DGRAF: Observation-Quality-Aware Reinforcement Learning for Dynamic Reconfigurable Batteries

Jingwei Hu **Jiasong Chen** **Zheng Fang** **Zhihong Zhang***

School of Informatics, Xiamen University

Abstract

Dynamic Reconfigurable Battery (DRB) systems enable flexible interconnection among cells through power electronic switches, crucial for electric vehicles and energy storage systems. However, sensor failures in practical deployments lead to partial observation loss, posing severe challenges for safe control. Existing methods inherently assume full state observability; consequently, when confronted with missing observations in practice, they typically resort to overly conservative heuristics that fail to balance safety and performance. More fundamentally, these methods lack mechanisms to adaptively adjust decision conservatism based on inference reliability, leading to unstable decisions under complex failure scenarios. We propose the **D**iffusion-**G**uided **R**isk-**A**daptive **F**ramework (DGRAF) to address this challenge. The framework handles spatiotemporally coupled failure patterns through joint diffusion inference with quality quantification, and integrates inference quality with distributional decision-making to enable automatic modulation between aggressive optimization and conservative maintenance. Through extensive simulations and hardware experiments, we demonstrate that DGRAF consistently outperforms baseline methods across complex failure patterns, achieving superior energy efficiency, decision stability, and risk resilience.

1 Introduction

Lithium-ion battery energy storage systems have become the core components in applications such as electric vehicles and grid energy storage, owing to their high energy density and long cycle life [1, 2]. Battery systems typically consist of multiple cells connected in series and parallel to meet power and capacity requirements. However, inevitable state inconsistencies among cells worsen with cycling, leading to reduced available capacity and safety hazards [3, 4]. Dynamic Reconfigurable Battery (DRB) systems provide an effective solution through power electronic switches that enable dynamic interconnection among cells, allowing real-time adjustment of series-parallel configurations for active balancing [5, 6].

Although DRB systems show great promise, sensor failures are inevitable in practical deployments [7]. Analysis of 300 electric vehicles over three years confirms that spatiotemporally coupled sensor failure patterns are pervasive in field deployments [8]. As shown in Figure 1, the left panel displays the observation status of 5 battery cells across 5 consecutive timesteps (k-4 to k), exhibiting two typical characteristics: (i) **Spatial correlation**: Batteries 2-3 fail simultaneously during k-3 to k-1, typically caused by shared communication bus faults or adjacent battery sampling circuit damage [9, 10]; (ii) **Temporal persistence**: Battery 1 remains unobservable for 5 consecutive steps, with industrial scenarios showing fault persistence of 500 seconds or longer [11]. Such spatiotemporally coupled failure patterns result in vastly different inference difficulties, and traditional methods struggle to

*Corresponding author: zhihong@xmu.edu.cn

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

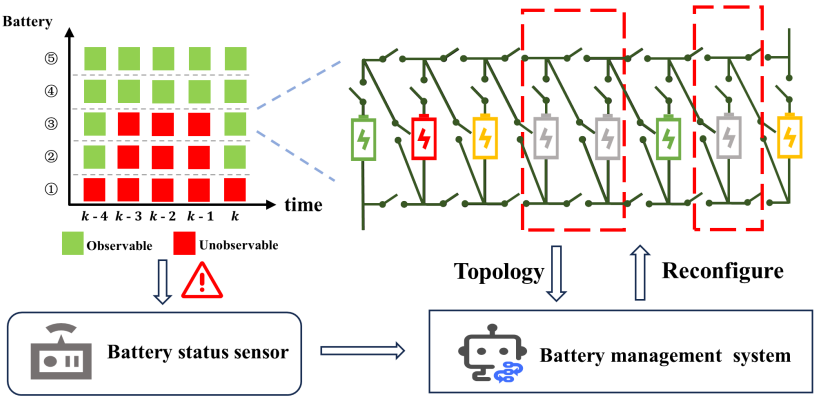


Figure 1: Illustration of DRB system with sensor failures. Left: battery pack with observable (green) and unobservable (red) cells due to sensor failures. Right: reconfigurable topology with switches enabling dynamic series-parallel connections.

simultaneously exploit spatial proximity constraints and historical spatiotemporal windows. As the number of cells increases, feasible topology configurations grow exponentially. Moreover, incorrect decisions based on uncertain state estimates may lead to low energy utilization or potential safety risks.

These sensor failure scenarios fundamentally constitute a partially observable Markov decision process (POMDP), where safe decisions must be made despite incomplete state information. One class of methods models belief states through variational inference or recurrent networks [12, 13], but struggles to simultaneously exploit spatial and temporal constraints under spatiotemporally coupled failures. Another class focuses on decision optimization under safety constraints. Distributional reinforcement learning provides a framework for modeling uncertainty [14, 15, 16]. Recent work has performed safe planning in belief space [17] and domain randomization techniques enhance robustness [18], yet both fail to establish the connection between observation quality and policy conservatism. The fundamental limitation of these methods is that they treat state inference and risk control as independent problems, failing to adaptively adjust decision conservatism based on the reliability of state estimates. This leads to unstable decision behaviors in complex failure scenarios.

This paper proposes the Diffusion-Guided Risk-Adaptive Framework (DGRAF), achieving an adaptive trade-off between risk and performance under incomplete observations. We leverage diffusion models to recover missing states from the spatiotemporal correlation structure of battery packs, whose iterative denoising process can fully utilize constraint information from adjacent batteries and historical time series to provide robust estimation under complex failures. More importantly, even in cases where inference is extremely difficult (such as when failures persist for long durations and excessive missing observations lead to degraded reconstruction quality), our framework can still automatically identify high-risk actions and adopt conservative strategies by learning complete value distributions. Under different failure patterns, DGRAF consistently improves battery balance states, effectively enhancing the robustness and energy utilization of DRB systems.

Our framework addresses two key research questions: (i) how to perform robust state inference and quantify inference quality under spatiotemporally coupled failure patterns; (ii) how to enable the decision module to adaptively adjust behavior based on inference reliability, pursuing aggressive optimization when observations are sufficient and adopting conservative maintenance when observations are missing. The main contributions of this paper are as follows:

- We identify the dual impact of spatiotemporally coupled sensor failures on inference difficulty and decision stability in DRB systems.
- We propose the first DRB safety control framework for sensor failure scenarios (DGRAF), unifying partially observable state inference with risk-sensitive decision-making.
- DGRAF achieves robust state reconstruction under spatiotemporally coupled failures and adaptive risk adjustment, effectively addressing safe decision-making under partial observability.

- We validate the effectiveness and robustness of the proposed method under complex failure patterns on both simulation environments and real hardware platforms.

2 Related Work

2.1 Dynamic Reconfigurable Battery Systems

DRB systems achieve flexible interconnection among battery cells through power electronic switches, enabling dynamic adjustment of battery pack topology according to operational requirements. Compared to traditional fixed-topology battery systems, DRB systems demonstrate significant advantages in battery balancing, fault tolerance, and energy efficiency optimization [5, 19].

Early research mainly employed traditional optimization methods to determine optimal topology configurations. Dynamic programming-based battery balancing schemes have been proposed [4], determining optimal switch sequences by solving shortest path problems. Model predictive control frameworks have also been adopted [20], formulating topology reconfiguration problems as constrained optimization problems solved online. These methods can theoretically obtain optimal solutions but typically require model simplification to reduce computational complexity. As the number of battery cells increases, the configuration space grows exponentially, making computational overhead of traditional optimization methods prohibitive.

Reinforcement learning methods provide new solutions for DRB system control. Deep Q-networks have been applied to DRB systems [21], leveraging reinforcement learning's model-free characteristics to avoid requirements for precise battery models. To handle large-scale discrete action spaces, control algorithms based on factored Q-learning have been proposed [22], decomposing composite actions into multiple sub-actions to reduce learning difficulty. Variational autoencoders have been combined with reinforcement learning [23], using generative models to learn low-dimensional representations of action spaces. Recent advances further introduce event-triggered mechanisms [24] and multi-objective optimization via Pareto optimality [25, 26].

These methods generally assume full state observability, but in practical deployments, sensor failures, communication delays, and packet losses inevitably lead to partial observation loss, which existing basic methods fail to address.

2.2 Partially Observable Safe Reinforcement Learning

Partially observable Markov decision processes (POMDPs) provide a standard mathematical framework for sequential decision problems with incomplete state observability [27]. The development of deep learning has brought new solution approaches to POMDP problems. Deep recurrent Q-networks (DRQN) have been proposed [12], using LSTM to extract hidden state representations from observation sequences. Deep variational reinforcement learning methods have also been developed [13], learning approximate posterior distributions of belief states through variational inference. Systematic comparisons reveal that networks with recurrent structures remain more advantageous for POMDP problems [28]. Recently, diffusion models have been introduced to state estimation due to their powerful distribution modeling capabilities, with methods using conditional diffusion models to recover true states from perturbed observations [29]. However, these methods mainly focus on state inference accuracy without considering the impact of inference uncertainty on downstream decisions.

When systems have safety constraints, POMDP problem solving becomes more complex. Safe reinforcement learning handles safety requirements through constrained optimization or risk measures [30, 31]. Distributional reinforcement learning [14, 32] provides a natural framework for modeling outcome uncertainty by learning complete return distributions rather than point estimates. Risk-sensitive objectives have been applied to guide safe exploration [15, 16], although these methods generally cannot modulate their decision sensitivity on a per-state basis during training.

POMDP inference has been combined with robust reinforcement learning [18], but such methods handle model uncertainty through domain randomization, not explicitly modeling observation quality's impact on decisions. Algorithms learning failure probabilities under given belief states through neural networks have been proposed [17], but require online execution of Monte Carlo tree search with large computational overhead. Shielding-based methods have also been developed [33] but rely on known environment models. However, existing methods struggle to adaptively adjust policy

conservatism according to dynamically changing observation quality, limiting their applicability in complex failure scenarios.

3 Problem Statement

A DRB system composed of n battery cells is considered, denoted as $(b_1, b_2, \ldots, b_n)$. The state of charge $\mathrm{SOC}_i^t \in [0, 1]$ of each battery at time t is calculated based on observed voltage and current, and updated as follows:

$$\mathrm{SOC}_i^t = \mathrm{SOC}_i^{t-1} + c \cdot u_i \cdot \Delta\mathrm{SOC}_i^{t-1} \tag{1}$$

where $c \in \{-1, 1\}$ indicates charge/discharge direction, $u_i \in \{0, 1\}$ indicates whether battery i is connected to the circuit, and $\Delta\mathrm{SOC}_i^{t-1} = \frac{I_i^{t-1} \cdot \Delta t}{Q_i}$ denotes the single-step SOC variation determined by cell current I_i^{t-1}, control interval Δt, and nominal capacity Q_i. The system contains $4n - 3$ switches (each cell requires 4 switching ports for bidirectional series-parallel connection, with shared endpoints at the two terminals of the pack reducing the count by 3), achieving different topology configurations by controlling switch states.

Under normal operation, the controller can observe all battery states, i.e., $o_t = s_t$. Sensor failures lead to partial observation losses, represented by mask $m_t \in \{0, 1\}^n$:

$$o_t = m_t \odot s_t \tag{2}$$

Beyond single-cell random failures, adjacent batteries may fail simultaneously with probability p_{adj}, with failures persisting for $T_{\text{fault}} \sim \mathcal{U}[T_{\min}, T_{\max}]$ steps.

