LiPM: Foundation Model for Lithium-Ion Battery Analysis

Juren Li
jrlee@zju.edu.cn
Zhejiang University
Hangzhou, China

Yang Yang*
yangy@zju.edu.cn
Zhejiang University
Hangzhou, China

Hanchen Su
hcsu@zju.edu.cn
Zhejiang University
Hangzhou, China

Jiayu Liu
ljymail@zju.edu.cn
Zhejiang University
Hangzhou, China

Youmin Chen
youminchen@zju.edu.cn
Zhejiang University
Hangzhou, China

Jianfeng Zhang
zhangjianfeng3@huawei.com
Huawei Noah's Ark Lab
Shenzhen, China

Lujia Pan
panlujia@huawei.com
Huawei Noah's Ark Lab
Shenzhen, China

Abstract

Current deep learning approaches for lithium-ion battery analysis are often specialized and limited to specific battery types or individual tasks. While recent advances in large language models (LLMs) highlight the potential of pretraining paradigms, existing time-series pretraining models inadequately address the physicochemical complexity and temporal irregularity inherent to battery operational data. We propose LiPM, a pretrained foundation model that unifies multi-dataset learning through physics-aware objectives and irregularity-tolerant temporal modeling. LiPM introduces three key innovations: (1) A Mix-Masked Autoencoder (MMAE) enforcing electrochemical consistency via joint reconstruction of temporally masked patches and cross-channel masked variables, (2) A Coulombic Integration Regression (CIR) task explicitly encoding charge conservation laws, and (3) A dual-scale temporal encoder combining irregular intra-patch processing (preserving raw timestamps) with regular inter-patch attention (capturing macroscopic dynamics). Trained on eight heterogeneous battery datasets without cycle-label annotations, LiPM demonstrates universal applicability across partial charge-discharge segments and irregular sampling protocols. Extensive experiments show remarkable improvements over 9 state-of-the-art baselines in critical downstream tasks.

CCS Concepts

• **Computing methodologies** → **Artificial intelligence**; **Neural networks**.

*Corresponding author.

KDD '25, Toronto, ON, Canada

ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3737027

Keywords

Pretrained model; Lithium-Ion battery; Irregular time series

ACM Reference Format:
Juren Li, Yang Yang, Hanchen Su, Jiayu Liu, Youmin Chen, Jianfeng Zhang, and Lujia Pan. 2025. LiPM: Foundation Model for Lithium-Ion Battery Analysis. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3711896.3737027

KDD Availability Link:
The source code of this paper has been made publicly available at https://doi.org/10.5281/zenodo.15526358.

1 Introduction

The rapid transition towards renewable energy systems has placed lithium-ion (li-ion) batteries at the forefront of energy storage technologies, with applications spanning portable electronics to electric vehicles (EVs) and grid-scale infrastructure [29]. By 2050, the cumulative capacity of in-service and retired EV batteries is projected to exceed 32–62 terawatt-hours, far surpassing earlier industry forecasts [33]. This growth highlights the urgent need for advanced battery modeling frameworks to address critical challenges such as performance optimization, safety assurance, and sustainable lifecycle management. For example, accurate battery modeling is essential for precise state-of-health (SOH) estimation, a metric that not only quantifies residual capacity and degradation patterns but also facilitates the effective reuse of retired batteries in secondary applications such as grid storage [29], promoting economic savings and environmental conservation by extending battery lifecycle. Moreover, remaining useful life (RUL) predictions are crucial for managing EV insurance risk and ensuring safety, as batteries with less than 80% SOH exhibit higher failure rates, necessitating proactive maintenance to mitigate potential secure problems [41].

Current research in li-ion battery modeling, despite showing promise [10, 15, 19, 32, 37], faces significant challenges when applied to different types of batteries and operational conditions. Different types refer to variations in battery chemistry (e.g., NMC-811 vs. LFP) and specifications, while conditions include diverse operational scenarios such as fast charging, slow discharge, or varying

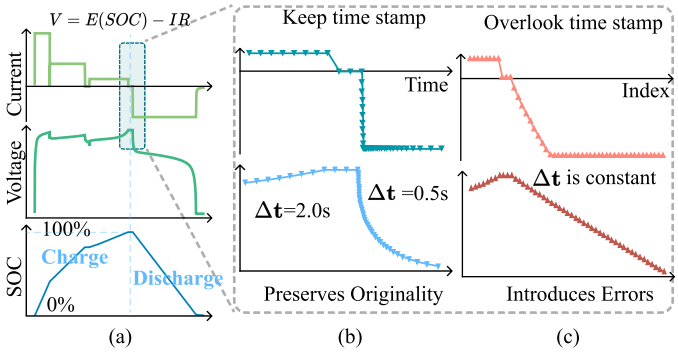

Figure 1: (a) A full charge-discharge cycle illustrating the complex physical relationships between current (I), voltage (V), and state-of-charge (SOC). These relationships are exemplified by the equation $V = E(SOC) - IR$, where $E(SOC)$ represents the open-circuit voltage that varies with SOC, and R is the battery's internal resistance, highlighting their correlation. (b) Partial cycle data showing irregular time series with sampling frequency increasing from sparse ($\Delta t = 2.0s$) to dense ($\Delta t = 0.5s$). (c) The same data as in (b), but ignoring timestamps, leading to significant errors.

temperature environments. Although various methodologies exist, they often fail to generalize well beyond the specific scenarios for which they are designed. For example, models optimized for one type of battery chemistry can produce highly inaccurate predictions when applied to another type under similar fast-charging conditions, leading to degradation over 300% in voltage prediction [30]. This highlights significant limitations in creating universally applicable models that can reliably predict battery performance across diverse settings.

Recent advances in pretraining models for time-series analysis [20, 35, 39, 42] have opened up new opportunities for addressing the limitations of battery data modeling. The emergence of these general-purpose pretraining models presents an unprecedented opportunity to develop more robust and general representations for downstream tasks of battery. The core idea behind pretraining is to leverage large-scale, multi-source datasets to learn generalized features that can be transferred to various downstream tasks. However, directly applying these generic time-series pretraining models to battery data introduces several significant challenges:

Lack of Physics-Aware Representation Learning. Existing pretraining methods primarily focus on optimizing temporal coherence while neglecting the inherent physical constraints embedded in battery data. Specifically, they fail to enforce fundamental relationships between measurable variables such as voltage (V), current (I), and time (t). As illustrated in Figure 1(a), the relationship between I, V, and SOC can be governed by the physical principle $V = E(SOC) - IR$, making them almost change synchronously. This oversight results in representations that capture statistical correlations but lack interpretable physical grounding, making them less effective for real-world applications.

Assumption of Regular Time Series. Most existing methods assume that the input data is regularly sampled, despite the intrinsic irregularity of real-world battery measurements. Notably, in our dataset collection, 7 out of 8 datasets exhibit irregular temporal sampling. As illustrated in Figure 1(b), voltage measurements during early charge stages are sparsely sampled due to stable dynamics, whereas rapid voltage drops in later discharge phases necessitate denser sampling. Such irregular temporal patterns, if unaccounted for, induce temporal aliasing errors in state-of-charge (SoC) estimation, leading to inaccurate predictions as shown in Figure 1(c).

Dependence on Cycle-Labeled Data. To better understand battery performance, it is essential to analyze *cycles*. As shown in Figure 1(a), a complete cycle of a battery refers to the complete process of charging from 0% SOC to 100% SOC and then discharging to 0% SOC, presenting a fundamental metric for assessing battery health and performance. Traditional battery modeling methods heavily rely on cycle-specific annotations. For example, the method proposed by Attia et al. [2] requires differences in the voltage-discharge quantity curve between specific cycles (e.g., 10th and 100th). Even cycle-agnostic methods like [18] need complete cycle data (e.g., current cycle plus preceding 29 cycles). However, in practical EV or smartphone applications, users rarely deplete batteries to 0% SoC before recharging, resulting in incomplete cycles like Figure 1(b). Therefore, models pretrained with complete cycles and cycle labels struggle to generalize to real-world usage and diverse downstream tasks, where charge-discharge cycles are often incomplete due to partial charging and discharging events, limiting their applicability in industrial settings.

