Safe and Efficient Control: A Subgraph-Augmented Hierarchical Reinforcement Learning Framework for Dynamically Reconfigurable Battery Systems

Kai Xie[1], **Jingwei Hu**[2], **Ri Huang**[1], **Xiaodong Li**[2], **Yanglin Zhou**[3], **Song Ci**[3,4], **Jun Cheng**[5], **Zhihong Zhang**[1,2*]

[1]Institute of Artificial Intelligence, Xiamen University, China
[2]School of Informatics, Xiamen University, China
[3]State Key Laboratory of Power System Operation and Control, Tsinghua University, China
[4]Department of Electrical Engineering, Tsinghua University, China
[5] College of Chemistry and Chemical Engineering, Xiamen University, China

Abstract

Dynamically Reconfigurable Battery (DRB) systems employ power electronic switches to create dynamic topologies. They enable effective management of cell inconsistencies through real-time adjustment of cell connections. However, existing DRB control methods struggle to learn effective strategies due to sparse rewards, which arise from blind exploration in large topological action spaces and complex operational constraints. This leads to ineffective policy learning, making safety and balancing performance difficult to ensure in practical applications. To this end, we propose a **S**ubgraph-**A**ugmented **H**ierarchical **R**einforcement **L**earning (SAHRL) framework. By combining hierarchical policies with topological structural knowledge, SAHRL effectively accelerates policy exploration and mitigates reward sparsity. Specifically, the high-level policy determines the strategic direction, while the subgraph-augmented low-level policy refines actions to meet operational constraints. The topological structural knowledge, extracted in the form of subgraphs and incorporated as an inductive bias, guides the agent focus on meaningful action patterns and reduce invalid exploration in the large action space. Extensive simulations and real-world experiments show that SAHRL achieves safe and efficient balancing. Notably, it increases the energy release by 10.56% compared to conventional methods in real-world applications.

1 Introduction

Lithium-ion battery energy storage systems, widely used in electric vehicles [Deng *et al.*, 2020], renewable energy integration [Abomazid *et al.*, 2022], and smart grids [Rouholamini *et al.*, 2022], rely on coordinated cell operation for efficient energy management and stable supply [Ma *et al.*, 2020]. However, conventional fixed-topology architectures are fundamentally limited by cell-to-cell inconsistencies [Feng *et al.*, 2019], as performance is constrained by the weakest cell, leading to early termination of discharge and significant energy waste [Wang *et al.*, 2023]. For example, as shown in Figure 1(a), the system is limited by the weakest cell b_5. To overcome this, Dynamically Reconfigurable Battery (DRB) systems employ power electronic switches to dynamically alter cell connections [Ci *et al.*, 2016], enabling adaptive balancing and improved energy utilization under varying load conditions [Viswanathan *et al.*, 2019]. Nevertheless, as shown in Figure 1(b), the introduction of reconfigurability expands the control action space combinatorially, making real-time optimal topology selection a challenging problem, especially within strict timing constraints [Han *et al.*, 2020].

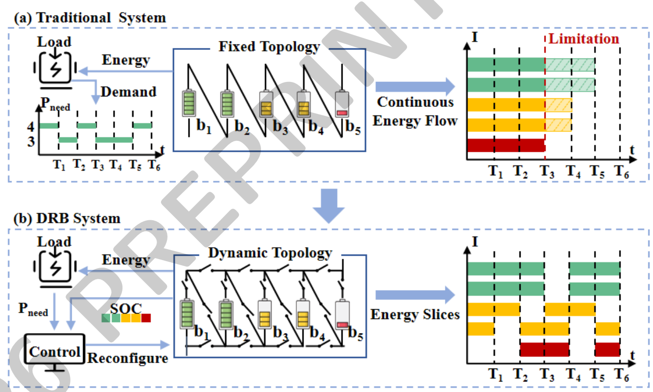

Figure 1: Comparison of different system operations. Fixed topologies are limited by the weakest cell b_5, while the dynamic topologies can reconfigure cell connections to ensure full energy utilization.

Recent advances have applied Reinforcement Learning (RL) to DRB systems for rapid decision-making [Jeon *et al.*, 2020], where agents learn adaptive control policies through environmental interaction and reward feedback [Yang *et al.*, 2022]. However, learning in large topological action spaces remains challenging due to sparse rewards [Ladosz *et al.*, 2022]. This sparsity stems from strict reward conditions that demand the simultaneous satisfaction of safety, load, and balancing constraints, worsened by the neglect of inherent topological relationships. Consequently, agents are limited to ineffective policy exploration, leading to frequent safety viola-

*Corresponding Author: zhihong@xmu.edu.cn

tions, premature convergence, and training failure. For example, in Figure 1(b), it may lead to unsafe outcomes, such as short circuits or open circuits, making the learned policy impractical. While existing works often restrict the action space [Baccari *et al.*, 2024] or simplify topologies [Meng *et al.*, 2023] to ensure training stability, such compromises limit the discovery of more efficient policies capable of flexible current routing and precise energy control [Bharath *et al.*, 2025]. This clear gap motivates our work to incorporate structural priors and hierarchical policy decomposition for safe, sample-efficient learning in DRB systems.

To address the aforementioned challenges, we explicitly incorporate topological structural knowledge into the DRB control paradigm. When inherent connectivity constraints are ignored, agents are forced to waste exploration effort on vast infeasible switch combinations. For instance, valid patterns like series are needed to meet voltage demand, while parallel patterns are essential for cell balancing. Without structural hints, it is difficult for the agent to identify these effective modes among random actions. This severely undermines the discovery of actions that meet the multi-faceted reward criteria. As we will show in the experiment section, approaches lacking this integration or using naive adaptation fail to resolve the sparse reward problem. Consequently, the core and novel challenge, which we tackle and validate experimentally, lies in adapting hierarchical RL to dynamically leverage this structural bias to ensure more efficient policy learning.

