming superiority over compact conventional observers. On the same drive-cycle split, a gated recurrent unit with 0.2 million parameters reaches 1.18% pooled MAE, and the adapter reaches 1.17%. Instead, the primary contribution is mechanistic. We demonstrate that 98.4% of the residual error is a constant per-window offset on that diagnostic. On this configuration, a constant per-window offset accounts for 0.98 of the residual energy. The estimate therefore behaves as a near-exact Coulomb integration after an initial-condition inverse problem at $t = 0$. Compact models trained on the same windows show the same offset-dominated residual, so this pattern is a property of the short-window task as well as of the backbone. This factorization is consistent with the observed failure regimes—concentration of error below 0 °C, and a zero-shot LFP error of 12% MAE on the reference range of about 0–47% state of charge, about ten times the nickel-based pool. Held-out files of an LFP cell partly seen in training reach 2.53% in a separate run. Consequently, we argue that the engineering target for LLM-based battery estimation should transition from expanding backbone capacity to explicitly conditioning the initial state.

Keywords: state of charge; lithium-ion battery; large language model; parameter-efficient fine-tuning; linear attention; gated delta network; open-circuit voltage; cross-chemistry generalization

Check for updates

Academic Editor: Yong-Joon Park

Received: 4 September 2026
Revised: 3 October 2026
Accepted: 6 October 2026
Published: 15 October 2026

Copyright: © 2026 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Lithium-ion batteries underpin the electrification of transport and the integration of variable renewable generation, and responsibility for operating them safely and for a long service life falls almost entirely on the battery management system (BMS) [1]. A BMS observes only terminal voltage, load current and surface temperature, from which it must continuously infer internal states that no sensor reports directly [1,2]. The most consequential is the state of charge (SoC), the charge remaining relative to usable capacity, which sets range prediction, instantaneous power limits and the margins that keep a cell away from over-charge and over-discharge [3]. Because SoC is unobservable, every practical estimator solves an inverse problem: an internal quantity is reconstructed from an electrical signature that simultaneously carries polarization, hysteresis and temperature effects [3,4]. Model-based estimators address it explicitly by pairing an equivalent-circuit or electrochemical model with a Kalman-type observer [5], which treats the initial condition in a principled way but requires parameters to be reidentified for each chemistry, format and aging state [4]. Over the same period in which machine learning reshaped adjacent battery problems—predicting cycle life before capacity fade becomes measurable [6], optimizing fast-charging protocols in closed loop [7] and inferring lifetime from field data [8]—SoC estimation was accordingly reformulated as supervised sequence regression [9]. Hybrid models that combine a neural network with a physics-based description have also been developed for vehicle dynamics, where the network identifies parameters and characterizes the modeling error of a lateral-dynamics model [10] or is combined with a physics-based estimator to model longitudinal dynamics from limited driving data [11]. Those studies concern vehicle motion rather than battery state of charge. The present analysis uses a comparable separation between a physically determined component, the charge balance, and a learned component, the initial state of charge.

Networks trained directly on voltage–current–temperature (V/I/T) sequences reach errors of a few percent across wide temperature ranges without any explicit cell model, first with feed-forward architectures [9] and then with long short-term memory networks [12]. Because SoC is an integral of current, architectures with long effective memory are a natural fit [13], and attention mechanisms further improve the selective use of distant context [14]; reviews of the field document the resulting displacement of hand-tuned observers by learned estimators [3,15]. The principal limitation for deployment is generalization [15]. Performance degrades sharply below 0 °C, where internal resistance distorts precisely the voltage signature that carries most of the SoC information [16], and it transfers poorly between cells, so that each new chemistry or format requires retraining [17] or an explicit domain-adaptation stage [18]. This fragility motivates the use of a sequence prior learned outside the battery domain in place of one learned from a small number of cells [15].

Transformer language models [19] scaled to billions of parameters acquire in-context and few-shot capabilities that extend well beyond their training distribution [20], which has made them attractive as general-purpose priors over sequences. Numeric series can be presented to a frozen large language model (LLM) as text [21,22], mapped into its embedding space by a learned reprogramming layer [23,24], or set aside in favor of dedicated time-series foundation models trained on large forecasting corpora [25–27]. Whether language pretraining improves such tasks is disputed: position papers argue that it supplies transferable structure for temporal reasoning [28], whereas controlled ablations show that on several forecasting benchmarks, removing the language model from the pipeline leaves accuracy unchanged [29]. This disagreement sets a methodological requirement for the present work: a claim that an LLM benefits a physical estimation task must rest on mechanistic evidence about what the model computes, and not on an aggregate error table alone [29].

Within the battery literature, LLM-based state estimation has developed along a single axis. Bian et al. introduced mixed prompt learning, converting V/I/T measurements into hard text prompts and combining them with learned soft prompt vectors to elicit SoC estimates from a frozen pretrained LLM [30], and subsequently proposed a hard–soft hybrid prompt with a side adapter network that reparameterizes adaptation into a low-dimensional subspace, together with a knowledge-infusion mechanism that conditions hidden states along the forward pass, reporting robust estimation for multiple cell types and sub-zero temperatures [31]. Related studies fine-tune prompt-driven LLMs for battery state estimation [32], apply prompt-guided LLMs to containerized energy-storage systems [33], and couple an LLM with a multiscale interval transformer for state-of-health estimation [34]. Across this body of work, the adaptation surface is effectively fixed: the trainable interface is a text or soft-prompt representation acting on the key–value projections and hidden states of a dense, softmax-attention decoder [30,31,33]—a choice inherited from the architecture that was standard when the line of work began [19], and the assumption this paper revisits. To keep the comparison direct, we deliberately vary nothing else: the cells, the temperature range and the drive-cycle holdout protocol are those of [30].

The scope of the claims is defined first. Operation across multiple cell types and at sub-zero temperature is already established by [31]. An accuracy advantage is not claimed either: the present estimator reaches parity with [30], and on the Panasonic cell used by both, a conventional long short-term memory network reported 0.573% mean absolute error at fixed ambient temperature and 1.606% across a 10–25 °C range some years earlier [12]. No published LLM-based SoC estimator, ours included, has yet shown that a billion-parameter backbone improves on a recurrent network of a few hundred thousand parameters on these datasets. The contribution lies on a different axis—what part of a frozen language model can be adapted at all, what the adapted part turns out to compute, and why that computation, rather than model capacity, sets the accuracy ceiling. On this split, a gated recurrent unit trained from scratch reaches 1.18% pooled MAE, and the adapter reaches 1.17%.

Here, we adapt a frozen four-billion-parameter hybrid-attention LLM, Qwen3.5-4B [35], a successor of the Qwen3 models [36], which interleaves gated delta-rule linear-attention blocks [37] with full-attention blocks in a 3:1 ratio, to SoC estimation by matching the injection mechanism to the algebra of each layer type. Full-attention blocks receive learned key–value prefixes produced by a bidirectional recurrent encoder from learned soft positions that are conditioned on a per-cell descriptor, following deep prefix tuning [38,39]. Linear-attention blocks instead receive a trainable initial recurrent state, which supplies what a prefix cannot express in a layer that has no keys or values to prepend [37,40]. Every weight of the language model remains frozen, and 22 million parameters—0.55% of the backbone—are trained. The recurrent-state path is the dominant term: adding it to the key–value adapter reduces mean absolute error (MAE) by a factor of 2.3 on a single-cell benchmark. On the final four-billion-parameter drive-cycle configuration, the corresponding factor is 1.86, with pooled mean absolute error decreasing from 2.18% under key–value prefixes alone to 1.17% under the full adapter. A single adapted model is trained on five cells spanning nickel–cobalt–aluminium, nickel–manganese–cobalt and lithium–iron–phosphate chemistries [41–44] from −20 °C to 45 °C and reaches, on the four nickel-based cells, 1.17% MAE over 18,712 windows held out by drive cycle, on a par with a dedicated dense-LLM estimator [30] while exposing a far smaller adaptation surface.

Because the adapted interface is an explicit initial condition, the resulting estimator can be analyzed directly, and this analysis yields the central result of this study. Decomposing the residual shows that 98.4% of it is a constant per-window offset rather than trajectory error on the single-cell diagnostic. On the final configuration, a constant per-window

offset accounts for 0.98 of the residual energy, which means the recurrent stack integrates charge almost exactly, and nearly all remaining error resides in the estimated SoC at the window start. SoC estimation with a recurrent LLM therefore factorizes into an inverse open-circuit-voltage problem at t = 0 followed by near-exact coulomb integration, and the accuracy ceiling belongs mainly to the former. Compact recurrent and convolutional models trained on the same 20 s windows show offset shares from 0.94 to 0.97, so the offset-dominated residual also reflects the short window and the coulomb-counted labels, and it does not by itself show what the language model computes internally. This decomposition makes two falsifiable predictions. Error should concentrate in cold strata, because internal resistance distorts the t = 0 voltage signature [16]; and it should increase sharply for a chemistry whose open-circuit-voltage curve is flat, because the inverse map is then ill-conditioned [45–48]. The cold-stratum prediction is consistent with the measured errors. The flat-plateau prediction is only partly supported once the A123 anchor is corrected. A held-out lithium–iron–phosphate cell, absent from training, is estimated at 35% MAE and recovers only 16% of that error when a single lithium–iron–phosphate cell is added to the training set, with neither feature ablation nor feature-wise conditioning of the soft prompt [49] closing the gap. That 35% value refers to the original anchor and is not used as evidence for the mechanism. The decomposition reframes the objective for LLM-based SoC estimation from lowering aggregate error to conditioning the estimate at t = 0. The 35% figure uses an anchor that sets the first label of each segment to 1. On the reference range of about 0–47% state of charge, the zero-shot error is 12%. The prediction mean is 0.30 and the span is 0.12. The label mean is 0.31. The slope of prediction on label is 0.16. These outputs are not the original-anchor predictions. Held-out files of an LFP cell that was partly seen in training reach 2.53% MAE in a separate run. The voltage plateau remains a plausible contributor to the zero-shot gap.

The contributions of this work are highlighted as follows:

- **Layer-type-aware adaptation:** We identify a structural gap in LLM-based battery estimation. Existing methods adapt dense softmax decoders via a key–value interface [30,31,33] that is absent in the majority of modern hybrid backbones [50–53]. We introduce a scheme that injects trainable key–value prefixes into the full-attention minority, and a trainable initial recurrent state into the gated-delta majority. On the single-cell diagnostic, this path reduces the error by a factor of 2.3. On the final four-billion-parameter configuration, the factor is 1.86. An initial-state-only control shows that the initial state accounts for the larger reduction, and that the key–value prefixes contribute a further reduction.
- **Parameter-efficient multicell estimator:** We deliver a single frozen-backbone estimator that spans five cells and three chemistries from −20 °C to 45 °C, training only 0.55% of the parameters. The held-out nickel-based pool is 1.17% MAE. The LFP cell whose files are shared by training and test fits at 1.22%. A separate run, with five of those files held out, reaches 2.53% on the held-out files.
- **Residual decomposition:** We give a decomposition-based account of what a language model contributes to SoC estimation. We mathematically decompose the residual error and show that the task factorizes into an inverse open-circuit-voltage problem at the window start, followed by near-exact charge integration.
- **Failure-regime analysis:** The factorization is used to state two regimes in which error should rise, and both are tested. Zero-shot transfer to an LFP cell is 12% MAE on the reference state-of-charge range. The prediction mean is 0.30, the label mean is 0.31, and the prediction span is 0.12. The slope on the label is 0.16. These windows are not the original-anchor windows. The voltage plateau is a plausible contributor to that

gap. Expanding backbone capacity does not remove the initial-condition limit on the nickel-based cells, where the offset accounts for 0.98 of the residual energy.

2. Background: Hybrid Attention and State-Based Adaptation

The argument of this paper turns on an algebraic difference between two kinds of transformer block and on what is known about adapting each. This section states both, for readers to whom the linear-attention literature is not familiar; readers who know it may proceed to Section 3.

The dense softmax-attention decoder assumed by existing LLM-based SoC estimators [30,31,33] is no longer the architecture being deployed. Softmax attention costs time and memory quadratic in sequence length, and the linear-attention family removes that cost by rewriting attention as a recurrence over a state matrix of fixed size [54]. Gating, decay, matrix-valued states and selective state-space formulations progressively narrowed the quality gap to softmax attention [55–60], culminating in delta-rule updates that maintain the recurrent state as an associative memory [37,61]. Because purely recurrent models remain weaker at precise long-range recall, several recent large-scale designs are hybrids that interleave a minority of full-attention blocks among a majority of linear-attention or state-space blocks [50–53]. The consequence for the battery-LLM literature is practical rather than theoretical: in a decoder where three of every four blocks are linear-attention layers, those blocks maintain no key–value cache, so prefix- and prompt-based adaptation [38,39,62]—the mechanism on which existing LLM-based SoC estimators depend [30,31], including a prompt-guided estimator of containerized storage [33]—is undefined for the majority of the network. Bottleneck adapters and low-rank weight updates remain applicable [63,64], but they act on the projections rather than on the recurrence and therefore leave untouched the state that carries temporal information [60].