To address these limitations, we propose LiPM (Lithium-ion battery Pretrained foundation model), a transformer-based architecture specifically designed to handle battery data's unique properties: physical consistency, temporal irregularity, and cycle-label scarcity. The framework comprises three key innovations:

Physics-Informed Pretraining Tasks: To ensure that the learned representations are physically meaningful, we introduce two novel pretraining tasks grounded in electrochemical principles governing **V** − **I** − **t** relationships:

- *Mix-Masked Autoencoder (MMAE)*: This method jointly reconstructs randomly masked patches across channels (current/voltage) and time, enforcing cross-variable physical correlations. By doing so, MMAE ensures that the learned features capture the intrinsic relationships between voltage, current, and time.
- *Coulombic Integration Regression (CIR)*: To explicitly incorporate charge conservation laws into our model, CIR predicts cumulative charge transfer $Q(t) = \int_{t_0}^{t} I(\tau)d\tau$ through trapezoidal rule approximation. This approach not only encodes fundamental electrochemical principles but also handles real-world operational patterns more effectively.

Irregularity-Aware Temporal Modeling: Recognizing the intrinsic irregularity of real-world battery data, we develop an innovative temporal modeling approach:

- *Intra-patch Irregular Processing*: Unlike conventional "patch by length" segmentation [20] (Figure 3 (a)), LiPM preserves raw timestamps with "patch by time" (Figure 3 (b)). This allows for accurate representation of variable sampling rates during different stages of charge/discharge cycles.

- *Inter-patch Regular Attention*: We compute cross-patch correlations using normalized relative regular time distances, ensuring that macroscopic temporal relationships are accurately captured.

Cycle-Agnostic Representation Learning: Given the scarcity of cycle-labeled data and the fragmented nature of real-world usage, LiPM is trained on universal variables $\{\mathbf{V}, \mathbf{I}\}$ and their corresponding timestamps without requiring cycle labels. This enables seamless handling of partial charging/discharging segments, making it highly adaptable to practical scenarios such as EV and smartphones.

We summarize the contributions of this work as follows:

- To the best of our knowledge, LiPM is first battery foundation model that unifies multi-dataset pretraining with physics-aware objectives, eliminating cycle-label dependencies.
- A dual-scale temporal encoder is proposed to natively handle irregular sampling through hybrid intra-patch and inter-patch processing.
- LiPM is pretrained on eight diverse datasets and evaluated on 62 downstream tasks, demonstrating an average improvement of **39.2**% in MSE compared to state-of-the-art baselines.

2 Methodology

Due to the variety in the types and amounts of variables among different battery datasets, we focus on current and voltage, which are present in all datasets. Although some datasets offer variables such as temperature, charge-discharge energy, etc., these variables may be absent in other datasets. Thus, we choose the current, voltage data that are nearly common to all datasets, along with their corresponding timestamps, as the input data. Consider a dataset of li-ion battery currents $\mathbf{I} = [I_0, I_1, \cdots, I_N]$, where $I > 0$ indicates charging and $I < 0$ indicates discharging, and voltages $\mathbf{V} = [V_0, V_1, \cdots, V_N]$. Here, I_i and V_i are recorded at time t_i. For simplicity, we use $\mathbf{x} = [x_1, x_2, \cdots, x_N]$ to represent $\mathbf{I}$ and $\mathbf{V}$, where $x_i = (V_i, I_i)$. The timestamp series $\mathbf{T} = [t_0, t_1, \cdots, t_N]$ is irregular and N is the total number of records.

2.1 Model Overview

As depicted in Figure 2, LiPM starts by partitioning the input $\mathbf{x}$ into patches of equal time spans according to the timestamps $\mathbf{T}$, that is, patch by time. Subsequently, LiPM conducts irregular time-series modeling within each patch (intra-patch) to obtain their respective embeddings. Then, LiPM processes these embeddings through a transformer encoder (inter-patch) to generate a representation. By using a Mix-Masked Autoencoder task (MMAE), LiPM models the underlying physical relationships between currents and voltages. Furthermore, to explicitly utilize timestamp information and directly incorporate charge/discharge conservation laws, we designed a Coulombic Integration Regression (CIR) task based on predicting cumulative charge/discharge. This enables the current and voltage representations generated by LiPM to contain temporal features, providing abundant information for multiple downstream tasks.

2.2 Patching for Irregular Time Series

Transformer architectures have become the cornerstone of modern time-series modeling, and they are also adopted as the core of our model design. Recent transformer-based time-series modeling methods commonly segment time-series data into patches. This approach has several advantages. It not only reduces the computational burden during data processing and improves training efficiency but also effectively captures and preserves the semantic information that could be lost if we treat each individual data point as a separate token. In the context of battery modeling, we follow a similar approach and divide the time-series data of battery currents and voltages into multiple patches.

Typically, these patches are generated by sliding a window of a fixed length, as shown in Figure 3 (a). However, such method cannot accurately retain the time information of irregular battery time series. Besides, positional encoding is crucial for capturing both the sequential and relative relationships. Patches that cover larger time intervals should have more substantial differences in positional encoding compared to those covering shorter intervals. Nevertheless, the conventional patching technique, which is based on a fixed number of timestamps, fails to account for these differences.

To address these issues, we propose the "patch by time" strategy, as shown in Figure 3. Specifically, we define the data points within each patch based on a fixed time span. Given a time span t_s, the i^{th} patch includes all the data points recorded within the time range from $(i-1) \times t_s$ to $i \times t_s$. Mathematically, we can represent the i^{th} patch as $\mathbf{x}_i = [x_j, x_{j+1}, \cdots, x_{j+k}]$ and the corresponding timestamps as $\mathbf{t}_i = [t_j, t_{j+1}, \cdots, t_{j+k}]$, where $(i-1) \times t_s \leq t_j < t_{j+1} < \cdots < t_{j+k} < i \times t_s$, and the length of the patch is $k+1$.

However, the length of each patch may vary, which makes it difficult for the model to perform parallel computations. To overcome this, we choose a predefined patch length that is long enough to accommodate all patches. For each patch, we generate a mask vector $\mathbf{t}_m = [1, \cdots, 1, 0, \cdots, 0]$, which indicates the actual number of valid data points within the patch. This mask is fed into the model together with the patch data, enabling parallel processing.

2.3 Irregular Intra-Patch Embedding

After performing the "patch by time" segmentation, LiPM maps each patch to embeddings of equal lengths. Traditional transformer-based methods typically use a linear layer for this mapping, expressed as $\mathbf{h} = \text{Linear}(\mathbf{x})$. Since each patch is processed independently in this step, we omit the subscript indicating the patch sequence number here and will continue to do so in the following descriptions. However, such a direct mapping fails to effectively utilize the crucial temporal information in irregular time series. Inspired by the positional encoding mechanism in transformers, we leverage the timestamps to measure the distances between points within a patch.

Specifically, we adopt a point-level Multi-Head Self-Attention (MHSA) mechanism to obtain the embeddings. First, we introduce an embedding token, which is initialized to zero, along with a corresponding timestamp also set to zero. Thus, the input to LiPM for each patch consists of $\mathbf{x} = [0, x_j, x_{j+1}, \cdots, x_{j+k}]$, $\mathbf{t} = [0, t_j, t_{j+1}, \cdots, t_{j+k}]$, and $\mathbf{t}_m = [1, \cdots, 1, 0, \cdots, 0]$. For each patch, the timestamps within the patch (excluding the embedding token) are re-initialized by subtracting $(i-1) * t_s$. This is because each patch is processed independently. Additionally, we convert each timestamp to seconds. In this way, within each patch, the difference between timestamps represents the relative time difference, which remains consistent regardless of the starting time of the patch. We

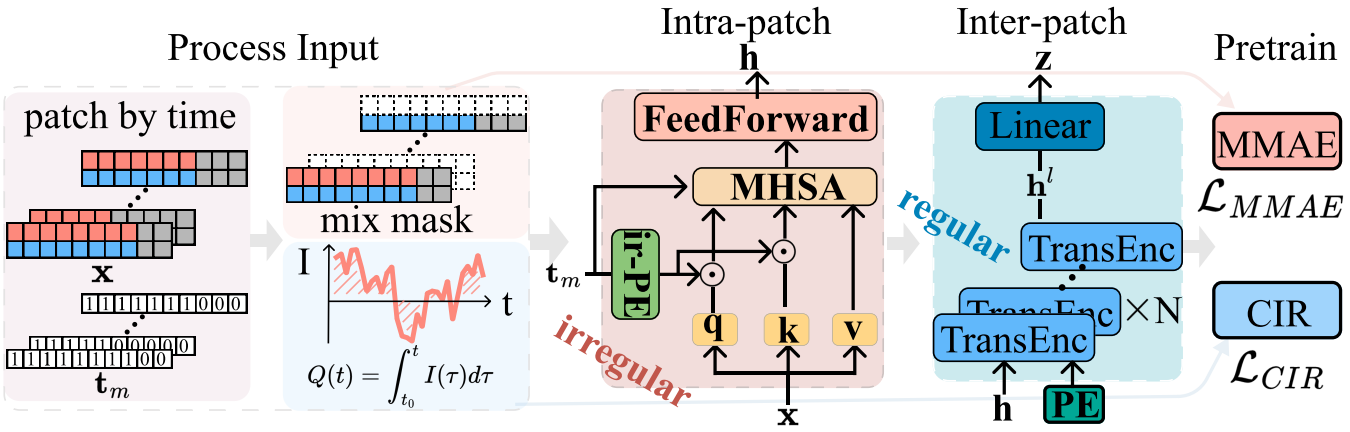

Figure 2: The pretraining framework of LiPM.

