Test-Time Learning-Enhanced Transformer for Lithium-Ion Battery SOH Prediction

Bin Ye, Vahid Abolghasemi, Amit Kumar Singh, Emil Baruch, and Ramesh Raoraje

***Abstract*—Accurate state-of-health (SOH) estimation is essential for battery management systems, but conventional data-driven models remain fixed after training and may lose accuracy when deployment conditions change. This paper proposes a Test-Time Learning-Enhanced Transformer (TTL–Transformer) for heterogeneous LCO, NMC, and LFP battery data. The key novelty is that the model remains adaptable after deployment: as new unlabelled voltage, current, and temperature measurements become available, the CNN denoising module can be periodically updated to suit new batteries, ageing stages, temperatures, operating profiles, and measurement conditions. Meanwhile, the Transformer encoder and SOH regression head remain frozen, preserving the supervised SOH mapping and avoiding full-model retraining. Only 2.17% of the complete model is updated during this process. Under pooled multi-chemistry evaluation, the plain Transformer achieves an average RMSE of 1.20%. Adding the denoiser reduces RMSE to 1.05%, while post-deployment adaptation further reduces it to 0.79%, giving an overall improvement of 34.4%. The model also achieves RMSE values of 0.56–0.73% in matched-chemistry evaluation. These results show that the proposed architecture enables lightweight, label-free adaptation to changing conditions after deployment without modifying the trained SOH predictor.**

***Index Terms*—state of health, lithium-ion batteries, test-time learning, domain shift, Transformer, cross-chemistry generalization**

I. Introduction

Battery state of health (SOH) is an important indicator of degradation and is commonly defined as the ratio between the currently available capacity and the nominal capacity. Reliable SOH estimation supports battery safety, available-energy assessment, maintenance planning, and end-of-life decisions in battery management systems (BMSs) [1]–[3]. Unlike voltage, current, and temperature, SOH cannot be measured directly during normal operation. Reference capacity measurement generally requires a controlled charge–discharge procedure, which is difficult to perform routinely in practical applications [2], [4]. Consequently, SOH must be estimated from electrical and thermal measurements collected during battery operation.

Model-based SOH-estimation methods use electrochemical models or equivalent-circuit models to relate measured battery responses to ageing-dependent parameters [5], [6]. These methods provide useful physical interpretation, but their accuracy depends on the selected model structure and the reliability of parameter identification. Data-driven methods instead learn the relationship between measured signals and SOH from battery-ageing data. Convolutional neural networks (CNNs) have been used to extract local degradation features from partial voltage and current profiles [7]. Recurrent architectures, including recurrent neural networks, long short-term memory networks, and gated recurrent units, can represent temporal dependencies in charging, discharging, and ageing sequences [8]. Transformer-based models use self-attention to capture relationships over longer sequences and have also demonstrated promising performance in battery-health estimation [9]. More general machine-learning pipelines have achieved accurate SOH estimation across cells operated under different experimental conditions [3].

Despite these advances, most data-driven SOH models assume that the training and deployment data follow similar distributions. This assumption may not hold when batteries differ in chemistry, manufacturer, cell format, temperature, cycling protocol, ageing behaviour, or measurement system. These differences can change the relationship between voltage, current, temperature, and capacity degradation. As a result, a model trained on one group of batteries may lose accuracy when it is deployed on batteries outside its original training distribution [10], [11]. Transfer-learning and domain-adaptation methods can reduce this problem, but they may require labelled target-domain samples, offline retraining, or a sufficiently large target dataset [10]. A static predictor that remains completely fixed after deployment also cannot respond to distribution changes that emerge as the target battery ages.

Test-time learning (TTL) provides a potential mechanism for adapting a deployed model using unlabelled test data. Existing TTL methods construct auxiliary objectives from the test inputs and update selected model parameters before prediction. Examples include self-supervised prediction, entropy minimisation, feature alignment, and masked-input reconstruction [12]–[14]. However, many of these methods were developed for image-classification tasks and adapt shared feature representations, normalisation parameters, or substantial parts of the prediction backbone. Updating task-related representations using a self-supervised objective may alter the supervised mapping required for SOH regression, particularly when only a small quantity of target-battery data is available.

Test-time adaptation has also been explored specifically for battery SOH estimation. For example, BatteryTTT uses masked reconstruction together with an equivalent-circuit-model-based physical constraint and adapts a shared feature extractor or learned prompt parameters [11]. However, this approach relies on constructed voltage–capacity features and a physics-guided auxiliary objective. A lightweight method that adapts directly to multivariate voltage, current, and temperature measurements while leaving the trained SOH predictor unchanged remains insufficiently explored.

To address this limitation, this paper proposes a Test-Time Learning-Enhanced Transformer (TTL–Transformer) for SOH estimation across heterogeneous lithium-ion battery datasets. The architecture contains a one-dimensional CNN denoising front-end, a Transformer encoder, and an SOH regression head. During supervised source-domain training, all three components

This work was supported by a Knowledge Transfer Partnership (KTP) from Innovate U.K. under Partnership 13744 between Baruch Enterprise and the University of Essex.

Bin Ye, Vahid Abolghasemi and Amit Kumar Singh are with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK (e-mail: by24457@essex.ac.uk; v.abolghasemi@essex.ac.uk; a.k.singh@essex.ac.uk).

Emil Baruch and Ramesh Raoraje are with Baruch Enterprises (e-mail: emil@baruch.co.uk; process@ultramax.co.uk).

are optimised using labelled SOH data. During deployment, Gaussian noise and random masking are applied to unlabelled target-battery sequences, and the CNN denoiser is updated through a reconstruction objective. The Transformer encoder and SOH regression head remain fixed.

This separation confines adaptation to the input-processing stage. It allows the denoiser to adjust to variations in the target measurements without directly modifying the supervised SOH mapping learned by the Transformer and regression head. The methodological contribution is therefore not the standard Transformer architecture or the reconstruction objective itself. Rather, it is the selective use of reconstruction-based adaptation in a dedicated battery-signal denoiser while preserving the downstream SOH predictor.

The main contributions of this work are summarised as follows:

- We propose a test-time learning framework for SOH estimation under heterogeneous lithium-ion battery deployment conditions.
- We develop a selective adaptation strategy that updates only the CNN denoiser using unlabelled target data after deployment, while keeping the Transformer encoder and SOH regression head fixed. This preserves the supervised SOH mapping without requiring target SOH labels or full-model retraining.
- We evaluate the method using cell-disjoint matched-chemistry and pooled multi-chemistry protocols covering LCO, NMC, LFP1, and LFP2 cells. The ablation study separately quantifies the benefits of the CNN denoiser and the test-time learning update.
- Only 11,779 parameters (2.17% of the complete model) are updated during deployment, providing a lightweight adaptation scope for resource-constrained battery-management applications.

The novelty of the proposed TTL–Transformer lies in its selective test-time adaptation strategy, which enables the model to evolve after deployment. During operation, only the CNN denoiser is updated using unlabelled voltage, current, and temperature measurements, while the Transformer encoder and SOH regression head remain fixed. This allows the method to adapt to changing battery operating conditions and data distributions without requiring target-domain SOH labels or altering the trained SOH prediction model.

II. Related Work

A wide range of methods have been developed for battery SOH estimation. Model-based approaches, including electrochemical models and equivalent circuit models (ECMs), infer SOH by fitting physical or semi-physical parameters, such as capacity and internal resistance, to measured data. These methods are often combined with Bayesian filtering methods such as Kalman or particle filters [6]. Plett [5], for example, introduced extended Kalman filtering for joint State of Charge (SOC)/SOH tracking by treating capacity as a slowly varying state. Yang [15] proposed a refined RC-network ECM and support vector regression for SOH prediction. Although model-based methods are interpretable, their accuracy can degrade when simplified model structures fail to capture complex ageing mechanisms such as SEI growth, lithium plating, and loss of active material.