The problem is formalized as a POMDP $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{O}, \gamma)$, where $\mathcal{A}$ denotes the switch configuration action space, $\mathcal{O}{=}m_t{\odot}s_t$ is the partial observation and $\mathcal{R}$ is defined in Appendix C.3. The policy operates on augmented surrogate $\mathbf{z}_t{=}[\hat{\mathbf{s}}_t; \mathbf{q}_t]$, where $\mathbf{q}_t$ is the quality score; without diffusion, $\mathbf{z}_t$ degrades to $\mathbf{o}_t$. The objective is to find a policy that balances energy efficiency and safety under incomplete observations.

4 Method

4.1 Framework Overview

In this section, the proposed DGRAF is introduced in detail. As shown in Figure 2, DGRAF consists of two main components: (i) **Joint Diffusion State Inference**, which reconstructs complete states from partial observations through window-based iterative denoising, outputting state estimates with quality scores that quantify inference reliability, and (ii) **Quality-Conditioned Distributional Decision**, which learns return distributions conditioned on quality-augmented states, enabling implicit risk adaptation where high quality triggers aggressive actions and low quality induces conservative behaviors. These two components are tightly coupled: inference quality scores directly guide the decision module to adaptively balance between performance optimization and safety assurance under varying observation conditions.

4.2 Joint Diffusion State Inference

4.2.1 Window Diffusion Modeling

To reconstruct complete system states from partial observations with spatiotemporal dependencies, we employ diffusion probabilistic models. Their iterative denoising process naturally captures nonlinear spatiotemporal constraints without requiring explicit prior specification, offering an advantage over conventional Bayesian or Gaussian state-space models in complex failure settings. This design enables joint inference within a temporal window, simultaneously reconstructing states across $L+1$ timesteps. Define stacked state $\mathbf{s} = [\mathbf{s}_{t-L}^\top, \ldots, \mathbf{s}_t^\top]^\top \in \mathbb{R}^{(L+1)n}$, where n is the state dimension per timestep, stacked observation $\mathbf{o} = [\mathbf{o}_{t-L}^\top, \ldots, \mathbf{o}_t^\top]^\top$, and historical actions $\mathbf{a} = [\mathbf{a}_{t-L}^\top, \ldots, \mathbf{a}_{t-1}^\top]^\top$. The observation mask $\mathbf{m} = [\mathbf{m}_{t-L}^\top, \ldots, \mathbf{m}_t^\top]^\top \in \{0, 1\}^{(L+1)n}$ stacks per-timestep masks $\mathbf{m}_t \in \{0, 1\}^n$, with $m_{i,j}$ denoting the mask value at timestep i and position j.

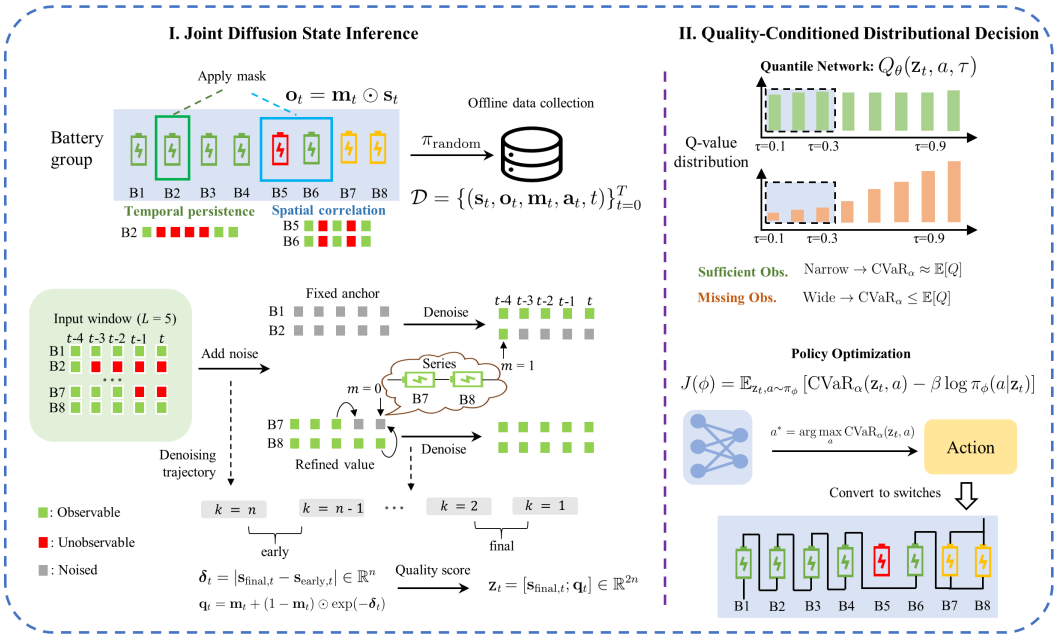


Figure 2: Overview of the DGRAF framework. The trajectories with simulated sensor failures (top-left) train the joint diffusion inference module (bottom-left), which processes observation windows through iterative denoising to produce state estimates and quality scores. These quality-augmented representations feed into the distributional decision module (right), where CVaR-based policy optimization enables automatic risk adaptation based on observation reliability.

We model the forward diffusion process as gradually adding Gaussian noise to states over K steps, formalized as:

$$q(\mathbf{s}^{(k)}|\mathbf{s}^{(k-1)}) = \mathcal{N}(\mathbf{s}^{(k)}; \sqrt{1-\beta_k}\mathbf{s}^{(k-1)}, \beta_k \mathbf{I}) \tag{3}$$

where $k \in \{1, \ldots, K\}$ denotes the diffusion step and $\{\beta_k\}_{k=1}^K$ is the predefined noise schedule, with $\alpha_k = 1 - \beta_k$. The reverse denoising process takes observations and actions as conditions, predicting the added noise component through denoising network ϵ_ψ to model the conditional distribution $p_\psi(\mathbf{s}^{(0)}|\mathbf{o}, \mathbf{m}, \mathbf{a})$. The training objective is the denoising score matching loss:

$$\mathcal{L}_{\text{DSM}} = \mathbb{E}_{k, \boldsymbol{\epsilon}, \mathbf{s}, \mathbf{o}, \mathbf{m}, \mathbf{a}} \left[\|\boldsymbol{\epsilon} - \epsilon_\psi(\mathbf{s}^{(k)}, k, \mathbf{o}, \mathbf{m}, \mathbf{a})\|_2^2 \right] \tag{4}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ represents the noise added in the forward process. By explicitly conditioning on observation mask $\mathbf{m}$, the network learns to distinguish between positions with and without observations.

During inference, we perform iterative denoising from pure noise $\mathbf{s}^{(K)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. At each iteration $k = K, K-1, \ldots, 1$, the network predicts noise $\hat{\boldsymbol{\epsilon}} = \epsilon_\psi(\mathbf{s}^{(k)}, k, \mathbf{o}, \mathbf{m}, \mathbf{a})$ and computes $\mathbf{s}^{(k-1)}$ via DDPM update. We then apply hard constraints:

$$\mathbf{s}_{i,j}^{(k-1)} \leftarrow \mathbf{o}_{i,j}, \quad \forall (i,j) \text{ s.t. } m_{i,j} = 1 \tag{5}$$

forcibly setting observed positions as fixed anchors; for unobserved positions ($m_{i,j} = 0$), no constraint is applied and their values are instead completed by the reverse diffusion process across all K iterations, leveraging the observed anchors as boundary conditions. The network progressively refines these missing estimates from random noise through iterative denoising, leveraging temporal continuity from adjacent timesteps, spatial correlations from other batteries informed by topological configurations encoded in action sequences, and system dynamics patterns. After K iterations, state estimates for all $(L+1)$ timesteps within the window converge to temporally consistent inference results.

4.2.2 Inference Quality Quantification

To quantify state estimate reliability at each position, we leverage the progressive refinement nature of diffusion denoising to extract quality indicators from the inference trajectory. Diffusion denoising iteratively refines estimates through multiple correction steps, where correction magnitudes reflect inference difficulty: positions near observations require small adjustments, while failure centers undergo substantial refinements.

We extract state estimates at multiple intermediate steps to capture this trajectory characteristic. For the current timestep t, we sample estimates at early stages $k \in \{k_1, k_2, \ldots, k_m\}$ where $k_i > K/2$ and late stages $k \in \{k_{m+1}, \ldots, k_M\}$ where $k_j < K/2$, obtaining $\mathbf{s}_t^{(k_i)} \in \mathbb{R}^n$ through the denoising network reparameterization. We then compute the average of early estimates $\mathbf{s}_{\text{early},t} = \frac{1}{m}\sum_{i=1}^{m} \mathbf{s}_t^{(k_i)}$ and late estimates $\mathbf{s}_{\text{final},t} = \frac{1}{M-m}\sum_{j=m+1}^{M} \mathbf{s}_t^{(k_j)}$. Their difference reflects the correction magnitude, formalized as:

$$\boldsymbol{\delta}_t = |\mathbf{s}_{\text{final},t} - \mathbf{s}_{\text{early},t}| \in \mathbb{R}^n \tag{6}$$

where $|\cdot|$ denotes element-wise absolute value. Small corrections indicate easy inference with nearby observations, while large corrections reflect high uncertainty in failure centers.

To construct a unified quality representation $\mathbf{q}_t$, we combine correction magnitudes with observation masks through an exponential mapping:

$$\mathbf{q}_t = \mathbf{m}_t + (1 - \mathbf{m}_t) \odot \exp(-\boldsymbol{\delta}_t) \tag{7}$$

where $\odot$ denotes element-wise multiplication and $\mathbf{m}_t \in \{0, 1\}^n$ is the observation mask at timestep t. Observed positions are assigned value 1; unobserved positions are mapped to $(0, 1)$ via $\exp(-\delta_{t,i})$, monotonically decreasing with correction magnitude. The inference module outputs augmented state representation $\mathbf{z}_t = [\mathbf{s}_{\text{final},t}; \mathbf{q}_t] \in \mathbb{R}^{2n}$, which the decision module takes as input to adjust policy behavior.

4.3 Quality-Conditioned Distributional Decision

4.3.1 Distributional Value Learning

To capture return uncertainty under varying inference quality, we adopt a distributional RL framework [14] that parameterizes complete return distributions, capturing tail risks imperceptible to expectation-based policies.

The quantile function $Q_\theta(\mathbf{z}_t, a, \tau)$ takes augmented state representation $\mathbf{z}_t = [\mathbf{s}_{\text{final},t}; \mathbf{q}_t]$ and quantile level $\tau \in (0, 1)$ as inputs, outputting return values at that quantile. The training objective is formalized as the quantile regression loss:

$$\begin{aligned}\mathcal{L}_{\text{QR}}(\theta) = \mathbb{E}_{\tau,(\mathbf{z}_t,a,r,\mathbf{z}_{t+1})}\Bigg[\rho_\tau\Bigg(r + \gamma Q_{\theta'}(\mathbf{z}_{t+1}, a', \tau) \\ - Q_\theta(\mathbf{z}_t, a, \tau)\Bigg)\Bigg]\end{aligned} \tag{8}$$

where $\rho_\tau(u) = |\tau - \mathbb{I}\{u < 0\}| \cdot |u|$ is the quantile regression loss with $\mathbb{I}\{\cdot\}$ being the indicator function, γ is the discount factor, θ' denotes the target critic parameters, and $a' = \pi_{\phi'}(\mathbf{z}_{t+1})$ denotes the action output by the target actor, where ϕ' denotes the target actor parameters. This loss ensures accurate quantile estimation through asymmetric penalties.

