Surrogate Model for Reconfigurable Battery Packs Using Graph Neural Networks

Ali Irshayyid and Jun Chen, *Senior Member, IEEE*

***Abstract*—This article presents a novel graph neural network (GNN)-based surrogate model (*GNN-SM*) for predicting state evolution in reconfigurable battery packs. By leveraging graph-based representations of battery cell interconnections, the proposed approach addresses the unique challenge of estimating the imbalance in state-of-charge (SOC) and temperature of cells of a battery pack in dynamic battery configurations. Unlike conventional methods that focus on instantaneous state estimation, our *GNN-SM* predicts future SOC and temperature distributions by considering both current system state and switch configuration. The model architecture combines graph attention networks with pooling operations to effectively capture cell-to-cell interactions and battery pack-level dynamics. Numerical results under constant current and constant power discharge conditions demonstrate that *GNN-SM*significantly outperforms baseline feedforward neural network (FNN) and FNN-attention models, achieving up to 73.7% reduction in root-mean-square error for temperature imbalance prediction and 46% reduction for SOC imbalance prediction. Furthermore, the model provides a 1629-fold speedup over high fidelity physics-based simulator while maintaining mean absolute percentage errors below 2% for temperature and 8% for SOC predictions. The scalability of *GNN-SM* is further validated on a 100-cell reconfigurable battery pack, where the proposed approach achieves high accuracy despite being trained on an extremely small fraction of all possible configurations. Finally, robustness analysis under sensor noise conditions demonstrates that *GNN-SM* maintains reliable predictions even under high noise levels.**

***Index Terms*—Attention networks, cell imbalance, graph neural network (GNN), reconfigurable battery packs, state-of-charge (SOC).**

Received 22 January 2026; revised 12 March 2026; accepted 20 March 2026. Date of publication 13 April 2026; date of current version 6 July 2026. This work was supported by National Science Foundation under Award #2237317 and Award #2430374. An earlier version of this paper was presented in part at the 2025 IEEE Conference on Control Technology and Applications, San Diego, CA, USA, August 25–27, 2025, [DOI: 10.1109/CCTA53793.2025.11151387]. This paper extends the conference version [?] with expanded literature review, additional implementation details, and new numerical results on constant power discharging scenario, real-time implementability assessment, and scalability analysis. Paper no. TII-26-0740. *(Corresponding author: Jun Chen.)*

The authors are with the Department of Electrical and Computer Engineering, Oakland University, Rochester, MI 48309 USA (e-mail: aliirshayyid@oakland.edu; junchen@oakland.edu).

Digital Object Identifier 10.1109/TII.2026.3676937

I. Introduction

THE increasing adoption of electric vehicles (EVs) [1], [2], the integration of renewable energy sources into smart grids [3], [4], [5], [6], and the growing demand for reliable backup power systems in various applications from mobile devices to electric planes all point to the critical role of battery packs in modern technologies [7]. Lithium-ion battery packs in EVs consist of hundreds of cells arranged in series and/or parallel configurations to meet the high voltage and energy requirements [8]. However, these complex battery systems face challenges in their design and management. A key challenge is cell imbalance, which arises from variations in battery cell characteristics due to manufacturing inconsistencies, uneven aging, and difference in operation conditions [9], [10]. This can lead to state-of-charge (SOC) and temperature imbalance, which can result in reduced battery pack lifespan, safety issues such as overheating and thermal runaway, and diminished energy efficiency [11], [12], [13], [14], [15].

While conventional battery management systems employ techniques like cell balancing to mitigate these issues, their effectiveness is often limited by the static nature of cell interconnections. To address these inherent limitations of fixed cell topologies, reconfigurable battery packs, enabled by networks of controllable switches, offer a promising solution [16], [17], [18], [19]. Consisting of battery cells equipped with switching circuits, reconfigurable battery packs allow dynamic reconfiguration of cell connections in series, parallel, or bypass configurations. Such a reconfigurable architecture enables flexible control over cell interconnections, allowing battery packs to be dynamically configured in different modes such as all-serial, all-parallel, or hybrid (serial–parallel-bypass) arrangements [20]. By dynamically altering the connections between battery cells, reconfigurable battery systems can improve cell balancing, enhance lifespan and safety, and optimize energy efficiency [21], [22]. For example, it is shown in [17] and [23] that reconfigurable battery packs for cell balancing can achieve up to 22.4% longer battery runtime.

The performance benefits of reconfigurable battery packs have motivated more research into optimal reconfiguration strategies [24], [25], [26]. Han et al. [24] investigated the problem of minimizing battery charge equalization (BCE) time through cell and module reconfiguration. The study reveals a fundamental computational challenge that the number of possible configurations scales exponentially with the increase of the number of cells/modules in the reconfigurable battery pack. While their proposed algorithms achieve near-optimal BCE times with significantly reduced computation compared to exhaustive search, effective real-time implementation requires a fast and accurate surrogate model capable of rapidly evaluating

how candidate configurations will affect cell state evolution. The study in [25] showed that reconfiguration can substantially improve energy capacity delivery by accounting for cell state-of-health (SOH). In series-connected strings, deliverable capacity is constrained by the weakest cell. Their SOH-aware algorithm minimizes the bottleneck effect of weak cells by grouping cells with similar SOH and improves the capacity delivery by 10%–60%. Collectively, the authors in [24] and [25] showed that balancing both speed and capacity requires models that can predict how different topologies affect cell states over time.

Beyond performance optimization, reconfigurable battery packs must satisfy safety-critical hardware constraints during topology transitions. Han et al. [26] analyzed the maximum switch current encountered when reconfiguring battery connections during series-to-parallel transitions. When cells with different terminal voltages are suddenly connected in parallel, substantial transient currents flow through the switches to equalize the voltages. Their study shows that the larger the SOC imbalance is the larger voltage spread, which yields higher switch current spikes. This relationship reveals an important connection between state prediction and hardware safety. While instantaneous safety verification can use currently measured cell states, predictive safety assessment requires a topology-aware model capable of forecasting how cell states will evolve under a given configuration.

To address this computational challenge, data-driven surrogate models offer a promising alternative to physics-based simulation. Attempts have been made in literature to address this issue. For example, the authors in [17] and [23] investigated several supervised learning approaches to predict optimal topology in a reconfigurable battery pack. To generate training data, a multiobjective optimization problem is formulated for the topology selection, with objectives to maximize the sum of all cell SOC values and to minimize the range of SOC values across cells. By solving the multiobjective optimization problem offline, the training data is obtained that is used to train various machine learning models including FNN, support vector machines (SVM) [27], adaptive boost (AdaBoost) [28], k-Nearest Neighbors (KNN) [29], and Random Forest (RF) [30] to predict the best topology for the next control period. In particular, the FNN model achieved the best performance with 72% testing accuracy. While the authors in [17] and [23] demonstrated the potential for modeling battery pack reconfiguration, flattening topology into feature vectors and relying on exhaustive simulation makes these approaches fundamentally limited in both scalability and generalization. First of all, exhaustive simulations, which enumerate all possible transitions between topology pairs, are required for training data generation. Furthermore, multiple simulations under different conditions are needed for each configuration to capture the patterns of SOC changes. Consequently, this approach developed by authors in [17] and [23] became computationally intractable when the number of cells increased, since the number of possible topologies grew exponentially with pack size.

To scale up for the large number of cells that commonly exist in EVs, we develop a graph neural network (GNN)-based surrogate model (*GNN-SM*) for reconfigurable battery packs to predict future variations of SOC and cell temperatures. Unlike FNNs that flatten topology into fixed-length vectors and discard physical connectivity, GNNs are specifically designed for graph-structured data [31], [32], [33], making them a natural fit for battery packs where cells and switches form an inherently graph-structured representation. Crucially, *GNN-SM* learn local cell-to-cell interaction patterns through message passing rather than memorizing global configuration mappings, enabling generalization to unseen topologies and scaling to packs with hundreds of cells without architectural changes. Recent works have applied GNNs specifically within battery management systems. For example, graph attention networks (GATs) have been employed for SOH estimation [34], graph convolutional learning has been used to capture degradation patterns from partial discharge data [35], and GNN-based frameworks have been developed for voltage fault diagnosis in EV battery packs [36]. Related advances in other networked dynamical systems have also highlighted the effectiveness of graph-based spatio-temporal learning [37], [38]. However, these studies do not consider the prediction of state evolution under dynamically reconfigurable topologies, where the graph structure itself changes with each switching decision.

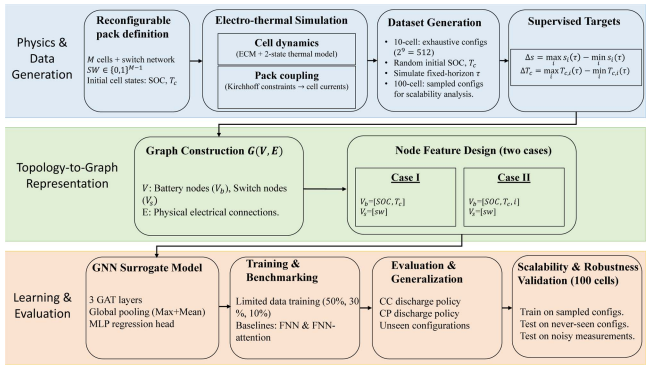

Fig. 1. Workflow of the proposed *GNN-SM* framework for reconfigurable battery packs.