Data-driven methods have therefore been widely explored. Early studies used classical regressors, including support vector machines and random forests, with handcrafted features extracted from charge/discharge curves, such as discharge capacity, incremental capacity, and differential-voltage signatures.

More recently, deep learning has enabled end-to-end modeling from raw or minimally processed signals. Zhang *et al.* [8] applied LSTM networks to NASA battery data and reported improved cycle-life prediction compared with traditional regression baselines. CNN-based methods have also been used to learn degradation features directly from voltage and current trajectories. Recent studies have further explored multi-task learning strategies to jointly estimate SOH and remaining useful life (RUL); for example, Tomar *et al.* [16] proposed an AttentiveLSTM framework that exploits temporal attention mechanisms for battery prognostics. However, these supervised approaches generally assume that the training and deployment data originate from similar distributions. However, many supervised approaches rely on large labeled datasets of batteries cycled to failure, which are expensive to collect.

A related trend is label-efficient and cross-domain learning. Lu *et al.* [10] proposed a domain-adaptation framework for SOH estimation that reduces reliance on target-domain labels by aligning representations across battery types. Other approaches include meta-learning, where models are trained across multiple battery trajectories to enable rapid adaptation to new cells with limited labeled data. These studies highlight the need for models that generalize across operating conditions, datasets, and chemistries while minimizing additional labeling effort.

In parallel, test-time adaptation has emerged as a practical response to distribution shift. Early work by Bottou and Vapnik [17] suggested refining predictions using neighborhood information at inference time. Modern test-time adaptation methods are usually unsupervised, where no test labels are available, or semi-supervised, where a small number of test labels may be used. Test-Time Training (TTT) [18] adapts a model at inference by optimizing an auxiliary self-supervised objective on test inputs. TENT [13] adapts models by minimizing prediction entropy and updating normalization statistics. For sequential data, Garg *et al.* [19] applied test-time training to time-series forecasting and demonstrated gains under distribution changes. These ideas are well suited to battery management, where input distributions evolve with aging, temperature, and operating profile.

Recent advances in sequence modeling have also introduced state-space models (SSMs) as alternatives or complements to Transformers [20]. SSMs maintain a latent state that evolves over time and can provide long-range memory with favorable computational scaling. Structured SSMs such as S4 enable parallelizable training on long sequences, and Mamba introduces selective mechanisms that improve efficiency while remaining competitive with attention-based models [20]. These properties are relevant to battery degradation modeling, where trajectories span hundreds to thousands of cycles and SOH evolves gradually over time. In this work, Mamba is included as a baseline model.

Recent SOH-estimation studies have also explored optimisation-assisted and hybrid approaches. For example, Zhang *et al.* [21] combined northern goshawk optimisation with a CNN–BiLSTM–self-attention network to improve SOH-estimation accuracy. In addition, recent reviews have systematically discussed hybrid SOH-estimation methods and the progress, advantages, and limitations of deep-learning-based battery state estimation [22], [23]. These studies indicate that optimisation and hybrid modelling can improve predictive performance; however, most methods are trained offline and remain fixed after deployment. Their accuracy may therefore decrease when the target battery chemistry, ageing trajectory, operating profile, temperature, or measurement conditions differ from the training distribution.

III. Methodology

A. Data Preprocessing

To estimate SOH, we use partial constant-current charging profiles and discharge segments as model inputs. During cycling, the BMS records voltage $V(t)$, current $I(t)$, and temperature $T(t)$. For charge-based features, a partial charging feature vector $\mathbf{c}^{\psi}$ is extracted over a fixed voltage window:

$$\mathbf{c}^{\psi} = \left[\frac{C_0^{\psi}}{C_p}, \frac{C_1^{\psi}}{C_p}, \ldots, \frac{C_K^{\psi}}{C_p}\right], \quad \psi \in \{S, T\}, \tag{1}$$

where ψ denotes the source or target domain and C_p is the nominal capacity used for normalization. The capacity segment C_i^{ψ} is computed over the time samples for which the voltage lies in the i-th voltage interval:

$$C_i^{\psi} = \int_{\mathcal{T}_i} |I(t)|\, dt, \tag{2}$$

where

$$\mathcal{T}_i = \big\{t : V_{\min} + i\Delta V \le V(t) < V_{\min} + (i+1)\Delta V\big\}. \tag{3}$$

where $V_{\min}$ is the lower bound of the voltage window and ΔV is the voltage-bin width.

At each time step, the raw multivariate input is formed by concatenating the measured channels:

$$\mathbf{x}_t = [I_t, V_t, T_t] \in \mathbb{R}^3. \tag{4}$$

The features are therefore not manually prioritized through a weighted sum. Instead, the CNN denoising module and Transformer encoder learn the relative importance of current, voltage, and temperature from data.

To reduce the effect of imbalanced SOH distributions in the training data, the source samples are binned into n_{bin} intervals of equal SOH width. Let n_i^{orig} be the number of original samples in bin i, n_i^{trim} the number of discarded samples, and $n_i^{\text{rem}} = n_i^{\text{orig}} - n_i^{\text{trim}}$ the number of retained samples. Trimming is performed by solving:

$$\begin{aligned} &\underset{0 \le n_i^{\text{trim}} \le n_i^{\text{orig}}}{\text{minimize}} && \alpha_1 g_1 + \alpha_2 g_2 \\ &\text{where} \\ &g_1 = \left| \mu_3\left(\bigcup_{i=1}^{n_{\text{bin}}} \{y_{ij}^S\}_{j=1}^{n_i^{\text{rem}}}\right) - \mu_3^* \right|, \\ &g_2 = \frac{\sum_{i=1}^{n_{\text{bin}}} n_i^{\text{trim}}}{\sum_{i=1}^{n_{\text{bin}}} n_i^{\text{orig}}}. \end{aligned} \tag{5}$$

Here, $\mu_3(\cdot)$ is the skewness of the SOH distribution, and $\mu_3^* = 0$ is the target skewness used to encourage a symmetric retained distribution.

The weights $\alpha_1 = 0.2$ and $\alpha_2 = 2$ were selected on the validation split through a small grid search. The first term controls the skewness of the retained SOH distribution, whereas the second term penalizes excessive sample removal. The larger value assigned to α_2 reflects the practical requirement that distribution balancing should not discard too much degradation data. These weights are used only for constructing the balanced training set and are not trainable parameters of the SOH prediction model. Sensitivity analysis to these weights is reported in Section VI-D.

The optimization is subject to:

$$\begin{aligned} &\max_i(n_i^{\text{rem}}) - \min_i(n_i^{\text{rem}}) \le \varepsilon \sum_{i=1}^{n_{\text{bin}}} n_i^{\text{orig}}, \\ &\sum_{i=2}^{n_{\text{bin}}-1} \text{sgn}(n_{i+1}^{\text{rem}} - n_i^{\text{rem}})\text{sgn}(n_i^{\text{rem}} - n_{i-1}^{\text{rem}}) = 1. \end{aligned} \tag{6}$$

where $\varepsilon = 0.045$ limits inter-bin imbalance and the second condition encourages a unimodal retained distribution.

B. Model Overview

The proposed architecture, shown in Figure 1, contains three components: a CNN-based denoising module, a Transformer encoder, and an SOH regression head. During supervised training, all components are optimized using labeled source-domain data. During test-time adaptation, only the denoising module is updated using an unlabeled reconstruction objective. The Transformer encoder and SOH regression head remain frozen.