To this end, we propose a Subgraph-Augmented Hierarchical Reinforcement Learning (SAHRL) framework, which incorporates hierarchical policy decomposition and topological structural knowledge for safe and efficient control. Specifically, we design a collaborative hierarchical architecture that decomposes the complex control task. The high-level policy captures real-time state changes to generate initial decisions, while the subgraph-augmented low-level policy performs parallel local optimization to efficiently satisfy complex operational constraints. Also, we integrate topological structural knowledge via subgraphs as an inductive bias. This enables the agent to explicitly capture relationships between switches and directs exploration toward meaningful connection patterns. Our major contributions are listed as follows:

- To address the safety issues and poor balancing performance of existing DRB control methods in large action spaces with sparse rewards, we propose a novel framework, SAHRL, which ensures safe and efficient control.
- We decompose the complex control task into a hierarchical process, where the high-level policy sets the strategic direction and the subgraph-augmented low-level policy refines actions, thereby satisfying complex operational constraints that one-step generation fails to meet.
- We extract topological structural knowledge as subgraphs and use it as an inductive bias to accelerate policy learning and adapt to changing system conditions.
- In tests across simulation and a real-world prototype, SAHRL achieves reliable cell balancing and demonstrates a 10.56% increase in actual energy release.

2 Related Work

2.1 Dynamically Reconfigurable Battery System

Manufacturing and operating variations cause differences in cell capacity and resistance, resulting in performance imbalance [Feng *et al.*, 2019]. Traditional balancing methods release extra charge as heat [Ismail *et al.*, 2017] or transfer it to cells with lower charge [Park *et al.*, 2023]. However, additional balancing currents reduce energy efficiency and accelerate degradation [Alvarez-Diazcomas *et al.*, 2020].

To address these limitations, DRB systems create dynamic topologies that can reconfigure cell connections for balancing without unnecessary discharge [Ci *et al.*, 2016]. Specifically, a system's balancing capacity depends on its topological flexibility [Han *et al.*, 2020], which increases with the number of switches but at the cost of higher control complexity. Consequently, the four-switch topology is regarded as an optimal compromise [Cui *et al.*, 2022], providing sufficient flexibility with minimal hardware redundancy. For control methods, previous studies have used lookup tables [Kim *et al.*, 2011] and dynamic programming [He *et al.*, 2016; Lin and Ci, 2017], which evaluate all feasible configurations and select those with the best balancing performance. However, they rely heavily on expert knowledge, and their high computational cost limits practical applicability [Bharath *et al.*, 2025]. Recently, RL has been widely applied for rapid decision-making, enabling agents to learn control strategies without explicit knowledge of complex balancing principles [Yang *et al.*, 2022]. Nevertheless, as the action space grows combinatorial with the number of switches, the learning process becomes difficult. Therefore, existing studies adopt single-switch topologies [Yang *et al.*, 2023; Meng *et al.*, 2023] or restrict reconfigurations in multi-switch topologies to an artificially defined finite set of actions [Baccari *et al.*, 2024], which limits system adaptability under complex operating conditions. In contrast, our proposed SAHRL framework trains the agent directly within the full action space of the four-switch topology to achieve superior control policies.

2.2 Reinforcement Learning

RL has demonstrated effectiveness for solving complex energy management problems [Shen *et al.*, 2022; Arroyo *et al.*, 2022]. Specifically, the task is formulated as a Markov Decision Process (MDP), where agents observe states, select actions, and learn policies to maximize cumulative rewards. However, in scenarios with sparse rewards, agents often suffer from inefficient and limited exploration, hindering policy learning [Ladosz *et al.*, 2022]. To overcome this issue, some methods improve learning efficiency by decomposing tasks and applying hierarchical policies, a paradigm known as Hierarchical Reinforcement Learning (HRL) [Pateria *et al.*, 2021]. A standard HRL framework employs a two-level structure bridged by goals: a high-level policy sets abstract goals, while a low-level policy generates concrete actions and provides feedback [Kulkarni *et al.*, 2016; Li *et al.*, 2019]. To enhance performance, some researchers have employed hindsight mechanisms [Levy *et al.*, 2017] for training stability, while introducing k-step mechanisms [Luo