The proposed GNN architecture naturally captures the physical connections and operational dynamics of reconfigurable battery packs by representing both battery cells and switches as nodes in a graph, with edges representing their physical connections. The battery cell nodes contain state information such as SOC and temperature, while switch nodes encode the mode of connections between two adjacent cells. By processing this graph-structured data through multiple GNN layers, our model learns to aggregate information from neighboring nodes and predict how different switch configurations affect the evolution of cell states. An overview of the proposed end-to-end surrogate modeling pipeline is shown in Fig. 1.

The *GNN-SM* is trained using simulation data generated from a detailed experimentally validated electro-chemical-thermal battery model that incorporates both electrical equivalent circuits and thermal dynamics [39], capturing the complex interactions between battery cells in various reconfigurable topologies. Numerical results demonstrate that, in all test conditions, the proposed *GNN-SM* significantly outperforms traditional FNNs and FNN-attention models in terms of estimation accuracy. This predictive capability is particularly valuable for reconfigurable battery systems, where understanding the future impact of different switching configurations can enable more effective reconfiguration strategies and thermal management. The key contributions of this article can be summarized as follows.

1) Unlike existing approaches that represent battery pack configurations as flattened feature vectors, [17], [23], a

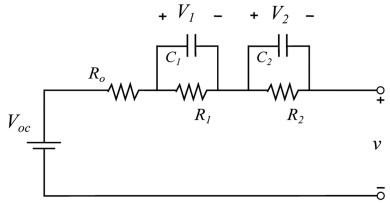

Fig. 2. ECM of a single cell.

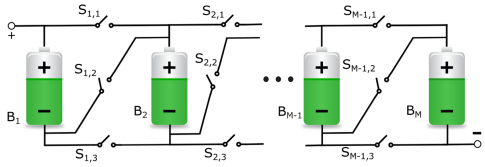

Fig. 3. Reconfigurable battery pack consisting of M cells.

novel graph representation is developed to model reconfigurable battery packs by treating both cells and switches as heterogeneous nodes and the electrical coupling as edges that capture the physical connections.

2) A key bottleneck in reconfigurable battery pack modeling is that the number of possible topologies grows as $2^{(M-1)}$ with pack size M, making exhaustive data collection intractable. Our attention-driven GNN architecture leverages message passing to learn transferable patterns of cell-to-cell interactions, enabling accurate predictions even when trained on an extremely small fraction (as low as $3.5 \times 10^{-25}\%$) of total possible configurations.
3) The proposed *GNN-SM* achieves 1629 times speedup over physics-based simulation while maintaining prediction errors below 2% for temperature and 8% for SOC imbalance. The proposed model achieved up to a 73.7% reduction in RMSE for temperature and a 46% reduction for SOC imbalance prediction.
4) To test scalability of the proposed surrogate model, the model is further evaluated on a large-scale 100-cell reconfigurable battery pack.
5) Model performance is examined under both constant current (CC) and constant power (CP) discharge policies to ensure robustness under diverse operating conditions.
6) To validate the proposed model performance for real-world applicability, we assess the robustness of the model to sensor measurement noise. The model maintains reliable predictions even under high noise conditions on completely unseen 100-cell configurations.

The rest of this article is organized as follows. Section II introduces the battery pack modeling methodology based on integrated electro-chemical-thermal modeling. Section III explains the graph representation for the reconfigurable battery pack system. Section IV elaborates on the architecture of the proposed *GNN-SM*, while Section V describes the experimental configuration and analyzes the numerical results. Lastly, Section VI concludes this article and outlines potential avenues for future research.

II. Dynamics of Battery Packs

A. Cell Dynamics

The cell model adopted in this work is based on the equivalent circuit model (ECM) [39], [40], [41], which combines an electrical circuit with a two-state thermal model to capture both the electrical and thermal dynamics of lithium-ion batteries (see Fig. 2). The electrical behavior is characterized by two RC pairs and a series resistance, as follows:

$$\dot{V}_1 = -\frac{V_1}{R_1 C_1} + \frac{I}{C_1} \tag{1a}$$

$$\dot{V}_2 = -\frac{V_2}{R_2 C_2} + \frac{I}{C_2} \tag{1b}$$

$$v = V_{OC} - V_1 - V_2 - I R_o \tag{1c}$$

where v is the terminal voltage, V_{OC} is the open-circuit voltage, and I is the current (positive for discharge, negative for charge). The RC pairs are characterized by voltages V_1, V_2, resistance R_1, R_2, and capacitance C_1, C_2, with R_o representing the series resistance. The cell's SOC is governed by

$$\dot{s} = -\frac{\eta}{3600 C_n} I \tag{2}$$

where s is the cell SOC, η is the coulombic efficiency and C_n is the nominal capacity of the cell in Amp-Hour.

The thermal behavior is modeled using core and surface temperatures

$$C_c \dot{T}_c = Q + \frac{T_s - T_c}{R_c} \tag{3a}$$

$$C_s \dot{T}_s = \frac{T_c - T_s}{R_c} + \frac{T_f - T_s}{R_u} \tag{3b}$$

where T_c is the core temperature, T_s is the surface temperature, T_f is the ambient temperature, C_c and C_s are the heat capacities of the core and surface, respectively, R_c is the conduction resistance between T_c and T_s, while R_u is the convection resistance between T_f and T_s. The heat generation Q is given by

$$Q = I(V_{oc} - v) - I \frac{T_s + T_c}{2} \frac{dV_{oc}}{dT}. \tag{4}$$

All model parameters (V_{oc}, R_o, R_1, R_2, C_1, C_2) are functions of SOC s and temperatures T_c and T_s, and can be expressed as

$$\sigma = f_\sigma(s, T_c, T_s) \tag{5}$$

where $\sigma = \{V_{oc}, R_o, R_1, R_2, C_1, C_2\}$. In this article, parameters in [39], which have been experimentally validated, are adopted for a nominal battery cell.

B. Battery Reconfiguration

Individual cell models are integrated into a reconfigurable battery pack consisting of M cells interconnected through a network of switches. Different switch architectures are reviewed in [42], each providing a different compromise between system flexibility and complexity. As shown in Fig. 3, a three-switch-per-connection architecture is adopted in this article, where switches can be either open or closed, enabling dynamic formation of series and parallel connections between adjacent

cells. This architecture provides sufficient degrees of freedom for topology reconfiguration while avoiding the limited flexibility of two-switch designs or the excessive complexity of four-switch arrangements. The resultant battery pack architecture enables dynamic adaptation of cell interconnections during operation.

Formally, let $\mathcal{B}$ be a reconfigurable battery pack with M cells $(B_1, B_2, \ldots, B_M \in \mathcal{M})$ connected through a network of switches as shown in Fig. 3, where each cell B_i has three switches $(S_{i,1}, S_{i,2}, S_{i,3})$ controlling its configuration. In other words, the switch configuration for cells B_i, $i = 1, \ldots, M-1$ determines its connection state sw_i with its adjacent cell B_{i+1}, with the following notation:

$$sw_i = \begin{cases} 1, & \text{for series connection between } B_i \text{ and } B_{i+1} \\ 0, & \text{for parallel connection between } B_i \text{ and } B_{i+1}. \end{cases}$$

For example, when switch $S_{i,2}$ is connected and both switches $S_{i,1}$ and $S_{i,3}$ are disconnected, cell B_i would be in series connection with cell B_{i+1} and so $sw_i = 1$. On the other hand, when switch $S_{i,2}$ is disconnected and both switches $S_{i,1}$ and $S_{i,3}$ are connected, cell B_i would be in parallel connection with cell B_{i+1} and so $sw_i = 0$. Note that other switch status combinations are prohibited to prevent short circuit or open circuit. Note also that sw_M does not necessarily need to be defined.

Finally, the dynamics of each cell B_i is given by (1)–(5). Given a prediction horizon τ, we define the SOC and temperature imbalance across cells as

$$\Delta s = \max_{1\le i\le M} s_i(\tau) - \min_{1\le i\le M} s_i(\tau) \tag{6a}$$

$$\Delta T_c = \max_{1\le i\le M} T_{c,i}(\tau) - \min_{1\le i\le M} T_{c,i}(\tau). \tag{6b}$$

Note that here the temperature imbalance is defined over the core temperature T_c only. However, the work presented in this article can be straightforwardly extended to the case of surface temperature (T_s) imbalance.

Remark 1: The simulation of the reconfigurable pack model described here requires significant computation, especially for embedded control systems, as it involves $5M$ coupled differential equations for a pack with M cells, representing three electrical states (V_1, V_2, s) and two thermal states (T_c, T_s) per cell. These differential equations must be integrated simultaneously, along with additional algebraic equations to calculate current, terminal voltage and heat generation for each cell (See Section II-C below). Such a computational complexity makes real-time prediction and control challenging for large reconfigurable battery packs with hundreds of cells. This motivates the need for computationally efficient surrogate models that can quickly predict state evolution while maintaining acceptable accuracy for real-time control applications.