Through training on augmented representations $\mathbf{z}_t$, the network encodes observation quality into distribution shapes: high-quality inputs yield narrow distributions; low-quality inputs yield wide distributions reflecting tail risks.

4.3.2 Adaptive Risk-Sensitive Optimization

The risk-aware policy objective is formalized as:

$$J(\phi) = \mathbb{E}_{\mathbf{z}_t, a\sim\pi_\phi}\left[\int_0^1 w(\tau) Q_\theta(\mathbf{z}_t, a, \tau) d\tau + \beta H[\pi_\phi(\cdot|\mathbf{z}_t)]\right] \tag{9}$$

where $w(\tau) = \frac{1}{\alpha}\mathbb{I}\{\tau \leq \alpha\}$ is a *predefined* weighting function with fixed $\alpha \in (0, 1)$, concentrating weight on the lowest α-quantiles to penalize downside risk, H denotes the entropy term, and β is the temperature coefficient. In discrete form: $\text{CVaR}_\alpha = \frac{1}{N}\sum_{i=1}^{N} Q_\theta(\mathbf{z}_t, a, \tau_i)$, with $\tau_i \sim \mathcal{U}[0, \alpha]$.

The fixed tail weighting combined with quality-conditioned distributional learning achieves implicit risk adaptation. High observation quality produces narrow distributions where tail and mean values align, leading to aggressive optimization; low quality produces wide distributions with large tail-mean gaps, automatically inducing conservative behaviors. This design allows risk sensitivity to adapt dynamically; CVaR is a coherent risk measure with well-established theoretical properties [34], providing principled tail-risk penalization suited to DRB safety requirements.

5 Experiments

We evaluate DGRAF across six questions: **Q1** performance versus state-of-the-art methods, **Q2** balancing performance under partial observability, **Q3** component ablation, **Q4** mechanism verification, **Q5** out-of-distribution (OOD) robustness, and **Q6** hardware validation (Appendix B.1).

Datasets and Environment. The DRB system is modeled and simulated in MATLAB/Simulink with 8 cells, 29 switches, and initial SOCs of 85%–100%. To respond to fluctuating external power demands, the system adjusts the number of series-connected cells, N. With the load as a constant current sink, N is sampled from a truncated normal distribution $\mathcal{N}(4, 1.5^2)$ and constrained to the integer range [1, 8], thereby modulating the total power output. The simulation is executed with 150 steps per discharge. Following prior work [10, 11], the simulated dataset covers typical spatiotemporally coupled failures: (i) Fully observable (30%), all sensors normal; (ii) Mild failures (40%), failure probability 0.01, duration 2–10 steps, adjacent failure probability 0.01; (iii) Severe failures (30%), failure probability 0.03, duration 5–20 steps, adjacent failure probability 0.03. For hardware validation, a physical DRB testbed equipped with INR 18650-20R cells is constructed. The trained policy is deployed on the real hardware, where control decisions are executed in real time on the actual battery system to verify practical deployability.

Evaluation Metrics. We use three complementary metrics: (i) Energy (Wh): Total energy release within the horizon, $E_{\text{total}} = \sum_{t=1}^{T} V_t \cdot I_t \cdot \Delta t$. (ii) Improvement Maintenance Rate (IMR, %): Percentage of steps where SOC difference decreases or remains unchanged. A step is improvement if $\Delta_{\text{SOC}}(t) \leq \Delta_{\text{SOC}}(t-1)$, where $\Delta_{\text{SOC}}(t) = \max_i(\text{SOC}_i(t)) - \min_i(\text{SOC}_i(t))$. (iii) Final SOC Diff (%): Final SOC imbalance $\Delta_{\text{SOC}}(T)$, lower is better.

Baselines. We select baselines from five categories: **Default**: Fixed topology, Naive (stops upon failure detection); **Traditional RL**: PPO [35], TD3 [36]; **Variational inference**: DVRL [13], Belief-RL [37]; **Risk-sensitive**: WCSAC [15]; **Diffusion inference**: DMBP [29].

5.1 Main Results (Q1)

Table 1: Performance comparison under mild and severe failures (mean ± std over 5 runs)

Metric (Failure)	PPO	TD3	DVRL	Belief-RL	WCSAC	DMBP	DGRAF
Energy (Wh, Mild)	3.58±0.31	4.25±0.27	5.08±0.18	5.47±0.16	5.69±0.14	6.11±0.10	**6.26±0.06**
Energy (Wh, Severe)	2.92±0.39	3.89±0.33	4.75±0.22	5.15±0.20	5.43±0.18	6.02±0.12	**6.08±0.08**
IMR (%, Mild)	46.75±4.23	60.25±3.54	73.92±2.67	75.84±2.41	79.12±1.85	84.14±1.32	**88.35±0.94**
IMR (%, Severe)	41.26±5.18	59.34±4.12	72.33±3.15	74.26±2.87	78.45±2.34	81.56±1.68	**85.64±1.23**
Final SOC Diff (%, Mild)	7.02±0.68	5.68±0.53	3.96±0.37	3.85±0.35	1.55±0.17	1.43±0.13	**1.00±0.08**
Final SOC Diff (%, Severe)	8.15±0.79	6.95±0.62	4.13±0.43	4.22±0.40	2.68±0.24	1.92±0.16	**1.03±0.10**

To address Q1, we evaluate DGRAF against SOTA baseline methods under both mild and severe failure conditions, with main results shown in Table 1 and Figure 3. We summarize the observations:

Obs. 1. DGRAF consistently outperforms baseline methods across all metrics. Under mild failures, DGRAF achieves 88.3% IMR and 6.26 Wh, outperforming DMBP by 4.2 points. Under severe failures, degradation is minimal (energy -2.9%, IMR -2.7 pts), whereas WCSAC's energy drops by 4.6%, indicating reduced adaptability under high uncertainty. Notably, DGRAF under severe failures (85.6%) surpasses WCSAC under mild failures (79.1%), validating the quality-aware mechanism.

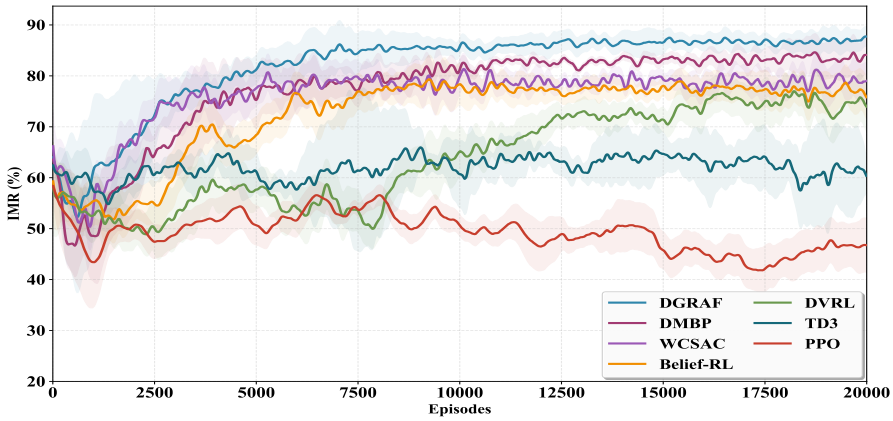


Figure 3: Improvement maintenance rate under different algorithms.

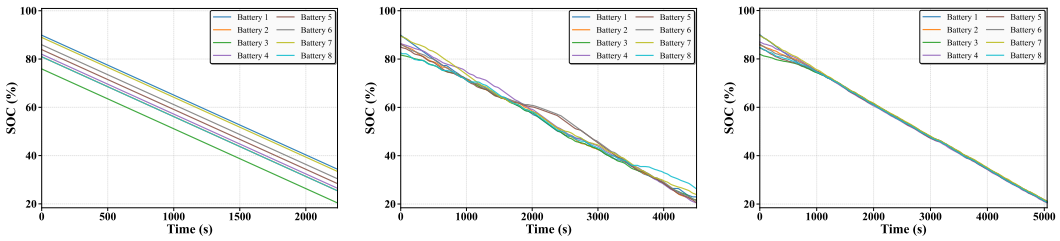
Obs. 2. Traditional reinforcement learning methods perform poorly in partially observable scenarios. PPO fails to converge (IMR ≈46.7%, declining to 41.3% under severe failures). TD3 adopts recurrent networks for partial observability but reaches only 60.3% with unstable training. Both methods lack explicit observation quality modeling, preventing them from distinguishing reliable from uncertain state estimates.

Obs. 3. Variational inference methods show improvement by modeling belief states, but training processes still exhibit oscillations. DVRL reaches 73.9% and Belief-RL 75.8% by decoupling belief modeling from policy optimization, but both exhibit training oscillations under complex failures. Under severe failures, Belief-RL's IMR decreases by 1.58 points, yet oscillations reflect insufficient decision consistency, as neither method leverages inference quality to guide conservatism.

Obs. 4. Diffusion inference methods demonstrate strong learning capabilities, but DGRAF's quality-aware mechanism provides a critical advantage. DMBP converges to 84.1% (81.6% under severe failures), demonstrating diffusion's reconstruction advantage. However, without quality modeling, the policy cannot perceive the reliability of state estimates. DGRAF maintains a 4.1-point advantage (85.6% vs. 81.6%) with a consistently smoother training curve.

Combining the above observations, DGRAF significantly outperforms existing methods in training efficiency, final performance, and robustness, with its performance degradation magnitude smaller than all baseline methods, fully validating the effectiveness of quality-aware and adaptive decision-making mechanisms.

5.2 System Operation Performance (Q2)



(a) Fixed (b) Naive (c) Ours

Figure 4: SOC trajectory comparison under different control strategies. (a) Fixed topology: all batteries in series, SOC differences persist throughout discharge. (b) Naive method: stops reconfiguration when sensor failures detected, resulting in 6.61% final SOC difference. (c) DGRAF: continues optimizing topology under partial observability, achieving effective balancing with final SOC difference $<1\%$.

For Q2, Figure 4 compares the complete discharge SOC trajectories of the three configurations under the same initial conditions until discharge stops at the lower limit.

Obs. 5. Fixed topology and Naive method fail to achieve effective balancing under partial observability. Fixed topology (Figure 4a) connects batteries in series, causing persistent SOC differences. Naive method (Figure 4b) stops reconfiguration upon failure detection, degrading to fixed-like behavior with 6.61% final SOC difference. This shows simple stop strategies sacrifice balancing capability.

Obs. 6. DGRAF maintains balancing capability under partial observability, achieving significant improvement. As shown in Figure 4c, DGRAF continues optimizing topology throughout operation. SOC differences decrease to low levels ($<$1%). This demonstrates quality-aware inference and adaptive mechanisms enable effective balancing under uncertainty.