C. CNN-Based Feature Denoising Module

The denoising module aims to suppress variations caused by sensor noise, temperature fluctuation, operating-condition differences, and manufacturing variability while preserving degradation-related information. Let the input feature vector at time t be denoted by $\mathbf{x}_t \in \mathbb{R}^d$. The denoising module $D_{\theta_d}(\cdot)$ maps the raw input to a denoised representation:

$$\tilde{\mathbf{x}}_t = D_{\theta_d}(\mathbf{x}_t). \tag{7}$$

In this work, $D_{\theta_d}(\cdot)$ is implemented as a one-dimensional CNN because convolutional layers are effective at extracting local temporal patterns from structured battery signals. The CNN consists of convolutional layers, nonlinear activation functions, and a projection layer. The output sequence is then passed to the Transformer encoder.

D. Transformer SOH Predictor

The Transformer encoder follows the standard architecture introduced by Vaswani *et al.* [24]. The positional encoding, scaled dot-product attention, multi-head attention, feedforward network, residual connections, and layer-normalization operations are included below for completeness and are not claimed as methodological contributions of this work. After denoising, the sequence of feature vectors is represented as:

$$\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_T]^{\top} \in \mathbb{R}^{T \times d}, \tag{8}$$

where T is the sequence length and d is the feature dimension.

Because the Transformer is non-recurrent, positional encodings $\mathbf{P} \in \mathbb{R}^{T \times d}$ are added to the input embeddings [24]:

$$\mathbf{Z}^{(0)} = \tilde{\mathbf{X}} + \mathbf{P}. \tag{9}$$

The Transformer encoder contains L layers. In each layer, the input $\mathbf{Z}^{(l-1)}$ is projected into queries, keys, and values:

$$\mathbf{Q} = \mathbf{Z}^{(l-1)}\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{Z}^{(l-1)}\mathbf{W}_K, \quad \mathbf{V} = \mathbf{Z}^{(l-1)}\mathbf{W}_V. \tag{10}$$

The scaled dot-product attention is:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{d_k}}\right)\mathbf{V}. \tag{11}$$

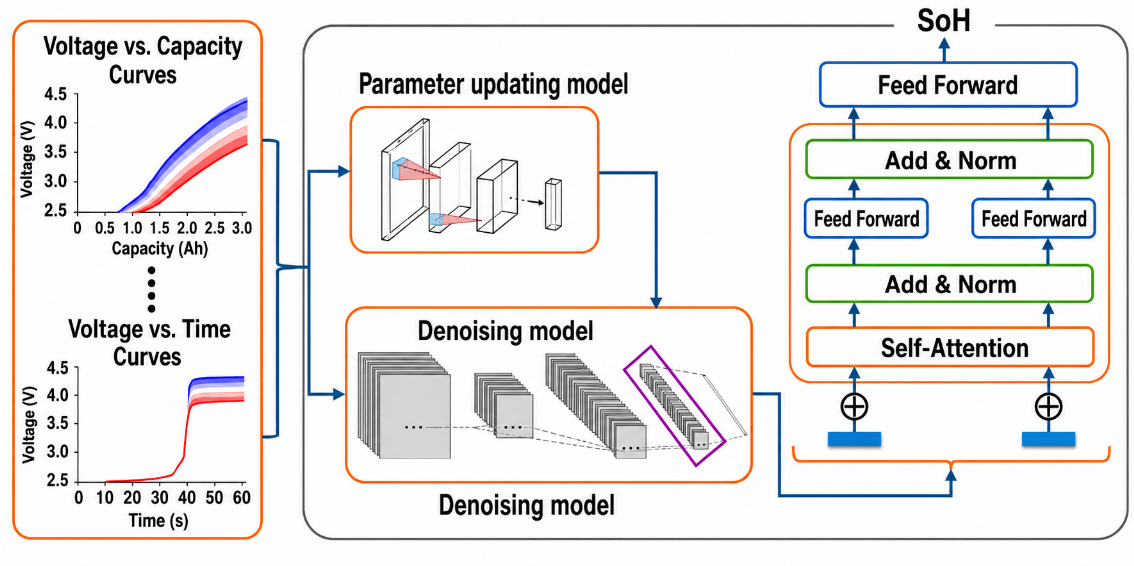


Figure 1. Overview of the proposed TTL–Transformer. The CNN denoising front-end processes voltage, current, and temperature sequences. During test-time learning, only the denoising module is updated using a self-supervised reconstruction loss, while the Transformer encoder and SOH regression head remain frozen.

For multi-head attention, the outputs from H heads are concatenated and linearly transformed:

$$\mathrm{MultiHead}(\mathbf{Z}) = \mathrm{Concat}(\mathrm{head}_1, \dots, \mathrm{head}_H)\mathbf{W}_O. \quad (12)$$

Each encoder layer also includes a position-wise feedforward network:

$$\mathrm{FFN}(\mathbf{z}) = \max(0, \mathbf{z}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2. \quad (13)$$

Residual connections and layer normalization are applied as:

$$\mathbf{Z}'^{(l)} = \mathrm{LayerNorm}\left(\mathbf{Z}^{(l-1)} + \mathrm{MultiHead}(\mathbf{Z}^{(l-1)})\right), \quad (14)$$

$$\mathbf{Z}^{(l)} = \mathrm{LayerNorm}\left(\mathbf{Z}'^{(l)} + \mathrm{FFN}(\mathbf{Z}'^{(l)})\right). \quad (15)$$

In this study, these standard Transformer operations are used to learn temporal SOH-related representations.

After the final encoder layer, a pooling operation and regression head produce the SOH estimate:

$$\widehat{\mathrm{SOH}} = \mathbf{W}_{\mathrm{reg}}\mathrm{Pool}(\mathbf{Z}^{(L)}) + b_{\mathrm{reg}}. \quad (16)$$

E. Test-Time Learning Adaptation

The full SOH predictor can be written as:

$$\hat{y} = g_{\theta_r}\left(T_{\theta_T}\left(D_{\theta_d}(\mathbf{x})\right)\right), \quad (17)$$

where $D_{\theta_d}(\cdot)$ is the CNN denoising module, $T_{\theta_T}(\cdot)$ is the Transformer encoder, and $g_{\theta_r}(\cdot)$ is the regression head. During supervised pretraining, all modules are optimized with the SOH regression loss:

$$\mathcal{L}_{\mathrm{SOH}} = \|\hat{y} - y\|_2^2. \quad (18)$$

At test time, the target SOH label y is unavailable. Following the general test-time-training paradigm [12], adaptation is therefore performed using an auxiliary self-supervised objective constructed from each unlabeled target input. The corrupt-and-reconstruct formulation is related to denoising autoencoders [25] and reconstruction-based test-time training with masked autoencoders [14]. In the proposed method, this objective is applied specifically to multivariate battery voltage, current, and temperature sequences and is used to update only the dedicated CNN denoising front-end. Given a target input sequence $\mathbf{x}^T$, a corrupted version is created:

$$\bar{\mathbf{x}}^T = \mathcal{A}(\mathbf{x}^T), \quad (19)$$

where $\mathcal{A}(\cdot)$ denotes random masking and additive Gaussian noise. The denoising module is updated to reconstruct the clean input:

$$\mathcal{L}_{\mathrm{TTL}} = \left\|D_{\theta_d}(\bar{\mathbf{x}}^T) - \mathbf{x}^T\right\|_2^2. \quad (20)$$

The reconstruction objective does not explicitly minimize the distribution discrepancy between the source and target battery chemistries. Gaussian noise and masking are used only to construct a self-supervised adaptation task and are not intended to simulate chemistry-induced differences in voltage curves, thermal behaviour, or current profiles. During adaptation, the CNN denoiser learns the local temporal structure of the unlabeled target measurements by recovering the original voltage, current, and temperature signals from their corrupted versions. Because the Transformer encoder and SOH regression head remain frozen, the adaptation recalibrates the target inputs while preserving the source-trained SOH mapping. Therefore, the proposed TTL mechanism should be interpreted as target-specific input adaptation rather than explicit source–target distribution alignment, and it does not theoretically guarantee the removal of all chemistry-related distribution shifts.

