ADAPTABLE AND TIMELY INCIDENT RESPONSE FOR INDUSTRIAL ENTERPRISE NETWORKS USING LLM-BASED PREDICTIVE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Adaptable and timely incident response is critical for industrial enterprise networks to recover from varying cyberattacks. However, existing works have two limitations: 1) performance degradation when facing varying attacks or deployed in different network environments; and 2) insufficient timeliness and effectiveness without jointly considering recovery delays and operational failures. Therefore, we develop an incident response method for industrial enterprise networks using large language model (LLM)-based predictive reinforcement learning, consisting of three parts: prior candidate-plan generation using LLMs, lookahead prediction using world-model rollouts, and posterior decision-making using proximal policy optimization (PPO). We propose selecting a plan by jointly considering its LLM-derived prior preference and the return and uncertainty from world-model rollouts. This approach captures regularities among prior preference, return, and uncertainty across incident scenarios, improving generalization to varying attacks and network environments. Furthermore, we propose a reward function that penalizes recovery delays and operational failures to guide plan-selection policy learning, improving response timeliness and effectiveness. Evaluations show that our method outperforms selected SOTAs under varying attacks and across different network environments, and our reward function achieves higher recovery precision, shorter recovery delay, and fewer operational failures than the reward functions used in selected SOTAs. The code is available at https://anonymous.4open.science/r/LWM-RL-1C52/.

1 INTRODUCTION

Industrial enterprise networks are becoming increasingly connected and digitized. Attacks against industrial enterprise networks continue to evolve, posing severe threats to their security and availability. Incident response is a critical measure that contains, mitigates, and recovers from attacks while minimizing operational failures (Schlette et al., 2024; Smith et al., 2021; Shaked et al., 2023). Incident response is a sequential decision-making process in which current actions can influence future network state and subsequent actions (Gao et al., 2026b; Hammar et al., 2026a). Varying attacks and network environments can make response outcomes difficult to predict, degrading response performance. Therefore, industrial enterprise networks require an incident response method that enables adaptable and timely recovery from varying attacks across network environments.

1.1 CHALLENGES

However, existing works have the following two limitations in achieving adaptable and timely incident response:

1) Performance degradation when facing varying attacks or deployed in different network environments. Most recent works (Purves et al., 2024; Li et al., 2026; Sahani & Liu, 2024; Hammar et al., 2024; Dutta et al., 2023; Hammar & Li, 2025; Hammar et al., 2026b; Gao et al., 2026b; Hammar et al., 2026a; Gao et al., 2026c;a) employ model-based reinforcement learning (MBRL) to improve incident response by considering the future effects of response actions. However, when these methods face varying attacks or are deployed in different network environme-nts, predictions

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

of these future effects may be highly uncertain, leading to suboptimal decisions. To address this issue, recent MBRLs (Dalmau-Moreno et al.; Webster, 2021) constrain the action distribution to regions of the state space where model uncertainty is low. But this approach may lead to conservative actions at the expense of performance. Therefore, existing work remains limited in generalization to varying attacks and network environments.

2) Insufficient timeliness and effectiveness without jointly considering recovery delay and operational failures. Most existing works (Dudman & Bull, 2025; Chen et al., 2026; Ren et al., 2025; Purves et al., 2024; Li et al., 2026; Sahani & Liu, 2024; Hammar et al., 2024; Dutta et al., 2023; Hammar & Li, 2025; Hammar et al., 2026b; Chevalier et al., 2019; Doroudi et al., 2021) minimize system compromise and operational failures. This may delay effective recovery actions, increasing the risk of attack propagation and subsequent operational availability loss. Some works (Dudman & Bull, 2025; Loevenich et al., 2025; Shen et al., 2023; Gao et al., 2026b; Hammar et al., 2026a; Gao et al., 2026c;a) minimize recovery delays, which may lead to aggressive response actions that reduce operational availability. Therefore, existing work remains limited in enabling timely and effective recovery while improving operational availability.