5.3 Ablation Study (Q3)

Table 2: Contribution of each component (mean $\pm$ std over 5 runs)

Metric (Failure)	w/o Diffusion	w/o Window	w/o Quality	w/o Tail-weight	Full DGRAF
Energy (Wh, Mild)	4.85±0.22	5.38±0.17	5.95±0.12	6.09±0.09	**6.26±0.06**
Energy (Wh, Severe)	4.69±0.26	5.15±0.20	5.58±0.15	6.02±0.10	**6.08±0.08**
IMR (%, Mild)	64.73±3.42	72.85±2.58	75.34±2.03	83.54±1.47	**88.35±0.94**
IMR (%, Severe)	68.36±3.87	71.52±3.12	73.35±2.64	81.34±1.85	**85.64±1.23**
Final SOC Diff (%, Mild)	4.83±0.45	2.45±0.25	1.87±0.18	1.41±0.12	**1.00±0.08**
Final SOC Diff (%, Severe)	4.74±0.51	3.52±0.33	2.59±0.24	1.15±0.14	**1.03±0.10**

To address Q3, we conduct systematic ablation experiments under both mild and severe failure conditions, generating 5 variants. Table 2 shows the results. We have the following observations:

Obs. 7. Removing the diffusion model causes the most significant performance degradation, validating the critical role of spatiotemporal reasoning. "w/o Diffusion" (LSTM) drops to 4.85 Wh and 64.7% IMR under mild failures (-22.5%, -23.6 pts), with degradation persisting under severe failures (-22.9%, -17.3 pts), confirming that diffusion's iterative denoising is crucial for exploiting spatiotemporal constraints under coupled failures.

Obs. 8. Removing window inference reduces the ability to handle persistent failures. "w/o Window" achieves 5.15 Wh under severe failures (-15.3%), with degradation intensifying with failure duration. Single-timestep inference lacks historical context during persistent failures, whereas window inference enables cross-timestep consistency that stabilizes inference quality.

Obs. 9. Removing quality quantification degrades performance, highlighting the necessity of feeding observation quality to the decision network. "w/o Quality" suffers a 12.3-point IMR drop under severe failures (85.6%$\rightarrow$73.4%) and an 8.2% energy decline. Without explicit quality signals, policies cannot distinguish reliable from uncertain estimates, leading to aggressive strategies under low-quality observations or excessive conservatism under high-quality ones.

Obs. 10. Removing tail-weighted aggregation has smaller but significant impact. "w/o Tail-weight" drops 4.8 and 4.3 IMR points under mild and severe failures. Quality-conditioned distributional learning shapes value distributions to reflect observation reliability, while tail-weighted CVaR aggregation translates this into risk-aware action selection, and both components are necessary.

Overall, removing any component leads to consistent performance degradation, which is more pronounced under severe failures. Imperfect decisions may further amplify sensor failures into system-level risks such as load mismatch or hazardous topologies. DGRAF achieves optimal performance under both conditions, confirming the necessity and synergy of all components.

5.4 Quality Calibration and Risk Adaptation (Q4)

To answer Q4, this work examines whether the model's gains stem from quality-optimized risk adaptation. First, we analyze whether the trajectory correction term $\boldsymbol{\delta}_t$ and the quality score $\mathbf{q}_t$ can effectively reflect the true reconstruction error e_t. Second, we analyze whether q_t indeed modulates the quantile distribution of the decision network in risk adaptation. Specifically, ρ denotes

the Spearman rank correlation coefficients with e_t and their p-values are used to assess statistical significance. Mean $\mathbf{q}_t$ denotes the population mean quality score, and IQR denotes the interquartile range of the reported predictions, indicating whether the majority of the decision distribution is concentrated.

Table 3: Verification of quality score calibration with true error e_t (Left) and return distribution spread across quality quartiles (Right).

Failure Regime	$\rho(\delta_t, e_t)$	$\rho(q_t, e_t)$	p-value
Mild Failure	0.63	-0.63	< 0.001
Severe Failure	0.76	-0.76	< 0.001
Overall	0.68	-0.68	< 0.001

Quartile Group	**Mean $\mathbf{q}_t$**	**IQR**
Q1 (Low)	0.36	0.42
Q2 / Q3	0.52 / 0.72	0.32 / 0.28
Q4 (High)	0.80	0.22

Obs. 11. Denoising trajectory discrepancy is well-calibrated with true reconstruction error. During inference, hard constraints preserve observed states and multi-step averaging for $\mathbf{s}_{\text{early},t}$ mitigates single-step stochasticity. Table 3 shows that $\boldsymbol{\delta}_t$ correlates with e_t, reaching $\rho = 0.63$ under mild failures, $\rho = 0.76$ under severe failures, and $\rho = 0.68$ overall with $p < 0.001$. Correspondingly, $\mathbf{q}_t$ exhibits an inverse correlation, confirming that it reliably reflects reconstruction accuracy without ground-truth supervision.

Obs. 12. Observation quality is directly reflected in the quantile distribution spread. As Mean $\mathbf{q}_t$ increases across quartiles from 0.36 in Q1 to 0.52 in Q2, 0.72 in Q3, and 0.80 in Q4, the IQR monotonically decreases from 0.42 down to 0.32, 0.28, and 0.22. A larger IQR under low $\mathbf{q}_t$ indicates high return dispersion and an unstable state, whereas a narrower IQR under high $\mathbf{q}_t$ reflects concentrated, stable predictions. This reduction in IQR confirms that DGRAF effectively distinguishes between stable and unstable operating conditions through its output distribution.

5.5 Out-of-Distribution Robustness (Q5)

In response to Q5, we conduct OOD tests under unseen stress conditions, including spatial fault propagation , specifically simulating fault propagation among four adjacent cells, extending fault duration to persist between thirty and fifty time steps, and combining them.

Table 4: Performance comparison under OOD failure regimes (IMR in %, mean $\pm$ std over 5 runs).

Stressor Regime	**Belief-RL**	**WCSAC**	**DMBP**	**DGRAF**
OOD-Spatial	70.58 ± 3.21	74.23 ± 2.68	78.82 ± 2.05	$\mathbf{82.22 \pm 1.42}$
OOD-Temporal	71.93 ± 3.04	75.91 ± 2.51	80.22 ± 1.38	$\mathbf{83.01 \pm 1.18}$
OOD-Combined	67.36 ± 3.47	70.84 ± 2.96	76.32 ± 2.15	$\mathbf{81.05 \pm 1.71}$

Obs. 13. DGRAF maintains robust control performance under unseen out-of-distribution stressors. As shown in Table 4, all baselines suffer performance degradation under unseen OOD patterns due to fixed uncertainty bounds. DGRAF sustains superior performance, achieving 82.22% IMR under spatial propagation, 83.01% under temporal durations, and 81.05% under the combined stressor. This confirms that coupling diffusion correction with distributional risk adaptation provides relatively robust generalization for complex degradation conditions.

6 Conclusion

This paper investigates safe control of Dynamic Reconfigurable Battery (DRB) systems under sensor failure scenarios. Through joint diffusion inference and quality-conditioned distributional decision-making, we address the contradiction between state uncertainty and safety under partial observability conditions, achieving observation quality-aware adaptive control. Experiments demonstrate that DGRAF maintains high energy utilization and balancing performance under complex failure patterns, with performance significantly superior to existing methods in severe failure scenarios. Additionally, we validated the practicality of the proposed method on real hardware platforms. Future work will extend this framework to larger-scale battery systems, explore distributed control and transfer learning, and investigate dynamic battery replacement scenarios.

Acknowledgments and Disclosure of Funding

This work is supported by the National Natural Science Foundation of China (62676337) and Xiamen Major Science and Technology Project (No. 3502Z20241028).

References

[1] Languang Lu, Xuebing Han, Jianqiu Li, Jianfeng Hua, and Minggao Ouyang. A review on the key issues for lithium-ion battery management in electric vehicles. *Journal of Power Sources*, 226:272–288, 2013.

[2] Jingyuan Zhao, Xuning Feng, Quanquan Pang, Michael Fowler, Yubo Lian, Minggao Ouyang, and Andrew F Burke. Battery safety: Machine learning-based prognostics. *Progress in Energy and Combustion Science*, 102:101142, 2024.

[3] Haifeng Dai, Bo Jiang, Xiaosong Hu, Xianke Lin, Xuezhe Wei, and Michael Pecht. Advanced battery management strategies for a sustainable energy future: Multilayer design concepts and research trends. *Renewable and Sustainable Energy Reviews*, 138:110480, 2021.

[4] 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.

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

[6] Taesic Kim, Wei Qiao, and Liyan Qu. Series-connected self-reconfigurable multicell battery. In *2011 Twenty-Sixth Annual IEEE Applied Power Electronics Conference and Exposition (APEC)*, pages 1382–1387. IEEE, 2011.

[7] Zhentong Liu and Hongwen He. Sensor fault detection and isolation for a lithium-ion battery pack in electric vehicles using adaptive extended Kalman filter. *Applied Energy*, 185:2033–2044, 2017.

[8] Hongao Liu, Chang Li, Xiaosong Hu, Jinwen Li, Kai Zhang, Yang Xie, Ranglei Wu, and Ziyou Song. Multi-modal framework for battery state of health evaluation using open-source electric vehicle data. *Nature Communications*, 16(1):1137, 2025.

[9] Nicolae Tudoroiu, Mohammed Zaheeruddin, Roxana-Elena Tudoroiu, and Sorin Mihai Radu. Fault detection, diagnosis, and isolation strategy in Li-ion battery management systems of HEVs using 1-D wavelet signal analysis. In *Wavelet Theory*, page 123. IntechOpen, 2020.

[10] Jingzhao Zhang, Yanan Wang, Benben Jiang, Haowei He, Shaobo Huang, Chen Wang, Yang Zhang, Xuebing Han, Dongxu Guo, Guannan He, and Minggao Ouyang. Realistic fault detection of Li-ion battery via dynamical deep learning. *Nature Communications*, 14(1):5940, 2023.

[11] Jingyuan Zhao, Xuning Feng, Junbin Wang, Yubo Lian, Minggao Ouyang, and Andrew F Burke. Battery fault diagnosis and failure prognosis for electric vehicles using spatio-temporal transformer networks. *Applied Energy*, 352:121949, 2023.

[12] Matthew J Hausknecht and Peter Stone. Deep recurrent Q-learning for partially observable MDPs. In *AAAI Fall Symposia*, volume 45, page 141, 2015.

[13] Maximilian Igl, Luisa Zintgraf, Tuan Anh Le, Frank Wood, and Shimon Whiteson. Deep variational reinforcement learning for POMDPs. In *Proceedings of the 35th International Conference on Machine Learning*, pages 2117–2126, 2018.

[14] Will Dabney, Mark Rowland, Marc Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pages 2892–2901, 2018.

[15] Qisong Yang, Thiago D Simão, Simon H Tindemans, and Matthijs TJ Spaan. WCSAC: Worst-case soft actor critic for safety-constrained reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10639–10646, 2021.

[16] Mehrdad Moghimi and Hyejin Ku. Beyond CVaR: leveraging static spectral risk measures for enhanced decision-making in distributional reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning*, pages 44571–44593, 2025.

[17] Robert J. Moss, Arec Jamgochian, Johannes Fischer, Anthony Corso, and Mykel J. Kochenderfer. ConstrainedZero: chance-constrained POMDP planning using learned probabilistic failure surrogates and adaptive safety constraints. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 6752–6760, 2024.

[18] Giacomo Arcieri, Cyprien Hoelzl, Oliver Schwery, Daniel Straub, Konstantinos G Papakonstantinou, and Eleni Chatzi. POMDP inference and robust solution via deep reinforcement learning: An application to railway optimal maintenance. *Machine Learning*, 113(10):7967–7995, 2024.