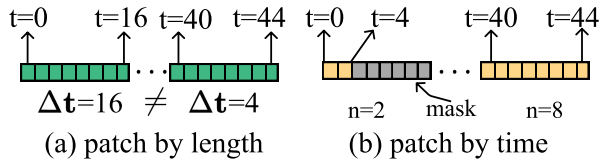

Figure 3: Two ways to segment irregular time series into patches.

utilize Rotary Position Embeddings (RoPE) [27] to encode the time information. First, each data point is linearly mapped to obtain the representations for queries, keys, and values, denoted as $\mathbf{q}$, $\mathbf{k}$, and $\mathbf{v}$ respectively. In the original RoPE, for a query located at position m, $\mathbf{q}_m$, and a key at position n, $\mathbf{k}_n$, the position encoding is defined as:

$$\mathbf{q}'_m = \mathbf{q}e^{im\theta}, \quad \mathbf{k}'_n = \mathbf{k}e^{in\theta} \tag{1}$$

For our irregular sequences, we replace m and n in the above equation with t_m and t_n, resulting in:

$$\mathbf{q}'_m = \mathbf{q}e^{it_m\theta}, \quad \mathbf{k}'_n = \mathbf{k}e^{it_n\theta} \tag{2}$$

The value of θ is chosen in accordance with that in the original RoPE to ensure a certain degree of decay over distance.

Subsequently, we employ multi-layer MHSA to obtain the patch embeddings. The computation for self-attention is given by:

$$\mathbf{o} = A_{ttn}\mathbf{v} = Softmax\left(\frac{\mathbf{q}'\mathbf{k}'^T}{\sqrt{d_k}}, \mathbf{t}_m\right)\mathbf{v} \tag{3}$$

We select the representation of the last-layer embedding token as the representation of the current patch, denoted as $\mathbf{h}$.

2.4 Transformer Encoder

The patch embedding sequence obtained from the previous irregular intra-patch embedding step is then directly input into a transformer encoder. To enhance the stability of the model, we employ RMSNorm (Root Mean Square Layer Normalization) [36] instead of the traditional layer normalization. RMSNorm's omission of mean subtraction can potentially expedite the model training process.

At the inter-patch level, since we model the patches in a regular manner, we use the standard Rotary Position Embeddings (RoPE). After stacking multiple layers of transformer encoder layers, we are able to generate the final representation $\mathbf{z}$. Mathematically, the process can be described as follows:

$$\mathbf{h}^l = TranEnc(\mathbf{h}^{l-1}), \quad l = 1, \cdots, L \tag{4}$$

$$\mathbf{z} = Linear(\mathbf{h}^L) \tag{5}$$

where the superscript l represents the l^{th} layer of the transformer encoder, and L is the total number of layers. The function $TranEnc(\cdot)$ denotes a single transformer encoder layer, and $Linear(\cdot)$ represents a linear layer. The resulting $\mathbf{z}$ is the overall representation.

2.5 Prtraining Task

Mix-Masked Autoencoder (MMAE). The Masked Autoencoder (MAE) paradigm has demonstrated efficacy in learning robust temporal representations through reconstruction of masked subsequences [5, 8, 28]. While conventional MAE implementations typically employ temporal masking (i.e., masking entire time patches), we extend this concept to address battery data's unique physical interdependencies. Specifically, the inherent relationship between voltage and current dynamics, where the direction of the current (charging or discharging) affects voltage changes, requires explicit modeling of these interactions. To achieve this, we propose a novel MMAE that integrates both temporal and cross-variable masking strategies, as illustrated in Figure 2. Our MMAE operates through two complementary masking mechanisms:

- *Cross-time Masking*: Randomly masks entire temporal patches to enforce learning of long-range electrochemical dynamics.
- *Cross-channel Masking*: Selectively masks individual variables (current or voltage) within patches to capture their physical correlations.

This dual masking strategy compels the model to reconstruct missing segments through both temporal continuity and physical consistency constraints. Formally, given input sequence $\mathbf{x}$ with timestamps $\mathbf{t}$, we generate three binary masks: $\mathbf{t}_m$: Natural sampling

mask (1=valid measurement). $\mathbf{p}_m$: Cross-time mask (0=patches to reconstruct). $\mathbf{c}_m$: Cross-channel mask (0=variables to reconstruct).

The effective mask $\mathbf{m} = \mathbf{t}_m \wedge \bar{\mathbf{p}}_m \wedge \bar{\mathbf{c}}_m$ determines reconstruction targets.

$$\hat{\mathbf{x}} = \text{MMAE}(\mathbf{z}_m, \mathbf{t}, \mathbf{t}_m) \tag{6}$$

where $\mathbf{z}_m$ denotes representations of masked data and $\hat{\mathbf{x}}$ is reconstructed data.

The reconstruction employs a dual-scale attention mechanism: Inter-patch MHSA: Restores global electrochemical patterns through transformer-based cross-patch attention. Intra-patch MHSA: Recovers temporal continuity within each irregular segment using timestamp-aware attention (like Section 2.3). The reconstruction loss emphasizes physically critical regions:

$$\mathcal{L}_{\text{MMAE}} = \frac{1}{\sum_{j=0}^{N} m_j} \sum_{i=0}^{N} (\hat{x}_i - x_i)^2 \cdot m_i \tag{7}$$

This design ensures three key advantages: i) Channel masking forces cross-variable reasoning; ii) Hybrid masking preserves both temporal dynamics and physical constraints; iii) Irregular timestamp encoding prevents aliasing artifacts in reconstructed signals. By simultaneously addressing temporal discontinuities and physical couplings, MMAE learns representations that inherently respect battery operational principles while maintaining reconstruction fidelity, a critical requirement for subsequent SOH/RUL estimation tasks.

Coulombic Integration Regression (CIR). The fundamental charge conservation principle ($Q(t) = \int_0^t I(\tau)d\tau$) governs all battery operations, where current polarity (positive/negative) directly reflects charging/discharging states. These bidirectional charge flows intrinsically correlate with critical battery health indicators, for instance, voltage hysteresis patterns during charge-discharge cycles provide essential degradation signatures [21]. To explicitly encode this physical law into our pretraining framework, we design a novel CIR task that predicts cumulative charge transfer through trapezoidal approximation [3].

Formally, given current measurements $\mathbf{I}$ with timestamps $\mathbf{t}$, the ground-truth cumulative charge at time t_i is calculated as:

$$Q_i = \frac{1}{2} \sum_{j=1}^{i} (I_j + I_{j-1})(t_j - t_{j-1}) \tag{8}$$

where $(t_j - t_{j-1})$ is converted to seconds to maintain unit consistency. This numerical integration method preserves current directionality while handling irregular sampling intervals [6].