The test-time update is performed for K_{TTL} gradient steps:

$$\theta_d^{(k+1)} = \theta_d^{(k)} - \eta_{\mathrm{TTL}}\nabla_{\theta_d}\mathcal{L}_{\mathrm{TTL}}, \quad k = 0, \dots, K_{\mathrm{TTL}} - 1. \quad (21)$$

Only θ_d is updated. The Transformer parameters θ_T and regression-head parameters θ_r remain fixed:

$$\theta_T^{(k+1)} = \theta_T^{(k)}, \qquad \theta_r^{(k+1)} = \theta_r^{(k)}. \quad (22)$$

After adaptation, prediction is performed as:

$$\hat{y}^T = g_{\theta_r}\left(T_{\theta_T}\left(D_{\theta_d^{(K_{\mathrm{TTL}})}}(\mathbf{x}^T)\right)\right). \quad (23)$$

This design limits test-time updates to the input-processing front-end, making adaptation more stable and computationally cheaper than updating the entire model. Reconstruction-based self-supervised objectives and partial-network adaptation have been investigated in prior test-time learning and test-time adaptation studies [14], [26]. However, the proposed approach is specifically designed for battery SOH estimation after deployment. Rather than updating the complete prediction model, the reconstruction objective is applied only to the CNN denoising front-end using unlabelled target voltage, current, and temperature sequences. The Transformer encoder and SOH regression head remain frozen, preserving the supervised SOH mapping learned from the source data. This selective denoiser-only update allows the input-conditioning stage to adapt to new battery and operating conditions without requiring target SOH labels or full-model retraining.

IV. Datasets

To evaluate the proposed TTL–Transformer under diverse conditions, we curate a hybrid dataset suite comprising public datasets, lab-collected data, and physics-based simulated data.

A. Public Datasets

Three open-access lithium-ion battery degradation datasets are used: the Oxford Degradation dataset [27], the CALCE battery repository [28], and the NASA Ames prognostics battery dataset [29]. Each dataset provides cycle-resolved voltage, current, and temperature measurements and supports cross-dataset generalization analysis. Table I summarizes the public datasets.

Table I
Summary of public datasets used in this work [27]–[29].

Dataset	Cell type
Oxford Degradation [27]	Pouch cells, 0.74 Ah
CALCE [28]	18650 lithium-ion cells, 1.1 Ah
NASA Ames [29]	18650 lithium-ion cells, 2 Ah

B. Lab-Collected Dataset

Degradation data were collected from four commercial lithium-ion cell types: BIXELL IFR21700 (LFP1), SINC IFR21700-3Ah (LFP2), BIXELL INR21700 (NMC), and Samsung SDI ICR18650-26A (LCO). Ten physical cells of each type were tested, giving 40 cells in total. Their main specifications are summarised in Table II.

Public model-specific datasheets were unavailable for the two BIXELL cells. Their nominal ratings were therefore obtained from the wrapper markings shown in Figure 2(d). The IFR21700 cell was classified as LFP based on its IFR designation, nominal voltage, and discharge profile. The INR21700 cell was conservatively classified as nickel-based lithium-ion and retained under the nominal dataset label “NMC” [30]. The SINC IFR21700-3Ah specifications were obtained from the manufacturer’s documentation [31]. The Samsung ICR18650-26A specifications and LCO classification were supported by model-specific documentation [32], [33]. Experimentally measured reference capacities, rather than wrapper ratings, were used to calculate SOH.

The cells were cycled from 100% to 80% SOH under 1C–2C charging conditions and ambient temperatures from $-10°$C to $40°$C. Voltage, current, and temperature were sampled at 1 Hz. Data were collected using a programmable battery cycler, temperature-control unit, and custom multi-cell test bench [34]. The measurements include practical non-idealities such as sensor noise, temperature gradients, and channel-specific offsets.

Figure 2 shows the experimental equipment, the four labelled cell types, and the chamber configurations used for sub-zero and elevated-temperature testing.

C. Simulated Dataset via PyBaMM

Synthetic degradation data were generated using PyBaMM version 25.10.1 [35]. The Doyle–Fuller–Newman model [36] was configured using the `OKane2022` parameter set, representing a 5.0-Ah graphite/NMC811 LG M50-type cell with voltage limits of 2.5 and 4.2 V.

The model included SEI growth, SEI-induced porosity change, reversible lithium plating, particle cracking, SEI growth on cracks, and stress-driven loss of active material. Simulations were solved using the IDAKLU solver. Ten virtual-cell degradation trajectories were generated under different combinations of ambient temperature (-10, 0, 10, 25, 40, and $45°$C) and ageing rate (0.5C, 1C, and 2C). Each trajectory was simulated for up to 500 cycles or until its capacity decreased to 80% of its beginning-of-life value.

A reference performance test was performed every 25 ageing cycles using a C/10 discharge at $25°$C from 4.2 to 2.5 V. The deliverable capacity at checkpoint k was calculated as:

$$Q_{\mathrm{sim}}^{(k)} = \frac{1}{3600} \int_{t_{\mathrm{start}}^{(k)}}^{t_{\mathrm{cut}}^{(k)}} |I(t)| \, dt. \tag{24}$$

The corresponding SOH was defined as:

$$\mathrm{SOH}_{\mathrm{sim}}^{(k)} = \frac{Q_{\mathrm{sim}}^{(k)}}{Q_{\mathrm{sim}}^{(0)}}, \tag{25}$$

where $Q_{\mathrm{sim}}^{(0)}$ is the beginning-of-life reference capacity.

Voltage, current, and temperature were sampled at 1 Hz and processed using the same procedure as the measured data. The 10 virtual cells were divided into seven training cells, one validation cell, and two test cells. All windows from the same virtual cell were retained within a single partition to prevent data leakage.

D. Discharge-Segment Construction for SOH Supervision

To obtain SOH labels without dedicated full-capacity calibration cycles, the voltage, current, and temperature signals are time-aligned and resampled at 1 Hz. Contiguous discharge segments are then extracted from the filtered current signal. Active discharge is defined as