C. Dynamics of Battery Packs

The electrical behavior of a reconfigurable battery pack is governed by Kirchhoff's current and voltage laws, which must be satisfied across all admissible topological configurations. Given a switch configuration vector $\mathbf{sw} = [sw_1, sw_2, \ldots, sw_{M-1}]$ and a total pack current demand I_p, the individual cell currents $\mathbf{I} = [I_1, I_2, \ldots, I_M]^T$ are obtained by solving a linear system that enforces the relevant circuit constraints. The current distribution problem is formulated as

$$\mathbf{AI} = \mathbf{C} \tag{7}$$

where $\mathbf{A} \in \mathbb{R}^{M\times M}$ is the topology-dependent coefficient matrix and $\mathbf{C} \in \mathbb{R}^M$ is the constraint vector. The matrix $\mathbf{A}$ encodes three categories of circuit relationships. First, for cells connected in series, rows of $\mathbf{A}$ enforce uniform current flow by imposing the condition $I_i = I_p$. Second, for cells connected in parallel, Kirchhoff's current law requires that the total current entering the parallel group equals the sum of the individual branch currents. This yields constraints of the form $\sum_{j\in\mathcal{P}} I_j = I_p$, where $\mathcal{P}$ denotes the set of parallel-connected cells within a parallel group. Third, voltage consistency is enforced across all cells within a parallel group. Specifically, for any two parallel-connected cells i and j, the following condition is imposed:

$$R_{o,i}I_i - R_{o,j}I_j = \bar{V}_{\mathrm{OC},i} - \bar{V}_{\mathrm{OC},j} \tag{8}$$

where $R_{o,i}$ and $\bar{V}_{\mathrm{OC},i} = V_{\mathrm{OC},i} - V_{1,i} - V_{2,i}$ denote the internal resistance and "modified" open-circuit voltage of cell i, respectively. The right-hand side vector $\mathbf{C}$ includes the corresponding constraint values: I_p for current flow and conservation constraints, and $V_{\mathrm{OC},i} - V_{\mathrm{OC},j}$ for voltage consistency constraints. Solving the resulting system (7) yields the current distribution vector $\mathbf{I}$, which is then used to update the battery states according to (1)–(5).

Example 1: For the pack shown in Fig. 4(a) with its configuration defined as $\mathbf{sw} = [1, 1, 2, 1, 1, 2, 2, 1, 1]$, the steady-state current distribution (7) becomes the following: (9) shown at the bottom of this page.

D. Constant Power Formulation

System (7) requires the pack current is known. When the battery pack is operated under constant power mode, pack

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & R_{o,3} & -R_{o,4} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & R_{o,6} & -R_{o,7} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & R_{o,7} & -R_{o,8} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \\ I_7 \\ I_8 \\ I_9 \\ I_{10} \end{bmatrix} = \begin{bmatrix} I_p \\ I_p \\ I_p \\ \bar{V}_{OC,3} - \bar{V}_{OC,4} \\ I_p \\ I_p \\ \bar{V}_{OC,6} - \bar{V}_{OC,7} \\ \bar{V}_{OC,7} - \bar{V}_{OC,8} \\ I_p \\ I_p \end{bmatrix}. \tag{9}$$

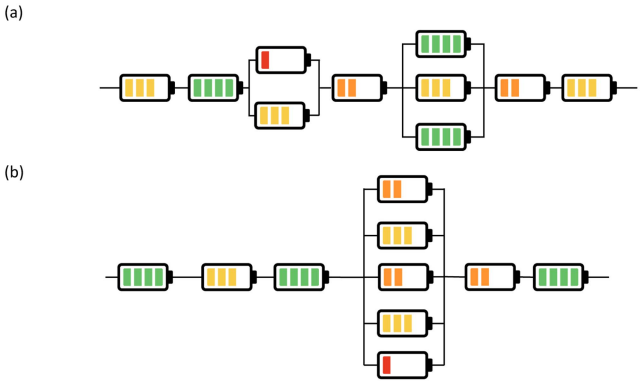

Fig. 4. Examples of 10-cells battery pack reconfiguration.

current needs to be calculated as the pack terminal voltage is time varying. Furthermore, for a reconfigurable battery pack, the change of cell topology further complicates the calculation of current for each cell. We first note that the solution of (7) can be represented as

$$\mathbf{I} = \mathbf{A}^{-1}\underbrace{\begin{bmatrix} 0 \\ c_2(1-sw_1) \\ \vdots \\ c_M(1-sw_{M-1}) \end{bmatrix}}_{\mathbf{a}} + \mathbf{A}^{-1}\underbrace{\begin{bmatrix} 1 \\ sw_1 \\ \vdots \\ sw_{M-1} \end{bmatrix}}_{\mathbf{b}} I_p$$

$$= \mathbf{a} + \mathbf{b} \cdot I_p \tag{10}$$

where $c_i = \bar{V}_{OC,i-1} - \bar{V}_{OC,i}$. For each cell i, the delivered power can be defined as

$$P_i = v_i I_i = (\bar{V}_{\mathrm{OC},i} - I_i R_i) I_i = \bar{V}_{\mathrm{OC},i} I_i - R_k I_k^2 \tag{11}$$

where v_i, p_i are the terminal voltage and power of cell i, respectively. The total pack power P_T is then

$$P_T = P_1 + P_2 + \ldots + P_M$$

$$[6pt] = \mathbf{V_{OC}}^T \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_M \end{bmatrix} - \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_M \end{bmatrix}^T \begin{bmatrix} R_{o,1} & & \\ & \ddots & \\ & & R_{o,M} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_M \end{bmatrix}$$

$$= \mathbf{V_{OC}}^T \mathbf{I} - \mathbf{I}^T \mathbf{R} \mathbf{I}$$

where $\mathbf{V_{OC}} = [\bar{V}_{\mathrm{OC},1}, \bar{V}_{\mathrm{OC},2}, \ldots, \bar{V}_{\mathrm{OC},M}]$ and $\mathbf{R} = \mathrm{diag}(R_{o,1}, R_{o,2}, \ldots, R_{o,M})$. Substituting $\mathbf{I} = \mathbf{a} + \mathbf{b} \cdot I_p$ yields

$$P_T = \mathbf{V_{OC}}^T(\mathbf{a} + \mathbf{b}I_p) - (\mathbf{a} + \mathbf{b}I_p)^T \mathbf{R}(\mathbf{a} + \mathbf{b}I_p)$$

$$= \mathbf{V_{OC}}^T(\mathbf{a} + \mathbf{b}I_p) - (\mathbf{b}^T\mathbf{R}\mathbf{b}I_p^2 + 2\mathbf{a}^T\mathbf{R}\mathbf{b}I_p + \mathbf{a}^T\mathbf{R}\mathbf{a}).$$

Rearranging terms yields

$$\mathbf{b}^T\mathbf{R}\mathbf{b}I_p^2 + (2\mathbf{a}^T\mathbf{R} - \mathbf{V_{OC}}^T)\mathbf{b}I_p + P_T + (\mathbf{a}^T\mathbf{R} - \mathbf{V_{OC}}^T)\mathbf{a} = 0 \tag{12}$$

which is a quadratic equation over the scalar variable I_p. Therefore, given requested pack power P_T, solving the quadratic equation (12) for the positive real root yields pack current I_p. Substituting I_P in (10) provides the individual currents of all cells within the pack that is required for the reconfigurable battery pack to deliver the total power P_T.

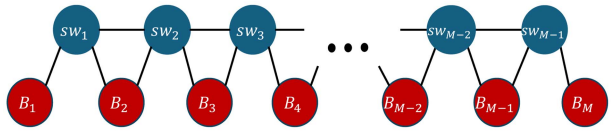

Fig. 5. Graph representation of a reconfigurable battery pack with M cells.

III. Graph Representation of Cell Topologies

Reconfigurable battery packs can be naturally represented as a graph structure that captures both the physical connectivity and the operational states of the system. As illustrated in Fig. 5, we model switches and battery cells as different types of nodes, where their interactions and physical connections are represented by edges. The switches, which control the series-parallel configurations, are represented as one type of node carrying binary state information. The battery cells form another type of nodes, containing their respective state variables such as s and T_c. The physical layout of battery packs determines the edge connections. For the reconfigurable battery pack shown in Fig. 3, edges exist between adjacent switches and between switches and their connected battery cells. The graph-based approach effectively captures the dynamic reconfigurability of the battery pack and allows us to model how the cell topology influences the state evolution of the battery pack.

Formally, we define our graph structure as follows. Let $G = (V, E)$ be an undirected graph representing the battery pack structure, where V is the set of nodes and E is the set of edges. The node set V consists of two disjoint subsets: $V = V_s \cup V_b$, where $V_s = \{v_1, v_2, \ldots, v_{M-1}\}$ represents the set of switch nodes and $V_b = \{v_1, v_2, \ldots, v_M\}$ represents the set of battery cell nodes. Each node $v_i \in V$ carries features such as switch status for $v_i \in V_s$ and SOC for $v_i \in V_b$. We denote the feature for node v_i as x_{v_i}. An undirected graph representation is adopted because the current direction is state-dependent and implicitly reflected in node features, while thermal propagation is inherently bidirectional. Thus, undirected message-passing naturally aligns with electro-thermal interactions in reconfigurable battery packs. More details of the features will be discussed in Section V-A. The edge set E represents physical connections, i.e.,

$$E = \{(i, j) | \text{ node } i \text{ is physically connected to node } j\}.$$

The adjacency matrix $\bar{A} = [\bar{A}_{ij}] \in \mathbb{R}^{|V|\times|V|}$ is defined as

$$\bar{A}_{ij} = \begin{cases} 1, & \text{if } (i,j) \in E \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

IV. GNN-Based Surrogate Model

GNNs have emerged as powerful architectures for modeling complex systems with dynamic relationships [43], [44]. For reconfigurable battery pack modeling, several GNN variants such as graph convolutional networks (GCNs) [43] and GATs [44] can be considered. While GCNs offer a straightforward approach

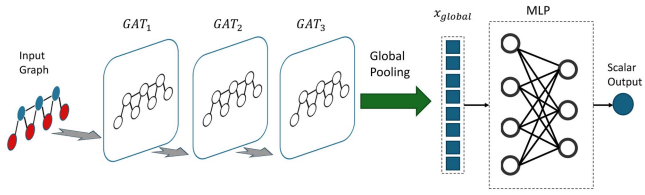

Fig. 6. *GNN-SM* architecture used in this article. The model consists of three sequential GAT layers that process the input graph representation of the battery pack. Each GAT layer performs message passing between nodes to capture cell-to-cell interactions. The processed features are then aggregated through global pooling operations (max and mean pooling) before being fed into an MLP to produce the final scalar output.