[19] Shaheer Muhammad, M Usman Rafique, Shuai Li, Zili Shao, Qixin Wang, and Xue Liu. Reconfigurable battery systems: A survey on hardware architecture and research challenges. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 24(2):1–27, 2019.

[20] Liang He, Linghe Kong, Yu Gu, Cong Liu, Tian He, and Kang G Shin. Extending battery system operation via adaptive reconfiguration. *ACM Transactions on Sensor Networks (TOSN)*, 15(1):1–21, 2019.

[21] 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.

[22] Bowen Jiang, Junfei Tang, Yujing Liu, and Luca Boscaglia. Active balancing of reconfigurable batteries using reinforcement learning algorithms. In *2023 IEEE Transportation Electrification Conference & Expo (ITEC)*, pages 1–6. IEEE, 2023.

[23] Jingwei Hu, Xinjie Li, Xiaodong Li, Zhensong Hou, and Zhihong Zhang. Optimizing reinforcement learning for large action spaces via generative models: Battery pattern selection. *Pattern Recognition*, 160:111194, 2025.

[24] David Flessner, Jun Chen, and Guojiang Xiong. Reinforcement learning-based event-triggered active-battery-cell-balancing control for electric vehicle range extension. *Electronics*, 13(5):990, 2024.

[25] Jingwei Hu, Kai Xie, Zheng Fang, Xiaodong Li, Junchi Yan, and Zhihong Zhang. Optimize battery control: A multi-objective evolutionary ensemble reinforcement learning approach. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pages 9214–9222, 2025.

[26] Jun Chen, Aman Behal, Zhaojian Li, and Chong Li. Active battery cell balancing by real-time model predictive control for extending electric vehicle driving range. *IEEE Transactions on Automation Science and Engineering*, 21(3):4003–4015, 2023.

[27] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, 1998.

[28] Chenhao Lu, Ruizhe Shi, Yuyao Liu, Kaizhe Hu, Simon S. Du, and Huazhe Xu. Rethinking transformers in solving POMDPs. In *Proceedings of the 41st International Conference on Machine Learning*, pages 33089–33112, 2024.

[29] Zhihe Yang and Yunjian Xu. DMBP: Diffusion model-based predictor for robust offline reinforcement learning against state observation perturbations. In *The 12th International Conference on Learning Representations*, 2024.

[30] Javier García and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

[31] Lukas Brunke, Melissa Greeff, Adam W Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5(1):411–444, 2022.

[32] Marc G Bellemare, Will Dabney, and Mark Rowland. *Distributional Reinforcement Learning*. MIT Press, 2023.

[33] Steven Carr, Nils Jansen, Sebastian Junges, and Ufuk Topcu. Safe reinforcement learning via shielding under partial observability. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 14748–14756, 2023.

[34] R Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–42, 2000.

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

[36] Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1587–1596, 2018.

[37] Andrew Wang, Andrew C Li, Toryn Q Klassen, Rodrigo Toro Icarte, and Sheila A McIlraith. Learning belief representations for partially observable deep RL. In *Proceedings of the 40th International Conference on Machine Learning*, pages 35970–35988, 2023.

[38] Oluleke Babayomi, Rafal Madonski, Zhenbin Zhang, Jose Rodriguez, Innocent Davidson, and Dong-Seong Kim. Robust model predictive control of converter-based microgrids. *IEEE Transactions on Power Electronics*, 40(11):17124–17146, 2025.

[39] Kashif Mushtaq, Hassan Nazir, Hassan Abdullah Khalid, Naseem Iqbal, Majid Ali, Aamir Khan, and Adeel Waqas. Thermal management of lithium-ion batteries using kraft paper honeycomb wall based novel cooling system for electric vehicles-an experimental investigation. *Journal of Energy Storage*, 97:112677, 2024.

A Frequently Asked Questions (FAQs)

A.1 Code and Dataset Availability

To promote transparency and reproducibility, we have made our code publicly available in a repository at `https://github.com/hujingwei/DGRAF.git`. The repository includes the complete source code of DGRAF, scripts for model training, and evaluation, as well as detailed instructions for reproduction, ensuring that researchers have full access to the resources required to reproduce and extend our work.

A.2 What are the main advantages of DGRAF over existing methods?

DGRAF demonstrates significant advantages in partially observable DRB systems, stemming from its deep understanding of the core challenges in sensor failure scenarios.

Dynamic balance between performance optimization and safety assurance. The key challenge in partial observability is not only "inferring missing states" but also "how to make decisions when inference quality is unknown." DGRAF explicitly quantifies inference quality, enabling policies to perceive their own uncertainty: pursuing aggressive optimization when observations are sufficient, and automatically shifting to risk avoidance when observations are missing. This adaptive mechanism allows DGRAF to maintain high energy utilization while controlling final SOC difference within 1%, without requiring manual trade-offs between performance and safety.

Superior robustness across failure intensities. In practical applications, sensor failure intensity varies dynamically, from occasional single-point failures to large-scale persistent outages. DGRAF's quality-aware mechanism automatically adapts to different failure intensities: achieving 88.3% IMR under mild failures and maintaining 85.6% under severe failures, with performance degradation of only 2.7 percentage points. More importantly, energy utilization drops by merely 2.9% when failure intensity increases, demonstrating stable performance under high uncertainty conditions. This robustness means DGRAF can be directly deployed to real environments without retuning for different failure scenarios.

Deep exploitation of spatiotemporal structural information. Sensor failures in DRB systems exhibit clear spatiotemporal coupling characteristics, with adjacent batteries failing simultaneously and failures persisting for extended durations. Through window-based joint inference, DGRAF can simultaneously leverage multi-source information including historical observations and adjacent batteries for state reconstruction. The iterative denoising process of diffusion models fully exploits these spatiotemporal constraints, while the correction magnitude during denoising naturally reflects the remaining uncertainty after utilizing these constraints. This deep exploitation of problem structure enables DGRAF to maintain stable decision quality and balancing performance even under complex failure patterns, such as long-duration persistent failures and large-scale adjacent failures.

A.3 Why focus on sensor failure scenarios in DRB systems?

Sensor failures are a pervasive challenge in practical DRB system deployments. A three-year tracking study of 300 electric vehicles confirms that spatiotemporally coupled sensor failure patterns are prevalent: adjacent batteries fail simultaneously due to shared communication buses, with failure durations exceeding 500 seconds. As system scale expands and operational years increase, factors such as sensor aging, connector loosening, and electromagnetic interference make partial observation loss a normalized issue.

The effective operation of DRB systems relies on accurate perception of each battery's state. Observation loss caused by sensor failures directly affects decision quality, potentially leading to battery overcharging/overdischarging or exacerbating imbalances. Traditional conservative strategies that stop reconfiguration upon failure detection can avoid erroneous decisions but completely abandon balancing capability, resulting in 6.61 % SOC difference. How to ensure both safety and maintain performance under partial observability is a key bottleneck for practical deployment. Existing DRB control methods almost all assume full observability, and partial observable reinforcement learning methods also fail to fully consider the spatiotemporal coupling characteristics of sensor failures. DGRAF is designed to fill this gap.

A.4 Can DGRAF scale to larger battery systems? What are the limitations?

DGRAF's core mechanisms are designed with good scalability, but scaling to larger systems (e.g., 100+ batteries) requires further engineering optimization.

DGRAF's diffusion inference processes battery state vectors, with complexity growing linearly with the number of batteries. For large-scale systems, hierarchical inference strategies can be adopted: dividing the battery pack into multiple subsystems, performing window inference within subsystems, and exchanging boundary state information between subsystems. This hierarchical architecture not only controls the computational scale of single inference but also leverages parallel computation to improve efficiency. However, the topology configuration space of DRB systems grows exponentially with battery count, a common challenge faced by all DRB control methods. For large-scale systems, hierarchical decision-making may be needed: high-level decisions determine inter-subsystem connection patterns, low-level decisions optimize intra-subsystem topology configurations, with quality information propagating across levels to guide decision conservatism.

Currently, DGRAF has been thoroughly validated on 8-battery systems, which represent common electric vehicle module configurations. For larger-scale energy storage systems, the core ideas of DGRAF, namely quality-aware inference and adaptive risk adjustment, remain applicable; however, engineering implementation requires combining hierarchical architectures and distributed computing. DGRAF's main limitations include the need for an offline pretraining stage and slightly higher computational overhead compared to traditional methods, although it still meets real-time requirements. We plan to explore distributed DGRAF variants in future work to achieve large-scale deployment through collaborative work of multiple local modules, and to investigate end-to-end training methods to simplify the training pipeline.

B Additional Experiments

B.1 Hardware Validation (Q6)

To address Q6, we constructed a real DRB system for practical evaluation. Ensuring absolute safety during testing in real environments is crucial. We conducted field tests on three configurations: the all-series configuration; a naive strategy that immediately stops topology reconfiguration upon detecting sensor failures; and the proposed DGRAF method.

Hardware Platform Specification. Our experimental platform comprises the following components:

Battery Cells. Eight INR 18650-20R lithium-ion batteries with nominal capacity 2000 mAh, nominal voltage 3.6V, and operating voltage range 2.5–4.2V. To establish pronounced initial imbalance conditions representative of real-world battery degradation scenarios, batteries were pre-discharged to different SOC levels with approximately 15 percentage points spread between the highest and lowest initial states.

Switch Array. 29 field-effect transistor (FET) switches are used to enable arbitrary series-parallel reconfiguration. Each channel employs an optocoupler to drive the MOSFET gate, together with a 20 Ω gate series resistor and a 10 kΩ gate pull-down resistor. This switch array supports rapid execution of topology reconfiguration commands with low switching losses.

Battery Management System (BMS). The custom-designed BMS is built around an STM32F103C8T6 microcontroller (ARM Cortex-M3, 72 MHz). Each board provides one battery-voltage measurement channel through a resistor-divider network, one ACS712 Hall-effect current-sensing channel, and one DS18B20 temperature-sensing channel. The eight BMS boards communicate with the STM32F407VET6 main controller through their TD301MCAN interfaces. The firmware initializes UART communication at 9600 baud and transmits BMS data via the CAN and UART interfaces. The BMS timer is configured at 4 Hz for periodic data transmission, sending sensor readings to the main controller.

Controller. The embedded control board is based on an STM32F407VET6 microcontroller (ARM Cortex-M4, 168 MHz), on which the trained DGRAF agent performs real-time control. The agent processes BMS measurements and graph-structured state representations to make topology reconfiguration decisions at 5-second intervals, balancing control responsiveness with computational overhead.

The entire loop, from information collection to decision output and command execution, is completed within 200 ms.

Load. A programmable constant current electronic load operating at a 2A discharge rate is employed. To emulate the fluctuating external power demands in real applications, the system dynamically adjusts the number of series-connected cells N. At each control step, N is sampled from a truncated normal distribution $\mathcal{N}(4, 1.5^2)$ and constrained to the integer range $[1, 8]$. To ensure operational safety and prevent over-discharge damage, the system enforces a strict 20 % SOC lower limit, immediately triggering a hardware cutoff to terminate discharge whenever any cell reaches this threshold.

Figure 5 presents the photograph of the hardware experimental platform, illustrating the interconnections among battery pack, reconfigurable switch matrix, BMS sensing circuits, and control computer. The platform enables systematic validation of our proposed DGRAF framework under realistic hardware constraints including sensor noise, communication delays, and actuator dynamics.