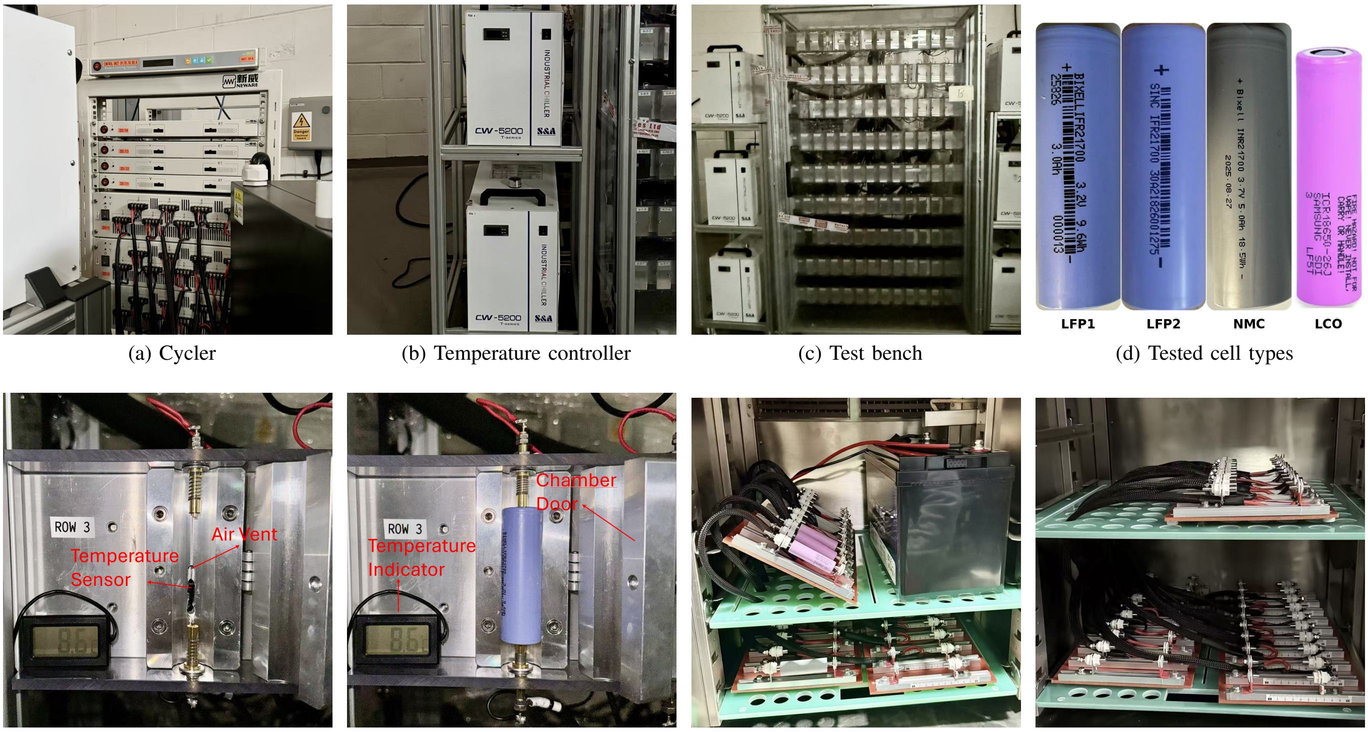
$$i[n] > I_{\mathrm{thr}}, \qquad I_{\mathrm{thr}} = 0.05C, \tag{26}$$

where positive current denotes discharge and C is the nominal cell capacity. A segment begins when the current remains above the threshold for at least 60 s and ends when it remains at or below the threshold for the same duration. Segments shorter than 60 s are discarded.

Table II
SPECIFICATIONS AND CHEMISTRY CLASSIFICATIONS OF THE LABORATORY-TESTED CELLS.

ID	Brand	Model	Format	Chemistry	Voltage	Capacity	Cells
LFP1	BIXELL [30]	IFR21700*	21700	$LiFePO_4$*	3.2 V*	3.0 Ah*	10
LFP2	SINC [31]	IFR21700-3Ah	21700	$LiFePO_4$	3.2 V	3.0 Ah	10
NMC	BIXELL [30]	INR21700*	21700	Nickel-based Li-ion*	3.7 V*	5.0 Ah*	10
LCO	Samsung SDI [32]	ICR18650-26A	18650	$LiCoO_2$†	3.7 V	2.6 Ah	10

*Public model-specific datasheets were unavailable for the BIXELL cells. Their model names and nominal ratings were transcribed from the cell wrappers in Figure 2(d); the NMC classification is nominal rather than independently verified. †The $LiCoO_2$/carbon classification of the Samsung ICR18650-26A is supported by [33].



(a) Cycler
(b) Temperature controller
(c) Test bench
(d) Tested cell types
(e) Thermal chamber
(f) Cell installed
(g) Chamber interior
(h) Battery holder

Figure 2. Experimental infrastructure and cell types used for laboratory data collection.

For each retained segment k, the discharged charge, available capacity, and SOH label are calculated as

$$\begin{aligned} Q^{(k)} &= \frac{1}{3600} \sum_{n=n_1^{(k)}}^{n_2^{(k)}} i[n]\Delta t, \\ C_{\text{cap}}^{(k)} &= \frac{mQ^{(k)}}{E_0[n_1^{(k)}] - E_0[n_2^{(k)}]}, \\ \text{SOH}^{(k)} &= \frac{C_{\text{cap}}^{(k)}}{C_n}, \end{aligned} \tag{27}$$

where $\Delta t = 1$ s, $E_0[n]$ is the observer-estimated no-load voltage state, m is the local linearisation slope relating the internal voltage state to SOC, and C_n is the rated cell capacity.

Each segment is divided into windows containing 60 samples of voltage, current, and temperature. At 1 Hz, each window represents 60 s of operation. A stride of 30 samples provides a 50% overlap, and every window inherits the SOH label of its corresponding segment. The resulting dataset is

$$\mathcal{D} = \left\{ \left(\mathbf{x}^{(k,j)}, \text{SOH}^{(k)} \right) \right\}, \tag{28}$$

where $\mathbf{x}^{(k,j)} \in \mathbb{R}^{60\times 3}$ is the jth voltage–current–temperature window from segment k. The same procedure is applied to the public, laboratory, and simulated datasets.

E. Scope of Evaluated SOH Range

This study evaluates SOH estimation within the 80–100% SOH range. This range is relevant to first-life battery operation and second-life screening, where 80% capacity retention is commonly used as a practical threshold for battery retirement from its original application. The reported results should therefore be interpreted within this evaluated SOH window.

The proposed denoiser-only test-time adaptation mechanism is not inherently restricted to the 80–100% range. However, extending the method to lower-SOH regimes requires representative end-of-life degradation data because the dominant degradation mechanisms, voltage trajectories, and measurement sensitivity may change as cells approach end of life. Evaluation across a broader SOH range is therefore identified as future work.

V. Experimental Setup

The objective is to develop an SOH model that generalises across LCO, NMC, and LFP cells with different formats and cycling conditions. The TTL–Transformer is evaluated under two protocols: chemistry-specific training and chemistry-agnostic pooled multi-chemistry training.

A. Data Splitting and Leakage Control

All data were split at the cell level before window extraction to prevent samples from the same cell appearing in different partitions. Each laboratory group contained 10 physical cells, divided into seven for training, one for validation, and two for testing. The same 70%/10%/20% split was applied to the 10 simulated cells. All windows from each physical or virtual cell were retained within one partition.

Normalization statistics were calculated using training data only:

$$x' = \frac{x - \mu_{\text{train}}}{\sigma_{\text{train}}}. \tag{29}$$

Chemistry-specific statistics were used for chemistry-aware models, whereas global statistics from the pooled training set were used for chemistry-agnostic models and subsequently applied to their validation and test data.

B. Evaluation Protocols

Let $\mathcal{C} \in \{\text{LCO}, \text{NMC}, \text{LFP}\}$ denote the set of battery chemistries. We define two cell-disjoint evaluation protocols.

1) P1: In-Chemistry Generalization: For each chemistry, the cells are split into training, validation, and test sets at the cell level. A separate model is trained for each chemistry using only training data from that chemistry and evaluated on unseen cells of the same chemistry. This protocol measures cell-level generalization without chemistry shift.

2) P2: Cross-Chemistry Generalization: A single unified model is trained using the union of all available training data across chemistries:

$$\mathcal{D}_{\text{train}} = \bigcup_{\mathcal{C} \in \{\text{LCO}, \text{NMC}, \text{LFP}\}} \mathcal{D}_{\text{train}}^{(\mathcal{C})}. \tag{30}$$

The trained model is then evaluated separately on held-out test cells from each chemistry:

$$\mathcal{D}_{\text{test}}^{(\mathcal{C})} = \mathcal{D}_{\text{test}}^{(\mathcal{C})}. \tag{31}$$

This protocol measures whether one model can handle chemistry-dependent shifts while maintaining accurate SOH prediction.

C. Baseline Models

The proposed method is compared with Linear Ridge (LR), CNN, LSTM, Transformer, and Mamba baselines. LR uses L2-regularised linear regression on engineered features [37]; CNN uses a convolutional encoder and regression head [38]; LSTM is trained on charge/discharge windows [39]; and Mamba is a selective state-space sequence model [20]. The Transformer baseline uses the same encoder and regression head as the proposed model but without test-time learning [24]. All baselines use identical data splits, preprocessing, and input representations.