1.2 Contributions

To address these limitations, we propose an adaptable and timely incident response method using LLM-based predictive reinforcement learning (RL). Our method consists of three parts: 1) prior plan generation. The large language model (LLM) generates a set of candidate multi-step plans and assigns a prior preference to each plan; 2) lookahead prediction. The world model rolls out each plan to predict its return and quantify the predictive uncertainty; and 3) posterior decision-making. Based on incident information and each plan's prior preference, return, and uncertainty, proximal policy optimization (PPO) optimizes a plan-selection policy to minimize recovery delays and operational failures. Our method executes the first action of the selected plan and replans at the next response step. The main contributions of our work are as follows:

1) We propose learning a posterior plan-selection policy based on the LLM-derived prior preference and the return and uncertainty from world-model rollouts. This approach captures regularities among prior preference, return, and uncertainty across incident scenarios, improving generalizability when facing varying attacks or deployed in different network environments.

2) We propose a reward function that penalizes both recovery delays and operational failures to guide the plan-selection policy learning, enabling timely and effective recovery from attacks while improving operational availability.

We evaluate our proposed method on the CAGE Challenge 4 environment. The experimental results show that 1) our method outperforms selected SOTAs when facing various attacks and deployed in different network environments, demonstrating its generalization capability; and 2) our reward function achieves higher recovery precision, shorter recovery time, and fewer operational failures than the reward functions used in selected SOTAs, demonstrating its ability to enable timely and effective incident response.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 introduces the proposed method. Section 4 conducts comparative experiments and evaluations. Finally, Section 5 concludes this paper.

2 Related Work

2.1 Incident Response

Approach. Recent works have begun to explore the automated generation of incident response policies. Three predominant approaches have emerged: 1) model-free RL (Dudman & Bull, 2025; Chen et al., 2026; Loevenich et al., 2025; Shen et al., 2023; Ren et al., 2025), 2) LLM (Lin et al., 2025), and 3) MBRL (Purves et al., 2024; Li et al., 2026; Sahani & Liu, 2024; Hammar et al., 2024; Dutta et al., 2023; Hammar & Li, 2025; Hammar et al., 2026b; Gao et al., 2026b; Hammar et al., 2026a; Gao et al., 2026c;a). The first two approaches select response actions based on the current state. However, they do not consider the future effects of the actions, limiting response effectiveness.

MBRL addresses this limitation by using a model to predict the future state and return of candidate actions.

Existing MBRLs can be categorized into two types: 1) MBRL with training-time planning (Purves et al., 2024; Li et al., 2026; Sahani & Liu, 2024), which uses a model (such as a causal graph and a dynamic model) to simulate trajectories and uses these trajectories to learn a policy; and 2) MBRL with decision-time planning (Hammar et al., 2024; Dutta et al., 2023; Hammar & Li, 2025; Hammar et al., 2026b; Gao et al., 2026b; Hammar et al., 2026a; Gao et al., 2026c;a), which uses a model (such as a world model and an LLM) to roll out candidate plans and executes the action with the highest return estimated from the rollouts. However, when these methods face varying attacks or are deployed in different network environments (such as different network topologies and host counts), the estimated return may be unreliable and thus lead to suboptimal decisions. Therefore, we propose a method that selects a plan by jointly considering its LLM-derived prior preference and the return and uncertainty from world-model rollouts, improving generalization to varying attacks and network environments.

Optimization Objective. Most existing incident response works (Dudman & Bull, 2025; Chen et al., 2026; Ren et al., 2025; Purves et al., 2024; Li et al., 2026; Sahani & Liu, 2024; Hammar et al., 2024; Dutta et al., 2023; Hammar & Li, 2025; Hammar et al., 2026b; Chevalier et al., 2019; Doroudi et al., 2021) minimize compromised components and/or operational failures. However, such optimization may delay effective response acting, increasing the risk of attack propagation and the resulting availability loss. Other works (Loevenich et al., 2025; Shen et al., 2023; Gao et al., 2026b; Hammar et al., 2026a; Gao et al., 2026c;a) minimize recovery delay. However, prioritizing rapid recovery may favor aggressive recovery actions, compromising operational availability. Therefore, our proposed method is designed to minimize recovery delays and operational failures, enabling timely and effective recovery while improving operational availability.

2.2 MBRLs Considering Model Uncertainty

To mitigate the impact of model uncertainty on decision-making, existing MBRL works either 1) select the action with the highest uncertainty-penalized value during trajectory simulation (Dalmau-Moreno et al.); or 2) execute the action with the highest uncertainty-penalized cumulative return estimated from trajectory rollouts (Webster, 2021). However, penalizing model uncertainty may lead to conservative decisions that favor low-uncertainty actions, thus sacrificing return (Dalmau-Moreno et al.; Hammar et al., 2025). Therefore, we propose making decisions based on the LLM-derived prior preference and the return and uncertainty from world-model rollouts. This approach captures regularities among prior preference, return, and uncertainty across incident scenarios, allowing high-return plans to be selected despite high model uncertainty when beneficial and thus improving decision-making under uncertainty.