Parameter-efficient fine-tuning (PEFT) of recurrent backbones has begun to be studied in its own right, and the findings bear directly on this gap. A systematic study of PEFT for state-space models finds that low-rank adaptation of the linear projections outperforms the other existing methods but is ineffective on the state-space modules, proposes a dedicated method for those modules, and evaluates initial-state tuning, to which prefix tuning on such models is bounded in expressivity [40]. An exploratory study of PEFT for Mamba evaluates Transformer methods, adapts them to the Mamba architecture, introduces Mamba-specific variants, and reports that a combination of methods performs best [65]. A third study reports that prompt-based methods such as prompt tuning and prefix tuning perform poorly on state-space models and proposes state-based methods that act on state-related quantities directly [66]. The same literature identifies the limitation of the naive formulation: the influence of a trained *initial* state decays as the recurrence is iterated, which motivated replacing it with a per-step state offset [66]. What remains unexamined is how state-based adaptation behaves inside a hybrid decoder, where recurrent layers coexist with attention layers requiring a different interface; whether it supports continuous regression from physical sensors; and what the trained state corresponds to in the target task. The distinction from that prior work is the split by layer type. Key–value prefixes are applied only to the full-attention minority. The trainable initial state is applied only to the gated-delta majority. The state-based methods cited above were developed for purely recurrent backbones, which contain no full-attention layers to which prefixes could be attached.

Three recurrent families are relevant to this distinction: selective state-space models of the Mamba type [59,60], RWKV models, whose later versions carry matrix-valued states [57,58], and gated delta-rule layers [37,61]. Each carries a fixed-size state that is updated once per token and keeps no key–value cache. Key–value prefix adaptation therefore has no point of attachment in any of them, although soft tokens can still be

prepended to the input, and the initial state is an object that each of them exposes. In the gated delta-rule layer, each token writes to the state through a delta-rule update, so the state acts as an associative memory. The present scheme applies state tuning to such layers inside a hybrid decoder, alongside key–value prefixes on the attention layers.

3. Materials and Methods

3.1. Problem Formulation

Let a measurement window be a sequence of T samples $x_{1:T}$, where each $x_t = (V_t, I_t, \vartheta_t)$ collects terminal voltage in volts, load current in amperes with discharge negative, and cell surface temperature in degrees Celsius. The estimator must produce the SoC sequence $y_{1:T}$ with $y_t \in [0, 1]$. We use $T = 20$ throughout.

Two properties of this task structure the method. First, SoC obeys the charge balance (1), so once the initial value y_1 is known, the remainder of the window is determined by the current signal alone, up to coulombic-efficiency and capacity-fade corrections that are negligible within a single window. Second, y_1 is *not* determined by the current signal; it must be inferred from the voltage and temperature signature, which is an inverse open-circuit-voltage (OCV) problem and is chemistry-dependent. The method and the analysis both follow from this separation, and Section 4.2 shows that the trained estimator respects it almost exactly.

Charge balance and task factorization.

Within a single window, the true SoC obeys

$$y_t = y_1 + \frac{1}{C_{\text{nom}}} \int_{t_1}^{t} I(\tau)\, \mathrm{d}\tau, \tag{1}$$

so the trajectory is determined by the initial value y_1 and the measured current. Writing $\hat{y}_t$ for the model prediction, define the *offset* and *shape* residuals

$$e^{\text{off}} := \hat{y}_1 - y_1, \qquad e_t^{\text{sh}} := (\hat{y}_t - \hat{y}_1) - (y_t - y_1). \tag{2}$$

The residual energy attributed to the offset is

$$\rho = \frac{\sum_w (e_w^{\text{off}})^2}{\sum_w \sum_{t=1}^{T} \left(\hat{y}_{w,t} - y_{w,t}\right)^2}, \tag{3}$$

where the outer sum runs over evaluation windows. Empirically, $\rho = 0.984$ on the single-cell diagnostic run (Section 4.2), i.e., the adapted recurrence integrates charge to within 1.6% of the residual energy.

Consequence (accuracy ceiling).

If $e_t^{\text{sh}} \approx 0$ for all t, then

$$\text{MAE} \;\approx\; \mathbb{E}\left|e^{\text{off}}\right| = \mathbb{E}\left|\hat{y}_1 - y_1\right|, \tag{4}$$

so enlarging sequence capacity cannot reduce error below the quality of the t=0 inverse map $V_1, \vartheta_1 \mapsto y_1$.

Conditioning of the OCV inverse.

Locally linearizing an open-circuit map $V = U(y)$ gives the sensitivity

$$\kappa(y) := \left|\frac{\mathrm{d}y}{\mathrm{d}V}\right| = \left|U'(y)\right|^{-1}. \tag{5}$$

On the LFP plateau, $U'(y) \to 0$, so κ diverges and a millivolt-level voltage error (or a .2f prompt quantization of size $\Delta V = 0.01\,\mathrm{V}$) maps to

$$\Delta y \;\approx\; \kappa(y)\,\Delta V, \tag{6}$$

which shows why a flat plateau can amplify a small voltage error. It is not, by itself, proof that the plateau caused the A123 error.

3.2. Datasets and Ground Truth

Six cylindrical cells across three cathode chemistries are used (Table 1). Five are in-distribution and one, the CALCE A123 lithium–iron–phosphate (LFP) cell, is held out entirely as a zero-shot target.

Table 1. Cells, chemistries and roles. Italic rows are group labels, in this table and in later tables. The shaded header is a column heading.

Cell	Chem.	Format	C_{nom}	T (°C)	Role
In-distribution (train + test)					
Panasonic 18650PF	NCA	18,650	2.9 Ah	−20–25	train+test
LG HG2	NMC	18,650	3.0 Ah	−20–40	train+test
CALCE INR-20R	NMC	18,650	2.0 Ah	0, 25, 45	train+test
Samsung 30T	NMC	21,700	3.0 Ah	−20–40	train+test
Overlap/held-out					
Malaya	LFP	18,650	2.57 Ah	5, 25, 45	train∩test
CALCE A123	LFP	18,650	1.1 Ah	−10	zero-shot

Ground-truth SoC is obtained by coulomb counting against each cell's nominal capacity,

$$y_t = 1 + \frac{Ah_t}{C_{\mathrm{nom}}},$$

where Ah_t is the cumulative ampere-hour count reported by the cycler and is negative during discharge, and the count is reanchored to $y = 1$ at the start of each extracted segment.

Capacity-label validation. The Malaya dataset is distributed as a 1100 mAh cell, but two independent checks on the data itself—coulomb-counting a full discharge, and decoding the constant-current file whose 1C rate is −2.5 A—both indicate a cell of approximately 2.57 Ah. Using the published label would have driven y_t outside $[0, 1]$ and corrupted the training targets. We therefore validate C_{nom} against the data before using any published value, and recommend the same for any reuse of public drive-cycle datasets.

3.3. Windowing and Prompt Construction

Each drive-cycle file is segmented, then windowed with a stride equal to the window span so that windows do not overlap. Within a window, every tenth raw sample is retained, so one window covers 200 raw samples and is represented by $T = 20$ points. The McMaster cylindrical-cell files [41–43] are logged at 10 Hz, so that block spans 20 s of wall-clock time and the retained payload is equivalent to 1 Hz. The CALCE workbooks expose an explicit `Test_Time(s)` column; the same stride-and-downsample rule is applied so that every window, regardless of source, is a 20-point sequence. The retained columns are voltage, current and battery temperature; no derived feature such as ΔV or ΔAh is supplied. An earlier configuration did include these two features; they were removed after the ablation of Section 4.6 showed no effect on either cross-chemistry transfer or in-distribution accuracy under the drive-cycle protocol.

The window is rendered as a single natural-language instruction. Voltage is serialized to two decimals; current and temperature retain their logged precision. A paired training prints voltage to three decimals and is reported with the precision check. The SoC target is formatted to four decimals, comma-separated and terminated by the end-of-sequence token. Instruction and target are concatenated under the model's chat template with reasoning disabled; loss is masked on the instruction tokens (Table 2).

Table 2. Hard prompt template. Instruction tokens are loss-masked; only answer tokens contribute to the training objective.

Instruction (loss masked)
`For a {cell description}, its (voltage V, current A, temperature C)` over 20 time steps are: (3.85, −1.20, 24.9), (3.84, −0.95, 24.9), . . . , (3.71, −2.10, 25.1). What are the SOC values? Reply with only 20 comma-separated numbers between 0 and 1.
Target (answer tokens)
`0.8123, 0.8110, 0.8094, ..., 0.7841<eos>`

3.4. Backbone

The frozen backbone is the four-billion-parameter Qwen3.5-4B decoder [35] in bfloat16, loaded with an eager attention implementation so that the attention forward pass can be intercepted. Layers are read from the model configuration and partitioned by declared type into `full_attention` and `linear_attention`; the latter are gated delta-rule blocks [37] that maintain a recurrent state instead of a key–value cache. The configuration used here yields 24 linear-attention layers and 8 full-attention layers, a 3:1 interleaving. All backbone parameters have gradients disabled and the model is kept in evaluation mode.

Writing L_a and L_g for the number of attention and gated-delta layers, H_{kv} and d_h for the key–value head count and head dimension of the attention layers, and H_v, d_k, d_v for the value-head count and key and value head dimensions of the gated-delta layers, the configuration gives $L_a = 8$, $L_g = 24$, $H_{kv} = 4$, $d_h = 256$, $H_v = 32$ and $d_k = d_v = 128$.

3.5. Layer-Type-Aware Adaptation

The adaptation scheme is shown in Figure 1.

Two trainable modules are attached, one per layer type, and nothing else in the network is modified.

3.5.1. Key–Value Prefix for Full-Attention Layers

For the attention minority we use deep prefix tuning [38,39], with the prefix produced by a small recurrent encoder rather than being a free parameter tensor. Let $D = 2L_aH_{kv}d_h = 16{,}384$ be the width needed to specify one key and one value vector in every attention layer. A learned embedding $E \in \mathbb{R}^{M\times D}$ with $M = 64$ soft positions, initialized from $\mathcal{N}(0, 0.01^2)$, is modulated feature-wise [49] by a per-cell descriptor c,

$$X = E \odot (1 + \gamma(c)) + \beta(c),$$

where γ and β come from a two-layer network $\mathbb{R}^6 \to \mathbb{R}^{64} \to \mathbb{R}^{2D}$ with SiLU activation whose output layer is zero-initialized, so that $X = E$ at initialization. The descriptor c is a six-dimensional vector holding a chemistry one-hot over (NCA, NMC, LFP), the nominal capacity normalized by 5 Ah, and the chemistry-specific voltage limits normalized as $(v - 2)/3$. The modulated embedding then passes through a bottleneck and a bidirectional recurrent encoder,

$$H = W_{\text{up}}\,\text{BiLSTM}(W_{\text{down}}X), \qquad W_{\text{down}} \in \mathbb{R}^{D\times 256},\ W_{\text{up}} \in \mathbb{R}^{128\times D},$$

with a single bidirectional layer of hidden size 64, Xavier initialization on W_{down} and zero initialization on W_{up}. A learned per-channel gate $g = \sigma(\theta)$ with θ initialized to zero mixes the two paths,

$$P = g \odot X + (1 - g) \odot H,$$

and P is reshaped to $(L_{\text{a}}, 2, H_{\text{kv}}, M, d_{\text{h}})$ to give one key and one value prefix per attention layer. In each patched attention forward pass the prefix is concatenated to the key and value tensors after rotary position embedding and cache update, and the additive attention mask is extended with M zero-valued columns so that every query position may attend to the prefix without violating causality among the real tokens. The remainder of the block—the output gate, the output projection and the attention interface itself—is untouched.

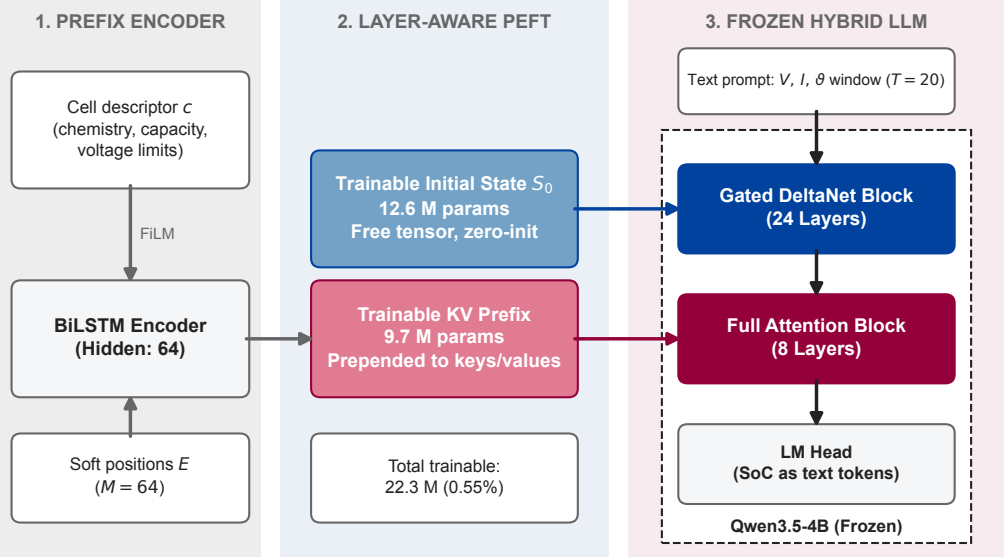


Figure 1. Layer-type-aware adaptation of a frozen hybrid decoder. The bidirectional encoder builds the key–value prefixes for the full-attention minority. The initial state is a separate trainable tensor on the gated-delta layers, not an output of that encoder. The encoder receives learned soft positions modulated by the cell descriptor. The measurement window enters the frozen model as a text prompt, and the state of charge is emitted as text through the language-model head. The 24 gated-delta and 8 full-attention layers are interleaved in a 3:1 pattern and are drawn as two blocks. Only 22.3 M of 4 B parameters are trained (0.55%).