D. Implementation Configuration and Model Scale

All models use voltage, current, and temperature sequences of 60 time steps. The Transformer-based models share the same architecture to ensure a fair comparison. The main hyperparameters are listed in Table III.

Table III
Main experimental hyperparameters.

Item	Setting	Item	Setting
Input	V, I, T	Sequence length	60
Optimizer	Adam	Learning rate	1×10^{-4}
Batch size	64	Weight decay	1×10^{-5}
Early stopping	20 epochs	Transformer layers	4
Attention heads	8	Embedding dimension	128
FFN dimension	256	Dropout	0.10
Denoiser	1-D CNN	TTL learning rate	1×10^{-5}
TTL steps	5	TTL corruption	Noise + masking
Noise level	0.03	Masking ratio	0.10

Table IV compares the total and updated parameters. The denoising front-end adds only 11,779 parameters, and TTL updates only this module while keeping the Transformer and regression head fixed.

Table IV
Model parameter comparison.

Model	Total	Updated	Ratio
Transformer	530,561	0	0%
Denoiser + Trans.	542,340	0	0%
TTL–Trans.	542,340	11,779	2.17%
Full-model TTL	542,340	542,340	100%

The proposed strategy therefore limits adaptation to 2.17% of the complete model, reducing the computational cost and the risk of disturbing the learned SOH mapping.

E. Computational Cost and Deployment Overhead

The model was trained and adapted on a laptop workstation, while forward SOH inference was deployed on an NXP FRDM–MCXN947 development board using its on-board neural accelerator. The hardware environments are summarised in Table V, and the embedded setup is shown in Figure 3.

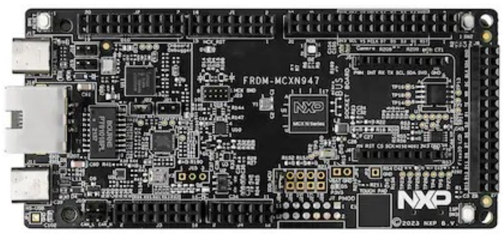


Figure 3. NXP FRDM–MCXN947 development board with a 150-MHz processor and 2-MB flash memory.

TTL introduces gradient-based computation, but only the CNN denoiser is updated; the Transformer encoder and regression head remain frozen. Adaptation is performed periodically after sufficient target data have been collected, rather than at every sampling instant. The adapted denoiser is then reused for subsequent predictions.

Runtime was averaged over 100 runs after 100 warm-up runs. Embedded inference was measured on the NXP board

Table V
TRAINING AND DEPLOYMENT HARDWARE.

Item	**Configuration**
CPU	Intel Core i7 laptop CPU
GPU	NVIDIA RTX 4060
Memory	32 GB RAM
Framework	PyTorch
Embedded board	NXP FRDM–MCXN947
Embedded accelerator	On-board neural engine
Embedded task	Forward SOH inference

using one 60×3 input window. Adaptation time was measured on the laptop CPU using $K_{\text{TTL}} = 5$ update steps.

Table VI
INFERENCE AND ADAPTATION OVERHEAD. INFERENCE WAS MEASURED ON THE NXP BOARD AND ADAPTATION ON THE LAPTOP CPU.

Method	**Updated parameters**	**Adapted part**	**Inference (ms)**	**Adaptation (ms)**
Transformer	0	None	24.8 ± 0.9	–
Denoiser + Trans.	0	None	27.1 ± 1.0	–
TTL–Trans.	11,779	Denoiser	27.2 ± 1.1	18.6 ± 1.2
Full TTL	542,340	Full model	–	74.9 ± 4.3

The proposed model required approximately 27.2 ms for embedded inference. Its denoiser-only adaptation required 18.6 ms, compared with 74.9 ms for full-model adaptation, representing a 75.2% reduction. Thus, updating only 11,779 parameters, or 2.17% of the model, provides lower adaptation overhead while preserving practical embedded inference.

In the current implementation, forward inference is performed on the NXP board, whereas gradient-based adaptation is performed on the NXP only when sufficent data are collected.

F. Evaluation Metrics

Performance is evaluated using MAE, RMSE, R^2, and tolerance accuracy. For measured and predicted SOH values y_i and $\hat{y}_i$ over N samples:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i| \qquad \text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}, \quad \bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i \quad \text{Acc}_{\pm\delta} = \frac{100}{N}\sum_{i=1}^{N}\mathbb{I}(|\hat{y}_i - y_i| \le \delta)$$

Here, $\delta = 3$ and 5 SOH percentage points are used for $\pm 3\%$ and $\pm 5\%$ accuracy, respectively. Metrics are reported as mean $\pm$ standard deviation across test cells.

VI. RESULTS AND ANALYSIS

The TTL–Transformer is evaluated under chemistry-specific and pooled multi-chemistry deployment scenarios.

A. Scenario 1: In-Chemistry Generalization

Under P1, a separate model is trained and tested on unseen cells of each chemistry. Table VII shows that the proposed model achieves an RMSE below 1% for all four cell groups.

Table VIII compares the proposed model with the baseline methods under the same protocol.

Table VII
TTL–TRANSFORMER PERFORMANCE UNDER P1.

Chemistry	MAE (%)	RMSE (%)	R^2	$\pm 3\%$	$\pm 5\%$
LCO	0.41	0.56	0.995	99.2	99.8
NMC	0.44	0.60	0.994	98.9	99.6
LFP1	0.49	0.66	0.991	96.3	99.3
LFP2	0.52	0.73	0.992	98.1	99.4

Table VIII
IN-CHEMISTRY RMSE COMPARISON UNDER P1.

Model	LCO	NMC	LFP1	LFP2
CNN	0.92	0.98	1.11	1.13
LSTM	0.84	0.90	1.02	1.05
Transformer	0.71	0.76	0.88	0.89
Mamba	0.68	0.72	0.83	0.85
TTL–Transformer	**0.56**	**0.60**	**0.66**	**0.73**

B. Scenario 2: Pooled Multi-Chemistry Generalization

Under P2, one model is trained using pooled multi-chemistry data and evaluated separately on held-out cells from each group. Table IX shows that the unified model maintains an RMSE between 0.71% and 0.84%.

Compared with the plain Transformer, the TTL–Transformer reduces RMSE by 29.6–38.5% across the four cell groups. This improvement combines the effects of the CNN denoiser and its unlabeled test-time adaptation, which are examined separately below.

C. Ablation Study

The ablation study compares the plain Transformer, the non-adapted Denoiser + Transformer, denoiser-only TTL, and full model TTL under P2.

The plain Transformer obtains an average RMSE of 1.20%. Adding the denoiser reduces it to 1.05%, a 12.1% improvement. Denoiser-only TTL further reduces the RMSE to 0.79%, improving upon the non-adapted model by 25.4% and the plain Transformer by 34.4%. Full-model TTL performs less effectively, suggesting that unrestricted adaptation may disturb the learned SOH mapping. Restricting updates to the denoiser therefore provides a better balance between adaptation and prediction stability.

D. Sensitivity to Trimming Weights

Table XII evaluates the trimming weights in Eq. (5). The validation RMSE remains between 0.79% and 0.82%, indicating limited sensitivity near the selected values. The setting $\alpha_1 = 0.2$ and $\alpha_2 = 2.0$ gives the lowest RMSE while retaining 90.8% of the training samples. Smaller α_2 values may discard too much data, whereas larger values may retain an imbalanced distribution.

E. Sensitivity to Test-Time Adaptation Steps

Table XIII evaluates the number of adaptation steps. Here, $K_{\text{TTL}} = 0$ represents the non-adapted Denoiser + Transformer.