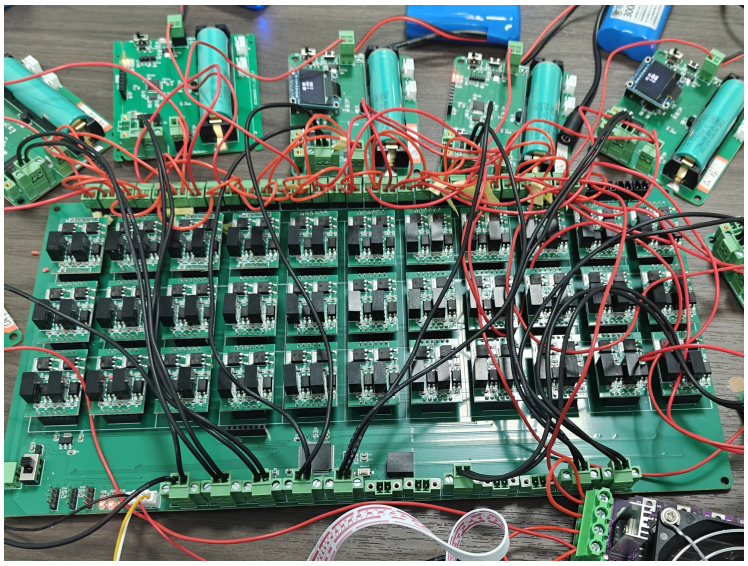


Figure 5: Experimental hardware platform with 8-cell reconfigurable battery system and switching array.

Software Masking Strategy and Safety Monitoring. To safely replicate sensor failure scenarios without compromising system safety, we employed a software masking strategy combined with hardware-level protection. The control algorithm operates on intentionally masked observation data to simulate sensor failures, while the complete ground-truth measurements remain accessible to an independent safety monitoring module. This separation ensures that failure scenarios can be injected into the decision-making process without affecting the underlying safety supervision.

At the hardware level, each BMS protection board integrates a DW01 protection IC with external MOSFETs to implement critical protection thresholds: (i) *Over-discharge protection*: cutoff at 2.4V, automatic recovery at 3.0V; (ii) *Over-charge protection*: cutoff at 4.3V, automatic recovery at 4.0V; (iii) *Over-current protection*: 2.5A limit for both charging and discharging. When any threshold is violated, the BMS autonomously disconnects the battery from the circuit until safe conditions are restored. The software safety monitor validates that control decisions do not command configurations exceeding hardware capabilities. This multi-layer safety architecture ensures that even when the control algorithm makes suboptimal decisions due to incomplete observations, the physical system remains protected from damage.

To evaluate robustness under varying uncertainty levels, we define two sensor failure modes during experiments:

Mild failure mode: Individual sensor random failure probability $p_f = 0.01$ per time step, failure duration uniformly sampled from [5, 15] steps, adjacent sensor simultaneous failure probability $p_{adj} = 0.01$. This represents occasional sensor glitches or intermittent communication losses.

Severe failure mode: Escalated failure probability $p_f = 0.03$, extended failure duration [10, 25] steps, and higher correlated failure probability $p_{adj} = 0.03$ for neighboring sensors. This simulates more challenging scenarios such as local BMS module malfunctions or partial communication bus failures.

The masked observations are generated by setting failed sensor readings to zero while preserving the observation quality indicator q_t that reflects current measurement reliability, enabling DGRAF to adaptively adjust its risk sensitivity based on information availability.

Hardware Experimental Results. Figure 6 shows the maximum SOC difference trajectories of the three strategies within a designated observation window, while Figure 7 compares the resulting energy utilization across the discharge process.

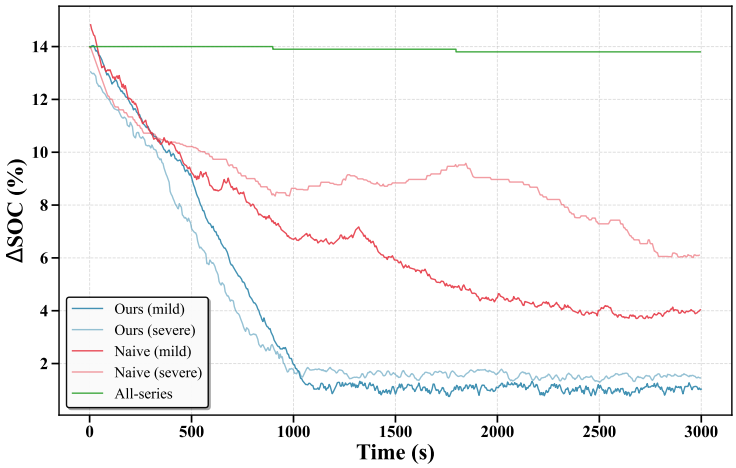


Figure 6: Maximum SOC difference comparison across three control strategies within a designated observation window, where poorly balanced baselines prematurely trigger the weakest cell safety cutoff and enter a static post-cutoff resting state.

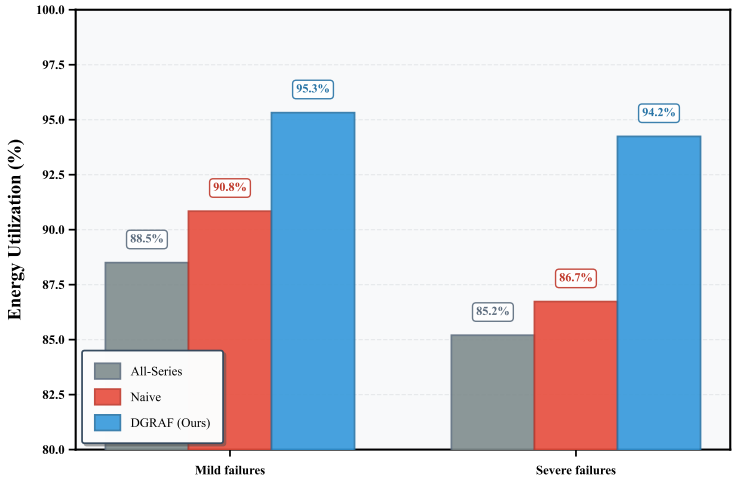


Figure 7: Energy utilization comparison of three control strategies under mild and severe failure modes.

Obs. 14. All-series configuration and Naive method fail to achieve effective balancing under partial observability, resulting in low energy utilization. As shown in Figure 7, under all-series configuration, all batteries experience the same current, causing persistent SOC differences throughout discharge, with final energy utilization of only 88.5% (mild failure) and 85.2% (severe failure). The Naive method stops topology reconfiguration once sensor failures are detected. Although improved compared to all-series, the system still remains in suboptimal configurations for most of the time,

with energy utilization increasing to 90.8% (mild failure) and 86.7% (severe failure). This indicates that simple failure-detection-stop strategies can avoid decisions based on incorrect estimates but also sacrifice the system's balancing capability and energy efficiency.

Obs. 15. DGRAF maintains the balancing capability of DRB systems under partial observability, achieving the highest energy utilization. As shown in Figure 6, after applying DGRAF to the DRB system, even under sensor failure scenarios, the system continues optimizing topology configuration throughout operation. SOC trajectories converge more effectively, with all batteries maintaining close SOC levels with a difference below 2 % in the later stages of operation. As shown in Figure 7, DGRAF achieves 95.3% energy utilization under mild failures and 94.2% under severe failures. Compared to the Naive strategy, it improves by 4.5 percentage points under mild failures (95.3% vs. 90.8%) and 7.5 percentage points under severe failures (94.2% vs. 86.7%). This demonstrates the effectiveness of quality-aware inference and adaptive decision-making mechanisms in sensor failure scenarios, with advantages becoming more pronounced as failure intensity increases.

Obs. 16. Hardware experimental results are highly consistent with simulation experiments, validating the practical deployability of the method. Hardware experiments demonstrate performance trends consistent with simulation experiments: DGRAF exhibits minimal performance degradation as failure intensity increases (95.3%→94.2%, decreasing by only 1.1 percentage points), while the Naive strategy degrades more significantly (90.8%→86.7%, decreasing by 4.1 percentage points). DGRAF's advantage over the Naive strategy becomes more pronounced under severe failures (improving by 4.5 percentage points under mild failures and 7.5 percentage points under severe failures). This indicates that the simulation environment can effectively reflect real system characteristics, and DGRAF's quality-aware mechanism remains effective under real sensor noise and system uncertainties. Additionally, no safety alarms were triggered throughout the experiments, validating the safety of the method.

Combining the above hardware experimental results, DGRAF fully validates its effectiveness, robustness, and safety in real environments, confirming the practical value and deployability of the method in real-world scenarios. The improvement in energy utilization directly translates to increased electric vehicle range or enhanced available capacity in energy storage systems, which has important practical application value.

B.2 Maintaining Performance from Full to Partial Observability

A good method for sensor failure scenarios should improve performance under partial observability while maintaining desirable performance under full observability. To verify whether DGRAF satisfies this requirement, we test its performance under four failure intensities: (i) **No failure** (full observability); (ii) **Mild failure** (failure probability 0.01, duration 2–10 steps); (iii) **Moderate failure** (failure probability 0.02, duration 3–15 steps); (iv) **Severe failure** (failure probability 0.03, duration 5–20 steps).

As shown in Table 5, DGRAF achieves 89.2% IMR and 6.31 Wh energy utilization under full observability, demonstrating strong baseline performance. As failure intensity increases, performance degradation exhibits a smooth gradient: IMR reaches 88.3% under mild failures (decreasing by only 0.9 percentage points), 86.9% under moderate failures (decreasing by 2.3 percentage points), and 85.6% under severe failures (decreasing by 3.6 percentage points). Energy utilization smoothly decreases from 6.31 Wh under no failure to 6.08 Wh under severe failure, with a total degradation of only 3.6%.

These results validate two key properties: (i) **No performance loss**: DGRAF's performance under full observability (89.2%) is even slightly higher than under mild failures (88.3%), indicating that the quality-aware mechanism does not introduce overhead in normal conditions. This slight improvement is attributed to the advantages of distributional value learning itself, which better evaluates potential action outcomes even under full observability. (ii) **Smooth adaptivity**: Performance degradation exhibits a predictable smooth curve rather than step-wise collapse. This benefits from the quality quantification module's ability to continuously capture inference reliability, enabling policies to smoothly adjust conservatism across different failure intensities.

Notably, the performance drop from moderate to severe failures (1.3 percentage points) is smaller than from mild to moderate (1.4 percentage points), showing diminishing marginal effects. This indicates that once failure intensity exceeds a certain threshold, conservative strategies are fully

activated, and further increases in failure intensity have limited impact. Even under the most severe conditions, the system maintains above 85% improvement maintenance rate and within 1% final SOC difference, validating DGRAF's stability across the entire failure intensity spectrum.

Table 5: Performance of DGRAF under different failure intensities (mean $\pm$ std over 5 runs)

Metric	**No Failure**	**Mild Failure**
Energy (Wh)	6.31±0.07	6.26±0.06
IMR (%)	89.2±0.87	88.3±0.94
Final SOC Diff (%)	0.95±0.09	1.00±0.08
Metric	**Moderate Failure**	**Severe Failure**
Energy (Wh)	6.17±0.07	6.08±0.08
IMR (%)	86.9±1.08	85.6±1.23
Final SOC Diff (%)	1.02±0.09	1.03±0.10

B.3 Sensitivity Analysis

To verify DGRAF's robustness to key hyperparameters, we systematically analyze two parameters that have the most significant impact on method performance: window length L and risk level α. As shown in Table 6, DGRAF exhibits good robustness to both hyperparameters.