3 Methodology

We detail our proposed incident response method in this section. Our method uses LLM-based predictive RL to enable adaptable and timely responses in industrial enterprise networks. Figure 1 presents the architecture of our method, which consists of the following three parts: 1) prior plan generation. The LLM generates multiple candidate multi-step plans and assigns a prior preference to each plan; 2) lookahead prediction. The world model rolls out each plan to estimate its return and quantify the predictive uncertainty; and 3) posterior decision-making. PPO optimizes a plan-selection policy based on the incident information and each plan's prior preference, predicted return, and predictive uncertainty, aiming to minimize recovery delays and operational failures. We next describe the three parts in detail.

3.1 Prior Plan Generation

LLMs encode cybersecurity knowledge that can be leveraged to generate several reasonable response plans, reducing the planning space and improving subsequent decision-making efficiency. Therefore, at each time t of the response, we use an LLM to generate N candidate multi-step plans $\boldsymbol{a}_t^{\boldsymbol{p}}$ and their prior preferences $\boldsymbol{p}_t$ based on the incident information $\boldsymbol{x}_t$:

$$\boldsymbol{a}_t^{\boldsymbol{p}}, \boldsymbol{p}_t = LLM(\boldsymbol{x}_t) \tag{1}$$

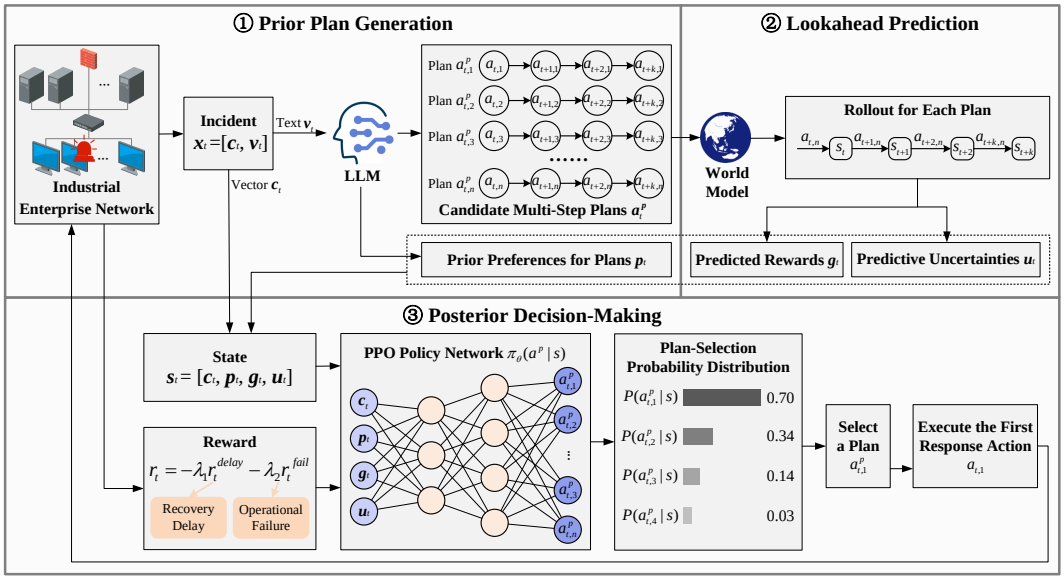


Figure 1: Architecture of our Proposed Method.

$$\boldsymbol{a}_t^{\boldsymbol{p}} = \left\{ a_{t,n}^p = (a_{t+k,n})_{k=0}^{K-1} \,|\, a_{t+k,n} \in \boldsymbol{A} \right\}_{n=1}^{N} \tag{2}$$

$$\boldsymbol{p}_t = \{p_{t,n}\}_{n=1}^{N} \tag{3}$$

where $\boldsymbol{x}_t = [\boldsymbol{c}_t, \boldsymbol{v}_t]$. $\boldsymbol{c}_t$ and $\boldsymbol{v}_t$ are the textual description and vector representation of the incident information, respectively. The incident information includes alarms, logs, and associated components; $a_{t,n}^p$ is a candidate plan; $p_{t,n} \in (0,1)$ is the prior preference of $a_{t,n}^p$; $\boldsymbol{A}$ is the set of response actions, including no-op (no operation), analyse (intrusion investigation), remove (user-level compromise removal), and restore (component reimaging); and K is the planning horizon.