Unlike the MMAE task that operates on masked inputs, CIR utilizes the original unmasked representations $\mathbf{z}$ to ensure numerical stability:

$$\hat{\mathbf{Q}} = \text{CIR}(\mathbf{z}, \mathbf{t}, \mathbf{t}_m) \tag{9}$$

The separation of reconstruction (MMAE) and physical regression (CIR) heads allows each component to specialize while sharing foundational representations. The CIR loss focuses on valid measurements through natural sampling masks:

$$\mathcal{L}_{\text{CIR}} = \frac{1}{\sum_{j=0}^{N} t_{mj}} \sum_{i=0}^{N} (\hat{Q}_i - Q_i)^2 \cdot t_{mi} \tag{10}$$

This design provides three key benefits: i) Enforces charge conservation laws through explicit current-time integration. ii) Maintains temporal causality by processing unmasked sequences. iii) Naturally handles real-world operational patterns (partial cycles, variable sampling).

2.6 Objective Function

The composite pretraining objective combines both physical consistency constraints:

$$\mathcal{L} = \mathcal{L}_{\text{MMAE}} + \mathcal{L}_{\text{CIR}} \tag{11}$$

This dual-task formulation ensures LiPM simultaneously learns: Cross-variable electrochemical relationships through MMAE's reconstruction. Temporal charge transfer dynamics via CIR's integration constraints. Robustness to irregular sampling through these two pretraining tasks. The synergistic optimization bridges data-driven learning with first-principles physics, establishing a foundation for generalizable battery modeling across diverse operational scenarios.

3 Experiment

3.1 Experiment setting

Datasets. Our study integrates eight publicly available lithium-ion battery datasets: NASA [23], CALCE [12], MATR [1], HUST [18], HNEI [7], ULPurdue [13], SNLLFP [22], and RWTH [16]. This comprehensive collection spans: 382 batteries covering major cathode chemistries ($LiCoO_2$, NMC, LFP, NCA) and 351 million data points totally. In addition, different charging and discharging protocols were used for these battery datasets. To ensure rigorous evaluation, we adopt a dataset-level separation strategy: i) Pre-training data: 85% batteries from each dataset. ii)Downstream tasks: Remaining 15% batteries held out for downstream tasks. iii) No battery overlap: All cycles from a given battery exclusively belong to one phase. This separation protocol prevents data leakage while maintaining chemical diversity, critical for assessing cross-domain generalization. Full dataset statistics and details are provided in Appendix.

Baseline Methods. We establish three comparison dimensions to validate LiPM's effectiveness:

- **Non-pretrained Models**: Crossformer [40]. iTransformer [17].
- **Time-series Pre-training**: GPT4TS [42], PatchTST [20], TS2Vec [35], TFC [39].
- **Irregular Time-series Models**: mTAN [26], Contiformer [4], tPatchGNN [38].

Implementation protocol during dowmstream tasks: i) For pretrained baselines: Frozen backbones + task-specific head fine-tuning. ii) For non-pretrained models: Full model training from scratch. iii) 10 experimental repetitions with different seeds (mean±std reported).

Methodological Distinction: Unlike conventional battery modeling approaches [2, 18, 25] requiring complete cycle labels or aligned cycle sequences, LiPM operates under *cycle-agnostic* conditions—processing arbitrary charge-discharge fragments without cycle boundary information. This aligns with real-world deployment constraints where full cycles are rarely observed (e.g., EV partial charging).

Table 1: The average experimental result in battery-split setting. The best result is in bold and second is in underlined. "Best" counts the number of times the method is best (62 in total). "p-value < 0.05" counts the number of times the method outperforms the second-best model with p-value < 0.05. "$\frac{1}{1000}$" indicates that the scale is too small, so the unit of the result is $\frac{1}{1000}$.

Task	Metrix	Crossformer	iTransformer	GPT4TS	PatchTST	TFC	TS2Vec	mTAN	CoutiFormer	tPatchGNN	LiPM
RUL	MSE	2.412	5.083	2.466	2.383	6.608	1.822	9.321	11.797	3.751	**1.744**
	MAE	0.866	1.309	0.971	1.027	1.275	0.779	1.635	1.692	1.259	**0.697**
SOH	MSE	0.004	0.097	0.201	0.006	0.148	0.104	0.498	0.358	0.148	**0.002**
	MAE	0.045	0.256	0.167	0.050	0.201	0.154	0.555	0.419	0.201	**0.027**
ΔQD	MSE	0.012	0.773	1.306	0.543	1.333	1.212	1.596	1.284	1.259	**0.002**
	MAE	0.057	0.558	0.407	0.278	0.527	0.444	0.835	0.616	0.638	**0.015**
ΔQC	MSE	0.009	0.238	1.125	0.339	1.209	1.142	2.394	2.110	2.517	**0.0002**
	MAE	0.048	0.345	0.400	0.233	0.515	0.394	1.031	0.866	0.920	**0.007**
IR($\frac{1}{1000}$)	MSE	0.206	10.3	0.045	9.119	8.682	0.106	306.3	104.7	0.328	**7.25_{e-04}**
	MAE	10.92	82.58	5.425	72.518	71.09	7.694	441.5	251.8	16.23	**0.539**
Average	MSE	0.488	1.240	1.019	0.656	1.862	0.856	2.823	3.131	1.535	**0.350**
	MAE	0.205	0.510	0.390	0.332	0.518	0.356	0.899	0.769	0.607	**0.149**
Best		3	0	8	3	0	6	0	0	0	**42**
p-value < 0.05		1	0	1	2	0	0	0	0	0	**37**

Table 2: The average experimental result in cycle-split setting.

Task	Metrix	Crossformer	iTransformer	GPT4TS	PatchTST	TFC	TS2Vec	mTAN	CoutiFormer	tPatchGNN	LiPM
RUL	MSE	2.603	4.414	2.640	3.101	5.097	1.887	10.164	9.954	3.957	**1.329**
	MAE	0.866	1.309	0.971	1.027	1.275	0.779	1.736	1.684	1.216	**0.590**
SOH	MSE	0.006	0.195	0.126	0.004	0.154	1.609	0.474	0.375	0.098	**0.001**
	MAE	0.048	0.344	0.159	0.044	0.208	0.159	0.538	0.468	0.170	**0.021**
ΔQD	MSE	0.005	0.296	1.150	0.386	1.095	1.098	1.562	1.230	1.242	**0.0003**
	MAE	0.036	0.366	0.395	0.236	0.479	0.381	0.820	0.686	0.634	**0.009**
ΔQC	MSE	0.009	0.238	1.125	0.339	1.209	1.142	2.553	2.208	2.495	**0.0001**
	MAE	0.048	0.345	0.400	0.233	0.515	0.394	1.048	0.947	0.919	**0.005**
IR($\frac{1}{1000}$)	MSE	0.34	5.201	0.141	0.347	3.926	0.149	305.1	78.08	0.507	**4.94_{e-04}**
	MAE	13.517	54.85	6.857	13.32	44.54	8.098	440.6	215.8	1.98	**0.448**
Average	MSE	0.525	1.030	1.008	0.766	1.512	1.147	3.012	2.769	1.559	**0.266**
	MAE	0.202	0.484	0.386	0.311	0.504	0.344	0.917	0.800	0.592	**0.125**
Best		0	0	1	3	0	3	0	0	0	**55**
p-value < 0.05		0	0	0	1	0	2	0	0	0	**53**

Battery State Estimation Tasks. Accurate state estimation is fundamental to li-ion battery management, requiring precise quantification of multiple interrelated variables:

- State of Health (SOH): Defined as the ratio between maximum available capacity at full charge and nominal capacity (manufacturer specification), SOH quantifies battery degradation:

$$\text{SOH} = \frac{Q_{\text{max}}}{Q_{\text{nominal}}} \times 100\% \tag{12}$$

  where Q_{max} is measured through constant-current discharge from 100% to cutoff voltage. SOH below 80% typically indicates end-of-life.
- Remaining Useful Life (RUL): The number of cycles until SOH reaches 80%, RUL reflects long-term degradation trends critical for maintenance planning and safety assurance.
- Internal Resistance (IR): Dynamic impedance measurements reveal electrochemical changes during operation. We evaluate IR estimation on MATR dataset.
- Charge/Discharge Quantity: Cumulative charge transfer $Q = \int Idt$ over arbitrary intervals enables state-of-charge (SOC) tracking and energy accounting. We separately evaluate charging (ΔQC) and discharging (ΔQD) quantities to: Explicitly model current directionality (charging: $I > 0$, discharging: $I < 0$); Support flexible operational scenarios (e.g., partial charge-discharge cycles); Support precise monitoring of energy flows in real-world battery management. This task evaluates the model's capacity to integrate temporal dynamics.