3.5.2. Initial Recurrent State for Gated-Delta Layers

A gated delta-rule layer carries a state matrix $S \in \mathbb{R}^{H_{\text{v}}\times d_{\text{k}}\times d_{\text{v}}}$ that is updated once per token and is conventionally initialized to zero, so there is no key–value tensor to prepend and prefix tuning has no point of attachment. We therefore make the initial state itself the trainable object. A parameter tensor $S_0 \in \mathbb{R}^{L_{\text{g}}\times H_{\text{v}}\times d_{\text{k}}\times d_{\text{v}}}$ is held in single precision, initialized to zero so that the adapted model is bit-identical to the unmodified backbone at the start of training, and broadcast across the batch. The chunked gated-delta-rule kernel of each linear-attention layer is wrapped so that when it is called with no initial state, the corresponding slice of S_0 is supplied. Because the wrapper only fills an argument that would otherwise default to zero, the recurrence, the gating and the delta-rule update are all unchanged.

Initial-state tuning and related state-based methods have been studied for purely recurrent backbones [40,66]; here, the trained state is combined with a prefix on the

attention layers, and the trained object admits a task interpretation (Section 5.1). Its known limitation—that the influence of a trained initial state decays as the recurrence is iterated [66]—is discussed in Section 5.3 and is the reason the window length matters.

Parameter budget. The recurrent-state path holds $L_g H_v d_k d_v = 12.58$ M parameters and the key–value path 9.66 M, of which the embedding contributes 1.05 M, the bottleneck 4.19 M, the up-projection 2.11 M, the modulation network 2.13 M, the recurrent encoder 0.16 M and the gate 0.02 M. The total of 22.3 M is 0.55% of the backbone.

3.6. Objective

Training minimizes a token-weighted cross-entropy over the answer tokens only,

$$\mathcal{L} = \frac{\sum_{t \in \mathcal{A}} w_t \, \mathrm{CE}(z_t, y_t)}{\sum_{t \in \mathcal{A}} w_t},$$

where $\mathcal{A}$ is the set of answer positions and z_t the logits. The weights implement the one intervention that the error decomposition motivates: positions belonging to the *first* emitted SoC value receive $w_t = \lambda$ and all others $w_t = 1$. The number of tokens covered is determined by tokenizing a representative answer and counting the tokens preceding the first comma, so the rule adapts to the tokenizer rather than assuming a fixed count. We use $\lambda = 10$; Section 4.1 quantifies the effect. Validation loss is computed with $\lambda = 1$ so that it remains a uniform likelihood.

3.7. Optimization

Only the two adapter modules receive gradients. We use Adam at a peak learning rate of 10^{-4}, linearly warmed up over 100 optimizer steps and then cosine-annealed to 10^{-5}, with gradient-norm clipping at 1.0, a micro-batch of 4 and gradient accumulation of 4 for an effective batch of 16. Five epochs were run. Non-finite losses and gradients are detected per step; the offending step is skipped, previously accumulated gradients are restored, and the run aborts if more than 200 such steps occur, so that numerical failures are detected and reported. Training used a single RTX PRO 6000 at roughly 1.9 h per epoch, about 9.5 h in total. The headline nickel-based metrics of this run are taken from epoch 5, the last scheduled epoch. The loss logged during that training was evaluated on the test partition and was not used to select the checkpoint (Section 3.11).

3.8. Inference and Parsing

Predictions are produced by greedy decoding, capped at 200 new tokens. The response is parsed by extracting the leading numeric literals and rejecting any window whose parse yields fewer than T values, a non-finite value, or a magnitude above 10; such windows are counted as parse failures and excluded from the error statistics. Over the final evaluation, 3 of 20,275 windows failed to parse (0.015%).

3.9. Metrics

Predictions and targets from all retained windows are pooled and the mean absolute error is computed over the pooled points, so every SoC sample contributes equally and cells with more windows carry proportionally more weight. This is the quantity reported as window-weighted MAE. Metrics are additionally broken down by file, by cell and by temperature.

3.10. Error Decomposition

To separate initial-condition error from trajectory error, each window residual is split into a constant and a shape component. Writing $\hat{y}$ for the prediction and y for the target,

$$\underbrace{\hat{y}_1 - y_1}_{\text{offset}}, \qquad \underbrace{(\hat{y}_t - \hat{y}_1) - (y_t - y_1)}_{\text{shape}},$$

and the share attributed to the offset is the fraction of total residual energy removed by subtracting the per-window constant. A value near unity means the estimator reproduces the trajectory shape and misplaces only its level.

3.11. Evaluation Protocol and Its Limitations

Train and test are split by named drive cycle, following [30] rather than randomly: for Panasonic, training uses the `Cycle_NN` and `Cycle_1`–`Cycle_4` files and testing the HWFET, UDDS, LA92 and US06 cycles; for LG HG2, training uses the mixed cycles and testing the four standard cycles; for CALCE INR-20R, training uses the 80% SoC files and testing the 50% SoC files; for Samsung 30T, training uses UDDS, HWFET and LA92 and testing US06 across all temperatures. CALCE A123 contributes no training files at all.

Four properties of this protocol limit the evidence and are stated explicitly.

1. **The Malaya LFP cell appears in both training and test.** Every Malaya segment is used for training, and the same segments are reported as an in-distribution LFP metric. The intent was to teach the LFP plateau while still reporting a per-cell number, but the consequence is that the Malaya figure is a fit statistic and not a generalization estimate. We therefore exclude it from the headline aggregate and report it as a separate row (Section 4.3); because it carries 1438 of 20,150 windows, doing so changes the pooled figure by less than 0.01 percentage points, and both values round to 1.17%. The table keeps the 1.22% fit and also the held-out-file measurement, 2.53% on 485 windows. A separate training run held out five Malaya files in full, with at least one file at 5, 25 and 45 degrees Celsius, and kept the other ten files for training. On 485 windows from the held-out files the MAE is 2.53%: 3.2% on 160 windows at 5 degrees Celsius, 2.1% on 170 windows at 25 degrees Celsius, and 2.3% on 155 windows at 45 degrees Celsius. The nickel-based pool of that run is 1.24% on 18,712 windows. The 1.17% headline is the run trained on all fifteen Malaya files.
2. **Validation aliases test.** No separate validation split exists in the run that supplies the headline. The training loop performs no early stopping and no learning-rate scheduling on validation, so the aliasing does not leak into optimization, but it does mean there is no clean model-selection signal for that run. A second training cuts whole files out of the training set for validation and leaves the test files in place. Malaya and A123 are not used as that validation set. The checkpoint is the epoch of lowest validation loss, which is epoch 4. Its test pool is 1.17%, on a different cell vector from the epoch-5 headline.
3. **Epoch selection.** The reported checkpoint is epoch 5, the last of the five scheduled epochs. Training used no early stopping. The checkpoint was not chosen by the loss on the test partition. The difference between that minimum and epoch 5 is 0.02 percentage points on the held-out pool.
4. **A123 segment anchoring.** The original extraction sets the first label of each segment to 1. The reference evaluation in [30] spans roughly 0–47% SoC and includes a charge phase. On 125 windows under that reference range, the MAE is 12%. The prediction minimum, mean and maximum are 0.24, 0.30 and 0.36, so the span is 0.12 and 0.30 is the mean. Labels run from 0.01 to 0.47, with mean 0.31. Forty-four labels fall below 0.20 and fifty fall above 0.37; those cuts are the original prediction limits, not the new span. The regression of prediction on label has slope 0.16, intercept 0.25 and an R-squared of 0.35. The original anchor remains 35.17% MAE, with predictions from

0.20 to 0.37 and slope 0.11. The two columns do not share one prediction vector. The 12% result is the one used in the plateau discussion.

Items 1 and 4, in particular, bound how strongly the corresponding results can be stated, and Section 5.6 returns to them.

4. Results

4.1. The Recurrent-State Path Carries the Improvement

The two injection paths were added cumulatively on a single Panasonic cell (Table 3, Figure 2). A key–value prefix alone—the mechanism available to prior work on this backbone—reaches 2.61% MAE. Adding the trainable initial recurrent state to the 24 gated-delta layers reduces this to 1.15%, a factor of 2.3 and 86% of all the error subsequently removed. Supplying ΔV and ΔAh features contributes a further 9%, and upweighting the loss on the first emitted SoC value by $\lambda = 10$ contributes a further 13%, reaching 0.91%.

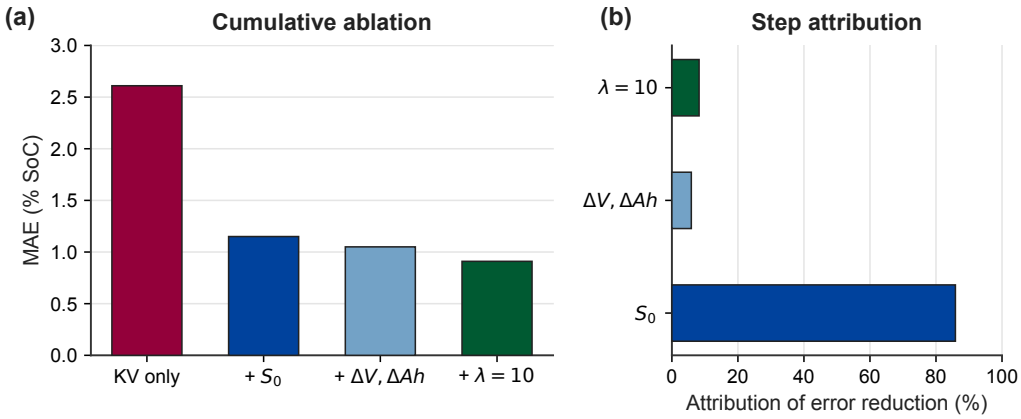


Figure 2. Cumulative ablation on a single Panasonic cell (9B backbone, random file-level split). (**a**) MAE after successive additions; adding S_0 yields a 2.3× reduction compared to KV-prefix alone. (**b**) Attribution of the removed error, showing that the recurrent state path handles the vast majority (86%) of the performance gain.

Table 3. Cumulative ablation (Panasonic, 9B, random split). Green = better vs. KV-prefix baseline. Single-cell diagnostic, not the final configuration. The ratio and change columns carry the same comparison without color.

Configuration	MAE (%)	Ratio	Δ vs. KV
Key–value prefix only	2.61	1.00×	—
+ initial state S_0	**1.15**	**0.44**×	−56%
+ ΔV, ΔAh	**1.05**	0.40×	−60%
+ step-0 loss ($\lambda = 10$)	**0.91**	0.35×	−65%

Two aspects of this table are considered separately. The 2.3-fold reduction establishes that adapting only the attention minority leaves most of the attainable error reduction unrealized, which is the central architectural claim. The step-0 weighting is a different kind of result: it is a single hyperparameter, motivated by the decomposition in Section 4.2 rather than by search, and it outperforms the feature-engineering step above it.

Two caveats attach to this table. It was measured in the earlier configuration—one Panasonic cell, a nine-billion-parameter backbone and a random file-level split—and the absolute values are therefore not comparable with the per-cell results and the reference comparison reported below. It also includes the ΔV and ΔAh features, which the final pipeline does not use: when those features were later removed under the drive-cycle protocol, in-distribution accuracy was unaffected, so the 9% contribution in the third row

did not replicate and the simpler three-channel prompt was adopted (Section 4.6). The rows that carry the argument of this paper are the first two, and their ratio is the quantity we rely on for the diagnostic.

Table 4 reports the same comparison for the configuration used in the main evaluation. The backbone is the frozen four-billion-parameter decoder, the checkpoint is epoch 5, and the evaluation set comprises 18,712 drive-cycle-held-out windows, with the Malaya and A123 cells excluded. Key–value prefixes alone yield a pooled mean absolute error of 2.18%. The initial state alone yields 1.50%. The two paths combined, at loss weight 1, yield 1.25%. The full adapter, which sets the weight on the first emitted state of charge to 10, yields 1.17%. The initial state accounts for the larger reduction. The key–value prefixes contribute a further reduction of 0.25 percentage points, and the step-0 loss weight a further reduction of 0.08 percentage points. The ratio of the prefix-only error to the full-adapter error is 1.86. On this checkpoint, a constant per-window offset accounts for 0.98 of the residual energy.

Table 4. Ablation of the injection paths on the final configuration. The backbone is the frozen four-billion-parameter decoder and the checkpoint is epoch 5. Entries are mean absolute error (%), evaluated on 18,712 drive-cycle-held-out windows with the Malaya and A123 cells excluded. The pooled value is the window-weighted mean. ρ is the fraction of residual energy attributable to a constant per-window offset. The full adapter follows the protocol of the main evaluation and is scored at the last scheduled epoch.