Performance improves rapidly over the first three steps and reaches its best level at five steps. Additional updates provide no meaningful gain and may increase latency or overfit the reconstruction objective. Therefore, $K_{\text{TTL}} = 5$ is used in the main experiments.

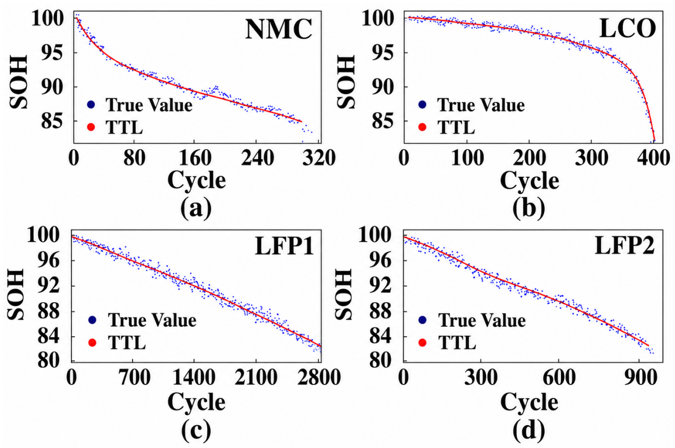


Figure 4. In-chemistry parity plots under P1.

Table IX
TTL–TRANSFORMER PERFORMANCE UNDER P2.

Chemistry	MAE (%)	RMSE (%)	R^2	±3%	±5%
LCO	0.53	0.71	0.993	98.6	99.5
NMC	0.56	0.76	0.992	98.2	99.3
LFP1	0.61	0.83	0.989	97.4	99.0
LFP2	0.62	0.84	0.990	97.6	99.2

F. Sensitivity to Self-Supervised Corruption

The reconstruction objective combines Gaussian noise with random feature masking. Table XIV evaluates the Gaussian noise level while retaining the selected masking ratio.

Moderate corruption provides a useful self-supervised adaptation signal without excessively distorting degradation-related patterns. The best performance is obtained at 0.03, whereas stronger corruption reduces SOH accuracy.

G. Sensitivity to Input Features

Table XV evaluates the effect of removing individual sensor channels under P2.

Table XV
SENSITIVITY TO INPUT FEATURES UNDER P2.

Input setting	**LCO**	**NMC**	**LFP1**	**LFP2**
V + I + T	**0.71**	**0.76**	**0.83**	**0.84**
Without voltage	1.46	1.58	1.82	1.87
Without current	1.12	1.20	1.39	1.42
Without temperature	0.78	0.83	0.92	0.94
Voltage + current only	0.76	0.81	0.89	0.91

Removing voltage causes the largest performance loss, followed by removing current. Temperature has a smaller effect but improves robustness to changing thermal conditions. The voltage–current-only configuration remains competitive, although all three channels give the best overall performance.

VII. CONCLUSION & FUTURE WORK

This paper proposes a denoiser-only test-time learning framework for lithium-ion battery SOH estimation under heterogeneous deployment conditions. The CNN denoiser adapts using unlabelled target data, while the Transformer encoder and SOH regression head remain fixed, preserving the learned SOH mapping without target labels or full-model retraining.

Under pooled multi-chemistry evaluation, RMSE decreases from 1.20% for the plain Transformer to 1.05% with the denoiser and 0.79% with TTL. This corresponds to a 12.1% denoiser gain, a further 25.4% TTL gain, and a total 34.4% reduction. Matched-chemistry and heterogeneous multi-chemistry RMSEs are 0.56–0.73% and 0.71–0.84%, respectively. Only 11,779 parameters (2.17% of the model) are updated during TTL.

TTL provides lightweight adaptation but requires gradient updates and reliable measurements. Future work will investigate uncertainty-aware and physics-informed objectives, adaptive update scheduling, embedded runtime, and robustness to degraded sensors.

Table X
RMSE COMPARISON UNDER P2.

Model	LCO	NMC	LFP1	LFP2
CNN	1.41	1.52	1.87	1.89
LSTM	1.28	1.33	1.65	1.68
Transformer	1.02	1.08	1.35	1.34
Mamba	0.94	1.01	1.21	1.22
TTL–Transformer	**0.71**	**0.76**	**0.83**	**0.84**

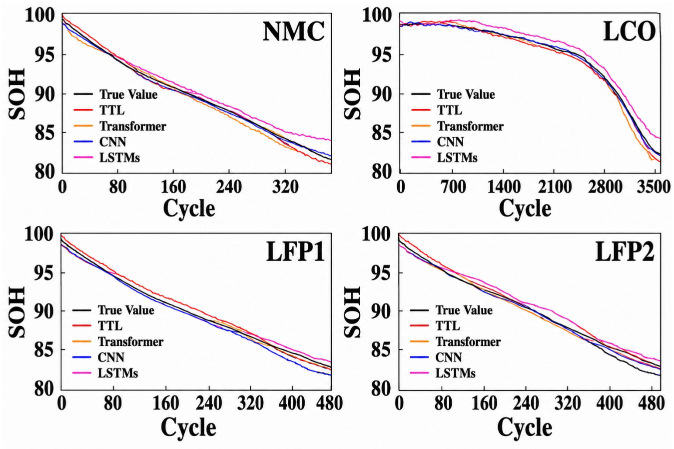


Figure 5. TTL-adapted parity plots under P2.

REFERENCES

[1] R. Xiong, L. Li, and J. Tian, “Towards a smarter battery management system: A critical review on battery state of health monitoring methods,” *Journal of Power Sources*, vol. 405, pp. 18–29, 2018.

[2] M. Berecibar, I. Gandiaga, I. Villarreal, N. Omar, J. Van Mierlo, and P. Van den Bossche, “Critical review of state of health estimation methods of li-ion batteries for real applications,” *Renewable and Sustainable Energy Reviews*, vol. 56, pp. 572–587, 2016.

[3] D. Roman, S. Saxena, V. Robu, M. Pecht, and D. Flynn, “Machine learning pipeline for battery state-of-health estimation,” *Nature Machine Intelligence*, vol. 3, pp. 447–456, 2021.

[4] A. Farmann, W. Waag, A. Marongiu, and D. U. Sauer, “Critical review of on-board capacity estimation techniques for lithium-ion batteries in electric and hybrid electric vehicles,” *Journal of Power Sources*, vol. 281, pp. 114–130, 2015.

[5] G. L. Plett, “Extended kalman filtering for battery management systems of lipb-based hev battery packs: Part 1. background,” *Journal of Power sources*, vol. 134, no. 2, pp. 252–261, 2004.

[6] P. A. Topan, M. N. Ramadan, G. Fathoni, A. I. Cahyadi, and O. Wahyunggoro, “State of charge (soc) and state of health (soh) estimation on lithium polymer battery via kalman filter,” in *2016 2nd International Conference on Science and Technology-Computer (ICST)*. IEEE, 2016, pp. 93–96.

[7] S. Bockrath, V. Lorentz, and M. Pruckner, “State of health estimation of lithium-ion batteries with a temporal convolutional neural network using partial load profiles,” *Applied Energy*, vol. 329, p. 120307, 2023.

Table XI
ABLATION STUDY UNDER P2.

Model variant	**LCO**	**NMC**	**LFP1**	**LFP2**
Transformer	1.02	1.08	1.35	1.34
Denoiser + Trans.	0.91	0.96	1.16	1.18
TTL–Transformer	**0.71**	**0.76**	**0.83**	**0.84**
Full-model TTL	0.78	0.82	0.96	1.01

Table XII
SENSITIVITY TO TRIMMING WEIGHTS.