We evaluate LiPM on these tasks across two scenarios:

- Battery-split: Train/test sets contain distinct batteries (no overlap). This method ensures that training and testing data are derived from entirely distinct batteries. This is critical because different batteries may exhibit varying charging/discharging protocols (e.g., HUST and MATR datasets have distinct discharge/charge protocols). Thus, battery-split mimics real-world scenarios where a model must generalize to unseen batteries.
- Cycle-Split: Train/test sets share batteries but use different cycles. This method evaluates the model's ability to predict future behavior of the same battery across its lifespan. Batteries degrade over cycles, leading to evolving distributions between early and late cycles. This setup tests the model's capacity to capture long-term degradation trends within a single battery.

Given the variation in labels across different downstream tasks of the datasets, our experiments encompass a total of 62 distinct setups. Each setup is evaluated based on both Mean Squared Error (MSE) and Mean Absolute Error (MAE), resulting in 124 experimental outcomes. Statistical significance is verified via paired t-tests ($p < 0.05$) across 10 experimental trials. More details can be found in Appendix.

3.2 Experimental Results

Overall Performance. Table 1 and Table 2 summarizes key findings across all tasks and scenarios, and more specific experimental results are in the Appendix.. LiPM achieves consistent superiority over baselines:

- Battery Split: **28.3**% MSE reduction (0.350 vs 0.488) and **27.3**% MAE improvement (0.149 vs 0.205) on average.
- Cycle-split: **49.3**% MSE reduction (0.266 vs 0.525) with **59.8**% MAE enhancement (0.125 vs 0.311).
- Achieves **97/124** optima across all experiments, of which **90/124** show significant improvements over other baselines (p-value < 0.05).

3.2.1 Task-Specific Analysis. . **SOH Estimation**: LiPM achieves 50.0% MSE reduction in battery-split (0.002 vs 0.004) compared to the best baseline Crossformer and achieves 75.0% MSE reduction in cycle-split (0.001 vs 0.004) compared to best baseline PatchTST. This validates our model's capacity to capture battery states.
RUL Prediction: LiPM achieves 4.2% MSE reduction in battery-split (1.744 vs 1.822) and 29.6% MSE reduction in cycle-split (1.329 vs 1.887) compared to best baseline TS2Vec. The relative gains are less pronounced than SOH (4.2% vs 50.0% and 29.6% vs 75%). This is because RUL prediction heavily relies on long-term degradation patterns, which the partial charge/discharge cycles used in our dataset do not fully capture. Consequently, LiPM's improvements are less significant.
Charge/Discharge Quantity Calculation. LiPM achieves orders-of-magnitude improvements in cumulative charge quantification, significantly outperforming both non-pretrained and pretrained baselines:

LiPM vs non-pretrained (Crossformer): For battery-split setting, charging MSE 0.002 vs 0.012 (6× improvements) / discharging MSE 0.0002 vs 0.009 (45× improvements). For cycle-split setting, charging MSE 0.0003 vs 0.005 (16× improvements) / discharging MSE 0.0001 vs 0.009 (90× improvements).

LiPM vs Pretrained method (PatchTST): For battery-split setting: charging MSE 0.002 vs 0.543 (271× improvements) / discharging MSE 0.0002 vs 0.339 (1,695× improvements). For cycle-split setting: charging MSE 0.0003 vs 0.386 (1,287× improvements) / discharging MSE 0.0001 vs 0.339 (3,390× improvements).

This dramatic improvement stems from LiPM's explicit modeling of temporal charge dynamics through the Coulombic Integration Regression (CIR) task. The orders-of-magnitude error reduction demonstrates that conventional temporal modeling approaches, even with pretraining, may not fully capture the physical-temporal interdependencies essential for battery charge accounting. Our approach with LiPM addresses these challenges, offering a refined method for charge/discharge monitoring.
Internal Resistance Estimation: Significant performance improvement (MSE from 0.45e-04 to 7.25e-07 in the battery-split setting; In the cycle-split setting, MSE drops from 1.49e-04 to 4.94e-07) confirms LiPM's effectiveness in learning voltage-current correlations critical for internal resistance dynamics.

3.2.2 Comparative Analysis and Insights. Our experiments reveal that time series pretraining models (GPT4TS, PatchTST, TS2Vec, TFC) do not consistently outperform non-pretrained models like Crossformer. While these pretrained models excel in capturing temporal dependencies, they may lack the domain-specific knowledge crucial for understanding battery behavior. Specifically, with their backbones frozen during downstream tasks, these models focus more on temporal features rather than incorporating physical constraints relevant to battery dynamics.

LiPM's success in various battery state estimation tasks and charge/discharge quantity calculation, stems from its MMAE and CIR pretraining tasks combined with a dual-scale temporal encoder. These features allow LiPM to effectively integrate temporal dynamics with critical physical insights, resulting in state-of-the-art performance in downstream tasks.

3.3 Ablation Study

Table 3: Ablation study of pretraining tasks.

Task	Metric	LiPM	Only MMAE	Only CIR
RUL	MSE	**1.744**	2.368	2.023
	MAE	**0.697**	0.901	0.772
SOH	MSE	**0.002**	0.007	0.007
	MAE	**0.027**	0.053	0.044
ΔQD	MSE	**0.002**	0.028	0.007
	MAE	**0.015**	0.078	0.035
ΔQC	MSE	**0.0002**	0.042	0.007
	MAE	**0.007**	0.090	0.034
IR($\frac{1}{1000}$)	MSE	**7.24e-04**	8.71e-04	8.80e-04
	MAE	**0.539**	0.636	0.623
Average	MSE	**0.350**	0.489	0.409
	MAE	**0.149**	0.225	0.177

In this section, we conduct ablation studies to investigate the impact of different components in proposed LiPM on downstream

Table 4: Ablation study of irregular time series modeling.

Task	Metric	LiPM	without **t**	with **t**
RUL	MSE	**1.744**	3.189	2.806
	MAE	**0.697**	0.967	0.772
SOH	MSE	**0.002**	0.122	0.179
	MAE	**0.027**	0.091	0.103
ΔQD	MSE	**0.002**	0.366	0.065
	MAE	**0.015**	0.156	0.076
ΔQC	MSE	**0.0002**	0.618	0.149
	MAE	**0.007**	0.223	0.110
IR($\frac{1}{1000}$)	MSE	**7.24e-04**	1.35e-03	1.27e-03
	MAE	**0.539**	0.924	0.903
Average	MSE	**0.350**	0.859	0.640
	MAE	**0.149**	0.288	0.212

Table 5: Ablation study of irregular time series modeling. "0.5, 0" indicates the ratio of cross-channel and cross-time masking is 0.5 and 0 respectively.

Tasks	Metric	LiPM	0.5, 0.5	0.5, 0	0, 0.5
RUL	MSE	**1.744**	2.361	2.677	2.207
	MAE	**0.697**	0.778	0.837	0.771
SOH	MSE	**0.002**	0.003	0.002	0.003
	MAE	**0.027**	0.029	0.028	0.028
QD	MSE	**0.002**	0.002	0.004	0.004
	MAE	**0.015**	0.015	0.019	0.021
QC	MSE	**0.0002**	0.0003	0.0004	**0.0002**
	MAE	**0.007**	**0.007**	0.010	**0.007**
IR	MSE	**7.24E**-04	3.22E-03	1.14E-03	9.70E-04
	MAE	**0.539**	0.730	0.709	0.670
Average	MSE	**0.350**	0.473	0.537	0.443
	MAE	**0.149**	0.312	0.321	0.299

task performance. The experiments are mainly conducted under a battery-split experimental setting.

3.3.1 Impact of Pre-training Tasks. To systematically evaluate the effectiveness of MMAE and CIR as pre-training tasks, we conducted ablations by individually excluding one of these tasks while keeping other settings constant for downstream tasks. The results are summarized in Table 3. Our findings demonstrate that removing either MMAE or CIR leads to significant performance degradation in LiPM. Notably, excluding the CIR task increases MSE for both ΔQC and ΔQD tasks by more than tenfold, highlighting its essential role in predicting battery state changes.