3.2 Lookahead Prediction

Then, we use world model rollouts to estimate the cumulative return and predictive uncertainty for each candidate plan. For each plan $a_{t,n}^p$, an ensemble of probabilistic dynamics models generate a trajectory $\hat{\mathcal{T}}_{t,n} = \{s_{t,n}, \hat{s}_{t+1,n}, ..., \hat{s}_{t+K-1,n}\}$ by recursively predicting the next state using the preceding state and the corresponding action in $a_{t,n}^p$. Then, cumulative return $g_{t,n}$ and predictive uncertainty $u_{t,n}$ are computed as

$$g_{t,n} = \sum_{k=1}^{K-1} \gamma^k r(\hat{s}_{t+k,n}, a_{t+k,n}) \tag{4}$$

where, γ is the discount factor and $r(\hat{s}_{t+k,n}, a_{t+k,n})$ is computed using Eq. (8).

$$u_{t,n} = \sum_{k=1}^{K-1} \sum_{m=1}^{M} \| p_m(\hat{s}_{t+k+1,n} | \hat{s}_{t+k,n}, a_{t+k,n}) - \bar{p} \|_2^2 \tag{5}$$

where, M is the number of probabilistic dynamics models in the ensemble; $p_m(\hat{s}_{t+k+1,n} | \hat{s}_{t+k,n}, a_{t+k,n})$ is the state transition distribution predicted by the m-th probabilistic dynamics model; and $\bar{p} = \sum_{m=1}^{M} p_m(\hat{s}_{t+k+1,n} | \hat{s}_{t+k,n}, a_{t+k,n})/M$.

3.3 Posterior Decision-Making

The preceding LLM and world model modules provide candidate plans with their prior preferences, predicted returns, and uncertainties. However, prior preferences may conflict with predicted returns, and predicted returns may be uncertain. Therefore, we use RL to optimize a posterior plan-selection

policy based on prior preferences, predicted returns, and uncertainties, enabling adaptable plan selection under diverse conditions of prior preference, return, and uncertainty. We execute the first response action of the selected plan and replan at the next response step.

Furthermore, incident response for industrial enterprise networks should eradicate attacks quickly and effectively to prevent attack propagation while minimizing disruptions to system operations. Therefore, we formulate minimizing recovery delays and operational failures as the learning objective, aiming to achieve effective, timely recovery from cyberattacks while improving operational availability.

Policy Network. The policy network maps the posterior representations of candidate plans to a probability distribution over them. The posterior representation of each candidate plan integrates its prior preference $p_{t,n}$, predicted return $g_{t,n}$, uncertainty $u_{t,n}$, and the incident information $\boldsymbol{v}_t$, and is defined as:

$$\boldsymbol{s}_{t,n} = [p_{t,n}, g_{t,n}, u_{t,n}, \boldsymbol{v}_t] \tag{6}$$

The policy network produces a probability distribution over candidate plans, as in Eq. (7). We execute the first response action a_t^* of the selected plan a_t^{p*} and replan at the next time step.

$$P_{t,n} = \mu_\theta(\boldsymbol{a}_t^{\boldsymbol{p}} | \boldsymbol{s}_t) = \frac{\exp(z_\theta(a_{t,n}^p))}{\sum_{j=1}^{N} \exp(z_\theta(a_{t,j}^p))} \tag{7}$$

where, $P_{t,n}$ is the probability of selecting the n-th candidate plan; $\mu_\theta(\boldsymbol{a}_t^{\boldsymbol{p}} | \boldsymbol{s}_t)$ is the policy parameterized by θ; and $z_\theta(a_{t,n}^p)$ is the logit output by the policy network for the posterior representation $\boldsymbol{s}_{t,n}$ of a candidate plan $a_{t,n}^p$.