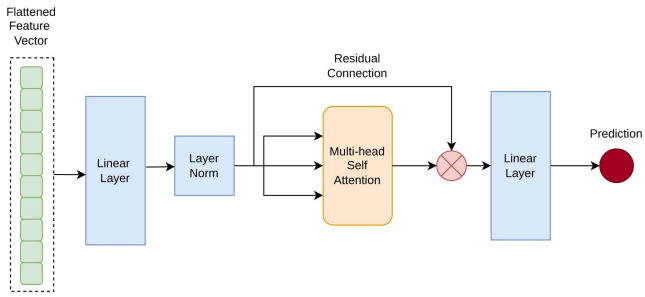

Fig. 7. Architecture of the FNN-attention model used for benchmarking.

to aggregate neighbor information using fixed weights based on graph structure, GATs dynamically assign different levels of importance to neighboring nodes using an attention mechanism. This property is particularly crucial for reconfigurable battery packs, where the influence of one cell on another can vary significantly based on their electrical connection state. Therefore, GATs are used in this article. In the rest of this article, we use "GNNs" and "GATs" interchangeably.

Given SOC variation Δs and temperature variation ΔT_c as defined in (6), the goal of the *GNN-SM* is to learn and predict Δs and ΔT_c while accounting for the dynamic reconfigurable topology. The proposed GAT architecture is shown in Fig. 6, which consists of three GAT layers that process both the battery cell and switch node features while leveraging the graph structure defined in Section III. The input of the first GAT layer is the original nodes features, $\mathbf{x} = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_{|V|}\}, \vec{x}_i \in \mathbb{R}^F$, where F is the number of features in each node. After the message passing process, new node features are produced as the output of this layer, denoted as $\mathbf{x}' = \{\vec{x}'_1, \vec{x}'_2, \ldots, \vec{x}'_{|V|}\}, \vec{x}'_i \in \mathbb{R}^{F'}$, where F' represents the embedding size.

In each GAT layer, the first step is to transform the features of each node into a high-level representation parametrized by $\mathbf{W} \in \mathbb{R}^{F' \times F}$. After that, a self-attention operation is applied at every node as follows:

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\vec{a}_1^T \mathbf{W}\vec{x}_i + \vec{a}_2^T \mathbf{W}\vec{x}_j\right)\right)}{\sum_{k \in \mathcal{N}_i \cup i} \exp\left(\text{LeakyReLU}\left(\vec{a}_1^T \mathbf{W}\vec{x}_i + \vec{a}_2^T \mathbf{W}\vec{x}_k\right)\right)}$$

where α_{ij} is the normalized attention coefficients representing the importance of node j's features to node i, $\mathcal{N}_i$ is a set of i's node neighbors, and $\vec{a} \in \mathbb{R}^{2F'}$ is the attention learnable weight vector. The final output features for each node are determined by computing a linear combination of the features using the normalized attention coefficients as $\vec{x}'_i = \tanh(\sum_{j \in \mathcal{N}_i \cup i} \alpha_{ij} \mathbf{W}\vec{x}_j)$, where $\tanh$ is used to add nonlinearity.

After the GAT layers transform the node features, both global max pooling and global mean pooling operations are employed to aggregate information across all nodes in the graph, as follows:

$$\vec{x}_{\max} = \text{MaxPool}(\vec{x}'_1, \vec{x}'_2, \ldots, \vec{x}'_{|V|}) \tag{14}$$

$$\vec{x}_{mean} = \text{MeanPool}(\vec{x}'_1, \vec{x}'_2, \ldots, \vec{x}'_{|V|}) \tag{15}$$

where x'_i represents the final node features from the last GAT layer. The pooled features are concatenated to form a fixed-size representation $\vec{x}_{global} = [\vec{x}^T_{\max}, \vec{x}^T_{mean}]^T$, which is then processed through FNN with ReLU activation

$$h = \text{ReLU}(\vec{W}_f \, \vec{x}_{global} + b_f) \tag{16}$$

where $\vec{W}_f$ and b_f are the learnable weights and bias of the multilayer perceptron (MLP) layer. To prevent overfitting, a dropout layer with rate of 0.2 is applied.

Remark 2: The proposed GNN architecture is implemented using PyTorch Geometric [45]. The first GAT layer projects the input features ($F = 2$ for Case I, $F = 3$ for Case II) to 24 dimensions, and the subsequent two GAT layers maintain the 24-D representation. Tanh activation is applied after each GAT layer. Layer normalization is applied after the final GAT layer to stabilize training. After the GAT layers, global mean pooling and global max pooling operations aggregate the node features to a 192-D vector. The MLP regression head consists of two fully connected layers with a dropout layer. The first layer reduces the dimension from 192 to 24 with ReLU activation, and the final layer outputs a scalar prediction. The model is trained using mean squared error (MSE) loss with the Adam optimizer and a learning rate of 0.0003 with a batch size of 1024. Hyperparameters were selected based on validation MSE. The hyperparameters search ranges are as follows. The learning rate: $\{0.001, 0.0005, 0.0003, 0.0001\}$, the number of GAT layers: $\{2, 3, 4\}$, the embedding dimension F' $\{24, 32, 64, 96, 128\}$, the number of attention heads: $\{2, 4, 8\}$, the MLP hidden dimension: $\{16, 24, 32\}$, and the dropout rate: $\{0.1, 0.2, 0.3\}$.

V. Numerical Results

A. Experimental Setup and Data Generation

To generate a comprehensive dataset for evaluating the *GNN-SM*, we simulate a 10-cell reconfigurable battery pack, where cells have different amp-hour capacities due to cell-to-cell variations. Given the series-parallel configuration options between adjacent cells, a 10-cell battery pack yields 512 (i.e., 2^9) unique configurations. For each configuration, we perform 10 different simulation runs, each with randomized initial conditions across all cells. The initial core temperature T_c of each cell is randomly sampled from the range [17.5 °C, 27.5 °C], while the initial SOC values are randomly assigned between 0.8 and 1.0, representing batteries in a high charge state. Each simulation episode runs for 500 s with a constant discharge current of 1.5 A, allowing sufficient time for thermal and SOC dynamics to evolve. At the end of each simulation, we record two key metrics: the final Δs and the final ΔT_c according to (6). This data generation approach resulted in 5120 total simulation runs (512 configurations with

TABLE I
EXPERIMENT PARAMETERS

Parameters	Value/Description
10-cell dataset	10 battery cells with 9 switches
Total configurations (10-cell)	2^9
# samples (Train/Test) splits (10-cell)	5120 (2560/1536, 1536/2150, 512/2560)
100-cell dataset	100 battery cells with 99 switches
Total configurations (100-cell)	2^{99}
# samples (Train/Test/Test*) (100-cell)	3000 (2000, 500, 500)
Simulation duration	500 s
Discharge Current (CC)	1.5 A
Discharge Power (CP)	30 W
Cell nominal capacity	2.3 Ah
Initial SOC range	[0.8, 1.0]
Initial T_c range	[17.5 °C, 27.5 °C]

TABLE II
MODEL PERFORMANCE COMPARISON UNDER CC DISCHARGE POLICY (CASE I)

Model	50% Train. Data		30% Train. Data		10% Train. Data	
	RMSE	%	RMSE	%	RMSE	%
ΔT_c						
FNN [17]	0.357	7.14	0.378	7.61	0.440	8.75
FNN-att. [46]	0.393	8.19	0.459	9.30	0.453	9.20
GNN-SM (Proposed)	**0.094**	**1.90**	**0.109**	**2.20**	**0.223**	**4.40**
Δs						
FNN [17]	0.026	13.37	0.028	14.00	0.033	16.56
FNN-att. [46]	0.026	13.42	0.028	14.48	0.034	17.77
GNN-SM (Proposed)	**0.020**	**10.46**	**0.020**	**10.80**	**0.023**	**12.00**

Bold indicated the best performance of the corresponding column.

Algorithm 1: *GNN-SM* Training for Reconfigurable Battery Packs.