3.3.2 Irregular Time Series Modeling. In order to verify the effectiveness of "patch by time", we directly remove this part. Considering that **t** may carry useful temporal information for irregular time series modeling, we also consider adding it to the input. Table 4 presents the results, distinguishing between experiments "without **t**" (without temporal data) and "with **t**" (including time data). These experiments reveal that modeling irregular time series is crucial

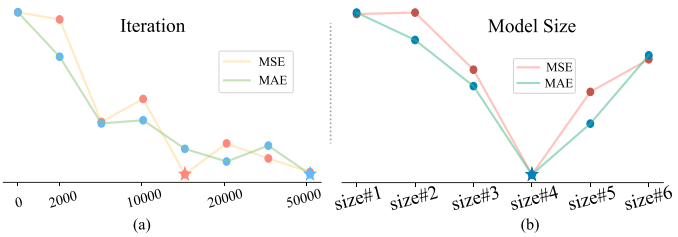

Figure 4: (a) Average results on downstream tasks with different pre-training iterations. (b) The average performance of LiPM on downstream tasks, where size#1 is the smallest model and size#6 is the largest model.

for each downstream task. Moreover, even when only employing the "patch by length" approach, incorporating additional temporal information enhances overall model performance.

3.3.3 MMAE Masking Strategy Analysis. In our proposed LiPM, each operation randomly masks 30% of the data independently, resulting in nearly half of the data being masked. To assess the effectiveness of these strategies, we conducted experiments using only cross-time masking or only cross-channel masking, setting the mask ratio to 0.5 to ensure half of the data remains masked. Additionally, we experimented with increasing mask ratio of both masking strategies to 0.5 to further challenge the model's learning capacity. Results in Table 5 show that relying solely on one masking strategy reduces the model's capability. Increasing the mask ratio to 0.5 also decreases performance, likely due to insufficient unmasked data for effective pre-training.

3.4 Model Analysis

3.4.1 Effect of Pre-training Iterations. Figure 4(a) illustrates the average performance on downstream tasks at various pre-training iterations, with detailed results provided in the Appendix. Optimal performance is marked with a star, showing that the lowest MSE is achieved at 14,000 iterations, closely followed by 50,000 iterations. For MAE, the best performance is observed at 50,000 iterations. Besides, the performance of LiPM is relatively stable from 14,000 iterations to 50,000 iterations, with no significant fluctuations.

3.4.2 Effect of Model Size. Considering LiPM's Transformer architecture, understanding the influence of model size on performance is critical. We adjusted the number of Transformer layers and the hidden size to vary model parameters, with specific configurations detailed in the Appendix. Figure 4(b) shows the performance of different model sizes on downstream tasks. Testing six models ranging from 9 million to 312 million parameters, we find that model size#4 (48M parameters) achieves the best performance. Further increases in model size lead to performance degradation, which we attribute to limitations in the pre-training dataset size, considering our use of data from only eight datasets. This suggests that current pre-training efforts may be constrained by the breadth and diversity of available battery operational data.

3.5 Efficiency Comparison

We provide a direct comparison of inference time for SOH prediction (100 batches, identical hardware and batch size), the results are show in Table 11. LiPM 's efficiency aligns with transformer baselines (e.g., Crossformer) while incorporating irregular-time modeling.

Table 6: Comparison of inference time (in seconds) for different methods.

Method	Crossformer	iTransformer	GPT4TS	PatchTST	TFC
Time (s)	2.51	1.49	0.12	5.87	0.41
Method	TS2Vec	mTAN	CountiFormer	tPatchGNN	LiPM
Time (s)	1.17	4.60	272.51	2.41	2.44

4 Related Work

Contemporary approaches to battery modeling fall into three categories: **Physics-Based Models**. Electrochemical models like the Pseudo-Two-Dimensional (P2D) framework [9] employ coupled PDEs to describe electrode interactions. While variants like SPMe [11] improve electrolyte dynamics, their practical adoption is limited by computationally intensive parameter identification and overfitting risks [34]. **Equivalent Circuit Models**. Ranging from basic Rint models to fractional-order networks with CPEs [10], these methods balance accuracy and computational efficiency for real-time applications. However, they struggle to capture nonlinear degradation patterns essential for SOH/RUL estimation [19]. **Data-Driven Approaches** Modern neural networks [37] and encoder-decoder architectures [24] bypass physical complexity but require extensive labeled data. Recent pre-training paradigms like [14, 31, 35, 39] demonstrate potential for temporal representation learning, yet remain agnostic to battery-specific physics.

Current pre-training methods face two critical limitations for battery data: **Physical Consistency Gap**. Standard temporal models (PatchTST [20], iTransformer [17]) optimize statistical correlations rather than electrochemical principles. The inherent $V - I - t$ relationships governed by charge conservation remain unenforced. **Regularity Assumption**. Most frameworks [40, 42] presume uniform sampling intervals, despite real-world battery data exhibiting adaptive sampling rates (Fig. 1(a)).

Emerging approaches address temporal irregularities through: mTAN [26] employs multi-time attention to capture irregular dependencies, while tPatchGNN [38] uses adaptive graph networks for asynchronous series. However, these generic methods fail to preserve battery-specific physical constraints during charge integration. Recent irregular TS models like ContiFormer [4] handle missing values but lack domain-specific adaptations for partial charge cycles.

5 Conclusion

We present LiPM, a novel universal pretrained model for li-ion battery modeling, LiPM, that bridges the gap between data-driven learning and electrochemical principles. By introducing two physics-aware pretraining tasks, MMAE and CIR, LiPM achieves superior performance across multiple downstream tasks. The model's dual-scale temporal encoder, combining irregular intra-patch processing with regular inter-patch attention, effectively handles real-world operational data with variable sampling rates.

Future work could focus on expanding pretraining datasets to support larger models for broader applicability, and exploring real-time online learning to enhance LiPM's adaptability and robustness in practical scenarios.

Acknowledgments

This work is supported by National Natural Science Foundation of China (No. 62176233) and National Science Fund of China for Excellent Young Scholars (No. 62322606).

A Dataset

The datasets used in this paper are as follows. Table 7 shows the dataset statistics.

- **NASA** [23]: A National Aeronautics and Space Administration (NASA) lithium-ion battery charge and discharge experimental dataset. The experiment process is as follows: In charging level, the batteries are charged with a constant current (CC) of 1.5A until the battery voltage reaches 4.2V, then they switch to constant voltage (CV) mode until the charging current drops to 20mA. The cathode material is lithium cobalt oxide. The dataset contains 7,203,084 records from 34 cells, each with a rated capacity of 2.0Ah.
- **CALCE** [12]: Comes from the Center for Advanced Life Cycle Engineering (CALCE) of the University of Maryland. This dataset includes two types of batteries: CS2 and CX2.
  For CS2 batteries:
  – Capacity Rating: 1.1Ah; Cell Chemistry: LiCoO2 cathode, EDS results also showed trace elements of Manganese.
  – Charging Profile: Standard constant current/constant voltage protocol with a constant current rate of 0.5C until the voltage reached 4.2V, then 4.2V was sustained until the charging current dropped to below 0.05A. Unless specified, the discharge cut-off voltage for these batteries was 2.7V.
  For CX2 batteries:
  – Capacity Rating: 1.35Ah; Cell Chemistry: LiCoO2 cathode (EDS showed trace elements of Manganese).
  – Charging Profile: Similar to CS2 batteries but with some minor variations incorporated in the standard charging profile for some CX2 cells.
  In total, we use 27 batteries from both types, and the dataset contains 16,852,982 records.
- **HUST** [18]: A dataset with 77 LFP/graphite cells (1.1 Ah nominal capacity and 3.3 V nominal voltage). The cells were cycled with an identical charge protocol but different multi-stage discharge protocols at a constant temperature of 30°C. The dataset contains 91,735,853 records.
- **MATR** [1, 25]: Consists of 185 commercial lithium-ion batteries cycled to failure under fast-charging conditions. All cells in this dataset are charged with a one-step or two-step fast-charging policy. This policy has the format "C1(Q1)-C2", where C1 and C2 are the first and second constant-current steps, respectively, and Q1 is the state-of-charge (SOC, %) at which the currents switch. The second current step ends at 80% SOC, after which the cells charge at 1C CC-CV. The upper and lower cutoff potentials are 3.6 V and 2.0 V, respectively. All cells discharge at 4C. The dataset contains four batches of lithium-ion phosphate (LFP)/graphite cells and 143,576,451 records.
- **RWTH** [16]: Contains time-series data (time, current, voltage, temperature) of a cyclic aging test of 48 lithium-ion battery cells. The experiment process involves aging 48 cells of the same type with the same profile under equal conditions. The battery uses a carbon anode and NMC as a cathode material. The dataset contains 48,625,944 records.
- **HNEI** [7]: Consists of 14 commercial 18650 cells with a graphite negative electrode and a blended positive electrode composed of NMC and LCO. The cell was cycled at 1.5C to 100% depth of discharge (DOD) for more than 1000 cycles at room temperature. The dataset contains 7,497,493 records.
- **SNLLFP** [22]: Consists of 18 commercial 18650 LFP cells cycled to 80% capacity (although cycling is still ongoing) from Sandia National Labs. This study examines the influence of temperature, depth of discharge (DOD), and discharge current on the long-term degradation of the commercial cells. The dataset contains 6,904,086 records. Additionally, datasets SNLNCA and SNLNMC are also available from the same source; however, due to their insufficient data length for our input requirements, they have not been included in this study.
- **ULPurdue** [13]: Consists of 22 commercial 18650 cells with a graphite negative electrode and an NCA positive electrode. The cells were cycled at 0.5C at 2.7-4.2V (0-100% SOC) or 2.9-4.0V (2.5-96.5% SOC, when the cells are fresh) at room temperature to various levels of capacity fade (10%, 15%, and 20%). The dataset contains 6,246,486 records.