Impact of window length L. Window length determines the amount of historical information utilized in joint inference. From Table 6, when L increases from 3 to 5, IMR improves from 86.8% to 88.3%, reaching the optimum at $L = 5$. Further increasing to $L = 9$ results in a slight decrease to 87.6%. This trend indicates that a moderate historical window ($L = 5$) provides sufficient temporal consistency constraints, while excessively long windows may introduce outdated information that degrades inference quality. Notably, within the range $L \in [3, 7]$, performance variation is only 1.5 percentage points, demonstrating good stability.

Impact of risk level α. Risk level α controls the strength of tail weighting in CVaR aggregation. Table 6 shows that as α increases from 0.1 to 0.4, IMR first rises to 88.3% ($\alpha = 0.3$), then slightly decreases to 87.7% ($\alpha = 0.4$). Too small α (e.g., 0.1) leads to overly conservative policies, sacrificing optimization opportunities under high-quality observations; further increasing α (e.g., 0.4) begins to weaken risk control effectiveness. Within the reasonable range $\alpha \in [0.2, 0.4]$, performance fluctuation does not exceed 0.6 percentage points, indicating the method has high tolerance to this parameter.

Overall, DGRAF demonstrates stable performance within reasonable ranges of key hyperparameters, which is important for practical deployment as extensive hyperparameter tuning may not be necessary across different application scenarios.

Table 6: Hyperparameter sensitivity analysis (IMR, %)

Parameter	**Value**			
	3	**5**	**7**	**9**
Window length L	86.8	**88.3**	88.1	87.6
	0.1	**0.2**	**0.3**	**0.4**
Risk level α	86.2	87.9	**88.3**	87.7

B.4 State Reconstruction Quality

To assess whether the diffusion component genuinely improves state inference, we report SOC reconstruction MAE for DGRAF and the *w/o Diffusion* (LSTM) ablation variant under mild and severe failures in Table 7.

DGRAF's diffusion inference achieves substantially lower state reconstruction error, confirming that the diffusion component improves state estimation fidelity and thereby contributes to DGRAF's overall performance advantage.

Table 7: SOC reconstruction MAE (%) under different failure intensities

Method	Mild Failure	Severe Failure
DGRAF (Diffusion)	0.6 ± 0.1	1.2 ± 0.2
w/o Diffusion (LSTM)	2.8 ± 0.3	4.1 ± 0.4

B.5 Computational Efficiency

Real-world deployment requires methods to not only achieve excellent performance but also meet real-time requirements. We evaluate DGRAF's computational efficiency in terms of inference time and decision time.

As shown in Table 8, for DRB systems with a typical control cycle of 5 seconds, DGRAF achieves efficient real-time performance. Under the current configuration ($L = 5$, $K = 20$), the state inference time is 17.2ms and the decision time is 0.35ms, totaling 17.55ms, which accounts for only 0.351% of the control cycle. This demonstrates that DGRAF fully meets real-time requirements for battery management systems.

We further analyze the impact of different configurations on inference time. Reducing denoising steps to $K = 10$ decreases inference time to 10.2ms, while increasing to $K = 30$ raises it to 24.5ms. Similarly, increasing window length to $L = 15$ results in 23.7ms inference time. However, even under the most conservative configuration ($L = 15$, $K = 20$), the total time is only 24.05ms (0.481% of the control cycle), still well within real-time constraints.

These results indicate that although the iterative denoising process of diffusion models is conceptually complex, it remains highly efficient in actual computation. The quality-aware mechanism introduces minimal computational overhead, making DGRAF suitable for real-time control applications in practical battery management systems.

Table 8: DGRAF inference time under different configurations

Window L	Denoising K	Inference (ms)	Total (ms)
5	10	10.2	10.55
5	**20**	**17.2**	**17.55**
5	30	24.5	24.85
15	20	23.7	24.05

B.6 Comparison with Model Predictive Control Baselines

Traditional Model Predictive Control methods [20, 26] require complete state observations and explicit system models, making them inapplicable directly under raw sensor failures. To evaluate non-learning industrial controllers under identical state estimation conditions, we equip Finite Control Set MPC, denoted as FCS-MPC, and Stochastic MPC, denoted as SMPC, with our pretrained diffusion reconstructor [38]. As shown in Table 9, the MPC methods only reconstruct states during operation without adopting quality-guided risk adaptation, yielding only 62.84% and 67.31% IMR under mild failures compared to 88.35% for DGRAF, while also exhibiting a substantial gap in performance retention under severe failures. Furthermore, due to online combinatorial search across the switching action space, their per-step inference speed is several times slower than that of the proposed method.

Table 9: Performance and online latency comparison with diffusion-augmented MPC baselines.

Method	Mild Energy ↑	Mild IMR ↑	Mild SOC Diff ↓	Severe Energy ↑	Severe IMR ↑	Severe SOC Diff ↓	Online Time / Step ↓
FCS-MPC	4.47 ± 0.25	62.84 ± 2.87	5.21 ± 0.45	4.33 ± 0.30	60.92 ± 3.41	5.48 ± 0.55	82.43 ± 19.12 ms
SMPC	4.82 ± 0.21	67.31 ± 2.46	4.63 ± 0.38	4.62 ± 0.27	65.48 ± 2.96	4.91 ± 0.47	368.08 ± 24.31 ms
DGRAF	$\mathbf{6.26 \pm 0.06}$	$\mathbf{88.35 \pm 0.94}$	$\mathbf{1.00 \pm 0.08}$	$\mathbf{6.08 \pm 0.08}$	$\mathbf{85.64 \pm 1.23}$	$\mathbf{1.03 \pm 0.10}$	$\mathbf{17.55 \pm 2.40}$ ms

C Implementation Details

C.1 Simulation Environment Justification

We adopt the standard lithium-ion battery blocks provided by MATLAB/Simulink to construct the battery simulation environment. The open-circuit voltage and internal resistance characteristics are directly governed by the built-in Simulink battery library under standard operating conditions. The simulation parameters are chosen to represent realistic operating conditions and typical DRB system settings. The initial SOC range of 85 %–100 % is selected to represent the typical state of a battery pack at the beginning of a discharge cycle in electric vehicle applications. To emulate fluctuating external power demands, the number of series-connected cells, N, is sampled from a truncated normal distribution $\mathcal{N}(4, 1.5^2)$ and constrained to the integer range $[1, 8]$. Since the load is modeled as a constant-current sink, adjusting N modulates the total power output. Failure probabilities ($p_f \in \{0.01, 0.03\}$) and persistence durations ($T_{\text{fault}} \in [2, 20]$ steps) are derived from the fault statistics of the 300-vehicle field study [8] and validated fault injection protocols in [10]. The 8-cell testbed scale matches the typical module configuration used in electric vehicle battery packs [39].

C.2 Hyperparameter Configurations

C.2.1 DGRAF Hyperparameters

Our DGRAF framework adopts a two-stage training strategy. The detailed hyperparameter configurations for each stage are as follows:

Diffusion Model Pretraining (Stage 1). We set window size $L = 5$ and denoising steps $K = 20$. The noise schedule follows a linear scheme with starting value $\beta_1 = 10^{-4}$ and terminal value $\beta_K = 0.02$. The network architecture employs a U-Net structure with timestep embedding layers for conditional generation. Training runs for 300 epochs with batch size 64 and learning rate 3×10^{-4}. We collect 5000 offline trajectories from the fully observable environment for pretraining, which include simulated various sensor failure patterns.

Distributional Reinforcement Learning (Stage 2). After freezing the diffusion model, we proceed with online policy learning. The quantile network samples $N = 32$ uniformly distributed quantile levels $\{\tau_i\}_{i=1}^{32}$. The network structure is a 3-layer MLP with 256 hidden units per layer and ReLU activation. The policy network adopts the same 3-layer MLP structure with 256 hidden units. Risk level $\alpha = 0.3$ is used for tail weighting when computing CVaR. Discount factor $\gamma = 0.99$, learning rate 3×10^{-4}, replay buffer size 10^5. Target networks use soft updates with rate $\rho = 0.005$. To encourage exploration, the entropy coefficient is set to $\beta = 0.01$.

C.2.2 Baseline Method Hyperparameters

PPO. Both policy and value networks adopt 3-layer MLP structures with 256 hidden units and ReLU activation. Learning rate 3×10^{-4}, discount factor $\gamma = 0.99$, GAE parameter $\lambda = 0.95$, clip range 0.2. Each update uses 2048 samples for 10 optimization epochs with minibatch size 64.

TD3. Both Actor and Critic networks are 3-layer MLPs with 256 hidden units. Actor learning rate 3×10^{-4}, Critic learning rate 3×10^{-4}, discount factor $\gamma = 0.99$. Policy update delay frequency is 2, target network soft update rate $\rho = 0.005$, exploration noise standard deviation 0.1, target policy smoothing noise standard deviation 0.2, noise clip range 0.5. To handle partial observability, a GRU layer with 128 hidden units is added before the policy network.

DVRL. Uses variational recurrent neural networks to model belief states. Latent state dimension 64, RNN hidden layer 128 units, 15 particles for importance sampling. VAE KL divergence weight $\lambda = 0.5$, learning rate 2×10^{-4}, discount factor $\gamma = 0.99$. Both encoder and decoder are 2-layer MLPs with 128 hidden units.

Belief-RL. Three-stage training: (i) Representation learning stage: state encoder and observation encoder output dimension 16, trained for 1000 epochs, KL coefficient $\lambda_{\text{KL}} = 0.3$, state prediction

loss coefficient 0.3, observation prediction loss coefficient 0.03, reward prediction loss coefficient 10. (ii) VAE pretraining stage: latent variable dimension 32, trained for 5000 epochs, learning rate 3×10^{-4}. (iii) Policy learning stage: learning rate 5×10^{-4}, minibatch 2048, 24 PPO update epochs, entropy coefficient 0.03. History encoder uses 3-layer stacked GRU with 256 hidden units.

WCSAC. Uses distributional safety critic to learn long-term cost distributions. Dual Q-networks to avoid overestimation, each Q-network is a 3-layer MLP with 256 hidden units. Safety critic is decomposed into mean network Q_μ^c and variance network V_η^c, both are 2-layer MLPs with 128 hidden units. Learning rate uniformly 3×10^{-4}, discount factor $\gamma = 0.99$, target network update rate $\rho = 0.005$.

DMBP. Diffusion model-based state denoising method. Diffusion steps $K = 20$, condition trajectory length $N = 4$, sample trajectory length $M = 6$. Denoising network consists of fully connected layers with hidden dimensions [256, 256, 256], Mish activation, dropout rate 0.1. Learning rate 3×10^{-4}, batch size 64, trained for 300 epochs. At test time, starting timestep k is selected based on noise scale, resampling times $U = 2$, and 5 parallel denoising samples are averaged. Since DMBP was originally designed for observation perturbations in offline RL, we adapt it to the sensor failure setting by replacing its perturbation noise injection with mask-based missing state completion: observed positions ($m_{i,j} = 1$) are kept fixed, and only unobserved positions are denoised. All other components, including the denoising network architecture and resampling strategy, remain unchanged.