Learning Objective. To enable effective and timely recovery while improving operational availability, the reward function consists of penalties for recovery delays and operational failures:

$$r_t = -\lambda_1 r_t^{delay} - \lambda_2 r_t^{fail} \tag{8}$$

where, $-r_t^{delay} = -(t - t^c)\mathbb{I}(c_t = 1, t^c \leq t)$ is the recovery delay penalty. t^c is the time at which the component becomes compromised, and $c_t = 1$ indicates that the component remains compromised at time t; $-r_t^{fail}$ is the penalty when the component fails to perform its normal operations (such as command issuing and measurement uploading). The values of $-r_t^{fail}$ follow the setting of component work fail penalty in the CAGE Challenge 4 (CAGE Challenge); and λ_1 and λ_2 are the balancing coefficients.

To capture the long-term impact of plan-selection decisions on subsequent responses, we optimize the policy to minimize the expected cumulative discounted return, as defined by

$$G_t(\theta) = \sum_{l=0}^{\infty} \gamma^l r_{t+l} \tag{9}$$

where γ^l is the discount factor.

Policy Optimization. We optimize the policy network using PPO with the clipped objective:

$$L(\theta) = \mathbb{E}_t \left[\min \left(\frac{\mu_\theta(\boldsymbol{a}_t^{\boldsymbol{p}} | \boldsymbol{s}_t)}{\mu_{\theta_{old}}(\boldsymbol{a}_t^{\boldsymbol{p}} | \boldsymbol{s}_t)} \hat{A}_t, \text{clip}(\frac{\mu_\theta(\boldsymbol{a}_t^{\boldsymbol{p}} | \boldsymbol{s}_t)}{\mu_{\theta_{old}}(\boldsymbol{a}_t^{\boldsymbol{p}} | \boldsymbol{s}_t)}, 1 - \varepsilon, 1 + \varepsilon) \hat{A}_t \right) \right] \tag{10}$$

where, $\mu_{\theta_{old}}$ is the policy before the update; $\hat{A}_t$ is the estimated advantage; clip$(\cdot)$ is the clipping function; and ε is the clipping threshold.

4 EVALUATION

In this section, we evaluate the proposed method and the reward function on the CAGE Challenge 4 environment. The code is available at https://anonymous.4open.science/r/LWM-RL-1C52/.

Table 1: Parameters Settings of Our Method.

Parameter	Value
Number of candidate plans N.	8
Number of steps in each candidate plan K.	4
Number of probabilistic dynamics models in the ensemble M.	5
World model ridge regularization coefficient.	0.001
Discount factor for computing the plan return γ.	0.99
PPO hidden dimension.	128
PPO learning rate.	0.0003
PPO Update Epochs.	4
Balancing coefficients in the reward function, λ_1 and λ_2.	1.0, 1.0
Clipping threshold ε.	0.2

4.1 Experiment Setup

Our method is implemented in Python 3.12. All experiments are conducted on a laptop with an AMD Ryzen 7 7745HX CPU, an NVIDIA GeForce RTX 4070 Laptop GPU, and 16GB of RAM. The parameter settings of our method are reported in Table 1. We adopt GPT-5.6 Sol (Chen et al., 2026) to generate candidate plans and assigns a prior preference to each plan. We evaluate our method over 100 test episodes of 500 timesteps each, and report the mean and standard deviation of the results. For a fair comparison, our method and all comparison methods use the same LLM and are tuned to achieve their best performance.

Dataset. CAGE Challenge 4 (CAGE Challenge) is a cyber defense environment for an industrial enterprise network, in which blue agents defend against red adversarial agents across four network zones. The network is stochastically initialized at each episode, with 5-15 hosts per zone, and each host performs local work or provides access services. Red agents conduct attacks through five stages: reconnaissance, privilege escalate, lateral movement, impact, and withdraw. About 200-4,000 attack incidents are observed at each stage. We deploy our method as a blue agent to respond to these incidents.