Table 7: Dataset Statistics

Dataset	Number of Cells	Number of Records
NASA	34	7,203,084
CALCE	27	16,852,982
HUST	77	91,735,853
MATR	185	143,576,451
RWTH	48	48,625,944
HNEI	14	7,497,493
SNLLFP	18	6,904,086
ULPurdue	22	6,246,486

B Experiment Setting

The implementation protocol during downstream tasks is as follows:

(1) For pretrained baselines: Frozen backbones + task-specific head fine-tuning.
(2) For non-pretrained models: Full model training from scratch.
(3) 10 experimental repetitions with different seeds (mean±std reported).

The implementation protocol during pretraining is as follows: For both our proposed LiPM and the baseline models requiring pretraining, we use the same dataset for pretraining. For baseline model pretraining, we follow their recommended settings. We ensure that all pretraining models reach convergence.

For LiPM, our default settings are as follows: a learning rate of 1e-04; using the AdamW optimizer with a weight decay set to 1e-03; a batch size of 256; a pretraining iteration count of 50,000; a ratio of cross-time masking set to 0.3; and a ratio of cross-channel masking set to 0.3. In the transformer encoder, the hidden layer dimension is set to 512, with 12 layers.

For input data settings, all baselines and our proposed LiPM maintain consistency as much as possible: the input data length is 256. If there is a need to split patches, the patch length is set to

16. Since LiPM splits patches by time, it cannot use a fixed sliding window method. Due to varying conditions across datasets, we set different time spans for each dataset to split patches, ensuring an average patch length close to 16. The maximum patch length is set to 64, and we use a masking method to indicate which values are valid, as described in Section 2.2.

B.1 Battery State Estimation Tasks

Given the variation in labels across different downstream tasks of the datasets, our experiments encompass a total of 62 distinct setups, as shown in Table 8.

Table 8: Dataset Usage for Different Tasks

Task	Datasets
RUL	NASA, MATR, HUST, HNEI ULPurdue, SNLLFP, RWTH
SOH	NASA, MATR, HUST, HNEI ULPurdue, SNLLFP, RWTH
ΔQC	NASA, MATR, HUST, HNEI ULPurdue, SNLLFP, RWTH, CALCE
ΔQD	NASA, MATR, HUST, HNEI ULPurdue, SNLLFP, RWTH, CALCE
IR	MATR

C Experimental Result

C.1 Effect of Model Size

Table 9: Experimental results with different model size.

	s1	s2	s3	s4	s5	s6
MSE						
RUL	2.73E+00	2.74E+00	2.39E+00	1.74E+00	2.25E+00	2.44E+00
SOH	3.19E-03	3.32E-03	3.52E-03	2.19E-03	2.89E-03	4.60E-03
QD	4.82E-03	2.36E-03	3.15E-03	1.50E-03	7.45E-03	8.88E-03
QC	6.87E-04	9.95E-04	8.19E-04	2.33E-04	1.72E-03	4.38E-03
IR	1.33E-06	1.10E-06	8.82E-07	7.24E-04	9.89E-07	8.71E-07
AVG	5.48E-01	5.50E-01	4.79E-01	3.50E-01	4.52E-01	4.92E-01
MAE						
RUL	9.37E-01	9.01E-01	8.25E-01	6.97E-01	7.55E-01	8.25E-01
SOH	3.46E-02	3.50E-02	3.26E-02	2.70E-02	3.06E-02	4.36E-02
QD	2.60E-02	1.64E-02	2.09E-02	1.51E-02	2.79E-02	4.00E-02
QC	9.68E-03	1.07E-02	9.40E-03	7.10E-03	1.39E-02	2.91E-02
IR	6.84E-04	6.85E-04	6.28E-04	5.39E-01	6.69E-04	5.74E-04
AVG	2.02E-01	1.93E-01	1.78E-01	1.49E-01	1.66E-01	1.88E-01

Considering LiPM's Transformer architecture, understanding the influence of model size on performance is critical. We adjusted the number of Transformer layers and the hidden size to vary model parameters. Table 9 and Table 10 show the performance of different model sizes on downstream tasks. Testing six models ranging from 9 million to 312 million parameters, we find that model size#4 (48M parameters) achieves the best performance. Further increases in model size lead to performance degradation, which we attribute to limitations in the pre-training dataset size, considering our use of data from only eight datasets. This suggests that current pre-training efforts may be constrained by the breadth and diversity of available battery operational data.

Table 10: Model Settings and number of parameters. d_model denotes the hidden layer dimension. n_layer denotes the number of hidden layers. n_head denotes the number of heads of the MHSA; n_param indicates the number of parameters in million (M).

	s1	s2	s3	s4	s5	s6
d_model	64	128	256	512	768	1024
n_layer	2	3	6	12	18	16
n_head	4	4	8	8	12	24
n_param (M)	9.7	10.3	15.9	47.6	137.5	312.4

C.2 Experiment Compute Resource.

Table 11: Pretraining time cost.

Model	S1	S2	S3	S4	S5	S6
Time (s)	10742	11191	12519	18767	39027	55253

The CPU we used is two Intel(R) Xeon(R) Gold 6226R, and the GPU used is two NVIDIA GeForce RTX 3090, and the training time is shown in Table 11.

D More Details

Owing to space constraints, additional information regarding the experimental methodology, detailed results, and comprehensive analysis is available at https://github.com/JuRenGithub/LiPM.

References

[1] Peter M Attia, Aditya Grover, Norman Jin, Kristen A Severson, Todor M Markov, Yang-Hung Liao, Michael H Chen, Bryan Cheong, Nicholas Perkins, Zi Yang, et al. 2020. Closed-loop optimization of fast-charging protocols for batteries with machine learning. *Nature* 578, 7795 (2020), 397–402.

[2] Peter M Attia, Kristen A Severson, and Jeremy D Witmer. 2021. Statistical learning for accurate and interpretable battery lifetime prediction. *Journal of The Electrochemical Society* 168, 9 (2021), 090547.

[3] Richard L. Burden, J. Douglas Faires, and Annette M. Burden. 2015. *Numerical Analysis* (10th ed.). Cengage Learning. 194–201 pages.

[4] Yuqi Chen, Kan Ren, Yansen Wang, Yuchen Fang, Weiwei Sun, and Dongsheng Li. 2024. Contiformer: Continuous-time transformer for irregular time series modeling. *Advances in Neural Information Processing Systems* 36 (2024).

[5] Jie Cheng, Xiaodong Mei, and Ming Liu. 2023. Forecast-mae: Self-supervised pre-training for motion forecasting with masked autoencoders. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 8679–8689.

[6] Philip J Davis and Philip Rabinowitz. 2007. *Methods of numerical integration*. Courier Corporation.

[7] Arnaud Devie, George Baure, and Matthieu Dubarry. 2018. Intrinsic variability in the degradation of a batch of commercial 18650 lithium-ion cells. *Energies* 11, 5 (2018), 1031.