C.3 Reward Function Design

Algorithm 1 Reward Function Computation

Require: System parameters: $V_{\min}$, $V_{\max}$, threshold, disturbance **Input:** $\mathbf{A}$, v_{total}, $\text{SOC}_{\text{before}}$, $\text{SOC}_{\text{after}}$
Ensure: r

if $\neg\text{IsValid}(\mathbf{A})$ **then**
  $r \leftarrow -0.1$ {Severe penalty for safety violations}
else if $v_{\text{total}} < V_{\text{min}}$ **or** $v_{\text{total}} > V_{\text{max}}$ **then**
  $r \leftarrow -0.05$ {Moderate penalty for voltage violations}
else
  $\delta_{\text{before}} \leftarrow \max(\text{SOC}_{\text{before}}) - \min(\text{SOC}_{\text{before}})$
  $\delta_{\text{after}} \leftarrow \max(\text{SOC}_{\text{after}}) - \min(\text{SOC}_{\text{after}})$
  $\Delta \leftarrow \delta_{\text{before}} - \delta_{\text{after}}$ {Balance improvement}
  if $\delta_{\text{before}} <$ threshold **and** $\delta_{\text{after}} <$ threshold + disturbance **then**
    $r \leftarrow 0.1$ {Maintain well-balanced state}
  else
    $r \leftarrow 0.05 + \Delta$ {Encourage balance improvement}
    $r \leftarrow \text{clip}(r, -0.1, 0.1)$ {Normalize reward}
  end if
end if
return r

To guide the reinforcement learning agent toward effective battery balancing strategies, a reward function is designed that comprehensively considers both safety and performance. The reward function follows three core principles: (i) illegal topology configurations must be severely penalized to prevent hardware damage, (ii) output voltage should satisfy load demand constraints to ensure stable power delivery, and (iii) SOC differences among batteries should be minimized to maximize energy utilization.

The reward structure embodies a hierarchical priority where safety violations receive the most severe penalties ($r = -0.1$), voltage violations receive moderate penalties ($r = -0.05$), and balancing performance receives positive rewards. This hierarchy ensures that the agent first learns to avoid catastrophic failures, then satisfies operational constraints, and finally optimizes performance. The reward magnitude is normalized to $[-0.1, 0.1]$ to maintain stable learning dynamics. When the

system reaches a well-balanced state, a fixed positive reward encourages the agent to maintain this state rather than making unnecessary reconfigurations.

Algorithm 1 presents the complete computation procedure. Given the voltage fluctuation range converted from the total power, including the minimum value $V_{\min}$ and the maximum value $V_{\max}$, the SOC expected constraint threshold, and the allowed threshold disturbance, and given the related inputs including the connection topology $\mathbf{A}$, total voltage v_{total}, pre-configuration SOC vector $\text{SOC}_{\text{before}}$, and post-configuration SOC vector $\text{SOC}_{\text{after}}$, the reward r is computed as follows:

Illegal Configuration: When IsValid($\mathbf{A}$) returns False, indicating short circuits or open circuits, the agent receives maximum penalty $r = -0.1$. This strong negative signal ensures rapid learning to avoid catastrophic states.

Voltage Violation: When the pack voltage v_{total} falls outside $[V_{\min}, V_{\max}]$, the agent receives $r = -0.05$. Based on a per-cell nominal voltage of 3.6 V with a ± 0.4 V margin, the system bounds are scaled by the series demand factor N as $V_{\min} = 3.2N$ V and $V_{\max} = 4.0N$ V. Enforcing a continuous pack-level window rather than a fixed cell count allows high-voltage cells to compensate for bypassed low-SOC cells, ensuring sufficient load drive and overvoltage protection without overly restricting balancing flexibility.

Valid Configuration: For valid configurations, the reward depends on balance improvement $\Delta = \delta_{\text{before}} - \delta_{\text{after}}$, where δ represents the SOC range. If already well-balanced ($\delta_{\text{before}} <$ threshold and $\delta_{\text{after}} <$ threshold + disturbance), $r = 0.1$ to maintain this state. Otherwise, $r = \text{clip}(0.05 + \Delta, -0.1, 0.1)$, where the base reward encourages valid actions and Δ provides proportional feedback on balancing effectiveness. Here, threshold $= 1.0\%$ and disturbance $= 0.02\%$ in practice.

To mitigate potential Q-function overfitting to the dense reward signal, all reward values are normalized to $[-0.1, 0.1]$, preventing reward magnitude from dominating the value estimates. The low standard deviations observed across 5 independent runs in Table 1 empirically confirm that no significant overfitting occurs in practice. This hierarchical design enables the agent to learn safe and effective balancing strategies while promoting stable convergence.

C.4 Algorithm Details

Algorithm 2 DGRAF Training Framework

Require: Environment Env, window length L, risk level α
Ensure: Diffusion model ϵ_ψ, quantile network Q_θ, policy network π_ϕ

Stage 1: Diffusion Model Pretraining
Collect trajectory dataset $\mathcal{D}_{\text{pretrain}}$ from fully observable environment
for epoch $= 1$ to N_{pretrain} **do**
  Sample batch $\{\mathbf{s}, \mathbf{o}, \mathbf{a}, \mathbf{m}\}$ from $\mathcal{D}_{\text{pretrain}}$
  Sample $k \sim \mathcal{U}(1, K)$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, compute $\mathbf{s}^{(k)} = \sqrt{\bar{\alpha}_k}\mathbf{s} + \sqrt{1-\bar{\alpha}_k}\boldsymbol{\epsilon}$
  Update ψ: $\mathcal{L}_{\text{DSM}} = \|\boldsymbol{\epsilon} - \epsilon_\psi(\mathbf{s}^{(k)}, k, \mathbf{o}, \mathbf{m}, \mathbf{a})\|^2$
end for
Stage 2: Online Reinforcement Learning (freeze ϵ_ψ)
Initialize replay buffer $\mathcal{D}_{\text{replay}}$
for episode $= 1$ to N_{episodes} **do**
  Reset environment, initialize window $\mathcal{W} = \{\mathbf{o}_{-L:0}, \mathbf{m}_{-L:0}, \mathbf{a}_{-L:-1}\}$
  for $t = 1$ to T **do**
    Sample denoising to obtain $\mathbf{s}_{\text{early},t}, \mathbf{s}_{\text{final},t}$
    Compute quality: $\mathbf{q}_t = \mathbf{m}_t + (1 - \mathbf{m}_t) \odot \exp(-|\mathbf{s}_{\text{final},t} - \mathbf{s}_{\text{early},t}|)$
    Construct $\mathbf{z}_t = [\mathbf{s}_{\text{final},t}; \mathbf{q}_t]$, sample $a_t \sim \pi_\phi(\cdot|\mathbf{z}_t)$
    Execute a_t, observe (o_{t+1}, m_{t+1}, r_t), store $(\mathbf{z}_t, a_t, r_t, \mathbf{z}_{t+1})$ to $\mathcal{D}_{\text{replay}}$
    if $t \bmod \text{update_freq} = 0$ **then**
      Sample batch from $\mathcal{D}_{\text{replay}}$, update Q_θ via $\mathcal{L}_{\text{QR}}$
      Sample $\{\tau_i\}_{i=1}^N \sim \mathcal{U}(0, \alpha)$, update π_ϕ: $\max \frac{1}{N}\sum_i Q_\theta(\mathbf{z}, a, \tau_i) + \beta H[\pi_\phi]$
      Soft update: $\theta' \leftarrow \rho\theta' + (1-\rho)\theta$
    end if
  end for
end for
return $\epsilon_\psi, Q_\theta, \pi_\phi$

Algorithm 2 presents the complete training procedure for DGRAF, which consists of two stages. In Stage 1, the diffusion model is pretrained on 5,000 offline trajectories with synthetically applied sensor failures, learning to reconstruct states from partial observations within temporal windows of length $L + 1$. In Stage 2, we freeze the diffusion model and train the distributional RL policy online. At each timestep, iterative denoising produces early-stage and final-stage estimates whose difference quantifies inference quality. The augmented representation $\mathbf{z}_t = [\mathbf{s}_{\text{final},t}; \mathbf{q}_t]$ enables the policy to learn quality-conditioned distributional value functions, adaptively adjusting risk sensitivity based on observation reliability.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 clearly state all four contributions: (i) identification of the dual impact of spatiotemporally coupled sensor failures, (ii) the first DRB safety control framework for sensor failure scenarios (DGRAF), (iii) robust state reconstruction with adaptive risk adjustment, and (iv) validation on simulation and real hardware. These claims are directly supported by the experimental results in Section 5 and Appendix B.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Section 6 and elaborated in detail in Appendix A (offline pretraining requirement, slightly higher computational overhead, and scalability to larger systems), which addresses scalability challenges and planned future directions including distributed variants and end-to-end training.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: This paper does not include theoretical results such as theorems or lemmas. The contributions are algorithmic and empirical in nature.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Full hyperparameter configurations for DGRAF and all six baselines are provided in Appendix C.2. The reward function is described in detail in Appendix C.3 with pseudocode (Algorithm 1). The complete training framework is presented in Algorithm 2. Environment setup, failure mode definitions, and evaluation metrics are specified in Section 5.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: A code repository is publicly available at `https://github.com/hujingwei/DGRAF.git`, containing the complete source code, data preprocessing scripts, training scripts, and reproduction instructions, as described in Appendix A.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5 specifies the simulation environment, dataset composition (fully observable 30%, mild failure 40%, severe failure 30%), evaluation metrics, and baseline selection. Appendix C.2 provides complete hyperparameters for all methods including optimizer (Adam), learning rates, network architectures, and training epochs.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: All quantitative results in Table 1, Table 2, Table 4 and Table 5 report mean $\pm$ standard deviation computed over 5 independent runs with different random seeds, as stated in the table captions.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Simulation and model training are conducted on a workstation equipped with an NVIDIA GeForce RTX 3090 GPU. The hardware validation platform (based on STM32F407VET6 and STM32F103C8T6 microcontrollers) is described in Appendix B. Inference latency is characterized in Table 8.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics?

   Answer: [Yes]

   Justification: This work involves battery management system control with no human subjects, no sensitive data collection, and no potential for misuse. The research aims to improve safety and efficiency of energy storage systems, which is broadly beneficial.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Improving battery energy utilization and fault tolerance supports sustainable grid storage and operational safety in electric vehicles. As a physical battery control framework, it poses no societal risks regarding privacy or malicious misuse, while mitigating overcharging or overdischarging risks through risk-averse control.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse?

    Answer: [N/A]

    Justification: The proposed method is a battery control algorithm for DRB systems and poses no meaningful risk of misuse. The released code is a research prototype for battery management and does not involve sensitive data, generative models, or dual-use capabilities.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All baseline methods (PPO, TD3, DVRL, Belief-RL, WCSAC, DMBP) are properly cited with their original papers. The DRB simulation environment is implemented in MATLAB/Simulink.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

Justification: The DGRAF codebase is released at `https://github.com/hujingwei/DGRAF.git` with documentation covering environment setup and evaluation scripts. The hardware platform design (BMS circuit, switch array) is described in detail in Appendix B.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects. All experiments are conducted on simulation environments and a self-constructed hardware battery testbed.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals were obtained?

    Answer: [N/A]

    Justification: This paper does not involve human subjects. No IRB approval is required.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research?

    Answer: [N/A]

    Justification: LLMs are not used as any component of the core methodology. The proposed DGRAF framework is based on diffusion models and distributional reinforcement learning, neither of which employs LLMs.