Initialize: Topologies $\mathcal{SW}$, horizon τ, discharge policy $\mathcal{P}$, Electro-chemical-thermal simulator Sim$(\cdot)$, feature case κ, GNN f_θ
Output: Trained parameters θ

Dataset $\mathcal{D} \leftarrow \emptyset$
for all sw $\in \mathcal{SW}$ **do**
  $s_i(0) \sim \mathcal{U}(0.8, 1.0)$ for all $i = 1, \ldots, M$
  $T_{c,i}(0) \sim \mathcal{U}(17.5\,^\circ\text{C}, 27.5\,^\circ\text{C})$ for all $i = 1, \ldots, M$
  $\{\mathbf{s}(\tau), \mathbf{T_c}(\tau)\} \leftarrow \text{Sim}(\mathbf{s}(0), \mathbf{T_c}(0), \mathbf{sw}, \mathcal{P}, \tau)$
  $y \leftarrow [\Delta s, \Delta T_c]$
  For all $i = 1, \ldots, M$:
  $\mathbf{x}(v_{b,i}) \leftarrow \begin{cases} [s_i(0),\ T_{c,i}(0)], & \kappa = \text{I} \\ [s_i(0),\ T_{c,i}(0),\ I_i(0)], & \kappa = \text{II} \end{cases}$
  For all $j = 1, \ldots, M-1$: $\mathbf{x}(v_{s,j}) \leftarrow [sw_j]$
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{(G, \mathbf{x}, y)\}$
end for
for epoch $n = 1$ to N **do**
  for all mini-batches $\mathcal{B} \subset \mathcal{D}_{train}$ **do**
    $\hat{y} \leftarrow f_\theta(G, \mathbf{x})$ for all $(G, \mathbf{x}, y) \in \mathcal{B}$
    $L \leftarrow \frac{1}{|\mathcal{B}|} \sum_{(G,\mathbf{x},y)\in\mathcal{B}} \|y - \hat{y}\|^2$
    Update θ to minimize L
  end for
end for

ten runs each). The training pipeline follows Algorithm 1, and the used simulation parameters are listed in Table I.

Two *GNN-SM*s are trained, each with different features for the battery cell nodes. In Case I, only the battery states such as s_i and $T_{c,i}$ are used as input features in the battery cell node, with s_i being used when the goal is to predict Δs and $T_{c,i}$ being used when the goal is to predict ΔT_c. In Case II, the discharge current of the individual battery cell is added as an additional input feature. In both scenarios, the switch status sw_i is used as an input feature for the switch node, as discussed in Section III. The final Δs and ΔT_c measurements serve as the target variables for both models.

For benchmarking, two baseline approaches are considered: an FNN consisting of four fully connected layers with ReLU activation function that processes battery cell states and switch configurations as a flattened vector input [17], and an FNN-attention model, proposed in literature for vehicle platooning control [46], which employs self-attention mechanism with four attention heads to capture cell-to-cell interactions. As shown in Fig. 7, the FNN-attention baseline model begins with a linear layer (feature extractor) that processes the flattened graph features (battery cell states and switch configurations) through a linear layer followed by layer normalization. This initial processing allows the network to extract relevant features from the input data independently. The processed features are then fed into the self-attention mechanism, comprising query, key, and value vectors, which feed into a multihead self-attention layer. The self-attention layer enables the network to dynamically weigh the importance of different battery cells' states and their interconnections. Following the attention mechanism, a residual connection combines the output of the attention layer with the processed features, allowing the network to preserve important individual cell information alongside the relational data. The final linear layer processes this combined information to predict the output.

For evaluation, both constant current and constant power discharge scenarios are considered. We start the simulation with 10 cells, and later scale it to a larger reconfigurable battery pack with 100 cells to test the scalability of the proposed *GNN-SM*.

B. Performance Evaluation With CC Discharge

Recall that the proposed *GNN-SM* is evaluated against two baseline approaches, FNN and an FNN-attention models, for predicting both Δs and ΔT_c. Our evaluation metrics include root-mean-square error (RMSE), which quantifies prediction accuracy with lower values indicating better performance, and mean absolute MAPE (MAPE, %), which provides a relative measure of prediction error as a percentage of the true value. The MAPE metric is particularly valuable for comparing prediction accuracy across different scales of measurements, such as between ΔT_c and Δs predictions. Please note that for brevity, we report MAPE in all tables as "%." The evaluation is conducted using different percentages of training data (50%, 30%, and 10%) to assess model robustness under limited data availability. For each training split, a validation set is constructed by withholding 20% of the training set size from the remaining (nontraining) data. Results are presented under two scenarios: (Case I) without individual cell currents as an input feature (see Table II) and (Case II) with individual cell currents as input feature for all models (see Table III).

For ΔT_c prediction, the proposed *GNN-SM* demonstrates superior performance across all training data configurations. As

TABLE III
MODEL PERFORMANCE COMPARISON UNDER CC DISCHARGE POLICY (CASE II)

Model	50% Train. Data		30% Train. Data		10% Train. Data	
	RMSE	%	RMSE	%	RMSE	%
ΔT_c						
FNN [17]	0.361	7.21	0.378	7.62	0.471	9.42
FNN-att. [46]	0.414	8.30	0.434	8.90	0.567	11.93
GNN-SM (Proposed)	**0.074**	**1.49**	**0.10**	**2.20**	**0.173**	**3.70**
Δs						
FNN [17]	0.028	14.08	0.03	15.54	0.033	17.00
FNN-att. [46]	0.03	16.14	0.0316	17.25	0.034	18.17
GNN-SM (Proposed)	**0.015**	**7.7**	**0.015**	**8.07**	**0.019**	**10.10**

Bold indicated the best performance of the corresponding column.

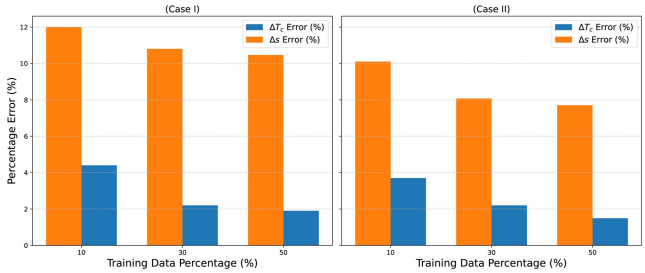

Fig. 8. Comparison of mean absolute MAPE for ΔT_c and Δs predictions by the proposed *GNN-SM*s with respect to training data percentages.

shown in Table II, *GNN-SM* achieved RMSE values of 0.094, 0.109, and 0.223 for 50%, 30%, and 10% training data, respectively, significantly outperforming both FNN and FNN-attention models. In the second experiment, the individual cell current is included as an additional input feature for both *GNN-SM* and baselines. The *GNN-SM*'s performance improved further, achieving RMSE values of 0.074, 0.1, and 0.173 across the three training data configurations. This improvement suggests that the GNN network effectively leverages the additional current information to enhance prediction accuracy.

For Δs prediction, the results follow a similar pattern, with the proposed *GNN-SM* demonstrating consistently superior performance. In Case I, the *GNN-SM* achieved RMSE values of 0.02, 0.02, and 0.023 for 50%, 30%, and 10% training data, respectively. These results represent a substantial improvement over both the FNN and FNN-attention models. Similarly, in the second experiment where the individual cell current is included as an input, the *GNN-SM* performance improved further, with RMSE values reducing to 0.015, 0.015, and 0.019 across the three training configurations. Notably, the *GNN-SM* architecture maintained robust performance even with limited training data (10%). This data efficiency comes from the fact that the *GNN-SM* learns transferable cell-to-cell interaction patterns at the neighborhood level rather than memorizing global configuration-specific mappings.

Fig. 8 provides a visual comparison of the proposed *GNN-SM*'s MAPEs for both ΔT_c and Δs predictions across different training data percentages. In Fig. 8 (Case I), we observe that Δs prediction errors show a gradual increase as training data decreases, ranging from 10.46% with 50% training data to 12% with 10% training data. The ΔT_c prediction errors follow a similar trend but with notably lower error rates, increasing from 1.9% to 4.4% as training data is reduced. With individual cell currents as input feature, Fig. 8 (Case II) demonstrates improved performance, particularly for Δs predictions, with errors decreasing to 7.7% with 50% training data. This improvement can be attributed to the addition of discharge current as an input feature, which either needs to be measured for each cell or needs to be calculated using the network topology [47] (hence requiring additional complexity for real-time implementation).

The performance advantage of the *GNN-SM* over baseline models is most evident in ΔT_c prediction, where the proposed *GNN-SM*s achieve significantly lower errors across both metrics—with RMSE values ranging from 0.094 to 0.223 and MAPE between 1.9% and 4.4%, compared to both FNN (RMSE: 0.357-0.44; MAPE: 7.14–8.75%) and FNN-attention models (RMSE: 0.393–0.453; MAPE: 8.19-9.20%) across all training conditions. The superior performance of the *GNN-SM* can be attributed to its graph-structured inductive bias. Unlike FNN, which flattens the topology into a fixed-length vector and discards the physical connectivity between cells, the *GNN-SM*'s message passing propagates state information along edges that directly reflect the series and parallel relationships between adjacent cells. This means the model's internal representations are grounded in the actual circuit structure, allowing it to reason about how a given switch configuration will redistribute current and affect state evolution across the pack. Furthermore, the addition of individual cell current as an input feature provides valuable information about the battery topology, enabling the model to better capture the behavior of the battery pack. However, as discussed earlier, it requires extra computation or sensors to obtain individual cell currents for real-time implementation.

C. Generalization to Unseen Configurations

While the previous results demonstrate strong performance across different training data percentages, the generalization to unseen configurations is not tested. In other words, in the previous evaluations, both training data and test data are randomly sampled from the entire dataset, giving the surrogate models the opportunity to train over all possible configurations. Though this is not an issue for a small battery pack, it may not be feasible to collect data for all configurations when the number M of cells increases.