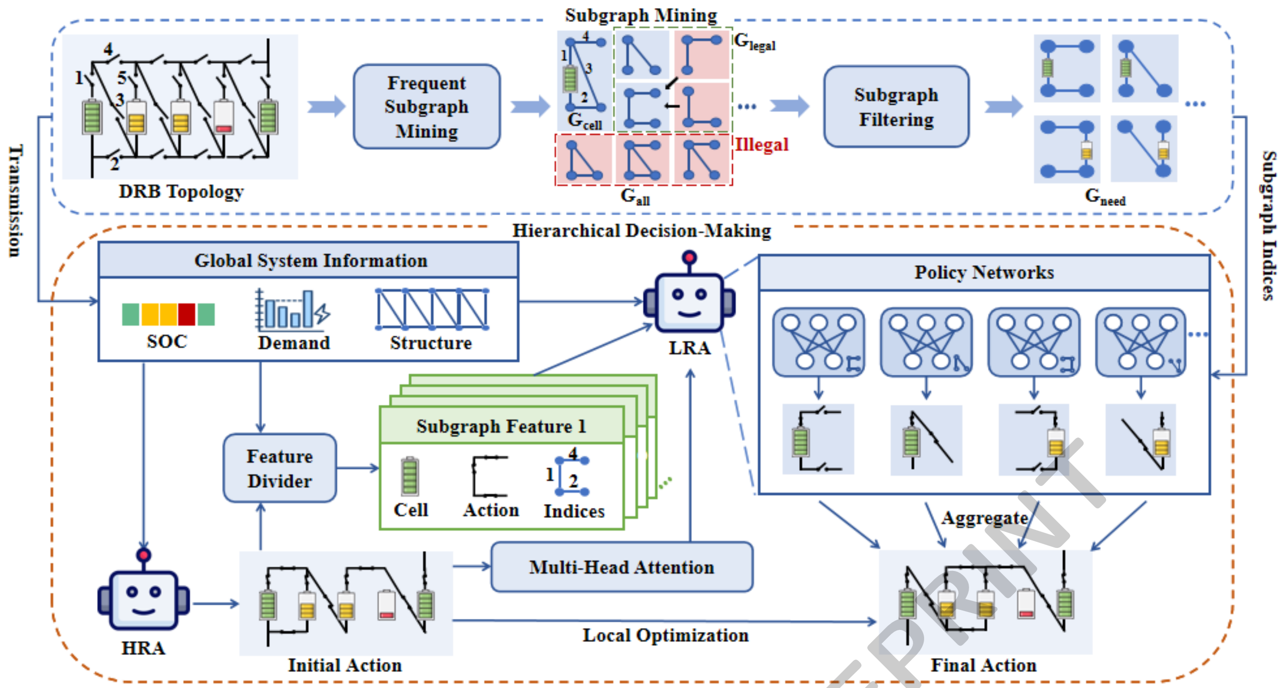

Figure 2: The overall architecture of the SAHRL framework. Recurring structural patterns, filtered by legality and inclusion, are extracted as subgraphs to serve as an inductive bias for accelerating policy exploration. For each decision, the HRA observes global system information and generates an initial action, which is processed via a multi-head attention layer. Subgraph features are extracted to directly provide localized information. The LRA generates the final action by aggregating local refinements, ensuring it meets complex operational requirements.

et al., 2024] to predict state transitions for enhanced goal reachability. Nevertheless, these methods rely on generating long-term goals that require multiple actions to accomplish, which is not suitable for DRB systems, as demonstrated in our experimental section. In DRB control, cell states can change within 100 milliseconds, rendering a goal derived from an earlier state may become invalid before the low-level policy can act. Furthermore, defining proper goals is hard, as hierarchical policies often get stuck in local optima. To overcome these limitations, our proposed SAHRL framework replaces goal guidance with a collaborative process where the high-level policy generates initial actions, and the low-level policy performs local refinement to complete the final decisions.

3 Problem Statement

The control task can be formally defined as an MDP, represented by the tuple (S, A, P, R), where S is the state space, A is the action space, P is the transition function and R is the reward function. At each time step t, the agent observes the current system state $s_t \in S$, selects an action $a_t \in A$, and receives the next state $s_{t+1} \in S$ together with a reward r_t.

In the context of an n-cell DRB system, the state s corresponds to the vector of state of charge (SOC) values, denoted as $(soc_1, soc_2, \ldots, soc_n)$. The current SOC for each cell is updated as follows:

$$SOC^t = SOC^{t-1} + c \cdot \sigma \cdot \delta \tag{1}$$

where $c \in \{1, -1\}$ represents the charging or discharging mode, $\sigma \in \{0, 1\}$ denotes the cell connection status, and δ represents the change in SOC. The topology is dynamically adjusted via m high-speed switches, whose discrete action space can be defined as $A = \{a_1, a_2, \ldots, a_{2^m}\}$, where each a_i represents a specific configuration. The primary physical objective is to select an optimal action a^* based on state s that minimizes the SOC inconsistency while strictly meeting operational constraints C. Using the above notations, this physical optimization problem is formulated as:

$$a^* = \arg\min_{a \in A} \mathbb{E}\left[\sum_{\tau=t}^{T} \Delta(SOC^\tau) \mid s, a\right] \quad \text{s.t. } C(a) > 0 \tag{2}$$

where $\Delta(SOC^\tau)$ denotes the SOC difference at time τ, and T is the duration. To solve this via RL, the objective is transformed into finding an optimal policy π^* that maximizes the expected cumulative discounted reward:

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0\right] \tag{3}$$

where r_t is designed to align with the minimization target in Eq. (2), s_0 is the initial state, and γ is the discount factor.

4 Methodology

4.1 SAHRL Overview

To address the issues of illegal decisions and suboptimal balancing performance observed in existing DRB control methods in large action spaces, we propose **SAHRL**, which decomposes the complex task via hierarchical policies to en-

Algorithm 1 Calculation of Hierarchical Rewards

Input: High-level Action a_h, Low-level Action a_l, Current State s
Output: High-level Reward r_h, Low-level Reward r_l

Stage I: Calculate reward of a_h
if Safety constraint violated **then** $r_h \leftarrow -1$
else if Power demand unmet **then** $r_h \leftarrow -0.15$
else if Unbalanced **then** $r_h \leftarrow 0.15$
else $r_h \leftarrow 1$
end if
Stage II: Calculate reward of a_l
Let $R_h(a_l)$ be the reward calculated using Stage I logic
Calculate improvement reward r_{imp}:
if $R_h(a_l) > r_h$ **then** $r_{imp} \leftarrow 1$
else if $R_h(a_l) = r_h$ **and** $R_h(a_l) = 1$ **then** $r_{imp} \leftarrow 1$
else if $R_h(a_l) = r_h$ **then** $r_{imp} \leftarrow 0$
else $r_{imp} \leftarrow -1$
end if
$r_l \leftarrow (R_h(a_l) + r_{imp})/2$
return r_h, r_l

sure constraint satisfaction, and integrates topological structural knowledge to accelerate policy learning. The overall architecture of SAHRL is shown in Figure 2, which comprises two main modules: (i) **Subgraph mining**, which extracts repeated and valid structural patterns from the topology via subgraphs. This topological knowledge serves as inductive bias that guides policy learning to focus on meaningful connection patterns, thereby reducing blind exploration. (ii) **Hierarchical decision-making**, composed of a high-level agent (HRA) and a low-level agent (LRA), that work together to complete the decision making process. The detailed computation of hierarchical rewards is given in Algorithm 1. The key concepts are defined as follows:

- The high-level state s_h describes the global information of the system, covering battery state parameters, external load demand, and topological structural features.
- The high-level action a_h denotes a preliminary decision and can be formalized as an m-dimensional binary vector $[sw_1, sw_2, \cdots, sw_m] \in \{0,1\}^m$, where each sw_i represents the state of the i-th switch, with 0 indicating disengagement and 1 indicating engagement.
- The high-level reward r_h evaluates the operational performance of a_h. Actions should reduce the SOC imbalance or maintain it within acceptable limits, while ensuring safety. In addition, the matching constraint $|P_{\text{load}} - P_{\text{supply}}| \le \varepsilon$ must be satisfied, where P_{load} denotes the external load demand, P_{supply} denotes the supplied power, and ε is the allowable deviation.
- The low-level state s_l describes more local information to support refinement, consisting of s_h, a_h and local features associated with each subgraph.
- The low-level action a_l denotes the final decision, sharing the same action space as a_h and incorporates local optimizations based on it.
- The low-level reward r_l evaluates the overall performance of a_l and consists of a base term identical to r_h and an improvement term quantifying the performance gain of a_l over a_h.

For each decision, the HRA first observes the high-level state s_h to give an initial action a_h, which acts as a distributional constraint. Conditioned on a_h, the LRA employs multiple subgraph-augmented policy networks that formulate adjustments based on the low-level state s_l. These terms are aggregated to refine a_h, yielding the final action a_l that satisfies operational requirements. Formally, the global policy is expressed as the expectation of the local refinement policy over the high-level guidance distribution:

$$\pi(a_l \mid s) = \mathbb{E}_{\pi_{\text{HRA}}(\cdot|s_h)} \left[\pi_{\text{LRA}}(a_l \mid s_l, a_h)\right] \tag{4}$$

4.2 Subgraph Mining

In DRB systems, the topological structure is highly regular, with each cell having an identical number and pattern of connected switches. Consequently, we can extract recurring structural patterns as subgraphs, which provide prior topological knowledge for effective policy learning.

The topology is abstracted as a weighted graph $G = (V, E, W)$, where V denotes the intersections of the topology, E represents the connected components, and W specifies the component types. A set of cell subgraphs is constructed as $G_{\text{cell}} = \{G_1, G_2, \ldots, G_n\}$, where G_i is the subgraph formed by the i-th cell and its adjacent switches. Subsequently, the gSpan algorithm is employed, which applies depth first search (DFS) encoding and the right-most path extension strategy to discover all frequent subgraphs:

$$G_{\text{all}} = \{g \subseteq G_i \mid \text{support}(g, G_{\text{cell}}) \ge \text{minsup}\} \tag{5}$$

where $\text{support}(g, G_{\text{cell}})$ denotes the support of the candidate subgraph g in G_{cell} and minsup represents the minimum support, which is set to $n-1$ to extract representative subgraphs, thereby reducing the set size while allowing for inherent structural variations at the topological boundaries.

Specifically, a subgraph can be encoded by its DFS sequence of edges. Each edge e is represented as a quintuple $(v_i, v_j, l_i, l_j, l_e)$, where v_i and v_j denote the visit orders of nodes i and j, l_i and l_j are their labels, and l_e is the edge label. Concatenating these quintuples in traversal order yields a DFS encoding for the subgraph. Since a subgraph may have multiple encodings due to different traversal orders, the lexicographically smallest sequence is chosen as the canonical form to ensure consistency. The mining process starts with one-order frequent edges and recursively expands candidate subgraphs. Expansion prioritizes backward extension along the rightmost path, followed by forward extension to introduce new nodes. Each expanded subgraph is retained if it meets the minsup threshold.

However, G_{all} may include unsafe configurations that could hinder policy exploration. Therefore, we remove all invalid subgraphs where series and parallel switches close simultaneously, as expressed by:

$$sw_{\text{series}} + sw_{\text{parallel}} \le 1 \tag{6}$$

where sw_{series} and sw_{parallel} represent the states of the series and parallel switches. Moreover, since the valid subgraph set

G_{legal} is still large, we filter it based on edge inclusion to reduce the training burden. For example, if g_p and g_q are in G_{legal} and their edge sets satisfy $E_p \subset E_q$, preference is given to g_q with more comprehensive coverage.

Finally, a small set of legal subgraphs is obtained, denoted G_{need}. These subgraphs serve as an inductive bias for building policy networks. This enables each network to focus on learning precise control logic for specific local structures, thereby accelerating policy convergence.

4.3 Hierarchical Decision-Making

To achieve safe and efficient control, we decompose the task into hierarchical policies where HRA and LRA collaborate in the decision-making process. Both agents are modeled using PPO combined with Generalized Advantage Estimation (GAE). In addition, an entropy regularization term is incorporated to promote sufficient exploration and prevent premature convergence. The objective function is defined as follows:

$$L(\theta) = \mathbb{E}_t \left[L^{CLIP}(\theta) - c_1 L^{VF}(\theta) + c_2 H[\pi_\theta(s_t)] \right] \tag{7}$$

where $L^{CLIP}(\theta)$ denotes the clipped surrogate objective and its calculation uses the advantage values from GAE, $L^{VF}(\theta)$ represents the value function loss and $H[\pi_\theta(s_t)]$ refers to the entropy of the policy distribution, with c_1 and c_2 being the corresponding weighting coefficients.