α_1	α_2	Retained samples (%)	Val. RMSE (%)
0.1	1.0	86.4	0.82
0.1	2.0	91.7	0.81
0.2	2.0	90.8	**0.79**
0.3	2.0	88.6	0.80
0.2	3.0	94.2	0.82

Table XIII
SENSITIVITY TO THE NUMBER OF TTL STEPS.

K_{TTL}	LCO	NMC	LFP1	LFP2
0	0.91	0.96	1.16	1.18
1	0.82	0.88	1.02	1.05
3	0.74	0.79	0.88	0.90
5	**0.71**	**0.76**	**0.83**	**0.84**
10	0.72	0.77	0.84	0.86

Table XIV
SENSITIVITY TO THE TTL CORRUPTION LEVEL.

Noise level	LCO	NMC	LFP1	LFP2
0.00	0.80	0.85	0.97	0.99
0.01	0.75	0.80	0.89	0.91
0.03	**0.71**	**0.76**	**0.83**	**0.84**
0.05	0.73	0.78	0.86	0.88
0.10	0.82	0.87	1.01	1.04

[8] Y. Zhang, R. Xiong, H. He, and M. G. Pecht, "Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 5695–5705, 2018.

[9] S. Zhang, Z. Cui, Q. Zhang, and Y.-Y. Li, "A parallel self-attention transformer for predicting the remaining useful life of lithium-ion batteries," *IEEE Journal of Emerging and Selected Topics in Industrial Electronics*, vol. 6, no. 4, pp. 1808–1818, 2025.

[10] J. Lu, R. Xiong, J. Tian, C. Wang, and F. Sun, "Deep learning to estimate lithium-ion battery state of health without additional degradation experiments," *Nature Communications*, vol. 14, no. 1, p. 2760, 2023.

[11] Y. Feng, G. Hu, X. Li, and Z. Zhang, "Adapting amidst degradation: Cross-domain li-ion battery health estimation via physics-guided test-time training," *arXiv preprint arXiv:2402.00068*, 2024.

[12] Y. Sun, X. Wang, Z. Liu, J. Miller, A. A. Efros, and M. Hardt, "Test-time training with self-supervision for generalization under distribution shifts," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 119, 2020, pp. 9229–9248.

[13] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell, "Tent: Fully test-time adaptation by entropy minimization," *arXiv preprint arXiv:2006.10726*, 2020.

[14] Y. Gandelsman, Y. Sun, X. Chen, and A. A. Efros, "Test-time training with masked autoencoders," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 29 374–29 385.

[15] J. Yang, C. Wang, Z. Wang, R. Niu, A.-L. Jin, and L. Shi, "A robust lithium-ion battery soh estimation method using refined rc-network ecm and svr," in *2024 IEEE PES 16th Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, 2024, pp. 1–5.

[16] A. Tomar, M. Gupta, J. Mittal, and A. Arya, "Prediction of soh and rul for li-ion batteries in ev based on attentivelstm multi-task model," *IEEE Journal of Emerging and Selected Topics in Industrial Electronics*, 2025, early Access.

[17] V. Vapnik and L. Bottou, "Local algorithms for pattern recognition and dependencies estimation," *Neural Computation*, vol. 5, no. 6, pp. 893–909, 1993.

[18] Y. Sun, X. Li, K. Dalal, J. Xu, A. Vikram, G. Zhang, Y. Dubois, X. Chen, X. Wang, S. Koyejo *et al.*, "Learning to (learn at test time): Rnns with expressive hidden states," *arXiv preprint arXiv:2407.04620*, 2024.

[19] R. Garg, S. Barpanda *et al.*, "Machine learning algorithms for time series analysis and forecasting," *arXiv preprint arXiv:2211.14387*, 2022.

[20] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *arXiv preprint arXiv:2312.00752*, 2023.

[21] L. Zhang, D. Liu, S. Wang, Y. Zhou, and C. Fernandez, "An innovative northern goshawk optimization-hybrid neural network algorithm for highly accurate state of health estimation of lithium-ion batteries," *Ionics*, vol. 32, no. 1, pp. 403–415, 2026.

[22] S. Wang, B. Hu, Q. Zhang, C. Fernandez, and J. M. Guerrero, "Hybrid models for predicting the state of health of lithium-ion batteries: A comprehensive review," *Journal of Energy Storage*, vol. 157, p. 121600, 2026.

[23] S. Wang, M. Zhang, L. Zhou, C. Fernandez, and F. Blaabjerg, "Critical review of battery health state estimation with deep learning methods," *Journal of Power Sources*, vol. 666, p. 239106, 2026.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[25] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*, 2008, pp. 1096–1103.

[26] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell, "Tent: Fully test-time adaptation by entropy minimization," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=uXl3bZLkr3c

[27] D. Howey and C. Birkl, "Oxford battery degradation dataset 1," Oxford University Research Archive (ORA) dataset, 2017, version 1. Accessed: 2026-02-13. [Online]. Available: https://ora.ox.ac.uk/objects/uuid:03ba4b01-cfed-46d3-9b1a-7d4a7bdf6fac

[28] Center for Advanced Life Cycle Engineering (CALCE), University of Maryland, "Calce battery data repository (lithium-ion aging datasets)," Open-access dataset, 2011, accessed: 2026-02-13. [Online]. Available: https://calce.umd.edu/battery-data

[29] NASA Ames Research Center, Prognostics Center of Excellence (PCoE), "Nasa ames prognostics center battery dataset (li-ion battery aging datasets)," Open-access dataset (DASHlink / NASA PCoE Prognostics Data Repository), 2010, download: https://phm-datasets.s3.amazonaws.com/NASA/5.+Battery+Data+Set.zip. Dataset citation commonly given as: Saha, B. and Goebel, K. (2007) "Battery Data Set", NASA Prognostics Data Repository. Accessed: 2026-02-13. [Online]. Available: https://c3.ndc.nasa.gov/dashlink/resources/133/

[30] Bixell Technology Limited, "Lithium battery products and oem/odm services," https://szbixell.com/, 2026, accessed: 8 September 2026.

[31] Batemo GmbH, "SINC IFR21700-3Ah Battery Data, Model and Report," https://www.batemo.com/products/batemo-cell-explorer/sinc-ifr21700-3ah/, 2024, independent experimental characterisation; model version 1.513, released 30 September 2024; accessed 8 September 2026.

[32] Samsung SDI Co., Ltd., "Specification of product for lithium-ion rechargeable cell: Icr18650-26a," Samsung SDI Co., Ltd., Mobile Energy Division, Tech. Rep. ICR18650-26A, Version 1.0, Jan. 2005, accessed: 8 September 2026. [Online]. Available: https://www.megacellmonitor.com/pdf/vendor_specs/SPEC_SAMSUNG_ICR18650-26A.pdf

[33] Texas Instruments, "Chemistry selection for the bq78pl114," Texas Instruments, Tech. Rep. SLUA505B, 2010, revised February 2010; accessed: 8 September 2026. [Online]. Available: https://www.ti.com/lit/an/slua505b/slua505b.pdf

[34] *BTS4000: The One for All Batteries (Battery Testing System)*, Neware Technology Limited, accessed: 2026-02-13. [Online]. Available: https://newarebattery.com/wp-content/uploads/2018/specs/Neware-BTS4000-compiled.pdf

[35] V. Sulzer, S. G. Marquis, R. Timms, M. Robinson, and S. J. Chapman, "Python battery mathematical modelling (pybamm)," *Journal of Open Research Software*, vol. 9, no. 1, p. 14, 2021. [Online]. Available: https://doi.org/10.5334/jors.309

[36] T. F. Fuller, M. Doyle, and J. Newman, "Simulation and optimization of the dual lithium ion insertion cell," *Journal of The Electrochemical Society*, vol. 141, no. 1, pp. 1–10, 1994.

[37] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[38] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[39] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.