In this section, we randomly select 51 configurations (10% of the total 512 possible configurations) and reserved them exclusively for testing, ensuring these configurations were never seen during training. From the remaining 4610 samples, we created three training scenarios using 50% (2305 data samples), 30% (1536 data samples), and 10% (461 data samples) of the data. It's important to note that when creating training dataset (for 50%, 30%, and 10%), data are sampled randomly (from the remaining 4610 samples) regardless of its configuration features. Tables IV and V present the models' performance on unseen configurations for Case I and Case II, respectively. While some performance degradation is observed when tested on completely unseen configurations, the *GNN-SM* maintains significantly better performance demonstrating that it has learned transferable local interaction rules rather than memorizing configuration-specific mappings.

TABLE IV
PREDICTIONS FOR UNSEEN CONFIGURATIONS UNDER CC DISCHARGE POLICY (CASE I)

Model	50% Train. Data		30% Train. Data		10% Train. Data	
	RMSE	%	RMSE	%	RMSE	%
ΔT_c						
FNN [17]	0.316	6.47	0.346	7	0.458	9.4
FNN-att. [46]	0.397	8.31	0.440	8.9	0.563	11.68
GNN-SM (Proposed)	**0.122**	**2.21**	**0.109**	**2.24**	**0.212**	**4.3**
Δs						
FNN [17]	0.026	14.28	0.028	14.43	0.034	17.83
FNN-att. [46]	0.026	12.98	0.026	14.16	0.037	18.92
GNN-SM (Proposed)	**0.02**	**11.25**	**0.022**	**12.42**	**0.022**	**11.20**

Bold indicated the best performance of the corresponding column.

TABLE V
GENERALIZATION FOR UNSEEN CONFIGURATIONS UNDER CC DISCHARGE POLICY (CASE II)

Model	50% Train. Data		30% Train. Data		10% Train. Data	
	RMSE	%	RMSE	%	RMSE	%
ΔT_c						
FNN [17]	0.311	6.18	0.368	7.29	0.489	9.81
FNN-att. [46]	0.374	7.49	0.447	8.67	0.565	11.18
GNN-SM (Proposed)	**0.1**	**2.02**	**0.109**	**2.24**	**0.151**	**3**
Δs						
FNN [17]	0.028	14.44	0.031	16.63	0.034	18.40
FNN-att. [46]	0.028	14.37	0.033	17.05	0.037	16.60
GNN-SM (Proposed)	**0.014**	**7.59**	**0.017**	**9.8**	**0.02**	**10.18**

Bold indicated the best performance of the corresponding column.

TABLE VI
MODEL PERFORMANCE COMPARISON UNDER CP DISCHARGE POLICY

Model	50% Train. Data		30% Train. Data		10% Train. Data	
	RMSE	%	RMSE	%	RMSE	%
ΔT_c						
FNN [17]	0.1287	0.96	0.251	1.25	0.305	1.77
FNN-att. [46]	0.121	0.87	0.16	1.22	0.17	1.3
GNN-SM (Proposed)	**0.103**	**0.8**	**0.115**	**0.799**	**0.107**	**0.818**
Δs						
FNN [17]	0.02	15.96	0.02	15.39	0.025	19.84
FNN-att. [46]	0.022	17.43	0.024	18.99	0.031	23.91
GNN-SM (Proposed)	**0.0063**	**4.6**	**0.0079**	**5.4**	**0.017**	**12.92**

Bold indicated the best performance of the corresponding column.

TABLE VII
PREDICTIONS FOR UNSEEN CONFIGURATIONS UNDER CP DISCHARGE POLICY

Model	50% Train. Data		30% Train. Data		10% Train. Data	
	RMSE	%	RMSE	%	RMSE	%
ΔT_c						
FNN [17]	0.144	1.07	0.265	1.56	0.33	2.2
FNN-att. [46]	0.13	0.88	0.154	1.212	0.166	1.31
GNN-SM (Proposed)	**0.106**	**0.8**	**0.104**	**0.822**	**0.139**	**1.05**
Δs						
FNN [17]	0.022	17.46	0.021	16.9	0.025	19.95
FNN-att. [46]	0.031	17.6	0.023	18.42	0.03	23.64
GNN-SM (Proposed)	**0.006**	**4.62**	**0.0065**	**4.89**	**0.0159**	**11.96**

Bold indicated the best performance of the corresponding column.

For Case I, without individual cell current as an input feature (see Table IV), the *GNN-SM* achieves remarkable ΔT_c prediction accuracy with RMSE values of 0.122, 0.109, and 0.212 across the three training scenarios, significantly outperforming both FNN and FNN-attention models. The MAPE values for ΔT_c prediction remain consistently low, ranging from 2.21% to 4.3%, while baseline models show errors up to 11.68%. For Case II with individual cell current (see Table V), the *GNN-SM*'s superior performance becomes even more evident. For ΔT_c prediction, the *GNN-SM* maintains excellent accuracy with RMSE values of 0.1, 0.109, and 0.151, while baseline models show considerably higher errors. Particularly noteworthy is the *GNN-SM*'s performance in Δs prediction in Case II, where it achieves remarkably low MAPE (7.59%, 9.8%, 10.18%) compared to the baseline models (14.44%–18.40%).

D. Performance Evaluation With CP Discharge

To evaluate the robustness of the proposed *GNN-SM* under more practical operating conditions, we conducted experiments using a CP discharge policy of 30 W. Unlike the CC scenarios, the CP discharge policy presents a more challenging test as individual cell currents vary dynamically based on the pack's reconfiguration and the cells' states. This dynamic current distribution makes CP discharge a more realistic and stringent evaluation, as it better represents real-world EV applications where the battery pack operates under varying current demands. As shown in Table VI, the proposed *GNN-SM* demonstrates great performance under the CP discharge policy across all training data settings. For temperature prediction ΔT_c, the *GNN-SM* achieves consistent RMSE of 0.103, 0.115, and 0.107 for 50%, 30%, and 10% training data respectively. The MAPE, in all cases, remains below 0.82%. In contrast, the FNN baseline shows significantly higher errors ranging from 0.96% to 1.77%, while the FNN-attention model shows errors between 0.87% to 1.3%. The *GNN-SM* performance is even better in SOC prediction, Δs, where it achieves RMSE values of 0.0063, 0.0079, and 0.017, with MAPEs of 4.6%, 5.4%, and 12.92%. The Δs results represents notable improvement over both baselines, which show MAPEs ranging from 15.39% to 19.84% and 17.43% to 23.91% for FNN and FNN-attention respectively.

The generalization capability of the *GNN-SM* to unseen configurations under CP discharge policy is shown in Table VII. The *GNN-SM* maintains robust performance when tested on completely unseen battery pack configurations, which is a similar behavior to previous experiments. For ΔT_c prediction, the *GNN-SM* achieves RMSE values of 0.106, 0.104, and 0.139 with MAPEs of 0.8%, 0.822%, and 1.05% across the three training scenarios. The baseline models show notably higher errors, with FNN reaching up to 2.2% and FNN-attention up to 1.31%. For Δs prediction on unseen configurations, the *GNN-SM* continues to demonstrate superior performance, achieving MAPEs of 4.62%, 4.89%, and 11.96%, while FNN and FNN-attention models show errors ranging from 19.95% to 23.64%, respectively.

The results clearly demonstrate that the *GNN-SM* sustains strong prediction performance even with limited training data, confirming that data efficiency is a key advantage for practical

TABLE VIII
MODEL ARCHITECTURE AND COMPUTATION COMPARISON

Model	Parameters	Dimensions	Training Time (50%)	Inference Time on CPU (ms)	Memory Size (KB)	Learning Rate
FNN [17]	25185	256, 64, 16, 1	∼27.45 min	0.01	103	0.00025
FNN-att. [46]	21009	Embedding = 52, heads=4	∼31.61 min	0.01	90	0.0001
GNN-SM	24337	F' = 24, heads = 4	∼87.77 min	5.5	103	0.0003

Note: All models were trained using the Adam optimizer with a batch size of 1024.

applications where collecting extensive training data across all possible configurations is expensive or time-consuming.

E. Comparison of Computational Requirements

This section discusses the architectural characteristics and computational requirements of the three models, as compared in Table VIII. The proposed *GNN-SM* has a moderate number of parameters (24 337) compared to the FNN (25 185) and FNN-attention (21,009) models. While the FNN uses a simple architecture with four layers of decreasing dimensions (256, 64, 16, 1), and the FNN-attention model employs an embedding size of 52 with 4 attention heads, the *GNN-SM* achieves superior performance using an embedding dimension F' of 24 and 4 attention heads. However, the *GNN-SM* does require longer training time (87.77 min for 50% training data) compared to the FNN (27.45 min) and FNN-attention model (31.61 min) when trained on a standard desktop computer with an NVIDIA RTX 3070 Ti GPU. All models were trained using the Adam optimizer with a batch size of 1024, with learning rates of 0.00025 (FNN), 0.0001 (FNN-attention), and 0.0003 (*GNN-SM*). At inference time on CPU, the two baselines are extremely lightweight (0.01 ms per sample), while the *GNN-SM* takes a little more time (55 ms) due to graph construction and message passing. Finally, memory usage remains small for all models: the FNN and *GNN-SM* require 103 KB, and the FNN-attention model is slightly more compact at 90 KB. Overall, although the *GNN-SM* introduces additional computational in training time and CPU inference latency, this cost is a reasonable tradeoff given its significantly improved prediction accuracy across both experiments. It is worth noting that the training cost is incurred offline only, whereas online deployment depends mainly on inference latency and memory usage.