Configuration	Panasonic	LG	INR	Samsung	Pooled	ρ
Key–value prefix only	2.38	1.83	2.83	4.14	2.18	—
Initial state only	1.63	1.26	2.24	2.72	1.50	—
Both paths, $\lambda = 1$	1.40	1.02	1.70	2.30	1.25	—
Full adapter, $\lambda = 10$	1.32	0.94	1.59	2.16	1.17	0.98

4.2. The Residual Is an Initial-Condition Error

Decomposing the residual as described in Section 3 shows that 98.4% of it is a constant per-window offset and only 1.6% is trajectory shape (Figure 3). The 98.4% share was measured on the single-cell diagnostic configuration of Table 3. On the final four-billion-parameter configuration the same share was measured again. The offset accounts for 0.98 of the residual energy and the shape term for 0.02. By cell, the share is 0.98 on Panasonic, 0.99 on LG, 0.97 on INR and 0.96 on Samsung. NCA, which is Panasonic alone, is 0.98. NMC, 9729 windows, is 0.98. Table 5 reports these shares.

Table 5. Offset share of residual energy on the final adapter, epoch 5, nickel-based holdout. Malaya and A123 are excluded. The three temperature rows are the groups that were scored. Their window counts were not recorded, and the gaps from 0 to 10 degrees Celsius and from 25 to 40 degrees Celsius are not in the table, so these rows are not a partition of the 18,712 windows.

Set	Windows	Offset Share
Panasonic, NCA	8983	0.98
LG, NMC	8812	0.99
INR, NMC	399	0.97
Samsung, NMC	518	0.96
Pooled	18,712	0.98
NCA	8983	0.98
NMC	9729	0.98
At or below 0 degrees Celsius	—	0.98
10 to 25 degrees Celsius	—	0.98
At or above 40 degrees Celsius	—	0.97

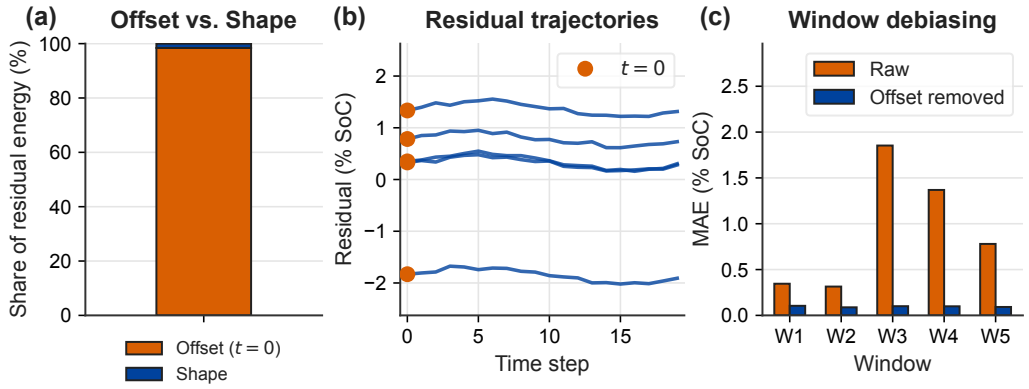


Figure 3. Residual decomposition on the single-cell diagnostic run. (**a**) 98.4% of residual energy is a constant per-window offset (initial SoC error). (**b**) Per-window residual trajectories against time: traces are essentially flat, indicating near-exact charge integration following an initial displacement. (**c**) Window MAE before and after subtracting the per-window offset, confirming that the estimator successfully tracks the trajectory shape but misplaces the level. This figure is the single-cell diagnostic. Panels (**b**,**c**) illustrate the residual pattern. The final-system shares are 0.98 pooled, and 0.98, 0.99, 0.97 and 0.96 by cell. Panel (**a**) shows the measured diagnostic share.

Within a window, the predicted and true SoC trajectories run parallel: the estimator reproduces the slope, which is the integral of the measured current, and misplaces the level, which is the value at $t = 0$.

This is the central empirical result of the paper. Within a window, the adapted recurrent stack reproduces the charge integral closely, which is consistent with the structure of a gated delta-rule state as a linear associative memory that accumulates its inputs. The remaining error therefore lies in the inverse mapping from the voltage and temperature signature at the start of the window to the initial state of charge. Given the same starting value, a coulomb integrator reproduces the remainder of the window.

Two qualifications bound this reading. First, the compact models trained on the same windows reach offset shares of 0.94 to 0.97. On 20 s windows with coulomb-counted labels, an offset-dominated residual is therefore expected from any estimator that follows the measured current, and the decomposition alone does not identify a computation specific to the language model. Second, the trained initial state is one tensor shared by all windows. It cannot store the initial state of charge of a particular window. The window-specific initial estimate is formed when the prompt tokens update the recurrence from that learned default.

On the same 18,712 nickel-based windows, coulomb counting that is given the true state of charge at the start of the window reaches 0.2% MAE. The same integrator with the start fixed at 1 reaches 19%. An extended Kalman filter reaches 2.6%. The epoch-5 adapter reaches 1.17%. The adapter is therefore not a more accurate integrator than coulomb counting with a known start. The 19% figure is the error of a wrong start on these windows of about 20 s, not a drift accumulated over an hour. Table 6 reports that comparison.

Table 6. Observers and the adapter on the 18,712 nickel-based windows. Windows are about 20 s long. The adapter row is the reported epoch-5 model. All rows use the same windows and the same coulomb-counted labels. EKF denotes an extended Kalman filter.

Method	MAE (%)
Coulomb counting, true start	0.2
Coulomb counting, start fixed at 1	19
Extended Kalman filter	2.6
4B adapter	1.17

Two testable consequences follow from this mechanism: error should concentrate below 0 °C, and transfer error should rise sharply on a flat-plateau chemistry. Sections 4.3–4.5 test both. The cold strata come from the three-cell comparison with the reference method, and the flat-plateau test uses the zero-shot A123 cell.

4.3. Nickel-Based Holdout, with LFP Reported Separately

The 1.17% pool is the four nickel-based cells. It is not an LFP result. Extending to five in-distribution cells under drive-cycle holdout gives a pooled MAE of 1.17% over the four cells that are held out (Figure 4). Per-cell values span 0.94% to 2.16%. Adding the Malaya LFP cell to a previously NCA/NMC-only training mix left three cells lower and Samsung 30T higher: Panasonic lower by 2%, LG HG2 lower by 3%, CALCE INR-20R lower by 22%, and Samsung 30T higher by 17% (epoch 4 of the five-cell run against epoch 5 of the four-cell run; the epoch-5 cell values were not used for these percentages), against an aggregate that stays within 1.13–1.18% across the whole experiment arc. The cell-level changes are mixed and, because the two epochs are not the same run, the comparison is not a controlled test of interference between chemistries in a single shared adapter.

The loss on the test partition is lowest at epoch 4. The held-out pool is 1.33% at epoch 3, 1.15% at epoch 4 and 1.17% at epoch 5. The corresponding errors on all 20,275 windows are 1.58%, 1.37% and 1.38%. Table 7 reports these epochs.

Table 7. Scheduled epochs of the final training run. The loss in the third column was evaluated on the test partition and was not used to select the checkpoint. The held-out pool is the window-weighted mean absolute error on the four drive-cycle-held-out cells. The reported model is epoch 5. Bold marks the reported epoch.

Epoch	Training Loss	Loss on the Test Partition	MAE, All Windows (%)	Held-Out Pool (%)
3	0.3105	0.1067	1.58	1.33
4	0.2943	0.1055	1.37	1.15
5	**0.2796**	**0.1079**	**1.38**	**1.17**

The repeated training, with validation files removed from training, records validation losses of 0.201, 0.104 and 0.108 at epochs 3, 4 and 5. The minimum is epoch 4. Test MAE on the 18,712 nickel-based windows is 1.16%, 1.17% and 1.20%. At epoch 4 the cells are Panasonic 1.35%, LG 0.92%, INR 1.43% and Samsung 2.22%, and the pool is 1.17%. The epoch-5 headline cells are 1.32%, 0.94%, 1.59% and 2.16%. The pools match at the reported precision. The cell vectors do not. The validation loss is the training objective, not the MAE. The ablation and the compact models were not repeated on this split. Table 8 reports that run.

Table 8. Repeated training with a validation set cut from the training files. The test files are unchanged. The reported checkpoint of this run is epoch 4, the minimum of the validation loss. Test MAE is the nickel-based pool on 18,712 windows. This pool is not the epoch-5 headline. Per-cell errors were recorded only for the selected epoch. Dashes mark values that were not recorded.

Epoch	Validation Loss	Test Pool (%)	Panasonic	LG	INR	Samsung
3	0.201	1.16	—	—	—	—
4	**0.104**	**1.17**	**1.35**	**0.92**	**1.43**	**2.22**
5	0.108	1.20	—	—	—	—

We report the held-out pool as the headline figure. Because the Malaya cell's test segments are also its training segments (Section 3.11), its 1.22% is a fit statistic, and pooling it with held-out cells would make the aggregate part fit and part generalization estimate. Excluding it changes the window-weighted pool by less than 0.01 percentage points, so both the four-cell holdout and the pool that includes the Malaya fit equal 1.17% at the

reported precision. Both values are reported so that the choice can be checked. Every row of Table 9 that belongs to the main run is taken from epoch 5. The Malaya holdout and the A123 reference-range row are separate evaluations. Relative to epoch 4, the largest cell-level change is CALCE INR-20R, from 1.44% to 1.59%. The held-out pool moves from 1.15% to 1.17%, because that cell contributes 399 of the 18,712 windows.

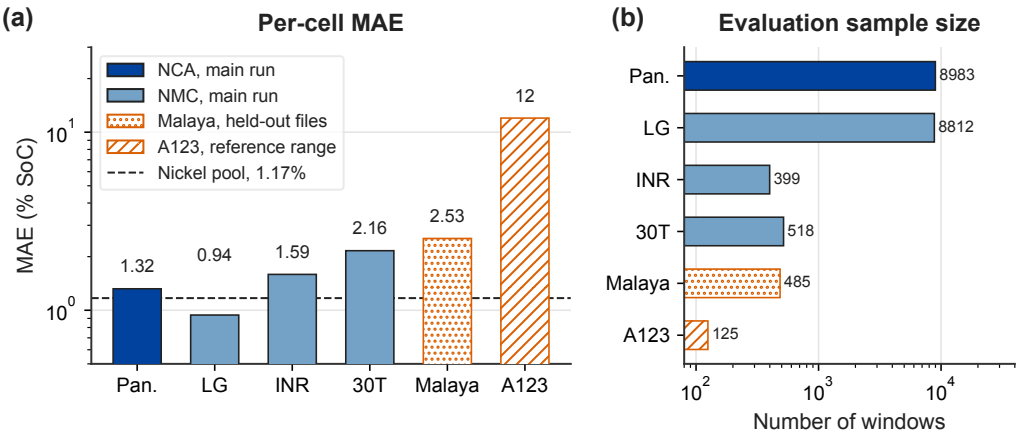


Figure 4. Per-cell mean absolute error for the reported evaluations. (**a**) Panasonic, LG, INR-20R and Samsung 30T are the epoch-5 main run; the dashed line is their pool, 1.17%. Malaya is the separate five-file holdout, 2.53% on 485 windows. A123 is the reference state-of-charge range, 12% on 125 windows. (**b**) Window counts for the same evaluations.

Table 9. Per-cell mean absolute error. The four nickel cells and their pool are the epoch-5 main run. The Malaya holdout row and the A123 reference-range row are the remaining bars in the figure above. The Malaya fit and the original A123 anchor are listed here and are not plotted.

Cell	Chem.	Windows	MAE (%)	Role
Held out by drive cycle				
Panasonic 18650PF	NCA	8983	1.32	held out
LG HG2	NMC	8812	0.94	held out
CALCE INR-20R	NMC	399	1.59	held out
Samsung 30T	NMC	518	2.16	held out
Pooled (4 cells)	—	**18,712**	**1.17**	—
Malaya LFP				
Malaya, fit	LFP	1438	1.22	same files in training and test
Pooled with Malaya fit	—	20,150	1.17	main run
Malaya, held-out files	LFP	485	2.53	separate run
A123 LFP, zero-shot				
A123, original anchor	LFP	125	35.17	first label set to 1
A123, reference range	LFP	125	12	about 0–47% SoC

The 1.22% Malaya entry is a fit on all fifteen files. The separate holdout, after those five files were removed from training, measures 2.53% MAE on its 485 windows. The nickel-based pool of that retraining is 1.24%. It is not the 1.17% headline. Table 10 reports both protocols.

Sample sizes differ by more than an order of magnitude, and the two smallest cells are the two extremes of the per-cell range. CALCE INR-20R and Samsung 30T carry a few hundred windows each against roughly nine thousand for Panasonic and LG HG2, so their individual figures are the least stable entries in the table and should not be read as evidence of chemistry- or format-specific behavior.

Table 10. Malaya LFP under two protocols. The fit uses the reported checkpoint, trained on all fifteen Malaya files. The holdout is a separate training run: five files are removed from training, with at least one file at each of 5, 25 and 45 degrees Celsius. The measured error on those 485 windows is 2.53%.