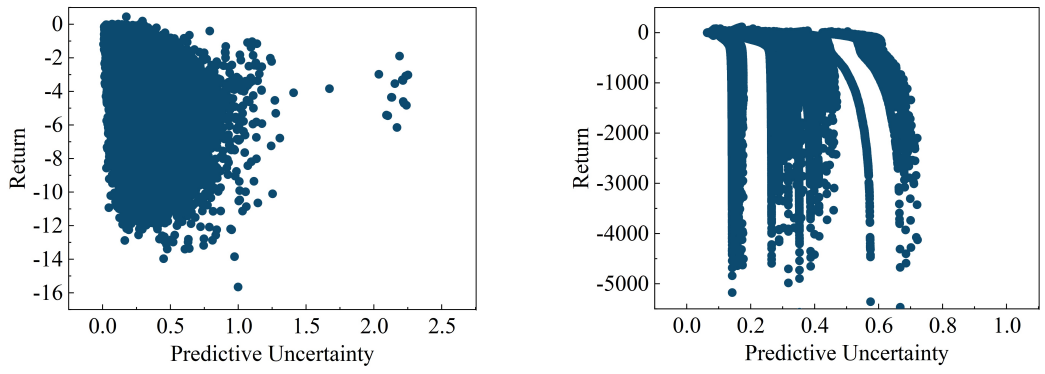
The State-of-the-Art Methods for Comparison. We evaluate our proposed method against two RL methods, two MBRL methods, and two uncertainty-constrained MBRL methods: 1) *TERLA* (Dudman & Bull, 2025), an incident response method based on PPO; 2) *PriorRL* (Yan et al., 2025), an RL method that incorporates LLM-generated action priors; 3) *CARL* (Purves et al., 2024), an MBRL method that uses a causal graph to simulate trajectories for policy training; 4) *DCA* (Sahani & Liu, 2024), an MBRL method that uses a dynamic simulator to generate trajectories for policy training; 5) *UAMCTS* (Dalmau-Moreno et al.), a tree search method that uses LLM priors to guide the initial search and constrains subsequent search to low-uncertainty regions; and 6) *RSMBRL* (Webster, 2021), an MBRL method that uses a dynamics model to generate trajectories and penalizes high-uncertainty predictions during trajectory rollouts. As PriorRL, UAMCTS, and RSMBRL do not specify reward functions, we apply our reward function to these methods.

We also evaluate our proposed reward function against four reward functions used in SOTAs: 1) penalties for system compromise and operational failures (Dudman & Bull, 2025); 2) a penalty for system compromise (Purves et al., 2024); 3) a penalty for operational failures (Hammar et al., 2024); and 4) a penalty for recovery time (Hammar et al., 2026a)]. To compare reward functions, we evaluate our method using each one.

Evaluation Metrics. We evaluate our method using four widely-used metrics (Hürten et al., 2024): 1) *CC4 official reward.* The sum of the penalties for local work failures, access service failures, and red impact/access. We report the average cumulative reward over test episodes; 2) *Operational failure penalty.* The penalty for local work failures caused by response actions and attacks. We report the average cumulative penalty over test episodes; 3) *Recovery precision.* The recovery precision is defined as $TP/(TP + FP)$, where TP and FP are the recovery actions performed on infected and clean hosts, respectively; and 4) *Recovery time.* The time required for the host to recover from a compromised state to a normal state, as defined by $ET = \sum_{k=1}^{K} (t^d - t^c)/K$. t^d is the time at which the host returns to the clean state and K is the incident count.

Table 2: Comparison of Our Method with SOTA Methods.

Method		CC4 Official Reward	Operational Failure Penalty	Recovery Precision	Recovery Time
RL Method	TERLA	-659.24 ± 143.50	548.70 ± 140.88	0.41 ± 0.13	301.06 ± 35.60
	PriorRL	-457.25 ± 70.39	181.83 ± 21.24	0.82 ± 0.06	74.30 ± 13.00
MBRL Method	CARL	-550.63 ± 116.63	468.58 ± 118.54	0.30 ± 0.14	228.51 ± 27.75
	DCA	-557.51 ± 132.57	426.31 ± 125.39	0.38 ± 0.08	245.05 ± 34.83
Uncertainty-Constrained MBRL Method	UAMCTS	-571.18 ± 131.96	438.90 ± 121.30	0.55 ± 0.09	240.75 ± 27.38
	RSMBRL	-627.94 ± 132.59	516.49 ± 142.51	0.24 ± 0.15	330.70 ± 20.31
Our Method		$\mathbf{-426.97 \pm 42.07}$	$\mathbf{31.53 \pm 80.73}$	$\mathbf{0.87 \pm 0.13}$	$\mathbf{73.58 \pm 13.53}$



(a) Our Method (b) UAMCTS

Figure 2: Return-Uncertainty Distribution of the Selected Candidate Plans.