F. Scalability Analysis

Since modern EVs typically employ battery packs consisting of hundreds of cells [8], this study conducts experiments on a significantly larger reconfigurable battery pack of 100 cells. The 100-cell reconfigurable battery pack presents greater complexity compared to the 10-cell reconfigurable battery pack studied in previous experiments, both in terms of the number of possible topologies and the cell-to-cell interactions that must be captured.

Exhaustively simulating a 100-cell battery pack ($2^{99} = 6.33 \times 10^{29}$ topologies) is computationally intractable, unlike a 10-cell pack ($2^9 = 512$), making a generalizable surrogate model essential. For training, we simulated 2000 unique topologies uniformly sampled from the full topology space, under randomized initial conditions. We then evaluated its performance across two 500-sample test sets: a standard test set drawn from the same 2000 training topologies but under new operating conditions, and a "Test*" set consisting of 500 topologies not included in the training set. This approach rigorously assesses the model's accuracy on both familiar and never-seen configurations.

TABLE IX
MODEL PERFORMANCE USING 100 CELLS RECONFIGURABLE BATTERY PACK

Model	RMSE	%	RMSE*	%*
ΔT_c				
FNN [17]	0.031	0.2129	1.486	7.923
FNN-att. [46]	0.052	0.381	1.369	6.983
GNN-SM (Proposed)	**0.026**	**0.165**	**0.857**	**4.53**
Δs				
FNN [17]	0.0026	0.839	0.0178	6.637
FNN-att. [46]	0.0075	2.7	0.015	5.607
GNN-SM (Proposed)	**0.0015**	**0.695**	**0.013**	**5.007**

Bold indicated the best performance of the corresponding column.

Table IX presents the results of both ΔT_c and Δs predictions of the proposed model in addition to the baseline models. For the standard test, the *GNN-SM* demonstrates exceptional performance. For temperature prediction (ΔT_c), the *GNN-SM* achieves an RMSE of 0.026 with a very low MAPE of 0.165%, substantially outperforming both the FNN (0.031 RMSE, 0.2129%) and FNN-attention (0.052 RMSE, 0.381%) models. Similarly, for SOC prediction (Δs), the *GNN-SM* achieves an RMSE of 0.0015 with a MAPE of only 0.695%, compared to 0.839% for FNN and 2.7% for FNN-attention. The results demonstrate that the *GNN-SM*'s superior performance observed in smaller battery packs successfully scales to larger packs. The *GNN-SM*'s performance on completely unseen configurations ["RMSE*" and "%*" in Table IX)] shows a consistent superiority of the *GNN-SM*. For ΔT_c prediction, the *GNN-SM* achieves as RMSE of 0.857 with MAPE of 4.53%, significantly outperforming the FNN (1.486, 7.923%) and FNN-attention (1.369, 6.983%) models. For Δs prediction on unseen configurations, the *GNN-SM* maintains robust performance with an RMSE of 0.013 and MAPE of 5.007%, while the baseline models show higher errors of 6.637% (FNN) and 5.607% (FNN-attention). The experiments confirm that the *GNN-SM* is scalable and generalizes exceptionally well even with training data of $3.5 \times 10^{-25}\%$ of the total possible topologies, because its message-passing mechanism learns local cell-to-cell interaction rules that remain valid across unseen configurations, rather than memorizing global topology-specific mappings.

To further validate the computational efficiency for real-time deployment, Table X compares inference times across different pack sizes. For the 100-cell pack, the high-fidelity electro-thermal model requires 104.3 s to simulate a 500-s horizon, whereas the proposed *GNN-SM* completes inference in only 64 ms, which is about 1629 times faster. This concludes that while the *GNN-SM* incurs marginally higher inference latency compared to the FNN baselines, its inference time remains

TABLE X
INFERENCE TIME COMPARISON ACROSS DIFFERENT PACK SIZES

Model	10-Cell (ms)	100-Cell (ms)
Electro-chemical-thermal Simulator	3430	104300
FNN [17]	0.01	0.1
FNN-Attention [46]	0.01	15
GNN-SM (Proposed)	5.5	64

TABLE XI
GNN-SM ROBUSTNESS TO SENSOR NOISE ON UNSEEN CONFIGURATIONS (100-CELL PACK)

Noise STD	RMSE*	%*
ΔT_c		
0.0 °C	0.857	4.53
0.5 °C	0.88	4.81
1.0 °C	0.981	6.51
1.5 °C	1.33	9.5
Δs		
0%	0.013	5.007
2%	0.0161	5.85
5%	0.0162	5.91
10%	0.017	6.38

well within real-time requirements while delivering improved prediction accuracy.

G. Robustness to Sensor Noise

To evaluate the practical applicability of the proposed *GNN-SM* under real-world conditions, we assess its robustness to sensor measurement noise on unseen configurations. We evaluate the trained *GNN-SM* from Section V-F by adding zero-mean Gaussian noise to the input features during inference on the 500 unseen configurations (Test* dataset). Table XI presents the results. For temperature prediction (ΔT_c), the model degrades as noise increases. With moderate noise ($\sigma = 0.5\,^{\circ}\text{C}$), the MAPE increases only marginally from 4.53% to 4.81%. And even under high noise conditions ($\sigma = 1.5\,^{\circ}\text{C}$), the model maintains a single-digit percentage error at 9.5%. For SOC prediction (Δs), the model exhibits remarkable robustness. With 2% SOC noise, the MAPE increases from 5.007% to 5.85%. Under 5% noise, the MAPE is 5.91%. Even with 10% SOC noise, the MAPE only reaches 6.38%, representing a mere 1.37 percentage point increase from the noise-free baseline. These results confirm that the proposed *GNN-SM* retains acceptable accuracy under realistic sensor noise on completely unseen configurations. The robustness comes from the GNN's neighborhood aggregation, hence, local measurement noise on any single node is naturally attenuated.

VI. CONCLUSION

This article introduced a novel *GNN-SM*) for predicting state evolution in reconfigurable battery packs. The proposed model utilized the graph representation of the reconfigurable battery pack by modeling both cells and switches as different types of nodes connected by edges representing their physical connections. The *GNN-SM* successfully addressed the challenges of modeling both SOC and temperature imbalance in dynamic battery configurations by leveraging the natural graph structure of the battery pack topology. Numerical results showed that *GNN-SM* consistently outperformed FNN and FNN-attention baselines across all evaluation metrics. For temperature prediction, the model achieves an RMSE as low as 0.074 and a MAPE less than 2%, while SOC prediction errors remained under 10% in most scenarios. The *GNN-SM* demonstrated 74% and 46% lower RMSE for temperature and SOC imbalance prediction, respectively, compared to conventional FNN approaches. Notably, the model maintained robust performance even with limited training data (10% of available data) and successfully generalized to completely unseen battery configurations. The scalability evaluation on a 100-cell battery pack further validated the approach, where the *GNN-SM* maintained exceptional accuracy despite being trained on merely $3.5 \times 10^{-25}\%$ of all possible configurations. The model's effectiveness was demonstrated with both constant current and constant power discharge policies. Additionally, robustness evaluation under sensor measurement noise demonstrated that the *GNN-SM* maintained reliable predictions on unseen configurations, with MAPE remaining below 6.5% for SOC and below 10% for temperature even under high noise conditions.

While the *GNN-SM* demonstrated significant improvements in prediction accuracy and scalability, the following limitations remain to be addressed. First, the computational overhead, while manageable for real-time applications, may require further optimization for ultralow-latency embedded systems. Second, the current study relied on data generated from an experimentally validated electro-chemical-thermal simulator rather than real-time data from a physical reconfigurable hardware testbed. Although the simulator captured complex electrical and thermal coupling, the model's transfer learning capabilities and its ability to generalize to real-world environmental unpredictability remain to be fully validated in hardware. Furthermore, the cost implications of hardware implementation, including switching circuits and sensors, will be addressed. Lastly, the *GNN-SM* estimated the state imbalance over a predefined fixed horizon. In practice, different lithium-ion battery applications may require varying prediction horizons from seconds to several minutes. Currently, the *GNN-SM* required retraining for each desired horizon length. Addressing these limitations through hardware validation, more comprehensive evaluation against alternative GNN architectures, architectural ablation studies on components such as the attention mechanism and graph depth, and horizon-conditioned architectures, as well as extending the framework to incorporate battery degradation mechanisms such as capacity fade and resistance growth, represents a promising direction for future research.