Specifically, the HRA is responsible for setting policy direction, such as deciding which battery units should be prioritized. Rather than generating fixed long-term goals, it dynamically produces a preliminary action vector a_h in real time based on the global state s_h. The LRA operates local optimizations by assigning an independent policy network π_i to each subgraph in G_{need}. It breaks the global topology into independent parts, turning the complex problem into smaller and parallel tasks. During the optimization process, a_h serves as a global constraint to ensure that all local actions align with the overall objective, thereby preventing independent networks from making conflicting decisions. To further enhance this information transmission between agents, a Multi-Head Attention (MHA) mechanism is used to encode a_h. Since changing one switch often affects the circuit state of others, a valid configuration relies on their coordination rather than independent control. MHA addresses this by calculating attention weights to identify strongly related components, allowing the model to capture these hidden dependencies and integrate them into a unified representation:

$$\text{MHA}(a_h) = \text{Concat}\left(h_1, h_2, \ldots, h_d\right) W^O \tag{8}$$

where h_i denotes the output of the i-th attention head, and W^O represents the output projection matrix. To ensure precise local optimization, we map global information into local receptive fields. Each π_i receives a specialized subgraph feature vector f_{sg}, which consist of the corresponding cell state, preliminary actions in a_h, and associated switch indices. Consequently, each π_i takes the s_h, the a_h and its specific subgraph feature $f_{sg}(i)$ as input, and outputs a local adjustment vector. The final decision a_l is then obtained by aggregating these parallel local refinements:

$$a_l = a_h + \sum_{i=1}^{|G_{\text{need}}|} \pi_i(s_h, a_h, f_{sg}(i))) \tag{9}$$

This decomposition enhances safety by enabling local networks to strictly enforce physical constraints that a global policy might overlook. Moreover, by generating action estimates through the aggregation of multiple policy networks, the framework ensures more robust decision-making and improves the efficient utilization of positive rewards.

5 Experiments

Extensive experiments are conducted to evaluate SAHRL in both simulation and real-world DRB systems. We aim to address the following questions: **(Q1)** How does SAHRL perform in comparison with SOTA Reinforcement Learning (RL) baselines and straight Hierarchical Reinforcement Learning (HRL) adaptions? **(Q2)** What contribution does each key component of SAHRL make to its performance? **(Q3)** Can SAHRL achieve cell balancing effectively in real-world scenarios? **(Q4)** Can SAHRL maintain robust and scalable balancing performance when deployed in larger-scale battery systems?

5.1 Experimental Setup

Environments. A DRB system simulation environment is developed in MATLAB/Simulink. Each cell is modeled from a predefined discharge curve [Zhu *et al.*, 2013], which provides real-time measurements of current, voltage, and SOC. A real-world test is also performed using INR18650-20R batteries with a rated capacity of 2000 mAh. All systems support interactive control through topological action signals.

Training Setup. The initial SOC difference is randomly initialized within 15%, and the load demand varies during operation, ranging from 1 to n. Each discharge cycle comprises 150 steps. The learning rates for HRA and LRA are 2e-4 and 1e-4. The number of attention heads is set to 8, and the entropy is set to 1e-2.

Baselines and Evaluation Metrics. We carefully select baselines for comprehensive evaluation. For single-level policy, DDQN [Van Hasselt *et al.*, 2016] and PPO [Schulman *et al.*, 2017] are included as representative and widely adopted algorithms for discrete high-dimensional action spaces. For hierarchical policies, H-DQN [Kulkarni *et al.*, 2016], HAC [Levy *et al.*, 2017], HiPPO [Li *et al.*, 2019], and HLMA [Luo *et al.*, 2024] are incorporated as representative methods of different hierarchical strategies. We evaluate practical performance using energy release, illegal operation rate, and SOC difference to respectively assess energy efficiency, system safety, and cell balancing capability.

5.2 Main Results (Q1)

To address **Q1**, we conduct a comprehensive comparative analysis of SAHRL and baselines. Figure 3 presents the training reward curves for each method, where shaded areas represent trajectories and dark curves denote smoothed results. Higher rewards indicate better balancing performance. To ensure robustness, each trained model is evaluated six times. The initial SOC values for cells are randomly sampled between 80% and 95%, with a mean difference of 11.18%. Table 1 summarizes the average performance. We summarize the observations as follows:

Algorithm	DDQN	PPO	H-DQN	HAC	HiPPO	HLMA	Ours
Energy / Wh	2.128E+01	2.064E+01	2.200E+01	2.144E+01	1.981E+01	2.238E+01	**2.272E+01**
illegal / %	4.331E+00	1.920E+00	1.783E+00	**0.000E+00**	**0.000E+00**	1.339E+00	**0.000E+00**
ΔSOC / %	8.949E+00	1.305E+01	3.582E+00	1.177E+01	1.367E+01	2.300E+00	**9.760E-01**

Table 1: Performance comparison among baselines and SAHRL in the simulated DRB system.

Ablation	w/o subgraphs	G_{cell}	G_{all}	G_{legal}	w/o f_{sg}	w/o MHA	Ours
Energy / Wh	2.256E+01	2.012E+01	2.247E+01	2.251E+01	2.142E+01	2.246E+01	**2.272E+01**
illegal / %	1.564E+01	**0.000E+00**	1.133E+01	**0.000E+00**	1.374E+00	3.724E+00	**0.000E+00**
ΔSOC / %	1.362E+00	1.556E+01	2.381E+00	1.476E+00	1.174E+01	1.446E+00	**9.760E-01**

Table 2: Ablation study on the effectiveness of each key component in SAHRL.

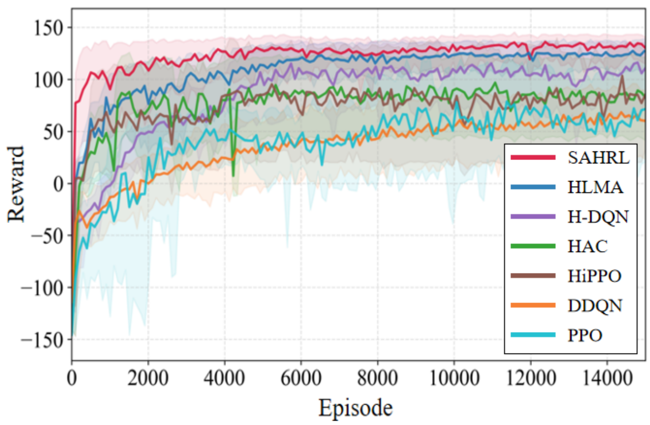

Figure 3: Comparison of reward curves for different algorithms.