[8] Jiaxiang Dong, Haixu Wu, Haoran Zhang, Li Zhang, Jianmin Wang, and Mingsheng Long. 2024. Simmtm: A simple pre-training framework for masked time-series modeling. *Advances in Neural Information Processing Systems* 36 (2024).

[9] Marc Doyle, Thomas F Fuller, and John Newman. 1993. Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell. *Journal of the Electrochemical society* 140, 6 (1993), 1526.

[10] Todd J Freeborn, Brent Maundy, and Ahmed S Elwakil. 2015. Fractional-order models of supercapacitors, batteries and fuel cells: a survey. *Materials for Renewable and Sustainable Energy* 4 (2015), 1–7.

[11] Thomas R. B. Grandjean, Liuying Li, Maria Ximena Odio, and Widanalage D. Widanage. 2019. Global Sensitivity Analysis of the Single Particle Lithium-Ion Battery Model with Electrolyte. In *2019 IEEE Vehicle Power and Propulsion Conference (VPPC)*. 1–7. doi:10.1109/VPPC46532.2019.8952455

[12] Wei He, Nicholas Williard, Michael Osterman, and Michael Pecht. 2011. Prognostics of lithium-ion batteries based on Dempster–Shafer theory and the Bayesian Monte Carlo method. *Journal of Power Sources* 196, 23 (2011), 10314–10321.

[13] Daniel Juarez-Robles, Judith A Jeevarajan, and Partha P Mukherjee. 2020. Degradation-safety analytics in lithium-ion cells: Part I. Aging under charge/discharge cycling. *Journal of The Electrochemical Society* 167, 16 (2020), 160510.

[14] Juren Li, Yang Yang, Youmin Chen, Jianfeng Zhang, Zeyu Lai, and Lujia Pan. 2024. DWLR: Domain Adaptation under Label Shift for Wearable Sensor. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, Kate Larson (Ed.). International Joint Conferences on Artificial Intelligence Organization, 4425–4433. doi:10.24963/ijcai.2024/489 Main Track.

[15] Shuangqi Li, Hongwen He, and Jianwei Li. 2019. Big data driven lithium-ion battery modeling method based on SDAE-ELM algorithm and data pre-processing technology. *Applied energy* 242 (2019), 1259–1273.

[16] Weihan Li, Neil Sengupta, Philipp Dechent, David Howey, Anuradha Annaswamy, and Dirk Uwe Sauer. 2021. One-shot battery degradation trajectory prediction with deep learning. *Journal of Power Sources* 506 (2021), 230024.

[17] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. 2023. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625* (2023).

[18] Guijun Ma, Songpei Xu, Benben Jiang, Cheng Cheng, Xin Yang, Yue Shen, Tao Yang, Yunhui Huang, Han Ding, and Ye Yuan. 2022. Real-time personalized health status prediction of lithium-ion batteries using deep transfer learning. *Energy & Environmental Science* 15, 10 (2022), 4083–4094.

[19] Hao Mu, Rui Xiong, Hongfei Zheng, Yuhua Chang, and Zeyu Chen. 2017. A novel fractional order model based state-of-charge estimation method for lithium-ion battery. *Applied Energy* 207 (2017), 384–393.

[20] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2022. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730* (2022).

[21] Gregory L Plett. 2015. *Battery management systems, Volume I: Battery modeling*. Artech House.

[22] Yuliya Preger, Heather M Barkholtz, Armando Fresquez, Daniel L Campbell, Benjamin W Juba, Jessica Romàn-Kustas, Summer R Ferreira, and Babu Chalamala. 2020. Degradation of commercial lithium-ion cells as a function of chemistry and cycling conditions. *Journal of The Electrochemical Society* 167, 12 (2020), 120532.

[23] B Sahaand and K Goebel. 2007. Battery Data Set, NASA ames prognostics data repository. *NASA Ames Research Center* (2007). http://ti.arc.nasa.gov/project/prognostic-data-repository

[24] Jakob Schmitt, Ivo Horstkötter, and Bernard Bäker. 2023. Electrical lithium-ion battery models based on recurrent neural networks: A holistic approach. *Journal of Energy Storage* 58 (2023), 106461.

[25] Kristen A Severson, Peter M Attia, Norman Jin, Nicholas Perkins, Benben Jiang, Zi Yang, Michael H Chen, Muratahan Aykol, Patrick K Herring, Dimitrios Fraggedakis, et al. 2019. Data-driven prediction of battery cycle life before capacity degradation. *Nature Energy* 4, 5 (2019), 383–391.

[26] Satya Narayan Shukla and Benjamin M Marlin. 2021. Multi-time attention networks for irregularly sampled time series. *arXiv preprint arXiv:2101.10318* (2021).

[27] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing* 568 (2024), 127063.

[28] Peiwang Tang and Xianchao Zhang. 2022. MTSMAE: masked autoencoders for multivariate time-series forecasting. In *2022 IEEE 34th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 982–989.

[29] Shengyu Tao, Ruifei Ma, Zixi Zhao, Guangyuan Ma, Lin Su, Heng Chang, Yuou Chen, Haizhou Liu, Zheng Liang, Tingwei Cao, et al. 2024. Generative learning assisted state-of-health estimation for sustainable battery recycling with random retirement conditions. *Nature Communications* 15, 1 (2024), 10154.

[30] Manh-Kien Tran, Andre DaCosta, Anosh Mevawalla, Satyam Panchal, and Michael Fowler. 2021. Comparative Study of Equivalent Circuit Models Performance in Four Common Lithium-Ion Batteries: LFP, NMC, LMO, NCA. *Batteries* 7, 3 (2021). doi:10.3390/batteries7030051

[31] Shihao Tu, Yupeng Zhang, Jing Zhang, Zhendong Fu, Yin Zhang, and Yang Yang. 2024. PowerPM: Foundation Model for Power Systems. In *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Eds.), Vol. 37. Curran Associates, Inc., 115233–115260. https://proceedings.neurips.cc/paper_files/paper/2024/file/d0a2279c9f7ded859bcbf878c3c3d1ed-Paper-Conference.pdf

[32] Rui Xiong, Jinpeng Tian, Weixiang Shen, and Fengchun Sun. 2018. A novel fractional order model for state of charge estimation in lithium ion batteries. *IEEE Transactions on Vehicular Technology* 68, 5 (2018), 4130–4139.

[33] Chengjian Xu, Paul Behrens, Paul Gasper, Kandler Smith, Mingming Hu, Arnold Tukker, and Bernhard Steubing. 2023. Electric vehicle batteries alone could satisfy short-term grid storage demand by as early as 2030. *Nature Communications* 14, 1 (2023), 119.

[34] Meng Xu, Rui Wang, Peng Zhao, and Xia Wang. 2019. Fast charging optimization for lithium-ion batteries based on dynamic programming algorithm and electrochemical-thermal-capacity fade coupled model. *Journal of Power Sources* 438 (2019), 227015.

[35] Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. 2022. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 8980–8987.

[36] Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. *Advances in Neural Information Processing Systems* 32 (2019).

[37] Chenbin Zhang, Yayun Zhu, Guangzhong Dong, and Jingwen Wei. 2019. Data-driven lithium-ion battery states estimation using neural networks and particle filtering. *International Journal of Energy Research* 43, 14 (2019), 8230–8241.

[38] Weijia Zhang, Chenlong Yin, Hao Liu, Xiaofang Zhou, and Hui Xiong. [n. d.]. Irregular Multivariate Time Series Forecasting: A Transformable Patching Graph Neural Networks Approach. In *Forty-first International Conference on Machine Learning*.

[39] Xiang Zhang, Ziyuan Zhao, Theodoros Tsiligkaridis, and Marinka Zitnik. 2022. Self-supervised contrastive pre-training for time series via time-frequency consistency. *Advances in Neural Information Processing Systems* 35 (2022), 3988–4003.

[40] Yunhao Zhang and Junchi Yan. 2022. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*.

[41] Feng-Ming Zhao, De-Xin Gao, Yuan-Ming Cheng, and Qing Yang. 2024. Application of state of health estimation and remaining useful life prediction for lithium-ion batteries based on AT-CNN-BiLSTM. *Scientific Reports* 14, 1 (2024), 29026.

[42] Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. 2023. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems* 36 (2023), 43322–43355.