REFERENCES

[1] A. Beaudet, F. Larouche, K. Amouzegar, P. Bouchard, and K. Zaghib, "Key challenges and opportunities for recycling electric vehicle battery materials," *Sustainability*, vol. 12, no. 14, 2020, Art. no. 5837.
[2] J. Chen, A. Behal, and C. Li, "Active cell balancing by model predictive control for real time range extension," in *Proc. 60th IEEE Conf. Decis. Control*, Austin, TX, USA, Dec. 2021, pp. 271–276.
[3] M. Cakir, I. Cankaya, I. Garip, and I. Colak, "Advantages of using renewable energy sources in smart grids," in *Proc. 10th Int. Conf. Smart Grid*, 2022, pp. 436–439.
[4] M. Ibrahim and R. Elhafiz, "Security analysis of smart grids," *Secur. Commun. Netw.*, vol. 2022, no. 1, 2022, Art. no. 7199301.
[5] M. Ibrahim and A. Alkhraibat, "Resiliency assessment of microgrid systems," *Appl. Sci.*, vol. 10, no. 5, 2020, Art. no. 1824.
[6] J. Chen and H. E. Garcia, "Economic optimization of operations for hybrid energy systems under variable markets," *Appl. Energy*, vol. 177, pp. 11–24, Sep. 2016.
[7] F. Yang, F. Gao, B. Liu, and S. Ci, "An adaptive control framework for dynamically reconfigurable battery systems based on deep reinforcement learning," *IEEE Trans. Ind. Electron.*, vol. 69, no. 12, pp. 12980–12987, Dec. 2022.

[8] R. Xiong, F. Sun, X. Gong, and H. He, "Adaptive state of charge estimator for lithium-ion cells series battery pack in electric vehicles," *J. Power Sources*, vol. 242, pp. 699–713, 2013.
[9] M. Dubarry, N. Vuillaume, and B. Y. Liaw, "Origins and accommodation of cell variations in li-ion battery pack modeling," *Int. J. Energy Res.*, vol. 34, no. 2, pp. 216–231, 2010.
[10] Q. Zhou, D. Anderson, and J. Sun, "State of health estimation for battery modules with parallel-connected cells under cell-to-cell variations," *eTransportation*, vol. 22, 2024, Art. no. 100346.
[11] F. Jin and K. G. Shin, "Pack sizing and reconfiguration for management of large-scale batteries," in *Proc. IEEE/ACM 3rd Int. Conf. Cyber- Phys. Syst.*, 2012, pp. 138–147.
[12] J. Chen, L. Zhang, and W. Gao, "Reconfigurable model predictive control for large scale distributed systems," *IEEE Syst. J.*, vol. 18, no. 2, pp. 965–976, Jun. 2024.
[13] T. Kim, W. Qiao, and L. Qu, "A series-connected self-reconfigurable multicell battery capable of safe and effective charging/discharging and balancing operations," in *Proc. 27th Annu. IEEE Appl. Power Electron. Conf. Expo.*, 2012, pp. 2259–2264.
[14] F. Altaf, *Thermal and State-of-Charge Balancing of Batteries Using Multilevel Converters*. Goteborg, Sweden: Chalmers Tekniska Hogskola, 2014.
[15] J. Chen, A. Behal, and C. Li, "Active battery cell balancing by real time model predictive control for extending electric vehicle driving range," *IEEE Trans. Autom. Sci. Eng.*, vol. 21, no. 3, pp. 4003–4015, Jul. 2024.
[16] S. Ci, J. Zhang, H. Sharif, and M. Alahmad, "A novel design of adaptive reconfigurable multicell battery for power-aware embedded networked sensing systems," in *Proc. IEEE Glob. Telecommun. Conf.*, 2007, pp. 1043–1047.
[17] Y. Weng and C. Ababei, "AI-assisted reconfiguration of battery packs for cell balancing to extend driving runtime," *J. Energy Storage*, vol. 84, 2024, Art. no. 110853.
[18] T. Morstyn, M. Momayyezan, B. Hredzak, and V. G. Agelidis, "Distributed control for state-of-charge balancing between the modules of a reconfigurable battery energy storage system," *IEEE Trans. Power Electron.*, vol. 31, no. 11, pp. 7986–7995, Nov. 2016.
[19] S. Jeon, J. Kim, J. Ahn, and H. Cha, "Optimizing discharge efficiency of reconfigurable battery with deep reinforcement learning," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 39, no. 11, pp. 3893–3905, Nov. 2020.
[20] G. Gunlu, "Dynamically reconfigurable independent cellular switching circuits for managing battery modules," *IEEE Trans. Energy Convers.*, vol. 32, no. 1, pp. 194–201, Mar. 2017.
[21] A. Škegro, C. Zou, and T. Wik, "Analysis of potential lifetime extension through dynamic battery reconfiguration," in *Proc. 25th Eur. Conf. Power Electron. Appl. Eur.*, 2023, pp. 1–11.
[22] T. Kacetl, J. Kacetl, N. Tashakor, and S. M. Goetz, "Ageing mitigation and loss control through ripple management in dynamically reconfigurable batteries," 2022, *arXiv:2211.03143*.
[23] Y. Weng and C. Ababei, "Battery pack cell balancing using topology switching and machine learning," in *Proc. IEEE Veh. Power Propulsion Conf.*, 2022, pp. 1–6.
[24] W. Han, C. Zou, L. Zhang, Q. Ouyang, and T. Wik, "Near-fastest battery balancing by cell/module reconfiguration," *IEEE Trans. Smart Grid*, vol. 10, no. 6, pp. 6954–6964, Nov. 2019.
[25] L. He, Z. Yang, Y. Gu, C. Liu, T. He, and K. G. Shin, "SOH-aware reconfiguration in battery packs," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 3727–3735, Jul. 2018.
[26] W. Han, A. Kersten, C. Zou, T. Wik, X. Huang, and G. Dong, "Analysis and estimation of the maximum switch current during battery system reconfiguration," *IEEE Trans. Ind. Electron.*, vol. 69, no. 6, pp. 5931–5941, Jun. 2022.
[27] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intell. Syst. Appl.*, vol. 13, no. 4, pp. 18–28, Jul./Aug. 1998.
[28] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1997.
[29] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "KNN model-based approach in classification," in *Proc. OTM Confederated Int. Conf. "On Move Meaningful Internet Syst."*, 2003, pp. 986–996.
[30] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, pp. 5–32, 2001.
[31] Z. Guo and H. Wang, "A deep graph neural network-based mechanism for social recommendations," *IEEE Trans. Ind. Inform.*, vol. 17, no. 4, pp. 2776–2783, Apr. 2021.
[32] W. Song, X. Chen, Q. Li, and Z. Cao, "Flexible job-shop scheduling via graph neural network and deep reinforcement learning," *IEEE Trans. Ind. Inform.*, vol. 19, no. 2, pp. 1600–1610, Feb. 2023.
[33] A. Ali, H. Y. Naeem, A. Sharafian, L. Qiu, Z. Wu, and X. Bai, "Dynamic multi-graph spatio-temporal learning for citywide traffic flow prediction in transportation systems," *Chaos, Solitons Fractals*, vol. 199, 2025, Art. no. 116898.
[34] T. Luo, Z. Bao, Z. He, and M. Gao, "FGNet: Feature engineering-guided attentive graph neural network for SOH estimation of lithium battery," *IEEE Trans. Transp. Electrific.*, vol. 11, no. 2, pp. 5880–5890, Apr. 2025.
[35] K. Q. Zhou, Y. Qin, and C. Yuen, "Graph neural network-based lithium-ion battery state of health estimation using partial discharging curve," *J. Energy Storage*, vol. 100, 2024, Art. no. 113502.
[36] J. Ouyang, Z. Lin, L. Hu, and X. Fang, "Voltage faults diagnosis for lithium-ion batteries in electric vehicles using optimized graphical neural network," *Sci. Rep.*, vol. 15, no. 1, 2025, Art. no. 27328.
[37] A. Ali, Y. Zhu, and M. Zakarya, "Exploiting dynamic spatio-temporal graph convolutional neural networks for citywide traffic flows prediction," *Neural Netw.*, vol. 145, pp. 233–247, 2022.
[38] A. Ali, I. Ullah, S. Ahmad, Z. Wu, J. Li, and X. Bai, "An attention-driven spatio-temporal deep hybrid neural networks for traffic flow prediction in transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 9, pp. 14154–14168, Sep. 2025.
[39] X. Lin et al., "A lumped-parameter electro-thermal model for cylindrical batteries," *J. Power Sources*, vol. 257, pp. 1–11, 2014.
[40] H. E. Perez, J. B. Siegel, X. Lin, A. G. Stefanopoulou, Y. Ding, and M. P. Castanier, "Parameterization and validation of an integrated electro-thermal cylindrical LFP battery model," in *Proc. Dyn. Syst. Control Conf.*, 2012, vol. 45318, pp. 41–50.
[41] J. Chen, Z. Zhou, Z. Zhou, X. Wang, and B. Liaw, "Impact of battery cell imbalance on electric vehicle range," *Green Energy Intell. Transp.*, vol. 1, no. 3, 2022, Art. no. 100025.
[42] W. Han, T. Wik, A. Kersten, G. Dong, and C. Zou, "Next-generation battery management systems: Dynamic reconfiguration," *IEEE Ind. Electron. Mag.*, vol. 14, no. 4, pp. 20–31, Dec. 2020.
[43] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.
[44] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.
[45] M. Fey and J. E. Lenssen, "Fast graph representation learning with pytorch geometric," 2019, *arXiv:1903.02428*.
[46] A. Irshayyid and J. Chen, "Highway merging control using multi-agent reinforcement learning," in *Proc. IEEE 3rd Int. Conf. Comput. Mach. Intell.*, 2024, pp. 1–2.
[47] G. L. Plett, *Battery Management Systems, Volume II: Equivalent-Circuit Methods*. Norwood, MA, USA: Artech House, 2015.