Quantity	Fit, All Files in Training	Five Files Held Out
Windows	1438	485
Files	15, also in test	5 in test, 10 in training
Shared segments	all	none
MAE (%)	1.22	2.53
MAE at 5 degrees Celsius (%)	—	3.2 (160 windows)
MAE at 25 degrees Celsius (%)	—	2.1 (170 windows)
MAE at 45 degrees Celsius (%)	—	2.3 (155 windows)
Nickel-based pool (%)	1.17	1.24

4.4. Comparison with the Reference Method

Measured on the cells, temperatures and split protocol of [30], the pooled three-cell result is 1.12% against 1.20% for the reference method—parity rather than an improvement (Table 11, Figure 5). The disaggregation is more informative than the aggregate: relative to the reference, the error is lower by 10% on Panasonic and by 3% on LG HG2, higher by 27% on CALCE INR-20R, and higher by 10% to 28% in every sub-zero stratum.

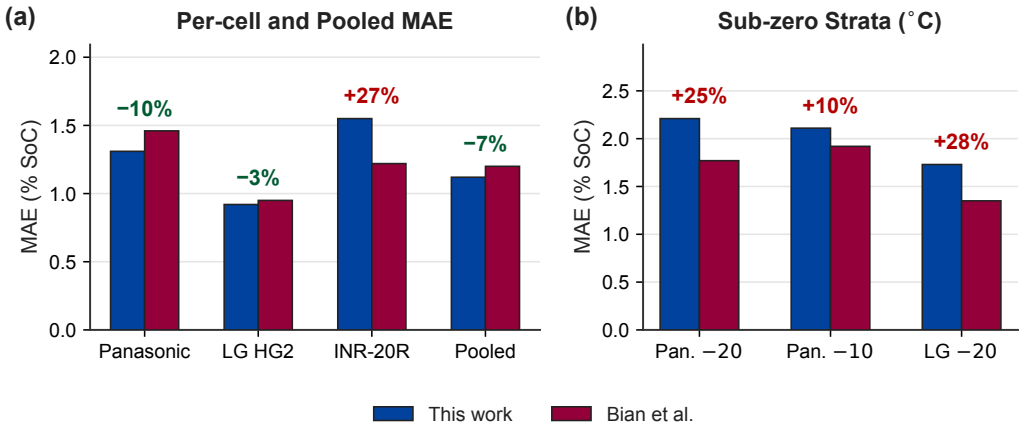


Figure 5. Comparison with Bian et al. [30] on identical cells and drive-cycle holdout. (**a**) Per-cell and pooled MAE; green labels are relative improvements, red labels are degradations. (**b**) Sub-zero strata, where this work is consistently worse. No superiority claim is drawn from the pooled relative change of minus 7 percent.

The Panasonic and CALCE INR-20R figures differ slightly from Table 9 (1.31 against 1.32, and 1.55 against 1.59) because this comparison is measured on the three-cell training mix that matches the cell set of [30], whereas Table 9 reports the final five-cell model. The reference values are those published in [30] for the same cells, temperatures and cycle holdout.

No claim of superiority is drawn from this table, and Section 5.2 examines why a newer and substantially larger backbone with a much smaller adaptation surface does not do better than parity.

4.5. Cold Strata and the Flat-Plateau Chemistry

The first prediction is that error should concentrate where the $t = 0$ voltage signature is least informative. Below 0 °C, internal resistance shifts the terminal voltage so that a cell reads as more discharged than it is [16], which distorts the quantity on which the inverse map depends. The cold rows of Table 11 are consistent with this: the sub-zero strata are both the worst absolute errors and the strata in which the reference method leads by the

widest margin. In that three-cell comparison, Panasonic at −20 degrees is 2.21 percent and at −10 degrees is 2.11 percent, against 1.31 percent over its five temperatures, and LG at −20 degrees is 1.73 percent, against 0.92 percent over its six temperatures. Internal resistance at low temperature pulls the terminal voltage at the window start away from the open-circuit value, so the inverse map that sets the initial state of charge is the step that degrades. These rows are not the five-cell headline.

Table 11. Comparison with [30] on identical splits (MAE, % SoC). Green = ours lower; red = higher. Yellow marks a smaller increase. Our values come from the three-cell training mix that matches the cell set of the reference, not from the five-cell model. The reference values are the published figures.

Stratum	MAE (%)		Relative	
	Ours	Ref.	Δ	%
In-distribution				
Panasonic (5 *T*)	1.31	1.46	−0.15	−10
LG HG2 (6 *T*)	0.92	0.95	−0.03	−3
CALCE INR-20R (3 *T*)	1.55	1.22	+0.33	+27
Three-cell pooled	**1.12**	1.20	−0.08	−7
Sub-zero				
Panasonic, −20 °C	2.21	1.77	+0.44	+25
Panasonic, −10 °C	2.11	1.92	+0.19	+10
LG HG2, −20 °C	1.73	1.35	+0.38	+28

The second prediction is that transfer error should rise sharply for a chemistry whose OCV curve is flat, because the inverse map is then ill-conditioned. On the reference state-of-charge range the measured error is 12%, not the original-anchor value of 35.17%. The mean prediction matches the mean label, and the plateau is not identified as the sole cause. NCA and NMC share a monotone OCV characteristic sloping from roughly 3.0 V to 4.2 V, whereas LFP exhibits a long plateau near 3.2 V across approximately 20–80% SoC [45–47], where a millivolt-level voltage error maps to a large SoC error (Figure 6).

Under the original anchor, a held-out A123 LFP cell is estimated at 35.17% MAE. On the reference range of about 0–47% state of charge the error is 12%, about ten times the 1.17% nickel-based pool and about five times the 2.53% error on held-out files of the Malaya cell.

The regression diagnostic below distinguishes this result from an error in the data pipeline (Figure 7).

The equation below is the original-anchor fit. The line in panel (a) is the reference-range fit, with slope 0.16 and intercept 0.25. Regressing prediction on ground truth across the 125 held-out windows of the original anchor gives

$$\hat{y} = 0.110\, y + 0.207, \qquad R^2 = 0.345,$$

a slope close to zero and an intercept near the lower end of the state-of-charge range; the predictions remain between 0.20 and 0.37 irrespective of the true value. A capacity-scaling error, by contrast, would have produced a slope of 0.38 through the origin, which is not observed. The nominal-capacity assignment, the label range, the prompt structure, the units, the sign convention and the sampling rate were checked and found to be consistent with the data files. This paragraph describes the original anchor. It is not the evidence for an ill-conditioned open-circuit-voltage map. That discussion uses the reference-range column of Table 12.

Re-evaluation on the reference range changes both the labels and the predictions, so the two columns are not a relabeling of one output vector.

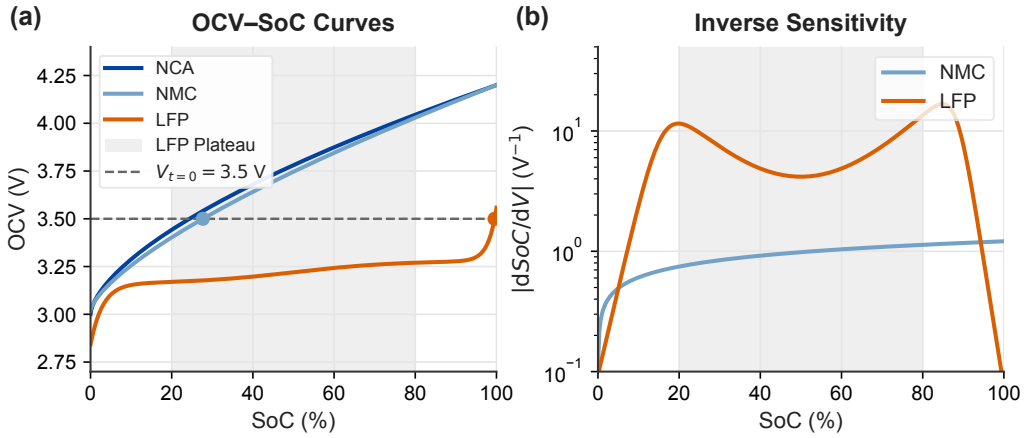


Figure 6. Conditioning of the $t = 0$ inverse map. (**a**) Open-circuit-voltage curves of NCA, NMC and LFP. A single voltage reading of 3.50 V maps to an unambiguous ≈28% SoC on NMC. On LFP that voltage lies near full charge, above the plateau. The plateau itself is the flat region near 3.2 V from about 20% to 80% SoC. (**b**) Inverse sensitivity $|\mathrm{d}SoC/\mathrm{d}V|$ (log scale). The plateau region causes inverse sensitivity to spike by an order of magnitude. The curves show that a flat plateau makes the initial map ill-conditioned. They do not establish that the A123 error is a constant offset: on the reference range, the mean prediction matches the mean label.

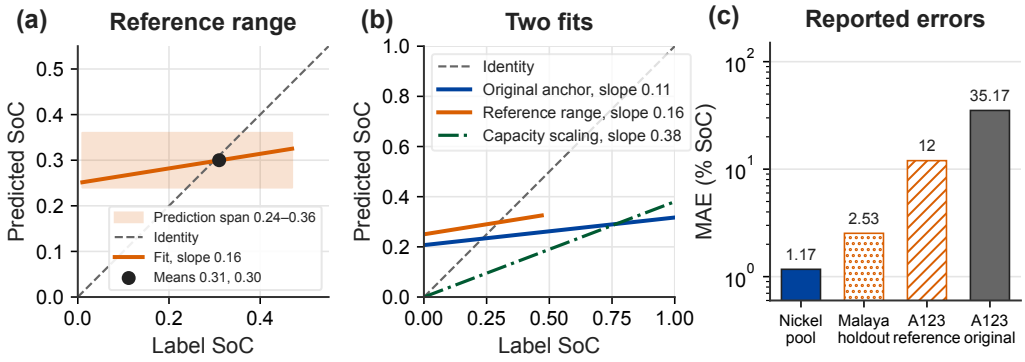


Figure 7. Zero-shot A123. (**a**) Reference range, about 0–47% state of charge. The band is the reported prediction span, 0.24 to 0.36. The marker is the pair of means, label 0.31 and prediction 0.30. The line has slope 0.16. Individual windows are not plotted. (**b**) The original-anchor fit, slope 0.11, and the reference-range fit, set against the identity line and a capacity-scaling line of slope 0.38. (**c**) Mean absolute error for the nickel pool, 1.17%, the Malaya file holdout, 2.53%, the reference range, 12%, and the original anchor, 35.17%.

Table 12. Zero-shot A123 at minus 10 degrees Celsius. Each column has 125 windows, and the windows are not the same: the reference-range predictions are not the original outputs with new labels. The original anchor sets the first label to 1. The reference range is about 0–47% state of charge. There, the prediction mean is 0.30, the label mean is 0.31, and the prediction span is 0.12. The label counts use the original prediction limits, 0.20 and 0.37. Dashes mark statistics that were not recorded for the original-anchor evaluation.

Quantity	Original Anchor	Reference Range
MAE (%)	35.17	12
Prediction minimum	0.20	0.24
Prediction mean	—	0.30
Prediction maximum	0.37	0.36

Table 12. *Cont.*

Quantity	Original Anchor	Reference Range
Prediction span	0.17	0.12
Label minimum	—	0.01
Label mean	—	0.31
Label maximum	—	0.47
Mean prediction minus mean label	—	−0.01
Labels below original prediction minimum 0.20	—	44 (share 0.35)
Labels above original prediction maximum 0.37	—	50 (share 0.40)
Slope	0.110	0.16
Intercept	0.207	0.25
R-squared	0.345	0.35

On the reference range the prediction mean is 0.30 and the label mean is 0.31, so the prediction mean is 0.01 below the label mean. The prediction span is 0.12. The slope of 0.16 shows that the estimate does not track the label. The plateau is a plausible contributor to this loss of tracking. The experiment does not isolate it as the sole cause. The original-anchor error of 35.17% is not used as that evidence.

4.6. What Does Not Fix It

Adding a single LFP cell to the training set moves the held-out LFP cell from 41.75% to 35.17%, a 16% relative improvement (Table 13). Exposure to the LFP plateau therefore has a measurable effect on transfer, but the residual gap to any in-distribution cell remains a factor of thirty under that anchor. Those A123 entries use the original anchor. Under the reference range the final model reaches 12%, about ten times the 1.17% nickel-based pool.

Table 13. Effect of training-set composition on in-distribution and zero-shot A123 MAE (%). The A123 column uses the original anchor, with the first label of each segment set to 1. All rows are the four-billion-parameter adapter at epoch 5. The in-distribution column excludes A123. In the five-cell row, it includes the Malaya fit. Green marks a lower error and red a higher error.

Run	Training Cells	In-Dist.	A123
Three + FiLM	Pan, LG, INR	**1.13**	**39.18**
Four cells	+ Samsung 30T	**1.18**	**41.75**
Five cells	**+ Malaya LFP**	1.17	**35.17**

Two interventions that target the trajectory rather than the initial condition do not reduce the zero-shot error, which is consistent with the factorization.

Removing the voltage-difference feature. The hypothesis was that an explicit ΔV input acts as a voltage-lookup shortcut that blocks cross-chemistry transfer, and that forcing the model to integrate current instead would help. With the feature removed the held-out LFP error stayed at 41.8–42.9%, with no reduction relative to the baseline. The hypothesis is rejected. Because in-distribution accuracy was unaffected, the simpler three-channel prompt was retained as canonical.