Obs. 1. SAHRL outperforms all baselines across every evaluation metric. During training, it demonstrates faster and more comprehensive policy learning. While ensuring operational safety, SAHRL shows the highest energy release, representing an improvement of about 1.5% over the next-best method. It also reduces the SOC difference to 0.98%, confirming that a hierarchical collaborative framework incorporating topological structural knowledge can significantly enhance balancing performance. By contrast, all baselines exhibit an SOC difference exceeding 2% and several of them generate illegal actions during operation, posing safety risks.

Obs. 2. RL methods perform poorly in DRB systems within the large action space. PPO ensures training stability through constrained policy updates, yet this constraint limits the scope of policy exploration, leading to noticeable SOC imbalance and nonzero constraint violations. Similarly, DDQN relies on explicit enumeration of discrete actions and Q-value updates, but accurately estimating the value of a large number of actions is difficult, as reflected by its higher illegal operation rate. Consequently, directly applying RL methods to explore the large topological action space is inefficient and often fails to learn effective control strategies.

Obs. 3. Hierarchical decomposition significantly improves learning efficiency, but existing HRL methods remain insufficient for practical applications. Specifically, H-DQN and HLMA perform well in cell balancing but they generate over 1% of invalid actions, indicating inadequate handling of safety constraints. In contrast, HAC and HiPPO prevent unsafe actions but fail to effectively reduce the SOC difference during operation, as they focus too much on satisfying safety constraints and neglect balancing performance. These methods often converge to suboptimal solutions that fail to meet both requirements simultaneously.

5.3 Ablation Studies (Q2)

To answer **Q2**, we conduct a series of ablation studies, with the results summarized in Table 2. By removing or altering specific modules, we analyze how they affect the safety and efficiency of the system. We have the following observations:

Obs. 4. Mining appropriate subgraphs is essential for learning comprehensive and safe control strategies, as improper subgraphs can lead to constraint violations or suboptimal convergence. Specifically, the effect of the subgraph mining module is evaluated by comparing models trained with different subgraph sets. When subgraphs are removed (w/o subgraphs), the illegal operation rate increases to 15.64%, clearly highlighting the importance of incorporating topological structural knowledge. While the G_{cell} variant avoids illegal operations, it yields a 15.56% SOC difference, indicating that cell subgraphs alone provide insufficient inductive bias. Additionally, the G_{all} variant generates 11.33% illegal actions, demonstrating that invalid subgraphs directly hinder policy learning. Finally, while the G_{legal} variant ensures operational safety, its balancing performance degrades due to redundant subgraphs increasing the complexity of the policy networks. This added complexity leads to premature convergence during training and prevents the discovery of more effective control strategies.

Obs. 5. Subgraph features and the multi-head attention mechanism are important for precise cell balancing and capturing switch dependencies. Specifically, removing subgraph features (w/o f_{sg}) leads to a significant drop in balancing performance, highlighting the necessity of features that directly represent the relevant cells. Furthermore, the performance degradation observed in the absence of multi-head attention (w/o MHA) proves that this mechanism is essential for understanding the relationships between switches. Since switches in DRB systems operate collaboratively to form circuits, MHA enables the agent to capture these structural connections and prioritize the most unbalanced cells.

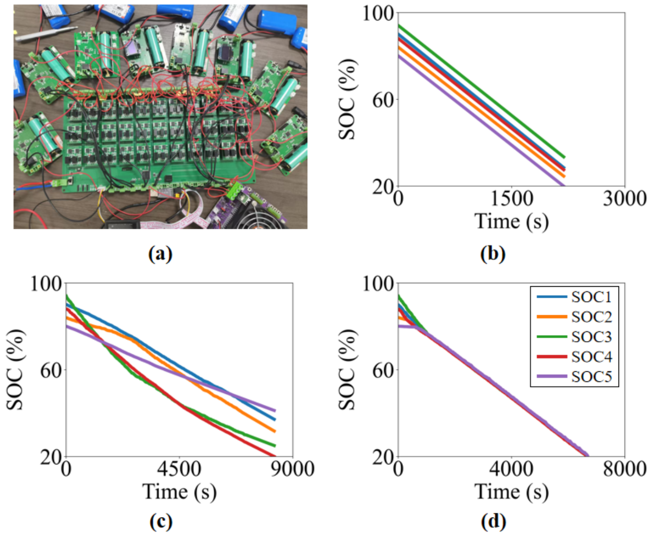

Figure 4: Testing in the real-world scenario. (a) represents hardware setup, (b) - (d) show the SOC variations of cells during system operation under the fixed topology, HAC, and SAHRL, respectively.

Algorithm	Fixed	HAC	SAHRL
Energy / Wh	1.838E+01	1.709E+01	2.032E+01
ΔSOC / %	1.349E+01	2.122E+01	6.325E-01

Table 3: Detailed practical performance of different methods.

5.4 Real-World Scenario Testing (Q3)

For **Q3**, a real hardware prototype is constructed for validation. Ensuring safety during operation in real systems is crucial. Therefore, to validate performance while ensuring safety, we conduct tests using three methods: the fixed topology, the HAC, and the SAHRL. Agents make decisions based on current and voltage feedback signals, along with SOC estimated using the ampere-hour integration method. To amplify the initial SOC difference, cells are discharged to different levels. The system terminates operation when the SOC of any cell drops below 20%. The hardware prototype and the SOC curves of different algorithms are shown in Figure 4, while their performance is summarized in Table 3. We have the following finding:

Obs. 6. SAHRL achieves superior energy utilization and precise cell balancing compared to other settings, while maintaining real-time computational efficiency. As shown in Figure 4(b), the SOC difference drops slightly from 14% to 13.49% when all cells discharge in series. This indicates that the available energy is not fully used, and the total output is limited to 18.38 Wh. For HAC control, an unexpected rise in SOC difference is observed in Figure 4(c). This issue is caused by the control policy that is too conservative, where only a few cells are selected for discharge at a time. This demonstrates that HAC fails to effectively rotate cell usage. In contrast, SAHRL successfully reduces the SOC difference to 0.63% as illustrated in Figure 4(d). This results in a total output of 20.32 Wh, representing a significant improvement of 10.56% over the fixed topology. Notably, SAHRL shows high computational efficiency with an inference time of only 1.00 ms, fully meeting real-time control requirements.