4.2 Evaluation of the Proposed Method

Response Performance. We compare our proposed method with SOTA methods, as reported in Table 2. Our method achieves the best results across all four evaluation metrics: CC4 official reward, operational failure penalty, recovery precision, and recovery time. These results show that our method enables effective, timely recovery from cyberattacks while improving operational availability. We provide a detailed comparison with SOTA methods below.

1) Compared with the RL-based methods TERLA (Dudman & Bull, 2025) and PriorRL (Yan et al., 2025), our method achieves the best performance across all evaluation metrics. TERLA shows the lowest performance, possibly because it relies solely on PPO for policy optimization without incorporating LLM priors or lookahead predictions. PriorRL outperforms TERLA, likely because it incorporates LLM-generated action priors into RL. PriorRL does not consider lookahead predictions, and thus performs worse than our method.

2) Compared with the MBRL-based methods CARL (Purves et al., 2024) and DCA (Sahani & Liu, 2024), our method outperforms both of them. These methods train their policies on the simulated trajectories but make decisions based solely on the current incident state. In contrast, our method leverages lookahead predictions in both training and decision-making, improving response-planning performance.

3) Compared with the uncertainty-constrained MBRL-based methods UAMCTS (Dalmau-Moreno et al.) and RSMBRL (Webster, 2021), our method also achieves better performance. UAMCTS and RSMBRL constrain planning to regions of the state space with low model uncertainty and may thus perform worse than CARL (Purves et al., 2024) and DCA (Sahani & Liu, 2024). In contrast, our method learns a planning policy based on prior preferences, predicted returns, and uncertainties. This enables adaptable planning under diverse return-uncertainty conditions, thereby improving performance under model uncertainty.

Generalization. To evaluate the generalizability of our method under different return-uncertainty conditions, we compare our method with UAMCTS. Figure 2 illustrates the return-uncertainty dis-

Table 3: Performance of Our Method in the Four Network Zones.

	CC4 Official Reward	Operational Failure Penalty	Recovery Precision	Recovery Time
Zone 0	-426.26 ± 43.42	32.54 ± 42.04	0.87 ± 0.14	72.62 ± 12.87
Zone 1	-422.14 ± 42.44	31.81 ± 45.25	0.90 ± 0.10	76.36 ± 14.12
Zone 2	-442.40 ± 36.74	26.74 ± 34.54	0.80 ± 0.16	70.18 ± 13.65
Zone 3	-417.12 ± 35.68	35.03 ± 41.03	0.93 ± 0.12	75.28 ± 13.50

Table 4: Comparison of Our Reward Function with Reward Functions used in SOTA Methods.

Reward Function	CC4 Official Reward	Operational Failure Penalty	Recovery Precision	Recovery Time
System Compromise and Operational Failure Penalties	-458.72 ± 91.22	43.32 ± 35.03	0.82 ± 0.13	181.59 ± 26.78
System Compromise Penalty	-518.36 ± 66.73	426.71 ± 112.38	0.72 ± 0.15	338.51 ± 22.75
Operational Failure Penalty	-560.27 ± 56.92	303.13 ± 43.68	N/A	387.90 ± 30.15
Recovery Time Penalty	-648.98 ± 71.50	326.55 ± 26.08	0.79 ± 0.20	331.92 ± 21.12
Our Reward Function	$\mathbf{-446.97 \pm 42.07}$	$\mathbf{31.53 \pm 30.73}$	$\mathbf{0.87 \pm 0.13}$	$\mathbf{73.58 \pm 13.53}$

tributions of the candidate plans selected by the two methods. As uncertainty increases, UAMCTS selects fewer high-return plans, while our method selects more high-return plans with high uncertainty. Moreover, Table 2 shows that our method outperforms UAMCTS. These results show that our method adapts its plan-selection behavior to different return-uncertainty conditions, improving generalizability across them.

Moreover, variations in attacks and network environments can increase model uncertainty. Therefore, we evaluate our method across the four network zones with different attacks and network environments. Attacks are generated stochastically in each episode. The four network zones differ in network environments, such as the number of hosts and network topology. As shown in Table 3, our method performs well across all four zones and outperforms the SOTA methods in Table 2. These results demonstrate the generalizability of our method across varying attacks and network environments.

4.3 Evaluation of the Reward Function

Table 4 compares our proposed reward function with reward functions used in SOTAs. Our reward function achieves the best results across all four evaluation metrics, demonstrating its capability in enabling effective, timely recovery while improving operational availability.