Feature-wise conditioning of the soft prompt. The hypothesis was that a single global adapter cannot serve several cells, and that modulating the soft prompt by a per-cell descriptor would allow the unseen chemistry to be addressed. Held-out LFP error moved only from about 42% to 39%, inside the run-to-run spread. The reason is structural: the modulation network receives gradient only at descriptor values corresponding to NCA and NMC cells, so it has no anchor in the region of feature space that the LFP cell occupies. A static specification vector cannot supply information that is absent from the training data. Unlike the ΔV and ΔAh channels, the conditioning path was nonetheless retained in

the final configuration described in Section 3.5, as it is zero-initialized and did not affect in-distribution accuracy either way; what is reported here as negative is specifically its failure to deliver the cross-chemistry transfer for which it was introduced.

Both negative results are reported because they constrain the design of subsequent studies and because they are consistent with the factorization.

4.7. Computational Cost

Because the estimator is several orders of magnitude larger than the recursive filters and compact networks it would have to displace, cost is reported alongside accuracy rather than left implicit. Table 14 gives the quantities that determine whether an estimator of this class is deployable and, if so, where.

Table 14. Computational and storage cost of the five-cell model, as measured on the training run.

Quantity	Value
Backbone parameters, frozen	4.0×10^9
Trainable parameters	22.3×10^6 (0.55%)
—initial recurrent states S_0	12.58×10^6
—key–value prefix encoder	9.66×10^6
Backbone storage, bfloat16	≈8 GB
Adapter checkpoint	≈90 MB
Training hardware	1 × RTX PRO 6000
Training time	≈1.9 h per epoch; ≈9.5 h for five epochs

Two ratios matter more than the absolute figures. The adapter is two orders of magnitude smaller than the backbone in storage, so serving many cell-specific adapters over one shared frozen model is inexpensive, and adapting to a new cell means distributing 90 MB rather than 8 GB. Against that, the estimator emits 20 numbers per window by autoregressive greedy decoding, which is the cost driver at inference and is what separates this class of estimator from an observer that evaluates a closed-form update per sample. Section 5.4 states the deployment scope that follows: a four-billion-parameter decoder is a cloud- or fleet-side instrument, not an on-board estimator. Inference was measured at batch size 8. The adapter takes 0.67 s per window, peaks at 11.6 GB, and processes 1.5 windows per second. The gated recurrent unit takes 0.007 s per window, stays under 2 GB, and processes about 140 windows per second. The 11.6 GB peak is the inference footprint, not the 8 GB weight storage and not the training allocation. The adapter is about 100 times slower than the gated recurrent unit at nearly the same pooled error. Table 15 records the comparison.

Table 15. Inference at batch size 8. Time is per window. The gated-recurrent-unit memory is an upper bound. Throughput of 1.5 windows per second is the reciprocal of 0.67 s, and about 140 windows per second is the reported rate for 0.007 s.

Model	Batch	Time per Window (s)	Peak GPU Memory	Windows per Second
4B adapter	8	0.67	11.6 GB	1.5
GRU	8	0.007	<2 GB	140

4.8. Compact Models on the Same Split

Three compact sequence models were trained from scratch on the same voltage, current and temperature windows and the same drive-cycle split as the epoch-5 adapter. Malaya and A123 are excluded from the pool, as in the headline. A gated recurrent unit with 0.2 million parameters reaches 1.18% pooled MAE. A bidirectional long short-term memory network with 0.3 million parameters reaches 1.39%. A temporal convolutional

network with 0.5 million parameters reaches 1.50%. The adapter, 22.3 million trained parameters on the frozen backbone, reaches 1.17%. Those two pools are the first of three independent trainings. Table 16 reports that comparison.

Table 16. Compact models and the adapter on one split. Entries are mean absolute error in percent on the 18,712 drive-cycle-held-out windows, with Malaya and A123 excluded. Pools are window-weighted. Rho is the fraction of residual energy in the constant per-window offset. The adapter row is the epoch-5 model. Each row is one training. The repeated trainings of the adapter and the gated recurrent unit are in the next table.

Model	Parameters	Panasonic	LG	INR	Samsung	Pool	ρ
GRU	0.2×10^6	1.38	0.88	1.43	2.45	1.18	0.97
BiLSTM	0.3×10^6	1.43	1.30	1.44	2.16	1.39	0.96
TCN	0.5×10^6	1.41	1.53	1.87	2.32	1.50	0.94
4B adapter	22.3×10^6	1.32	0.94	1.59	2.16	1.17	0.98

Repeating the adapter and the gated recurrent unit three times gives pooled MAE of 1.17%, 1.15% and 1.23% for the adapter, and 1.18%, 1.12% and 1.24% for the gated recurrent unit. The sample means and sample standard deviations, with three trainings, are 1.18% plus or minus 0.04% and 1.18% plus or minus 0.06%. The intervals overlap. The bidirectional LSTM, the temporal convolutional network and the ablation remain one training each. The cell table keeps the first training, not the mean. Table 17 lists the three trainings. With three trainings per model no significance test is reported. The difference of 0.01 percentage points between the first trainings is smaller than either sample standard deviation, so the two models are not distinguished on this split. Single-training differences elsewhere in this section that are smaller than about 0.06 percentage points should be read in the same way.

The pooled error of the gated recurrent unit equals that of the adapter within the training-to-training spread. The other two compact models have higher pooled errors. LG is the lowest cell error in every row except the temporal convolutional network, where LG is 1.53% and Panasonic is 1.41%. That reversal is not used as a chemistry result. The offset share of residual energy is 0.97, 0.96, 0.94 and 0.98. The residual remains an initial-value error for the compact models as well.

Table 17. Three independent trainings of the epoch-5 adapter and of the gated recurrent unit. Entries are pooled mean absolute error in percent on the 18,712 nickel-based windows. The mean and the spread are the sample statistics for three trainings. The first training is the one reported in the tables above.

Model	Training 1	Training 2	Training 3	Mean $\pm$ Sample SD
4B adapter	1.17	1.15	1.23	1.18 ± 0.04
GRU	1.18	1.12	1.24	1.18 ± 0.06

5. Discussion

5.1. Role of the Language Model in the Estimate

The decomposition in Section 4.2 addresses a question that the literature on language models for time series has largely left open, namely, which part of the task the model performs [28,29]. The frozen recurrent stack, once given a trainable initial state, integrates charge to within 1.6% of the residual on the diagnostic and to within 2% of the residual energy on the final model, and the remaining error is concentrated in a single inverse mapping—voltage and temperature at the window start to SoC at the window start. This role is narrower than end-to-end state-of-charge estimation, and it offers one way to reconcile the skeptical result of [29] with the optimistic position of [28]: the model may

supply a useful prior for one inference, while the remainder of the window follows from the charge balance.

This interpretation is also consistent with the observation that the recurrent-state path dominates the key–value path by a factor of 2.3 on the diagnostic. On the final configuration the factor is 1.86. The initial-state-only row attains a pooled mean absolute error of 1.50%, compared with 2.18% for key–value prefixes alone. The initial state therefore accounts for the larger reduction, and the prefixes contribute a further reduction. The state S_0 is the initial condition of the recurrence, and the initial state of charge is the only degree of freedom of the task that the input current does not determine. Prefix tuning on the attention layers acts on the estimate through what those layers write into the residual stream. The initial state acts on the recurrence before the first token. Because it is shared by all windows, it supplies a learned default state from which the prompt moves the recurrence to a window-specific estimate. The ablation shows that this path carries the larger reduction. It does not show which internal quantity encodes the initial value, and the gated recurrent unit reaches the same pooled error without either path.

5.2. Accuracy Relative to Compact Estimators

For any language-model estimator, the accuracy obtained for the additional parameters has to be quantified. On the Panasonic 18650PF cell used here, a long short-term memory network of conventional size reported 0.573% MAE at fixed ambient temperature and 1.606% on a dataset with ambient temperature rising from 10 °C to 25 °C [12]. Our figure for the same cell is 1.31% over five temperature strata from −20 °C to 25 °C, and the reference prompt-based LLM estimator reports 1.46% over the same five strata [30]. The temperature ranges are not identical and the splits differ, so these numbers do not constitute a controlled comparison; they indicate the range of errors already reported for this cell. A recurrent network of a few hundred thousand parameters, published in 2018, already estimates SoC on this cell to within roughly one percent. Neither this work nor the LLM-based estimators that precede it [30,31,33] has demonstrated that a billion-parameter backbone improves on that. A controlled comparison on the present split is given in Table 16: the gated recurrent unit reaches 1.18% and the adapter reaches 1.17%.

No accuracy advantage is therefore claimed. This outcome is consistent with the factorization. If 98.4% of the diagnostic residual is the $t = 0$ inverse map, then additional sequence-modeling capacity—more parameters, better long-range recall, a richer adapter, a more carefully designed prompt—acts on the component that contributes 1.6% of that residual. On the final configuration the offset accounts for 0.98 of the residual energy, so the same bound applies to a shape term of 0.02. The binding constraint lies in the conditioning of an inverse problem in the data, not in the capacity of the estimator. Under that reading, the convergence of methods spanning four orders of magnitude in parameter count onto the same one-percent band is consistent with a task-imposed ceiling, although the present data do not exclude other explanations, such as the label construction or the window length, and the large model serves here mainly as an instrument that makes that ceiling measurable.

Two consequences follow from this result. For the field, the research objective shifts: effort spent on larger backbones or richer prompt semantics is unlikely to shift the aggregate, whereas effort spent on conditioning the initial estimate—a relaxation-window feature, OCV-anchored initialization, an explicit low-temperature resistance correction, or a prior supplied by a model-based observer at $t = 0$—acts directly on the dominant term. For this paper, the contribution is relocated where the evidence supports it: the architectural result that adaptation must be defined per layer type (Section 5.1), the mechanistic result that identifies what the model computes (Section 4.2), and the two failure regimes that

the factorization predicts (Section 4.5). None of those requires the estimator to be the most accurate one available, and all three transfer to the compact estimators that a battery management unit can run.

5.3. Relation to State-Based Parameter-Efficient Fine-Tuning

Trainable initial recurrent states are not new. Initial-state tuning has been evaluated within a systematic study of PEFT for state-space models [40], PEFT methods for Mamba have been benchmarked and extended [65], and state-based methods that act on the recurrent state at every step have been proposed [66]. The present work does not introduce that mechanism, and its claims are narrower: that it works inside a hybrid decoder where recurrent layers coexist with attention layers requiring a different interface; that it supports continuous regression from physical sensor streams rather than language classification; and that in this task the trained state admits a working interpretation as a learned default initial state from which testable predictions follow.

The most relevant prior finding also qualifies the present result. Kang et al. report that the influence of a trained initial state diminishes as the recurrence proceeds, and they replace it with a per-step state offset [66]. Our windows are 20 tokens of numeric payload, short enough that the decay has little opportunity to act, which is a likely reason why the initial-state formulation suffices here. It also delimits the method: applied to substantially longer windows, initial-state tuning should be expected to weaken, and the state-offset formulation of [66] becomes the appropriate choice. This expectation is a prediction of the present analysis and can be tested by varying T. No such sweep was run. Every reported window has 20 steps, about 20 s. On that horizon, the initial state still reaches the tokens that carry the estimate. On substantially longer windows the same state is expected to fade, and the per-step state offset is the appropriate replacement.

5.4. Implications for Practice

Three findings transfer beyond this architecture.

First, the loss-weighting rule. Because the residual is dominated by one predicted value, upweighting the loss on the tokens that emit it improved accuracy by 13% at zero parameter cost—more than the feature engineering it followed on the single-cell diagnostic. On the final configuration the same weighting lowers the pooled error from 1.25% to 1.17%, a relative reduction of about 6%. Any sequence estimator whose error decomposes this way admits the same intervention, and the decomposition of Section 3.10 is cheap to compute on an existing model.

Second, the capacity-label check. A public dataset used here is distributed with a nominal capacity that disagrees with its own coulomb-counted content by a factor of 2.3. Because SoC labels are constructed by dividing by that number, using the published label would have pushed targets outside $[0, 1]$ and corrupted training without raising an error. Validating C_{nom} against the data—by coulomb-counting a full discharge and by decoding a known-rate constant-current file—requires only minutes and is recommended as a routine check.

Third, split protocol. An earlier configuration of this work reported 0.91% on a random file-level split, and that figure was not reproduced after the change to drive-cycle holdout with the cold strata restored. We state this as a caution rather than a measurement, because the backbone size and the cell set changed over the same transition, and the protocol is therefore not isolated. The caution is recorded because a random file-level split places windows from the same drive cycle on both sides of the partition, so it tests interpolation within a cycle rather than generalization to an unseen one, and headline errors obtained under the two protocols should not be compared without saying which was used.

Finally, scope. A four-billion-parameter backbone in bfloat16 does not run on a battery management unit, and no such use is claimed; the realistic targets are fleet- or cloud-side estimation, offline label generation for smaller models, and use as a diagnostic instrument. The central result, however, is a statement about the task and not about the backbone. If accuracy is governed by the initial-condition inverse problem, that holds for a compact recurrent estimator on a microcontroller as much as for a four-billion-parameter decoder, and it is on the compact estimator that the reordering of priorities in Section 5.2 is most useful.