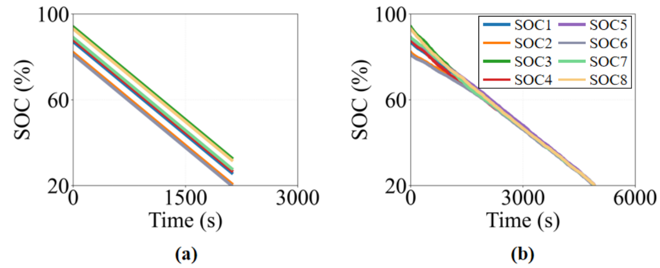

Figure 5: Testing in the larger-scale battery systems. (a) and (b) show the SOC variations of the cells during system operation under the fixed topology and SAHRL, respectively.

5.5 Larger-Scale Battery Systems Testing (Q4)

In response to **Q4**, the SAHRL framework is extended to an eight-cell system to evaluate its scalability and effectiveness under increased system complexity. In practice, an eight-cell configuration can serve as a functional battery cluster, making it a representative scale for deployment. As the number of cells increases, the action space grows rapidly, posing a greater challenge for the agent to select effective control actions. Figure 5 presents the SOC curves of a representative run. Based on the results, we have the following conclusion:

Obs. 7. SAHRL maintains effective and scalable balancing performance in larger-scale battery systems. Under fixed topology operation, all cells are constrained to share the same current, preventing effective energy redistribution and causing the initial SOC difference to persist throughout the discharge process. In contrast, SAHRL actively coordinates multiple cells to allow distinct SOC values to gradually converge. Consequently, the SOC difference is reduced from an initial 13% to 0.77%, and the total released energy increases from 30.84 Wh to 35.75 Wh. This represents a significant improvement of 15.92%, demonstrating the superior scalability of the proposed framework.

6 Conclusion

In this paper, we propose a novel HRL framework for DRB systems that addresses insufficient policy learning caused by sparse rewards while achieving safe and efficient control. The framework decomposes the control task into the high-level that sets strategic directions and the subgraph-augmented low-level policy refines actions. By extracting topological structural knowledge via subgraphs and using them as an inductive bias, SAHRL effectively accelerates policy learning. Experimental results show that SAHRL achieves superior performance across multiple metrics, ensuring both effective balancing and system safety. Furthermore, real-world scenario testing confirms the practical applicability. Future work will focus on applying SAHRL to address other practical challenges for further optimization.

Acknowledgements

This research is supported by the National Natural Science Foundation of China (no. 62176227); Xiamen Major Science and Technology Project (no. 3502Z20241028); Xiamen Science and Technology Project (no. 3502Z202571028); The Fundamental Research Funds for the Central Universities, Xiamen University (no. 20720250171); National Key Research and Development Program of China (no. 2024YFE0200502, 2023YFB2407900); National Natural Science Foundation of China (no. U22A20227).

References

[Abomazid *et al.*, 2022] Abdulrahman M Abomazid, Nader A El-Taweel, and Hany EZ Farag. Optimal energy management of hydrogen energy facility using integrated battery energy storage and solar photovoltaic systems. *IEEE Transactions on Sustainable Energy*, 13(3):1457–1468, 2022.

[Alvarez-Diazcomas *et al.*, 2020] Alfredo Alvarez-Diazcomas, Adyr A Estévez-Bén, Juvenal Rodríguez-Reséndiz, Miguel-Angel Martínez-Prado, and Jorge D Mendiola-Santíbañez. A novel rc-based architecture for cell equalization in electric vehicles. *Energies*, 13(9):2349, 2020.

[Arroyo *et al.*, 2022] Javier Arroyo, Carlo Manna, Fred Spiessens, and Lieve Helsen. Reinforced model predictive control (rl-mpc) for building energy management. *Applied Energy*, 309:118346, 2022.

[Baccari *et al.*, 2024] Silvio Baccari, Massimo Tipaldi, and Valerio Mariani. Deep reinforcement learning for cell balancing in electric vehicles with dynamic reconfigurable batteries. *IEEE Transactions on Intelligent Vehicles*, 2024.

[Bharath *et al.*, 2025] YK Bharath, VP Anandu, U Vinatha, and GK Purushothama. Single-port and multi-port self-reconfigurable battery topologies for dynamic cell balancing. *Journal of Energy Storage*, 119:116402, 2025.

[Ci *et al.*, 2016] Song Ci, Ni Lin, and Dalei Wu. Reconfigurable battery techniques and systems: A survey. *IEEE access*, 4:1175–1189, 2016.

[Cui *et al.*, 2022] Haoyong Cui, Zhongbao Wei, Hongwen He, and Jianwei Li. Novel reconfigurable topology-enabled hierarchical equalization of lithium-ion battery for maximum capacity utilization. *IEEE Transactions on Industrial Electronics*, 70(1):396–406, 2022.

[Deng *et al.*, 2020] Jie Deng, Chulheung Bae, Adam Denlinger, and Theodore Miller. Electric vehicles batteries: requirements and challenges. *Joule*, 4(3):511–515, 2020.

[Feng *et al.*, 2019] Fei Feng, Xiaosong Hu, Lin Hu, Fengling Hu, Yang Li, and Lei Zhang. Propagation mechanisms and diagnosis of parameter inconsistency within li-ion battery packs. *Renewable and Sustainable Energy Reviews*, 112:102–113, 2019.