5.5. A Candidate Precision Ceiling in the Prompt Format

One consequence of the factorization appears not to have been examined in the literature, and it bears on the accuracy ceiling of prompt-based estimation as a class rather than on this architecture in particular.

The estimator observes its measurements as text. In the prompt of Section 3.3, voltage is rendered with two decimal places, so the model never sees voltage more finely than 0.01 V. As almost all error is the $t = 0$ inverse map, and that map converts a voltage reading into an SoC, the quantization of the input propagates directly into a floor on the recoverable initial value. Its size is $\Delta V / (\mathrm{d}V / \mathrm{d}\,\mathrm{SoC})$. Taking mid-range slopes of roughly 0.6 V per unit SoC for NCA and NMC and roughly 0.2 V per unit SoC on the LFP plateau gives implied resolutions of about 0.017 and about 0.05 SoC, respectively.

The first of these values is of the same order as the measured error. The held-out aggregate is 1.17% MAE. The offset share of 98.4% was obtained on the diagnostic configuration. On the final configuration the offset accounts for 0.98 of the residual energy. The quantity being measured therefore still sits at the same order of magnitude as the resolution at which the input is presented. This raises the possibility that the reported accuracy—ours and, by the same argument, that of any estimator serializing measurements at two decimal places—is limited by the numerical precision of the text representation rather than by the capacity of the model or the design of the adapter.

A paired training tests this possibility. The configuration is the epoch-5 adapter, the same 18,712 windows, and with Malaya and A123 excluded. Only the number of voltage decimals changes, and the three-decimal prompt is used in training, not only at evaluation. The pool moves from 1.17% to 1.13%. The offset share stays 0.98. A third digit therefore does not remove the initial-value error, and it is not the binding constraint on this split. The headline and the factor 1.86 remain the two-decimal model. The 1.13% entry in the composition table is a different run. Table 18 shows that comparison: two decimal places against three.

Table 18. Voltage precision in the prompt. Both rows are the epoch-5 adapter on the 18,712 nickel-based windows. Row 2 is the reported model. Row 3 is a separate training. Rho is the offset share of residual energy.

Decimal Places	Role	Pooled MAE (%)	Offset Share
2	reported model	1.17	0.98
3	paired training	1.13	0.98

5.6. Limitations

The evaluation protocol carries the four weaknesses set out in Section 3.11. The 1.22% Malaya figure remains a fit. Held-out files of that cell reach 2.53% after a separate training run, whose nickel-based pool is 1.24%. The claim that one model spans three chemistries at 1.17% is therefore not made. Nickel-based cells are reported at 1.17% on the main run. LFP is reported separately. The 35.17% A123 figure is the original anchor, and its predictions are

not the reference-range predictions. On the reference range the error is 12%, the prediction mean is 0.30, the label mean is 0.31, the prediction span is 0.12, and the slope is 0.16. The plateau is a plausible contributor. It is not identified as the sole cause.

Further limitations are stated here. A gated recurrent unit, a bidirectional LSTM and a temporal convolutional network were trained from scratch on this split. Their pooled errors are 1.18%, 1.39% and 1.50%. The adapter reaches 1.17%. On these windows the gated recurrent unit matches the adapter. For the final configuration, the initial-state-only control gives a pooled mean absolute error of 1.50%, compared with 2.18% for key–value prefixes alone and 1.17% for the full adapter. On this configuration, the offset accounts for 0.98 of the residual energy. The factor of 2.3 and the offset share of 98.4% refer to the single-cell, nine-billion-parameter diagnostic.

Repeated trainings exist only for the adapter and the gated recurrent unit. The ablation, the bidirectional LSTM, the temporal convolutional network, the validation-split run and the three-decimal run are single trainings, and differences between them below the observed spread of about 0.06 percentage points are not resolved. The temperature groups of the offset share have no recorded window counts. Every window has 20 steps, so the behavior of the initial-state path on longer windows is a prediction, not a measurement. Labels are coulomb-counted against nominal capacity without a temperature- or aging-dependent capacity correction, and the cells were not studied across aging, so the results do not extend to state-of-health variation.

6. Conclusions

We adapted a frozen hybrid-attention large language model to lithium-ion state-of-charge estimation by matching the parameter-efficient adaptation mechanism to each specific layer type, and used the resulting interpretability to analyze how the estimator operates.

1. In hybrid decoders interleaving linear-attention and full-attention blocks (e.g., 3:1 ratio), standard KV-prefix tuning ignores the majority of the network. Injecting a trainable initial recurrent state (S_0) into the linear-attention majority, alongside a KV-prefix in the attention minority, reduces error by a factor of 2.3 compared to adapting attention layers alone on the diagnostic. On the final four-billion-parameter configuration the factor is 1.86.
2. A single adapter representing just 0.55% of the frozen 4B backbone reports the nickel-based cells from $-20\,^{\circ}\mathrm{C}$ to $45\,^{\circ}\mathrm{C}$ at 1.17% pooled MAE over 18,712 drive-cycle-held-out windows. The LFP fit is 1.22%. Held-out files of that cell reach 2.53% in a separate run, whose nickel-based pool is 1.24%. This result shows that a hybrid LLM can reach parity with dense LLM estimators on the nickel-based task.
3. **Mechanistic factorization:** The residual error is dominated (98.4% on the diagnostic; 0.98 on the final configuration) by a constant per-window offset. Within a window the estimate follows the integrated current closely, and the remaining error is set mainly by an initial-condition inverse problem at the window start. Compact models on the same windows show the same pattern. The task's accuracy ceiling is therefore bounded mainly by the conditioning of the $V_{t=0} \mapsto \mathrm{SoC}_{t=0}$ mapping. On this split, the gated recurrent unit reaches 1.18% pooled MAE and the adapter reaches 1.17%. The additional backbone parameters do not lower the pool.
4. This factorization is consistent with the observed failure regimes. Error is highest below $0\,^{\circ}\mathrm{C}$, where internal resistance corrupts the initial voltage signature. Zero-shot transfer to an LFP cell is 12% MAE on the reference state-of-charge range. The prediction mean is 0.30, the label mean is 0.31, and the prediction span is 0.12. The

slope on the label is 0.16. The voltage plateau is a plausible contributor to that gap and is not identified as its sole cause.

5. **Practical engineering impact:** Identifying the $t = 0$ bottleneck indicates where optimization effort should be directed. For example, upweighting the training loss exclusively on the first emitted value improved accuracy by 13% at no parameter cost on the diagnostic. On the final configuration the same weighting reduces the pooled mean absolute error from 1.25% to 1.17%. Future engineering should focus on conditioning the initial estimate—through relaxation features, OCV-anchored initialization, or incorporating model-based priors—rather than scaling the sequence backbone.

A sweep of window length remains for future work: the initial-state path is evaluated only at 20 steps, and longer windows are expected to require a per-step state offset.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The Panasonic 18650PF [41], LG 18650HG2 [42], Samsung INR21700-30T [43] and Malaya [44] datasets are publicly available at the cited sources. The CALCE INR 18650-20R and CALCE A123 dynamic-profile files are distributed by the Center for Advanced Life Cycle Engineering, University of Maryland [67,68]. Training and evaluation code will be deposited upon publication.

Conflicts of Interest: The author declares no conflicts of interest.

Nomenclature

The following symbols are used in this manuscript:

C_{nom}	nominal cell capacity (Ah)
c	per-cell descriptor vector, $\mathbb{R}^6$
d_{h}	head dimension of the full-attention layers
$d_{\mathrm{k}}, d_{\mathrm{v}}$	key and value head dimensions of the gated-delta layers
D	width of the key–value prefix, $2L_{\mathrm{a}}H_{\mathrm{kv}}d_{\mathrm{h}}$
E	learned soft-prefix embedding, $\mathbb{R}^{M\times D}$
g	learned per-channel gate
H_{kv}	key–value head count of the full-attention layers
H_{v}	value-head count of the gated-delta layers
I_t	load current at step t (A; negative on discharge)
$L_{\mathrm{a}}, L_{\mathrm{g}}$	number of full-attention and gated-delta layers
$\mathcal{L}$	training objective
M	number of soft prefix positions
P	key–value prefix delivered to the attention layers
r	bottleneck width of the prefix encoder
S	recurrent state of a gated-delta layer, $\mathbb{R}^{d_{\mathrm{k}}\times d_{\mathrm{v}}}$
S_0	trainable initial recurrent state
T	number of retained samples per window
V_t	terminal voltage at step t (V)
w_t	per-token loss weight
$y_t, \hat{y}_t$	true and estimated SoC at step t
γ, β	feature-wise modulation scale and shift
ϑ_t	cell surface temperature at step t (°C)
λ	loss multiplier on the first emitted SoC value
BMS	Battery management system

EKF	Extended Kalman filter
FiLM	Feature-wise linear modulation
GDN	Gated DeltaNet
KV	Key–value
LFP	Lithium iron phosphate
LLM	Large language model
LSTM	Long short-term memory
GRU	Gated recurrent unit
BiLSTM	Bidirectional long short-term memory
TCN	Temporal convolutional network
MAE	Mean absolute error
NCA	Lithium nickel cobalt aluminium oxide
NMC	Lithium nickel manganese cobalt oxide
OCV	Open-circuit voltage
PEFT	Parameter-efficient fine-tuning
SoC	State of charge
V/I/T	Voltage, current and temperature

References

1. Lu, L.; Han, X.; Li, J.; Hua, J.; Ouyang, M. A review on the key issues for lithium-ion battery management in electric vehicles. *J. Power Sources* **2013**, *226*, 272–288. [CrossRef]
2. Hu, X.; Xu, L.; Lin, X.; Pecht, M. Battery Lifetime Prognostics. *Joule* **2020**, *4*, 310–346. [CrossRef]
3. How, D.N.T.; Hannan, M.A.; Hossain Lipu, M.S.; Ker, P.J. State of Charge Estimation for Lithium-Ion Batteries Using Model-Based and Data-Driven Methods: A Review. *IEEE Access* **2019**, *7*, 136116–136136. [CrossRef]
4. Hu, X.; Li, S.; Peng, H. A comparative study of equivalent circuit models for Li-ion batteries. *J. Power Sources* **2012**, *198*, 359–367. [CrossRef]
5. Plett, G.L. Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs. *J. Power Sources* **2004**, *134*, 262–276. [CrossRef]
6. Severson, K.A.; Attia, P.M.; Jin, N.; Perkins, N.; Jiang, B.; Yang, Z.; Chen, M.H.; Aykol, M.; Herring, P.K.; Fraggedakis, D.; et al. Data-driven prediction of battery cycle life before capacity degradation. *Nat. Energy* **2019**, *4*, 383–391. [CrossRef]
7. Attia, P.M.; Grover, A.; Jin, N.; Severson, K.A.; Markov, T.M.; Liao, Y.; Chen, M.H.; Cheong, B.; Perkins, N.; Yang, Z.; et al. Closed-loop optimization of fast-charging protocols for batteries with machine learning. *Nature* **2020**, *578*, 397–402. [CrossRef] [PubMed]
8. Sulzer, V.; Mohtat, P.; Aitio, A.; Lee, S.; Yeh, Y.T.; Steinbacher, F.; Khan, M.U.; Lee, J.W.; Siegel, J.B.; Stefanopoulou, A.G.; et al. The challenge and opportunity of battery lifetime prediction from field data. *Joule* **2021**, *5*, 1934–1955. [CrossRef]
9. Chemali, E.; Kollmeyer, P.J.; Preindl, M.; Emadi, A. State-of-charge estimation of Li-ion batteries using deep neural networks: A machine learning approach. *J. Power Sources* **2018**, *400*, 242–255. [CrossRef]
10. Zhou, Z.; Wang, Y.; Zhou, G.; Liu, X.; Wu, M.; Dai, K. Vehicle Lateral Dynamics-Inspired Hybrid Model Using Neural Network for Parameter Identification and Error Characterization. *IEEE Trans. Veh. Technol.* **2024**, *73*, 16173–16186.. [CrossRef]
11. Zhou, Z.; Wang, Y.; Liu, X.; Li, Z.; Wu, M.; Zhou, G. Hybrid of Neural Network and Physics-Based Estimator for Vehicle Longitudinal Dynamics Modeling Using Limited Driving Data. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 16735–16746.. [CrossRef]
12. Chemali, E.; Kollmeyer, P.J.; Preindl, M.; Ahmed, R.; Emadi, A. Long Short-Term Memory Networks for Accurate State-of-Charge Estimation of Li-ion Batteries. *IEEE Trans. Ind. Electron.* **2018**, *65*, 6730–6739. [CrossRef]
13. Feng, X.; Chen, J.; Zhang, Z.; Miao, S.; Zhu, Q. State-of-charge estimation of lithium-ion battery based on clockwork recurrent neural network. *Energy* **2021**, *236*, 121360. [CrossRef]
14. Wu, L.; Zhang, Y. Attention-based encoder-decoder networks for state of charge estimation of lithium-ion battery. *Energy* **2023**, *268*, 126665. [CrossRef]
15. Vidal, C.; Malysz, P.; Kollmeyer, P.; Emadi, A. Machine Learning Applied to Electrified Vehicle Battery State of Charge and State of Health Estimation: State-of-the-Art. *IEEE Access* **2020**, *8*, 52796–52814. [CrossRef]
16. Vidal, C.; Gross, O.; Gu, R.; Kollmeyer, P.; Emadi, A. xEV Li-Ion Battery Low-Temperature Effects—Review. *IEEE Trans. Veh. Technol.* **2019**, *68*, 4560–4572. [CrossRef]
17. Vidal, C.; Kollmeyer, P.; Chemali, E.; Emadi, A. Li-ion Battery State of Charge Estimation Using Long Short-Term Memory Recurrent Neural Network with Transfer Learning. In Proceedings of the 2019 IEEE Transportation Electrification Conference and Expo (ITEC), Novi, MI, USA, 19–21 June 2019; pp. 1–6. [CrossRef]

18. Oyewole, I.; Chehade, A.; Kim, Y. A controllable deep transfer learning network with multiple domain adaptation for battery state-of-charge estimation. *Appl. Energy* **2022**, *312*, 118726. [CrossRef]
19. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762. [CrossRef]
20. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. *arXiv* **2020**, arXiv:2005.14165. [CrossRef]
21. Xue, H.; Salim, F.D. PromptCast: A New Prompt-Based Learning Paradigm for Time Series Forecasting. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 6851–6864. [CrossRef]
22. Gruver, N.; Finzi, M.; Qiu, S.; Wilson, A.G. Large Language Models Are Zero-Shot Time Series Forecasters. *arXiv* **2023**, arXiv:2310.07820. [CrossRef]
23. Jin, M.; Wang, S.; Ma, L.; Chu, Z.; Zhang, J.Y.; Shi, X.; Chen, P.; Liang, Y.; Li, Y.; Pan, S.; et al. Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. *arXiv* **2023**, arXiv:2310.01728. [CrossRef]
24. Sun, C.; Li, H.; Li, Y.; Hong, S. TEST: Text Prototype Aligned Embedding to Activate LLM's Ability for Time Series. *arXiv* **2023**, arXiv:2308.08241. [CrossRef]
25. Ansari, A.F.; Stella, L.; Turkmen, C.; Zhang, X.; Mercado, P.; Shen, H.; Shchur, O.; Rangapuram, S.S.; Arango, S.P.; Kapoor, S.; et al. Chronos: Learning the Language of Time Series. *arXiv* **2024**, arXiv:2403.07815. [CrossRef]
26. Woo, G.; Liu, C.; Kumar, A.; Xiong, C.; Savarese, S.; Sahoo, D. Unified Training of Universal Time Series Forecasting Transformers. *arXiv* **2024**, arXiv:2402.02592. [CrossRef]
27. Das, A.; Kong, W.; Sen, R.; Zhou, Y. A decoder-only foundation model for time-series forecasting. *arXiv* **2023**, arXiv:2310.10688. [CrossRef]
28. Jin, M.; Zhang, Y.; Chen, W.; Zhang, K.; Liang, Y.; Yang, B.; Wang, J.; Pan, S.; Wen, Q. Position: What Can Large Language Models Tell Us about Time Series Analysis. *arXiv* **2024**, arXiv:2402.02713. [CrossRef]
29. Tan, M.; Merrill, M.A.; Gupta, V.; Althoff, T.; Hartvigsen, T. Are Language Models Actually Useful for Time Series Forecasting? *arXiv* **2024**, arXiv:2406.16964. [CrossRef]
30. Bian, C.; Duan, Z.; Hao, Y.; Yang, S.; Feng, J. Exploring large language model for generic and robust state-of-charge estimation of Li-ion batteries: A mixed prompt learning method. *Energy* **2024**, *302*, 131856. [CrossRef]
31. Bian, C.; Han, X.; Duan, Z.; Deng, C.; Yang, S.; Feng, J. Hybrid Prompt-Driven Large Language Model for Robust State-of-Charge Estimation of Multitype Li-ion Batteries. *IEEE Trans. Transp. Electrif.* **2025**, *11*, 426–437. [CrossRef]
32. Qiu, T.; Hou, L.; Shang, Y. Prompt-Driven Fine-Tuning of Large Language Model for Li-ion Battery State Estimation. In Proceedings of the 2024 8th CAA International Conference on Vehicular Control and Intelligence (CVCI), Chongqing, China, 25–27 October 2024; pp. 1–6. [CrossRef]
33. Liu, Y.; Mao, Z.; Chen, Z. Prompt-guided large language model for estimating the state of charge of containerized lithium-ion battery energy storage systems. *J. Energy Storage* **2026**, *174*, 123069. [CrossRef]
34. Zhang, W.; Feng, H.; Ren, M.; Liu, K.; Yan, G. LLM-MSIformer: A method for lithium-ion battery state of health estimation based on a large language model and multi-time scale interval transformer. *Energy* **2026**, *342*, 139417. [CrossRef]
35. Qwen Team. Qwen3.5-4B, Model Card and Configuration. Hugging Face. Available online: https://huggingface.co/Qwen/Qwen3.5-4B (accessed on 3 October 2026).
36. Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. Qwen3 Technical Report. *arXiv* **2025**, arXiv:2505.09388. [CrossRef]
37. Yang, S.; Kautz, J.; Hatamizadeh, A. Gated Delta Networks: Improving Mamba2 with Delta Rule. *arXiv* **2024**, arXiv:2412.06464. [CrossRef]
38. Li, X.L.; Liang, P. Prefix-Tuning: Optimizing Continuous Prompts for Generation. *arXiv* **2021**, arXiv:2101.00190. [CrossRef]
39. Liu, X.; Ji, K.; Fu, Y.; Tam, W.L.; Du, Z.; Yang, Z.; Tang, J. P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks. *arXiv* **2021**, arXiv:2110.07602. [CrossRef]
40. Galim, K.; Kang, W.; Zeng, Y.; Koo, H.I.; Lee, K. Parameter-Efficient Fine-Tuning of State Space Models. *arXiv* **2024**, arXiv:2410.09016. [CrossRef]
41. Kollmeyer, P. Panasonic 18650PF Li-ion Battery Data [Dataset]. Mendeley 2018. Available online: https://data.mendeley.com/datasets/wykht8y7tg/1 (accessed on 4 September 2026).
42. Naguib, M. LG 18650HG2 Li-ion Battery Data [Dataset]. Mendeley 2020. Available online: https://data.mendeley.com/datasets/b5mj79w5w9/2 (accessed on 4 September 2026).
43. Kollmeyer, P. Samsung INR21700 30T 3Ah Li-ion Battery Data [Dataset]. Mendeley Data 2023. Available online: https://data.mendeley.com/datasets/9xyvy2njj3/2 (accessed on 4 September 2026).
44. Shrivastava, P.; Soon, T.K.; Idris, M.Y.I.B.; Mekhilef, S.; Adnan, S.B.R.S. Comprehensive co-estimation of lithium-ion battery state of charge, state of energy, state of power, maximum available capacity, and maximum available energy. *J. Energy Storage* **2022**, *56*, 106049. [CrossRef]

45. Dong, G.; Wei, J.; Zhang, C.; Chen, Z. Online state of charge estimation and open circuit voltage hysteresis modeling of LiFePO4 battery using invariant imbedding method. *Appl. Energy* **2016**, *162*, 163–171. [CrossRef]
46. Xiong, R.; Duan, Y.; Zhang, K.; Lin, D.; Tian, J.; Chen, C. State-of-charge estimation for onboard LiFePO4 batteries with adaptive state update in specific open-circuit-voltage ranges. *Appl. Energy* **2023**, *349*, 121581. [CrossRef]
47. Zhang, K.; Xiong, R.; Li, Q.; Chen, C.; Tian, J.; Shen, W. A novel pseudo-open-circuit voltage modeling method for accurate state-of-charge estimation of LiFePO4 batteries. *Appl. Energy* **2023**, *347*, 121406. [CrossRef]
48. Schmitt, J.; Horstkötter, I.; Bäker, B. State of charge estimation with hysteresis-prone open circuit voltage in lithium-ion batteries using the trajectory correction hysteresis (TCH) model. *J. Power Sources Adv.* **2024**, *28*, 100151. [CrossRef]
49. Perez, E.; Strub, F.; Vries, H.d.; Dumoulin, V.; Courville, A. FiLM: Visual Reasoning with a General Conditioning Layer. *arXiv* **2017**, arXiv:1709.07871. [CrossRef]
50. Lieber, O.; Lenz, B.; Bata, H.; Cohen, G.; Osin, J.; Dalmedigos, I.; Safahi, E.; Meirom, S.; Belinkov, Y.; Shalev-Shwartz, S.; et al. Jamba: A Hybrid Transformer-Mamba Language Model. *arXiv* **2024**, arXiv:2403.19887. [CrossRef]
51. Ren, L.; Liu, Y.; Lu, Y.; Shen, Y.; Liang, C.; Chen, W. Samba: Simple Hybrid State Space Models for Efficient Unlimited Context Language Modeling. *arXiv* **2024**, arXiv:2406.07522. [CrossRef]
52. NVIDIA; Blakeman, A.; Basant, A.; Khattar, A.; Renduchintala, A.; Bercovich, A.; Ficek, A.; Bjorlin, A.; Taghibakhshi, A.; Deshmukh, A.S.; et al. Nemotron-H: A Family of Accurate and Efficient Hybrid Mamba-Transformer Models. *arXiv* **2025**, arXiv:2504.03624. [CrossRef]
53. Zuo, J.; Velikanov, M.; Chahed, I.; Belkada, Y.; Rhayem, D.E.; Kunsch, G.; Hacid, H.; Yous, H.; Farhat, B.; Khadraoui, I.; et al. Falcon-H1: A Family of Hybrid-Head Language Models Redefining Efficiency and Performance. *arXiv* **2025**, arXiv:2507.22448. [CrossRef]
54. Katharopoulos, A.; Vyas, A.; Pappas, N.; Fleuret, F. Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention. *arXiv* **2020**, arXiv:2006.16236. [CrossRef]
55. Yang, S.; Wang, B.; Shen, Y.; P.; Panda, R.; Kim, Y. Gated Linear Attention Transformers with Hardware-Efficient Training. *arXiv* **2023**, arXiv:2312.06635. [CrossRef]
56. Sun, Y.; Dong, L.; Huang, S.; Ma, S.; Xia, Y.; Xue, J.; Wang, J.; Wei, F. Retentive Network: A Successor to Transformer for Large Language Models. *arXiv* **2023**, arXiv:2307.08621. [CrossRef]
57. Peng, B.; Alcaide, E.; Anthony, Q.; Albalak, A.; Arcadinho, S.; Biderman, S.; Cao, H.; Cheng, X.; Chung, M.; Grella, M.; et al. RWKV: Reinventing RNNs for the Transformer Era. *arXiv* **2023**, arXiv:2305.13048. [CrossRef]
58. Peng, B.; Goldstein, D.; Anthony, Q.; Albalak, A.; Alcaide, E.; Biderman, S.; Cheah, E.; Du, X.; Ferdinan, T.; Hou, H.; et al. Eagle and Finch: RWKV with Matrix-Valued States and Dynamic Recurrence. *arXiv* **2024**, arXiv:2404.05892. [CrossRef]
59. Gu, A.; Dao, T. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. *arXiv* **2023**, arXiv:2312.00752. [CrossRef]
60. Dao, T.; Gu, A.Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality. *arXiv* **2024**, arXiv:2405.21060. [CrossRef]
61. Yang, S.; Wang, B.; Zhang, Y.; Shen, Y.; Kim, Y. Parallelizing Linear Transformers with the Delta Rule over Sequence Length. *arXiv* **2024**, arXiv:2406.06484. [CrossRef]
62. Lester, B.; Al-Rfou, R.; Constant, N. The Power of Scale for Parameter-Efficient Prompt Tuning. *arXiv* **2021**, arXiv:2104.08691. [CrossRef]
63. Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; Laroussilhe, Q.d.; Gesmundo, A.; Attariyan, M.; Gelly, S. Parameter-Efficient Transfer Learning for NLP. *arXiv* **2019**, arXiv:1902.00751. [CrossRef]
64. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv* **2021**, arXiv:2106.09685. [CrossRef]
65. Yoshimura, M.; Hayashi, T.; Maeda, Y. MambaPEFT: Exploring Parameter-Efficient Fine-Tuning for Mamba. *arXiv* **2024**, arXiv:2411.03855. [CrossRef]
66. Kang, W.; Galim, K.; Zeng, Y.; Lee, M.; Koo, H.I.; Cho, N.I. State-offset Tuning: State-based Parameter-Efficient Fine-Tuning for State Space Models. *arXiv* **2025**, arXiv:2503.03499. [CrossRef]
67. Zheng, F.; Xing, Y.; Jiang, J.; Sun, B.; Kim, J.; Pecht, M. Influence of different open circuit voltage tests on state of charge online estimation for lithium-ion batteries. *Appl. Energy* **2016**, *183*, 513–525. [CrossRef]
68. Xing, Y.; He, W.; Pecht, M.; Tsui, K.L. State of charge estimation of lithium-ion batteries using the open-circuit voltage at various ambient temperatures. *Appl. Energy* **2014**, *113*, 106–115. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.