[Han *et al.*, 2020] Weiji Han, Torsten Wik, Anton Kersten, Guangzhong Dong, and Changfu Zou. Next-generation battery management systems: Dynamic reconfiguration. *IEEE Industrial Electronics Magazine*, 14(4):20–31, 2020.

[He *et al.*, 2016] Liang He, Zhe Yang, Yu Gu, Cong Liu, Tian He, and Kang G Shin. Soh-aware reconfiguration in battery packs. *IEEE Transactions on Smart Grid*, 9(4):3727–3735, 2016.

[Ismail *et al.*, 2017] Kristian Ismail, Asep Nugroho, Sunarto Kaleg, et al. Passive balancing battery management system using mosfet internal resistance as balancing resistor. In *2017 International Conference on Sustainable Energy Engineering and Application (ICSEEA)*, pages 151–155. IEEE, 2017.

[Jeon *et al.*, 2020] Seunghyeok Jeon, Jiwon Kim, Junick Ahn, and Hojung Cha. Optimizing discharge efficiency of reconfigurable battery with deep reinforcement learning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(11):3893–3905, 2020.

[Kim *et al.*, 2011] Younghyun Kim, Sangyoung Park, Yanzhi Wang, Qing Xie, Naehyuck Chang, Massimo Poncino, and Massoud Pedram. Balanced reconfiguration of storage banks in a hybrid electrical energy storage system. In *2011 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 624–631. IEEE, 2011.

[Kulkarni *et al.*, 2016] Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

[Ladosz *et al.*, 2022] Pawel Ladosz, Lilian Weng, Minwoo Kim, and Hyondong Oh. Exploration in deep reinforcement learning: A survey. *Information Fusion*, 85:1–22, 2022.

[Levy *et al.*, 2017] Andrew Levy, George Konidaris, Robert Platt, and Kate Saenko. Learning multi-level hierarchies with hindsight. *arXiv preprint arXiv:1712.00948*, 2017.

[Li *et al.*, 2019] Alexander C Li, Carlos Florensa, Ignasi Clavera, and Pieter Abbeel. Sub-policy adaptation for hierarchical reinforcement learning. *arXiv preprint arXiv:1906.05862*, 2019.

[Lin and Ci, 2017] Ni Lin and Song Ci. Toward dynamic programming-based management in reconfigurable battery packs. In *2017 IEEE Applied Power Electronics Conference and Exposition (APEC)*, pages 2136–2140. IEEE, 2017.

[Luo *et al.*, 2024] Yu Luo, Tianying Ji, Fuchun Sun, Huaping Liu, Jianwei Zhang, Mingxuan Jing, and Wenbing Huang. Goal-conditioned hierarchical reinforcement learning with high-level model approximation. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[Ma *et al.*, 2020] Dajun Ma, Wu Chen, and Xinbo Ruan. A review of voltage/current sharing techniques for series–parallel-connected modular power conversion systems. *IEEE Transactions on Power Electronics*, 35(11):12383–12400, 2020.

[Meng *et al.*, 2023] Jinhao Meng, Feng Yang, Jichang Peng, and Fei Gao. A proximal policy optimization based control framework for flexible battery energy storage system. *IEEE Transactions on Energy Conversion*, 39(2):1183–1191, 2023.

[Park *et al.*, 2023] Kyung-Hwa Park, Minsu Lee, and Gun-Woo Moon. A new phase shift full bridge dc/dc converter with integrated intermodule battery equalization circuit (ibec). *IEEE Transactions on Transportation Electrification*, 10(3):6618–6633, 2023.

[Pateria *et al.*, 2021] Shubham Pateria, Budhitama Subagdja, Ah-hwee Tan, and Chai Quek. Hierarchical reinforcement learning: A comprehensive survey. *ACM Computing Surveys (CSUR)*, 54(5):1–35, 2021.

[Rouholamini *et al.*, 2022] Mahdi Rouholamini, Caisheng Wang, Hashem Nehrir, Xiaosong Hu, Zechun Hu, Hirohisa Aki, Bo Zhao, Zhixin Miao, and Kai Strunz. A review of modeling, management, and applications of grid-connected li-ion battery storage systems. *IEEE Transactions on Smart Grid*, 13(6):4505–4524, 2022.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[Shen *et al.*, 2022] Rendong Shen, Shengyuan Zhong, Xin Wen, Qingsong An, Ruifan Zheng, Yang Li, and Jun Zhao. Multi-agent deep reinforcement learning optimization framework for building energy system with renewable energy. *Applied Energy*, 312:118724, 2022.

[Van Hasselt *et al.*, 2016] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

[Viswanathan *et al.*, 2019] Vijayaragavan Viswanathan, Lakshimi Narayanan Palaniswamy, and Padma Balaji Leelavinodhan. Optimization techniques of battery packs using re-configurability: A review. *Journal of Energy Storage*, 23:404–415, 2019.

[Wang *et al.*, 2023] Yongqi Wang, Yujie Zhao, Siyuan Zhou, Qingzhong Yan, Han Zhan, Yong Cheng, and Wei Yin. Impact of individual cell parameter difference on the performance of series–parallel battery packs. *ACS omega*, 8(11):10512–10524, 2023.

[Yang *et al.*, 2022] Feng Yang, Fei Gao, Baochang Liu, and Song Ci. An adaptive control framework for dynamically reconfigurable battery systems based on deep reinforcement learning. *IEEE Transactions on Industrial Electronics*, 69(12):12980–12987, 2022.

[Yang *et al.*, 2023] Xu Yang, Pei Liu, Fang Liu, Zhicheng Liu, Daoqi Wang, Jin Zhu, and Tongzhen Wei. A dod-soh balancing control method for dynamic reconfigurable battery systems based on dqn algorithm. *Frontiers in Energy Research*, 11:1333147, 2023.

[Zhu *et al.*, 2013] Cong Zhu, Xinghu Li, Lingjun Song, and Liming Xiang. Development of a theoretically based thermal model for lithium ion battery pack. *Journal of Power Sources*, 223:155–164, 2013.