The recovery time penalty (Hammar et al., 2026a) performs better than the operational-failure penalty (Purves et al., 2024) in terms of operational failure penalty, recovery precision, and recovery time. This may be because the recovery-delay penalty captures not only recovery delay but also unsuccessful recovery. The recovery precision of operational failure penalty is N/A because it completes no recoveries. Our reward function's better performance over the system-compromise and operational-failure penalties (Dudman & Bull, 2025) further supports the benefit of incorporating recovery delay into the reward.

Our reward function consistently outperforms the recovery-time penalty (Hammar et al., 2026a) and the operational failure penalty (Hammar et al., 2024). Prolonged attacks may cause additional operational failures, while aggressive responses may reduce operational availability. By jointly incorporating recovery delay and operational failures, our reward function encourages timely and effective recovery while reducing unnecessary operational disruptions.

4.4 Ablation Study

We conduct ablation experiments to evaluate the contribution of each component of our method. We consider three variants: 1) RL-Only, which uses only RL; 2) LLM-RL, which incorporates LLM priors into RL; and 3) WM-RL, which incorporates world-model predictions into RL. As reported in

Table 5: Ablation Study on Our Method.

	CC4 Official Reward	Operational Failure Penalty	Recovery Precision	Recovery Time
RL-Only	-612.00 ± 54.19	34.49 ± 43.46	0.85 ± 0.05	121.47 ± 17.30
LLM-RL	-557.57 ± 88.04	32.81 ± 32.78	0.85 ± 0.06	118.46 ± 20.60
WM-RL	-452.04 ± 73.98	34.74 ± 42.14	0.86 ± 0.06	73.77 ± 11.36
Our Method	$\mathbf{-426.97 \pm 42.07}$	$\mathbf{31.53 \pm 40.73}$	$\mathbf{0.87 \pm 0.13}$	$\mathbf{73.58 \pm 13.53}$

Table 6: Ablation Study on Our Reward Function.

	CC4 Official Reward	Operational Failure Penalty	Recovery Precision	Recovery Time
Delay-Only	-648.98 ± 71.50	326.55 ± 26.08	0.79 ± 0.20	331.92 ± 21.12
Fail-Only	-650.27 ± 96.18	430.13 ± 31.90	N/A	353.49 ± 20.01
Our Reward Function	$\mathbf{-446.97 \pm 42.07}$	$\mathbf{31.53 \pm 30.73}$	$\mathbf{0.87 \pm 0.13}$	$\mathbf{73.58 \pm 13.53}$

Table 5, all three variants exhibit degraded performance across the four evaluation metrics, indicating that each component contributes to the overall response performance of our method.

Moreover, we evaluate each penalty term's contribution in our reward function. We consider two variants: 1) Delay-Only, which includes only the recovery-delay penalty; and 2) Fail-Only, which includes only the operational-failure penalty. Table 6 shows that our reward function outperforms the two variants across the four evaluation metrics. This is because the two penalty terms provide complementary learning signals that encourage timely, effective recovery from attacks while reducing unnecessary operational failures. In addition, Fail-Only recovery precision is N/A because it completes no recoveries.

5 Conclusion

Adaptable and timely incident response is key for industrial enterprise networks to recover effectively from varying attacks across network environments. However, existing works fail to achieve this. To address this gap, we develop an LLM-based predictive RL method. The posterior decision-making approach leverages regularities among prior preference, return, and uncertainty across incident scenarios to improve generalization to varying attacks and network environments. The reward function penalizes recovery delays and operational failures to enable timely, effective recovery while improving operational availability. Evaluations show that 1) our method outperforms selected SOTAs under varying attacks and across different network environments; and 2) our reward function achieves higher recovery precision, shorter recovery delay, and fewer operational failures than the reward functions used in selected SOTAs.

AI use statement

In this work, we have not used generative AI tools for the tasks with required and recommended disclosures.

Reproducibility statement

To facilitate reproducibility, we provide an anonymous GitHub repository containing the source code for our proposed method and comparison baselines, experimental configurations, evaluation code, result summaries, and the constructed dataset. The repository is available for review at the following anonymous link: `https://anonymous.4open.science/r/LWM-RL-1C52/`.

References

CAGE Challenge. Cage challenge 4. `https://cage-challenge.github.io/cage-challenge-4/`. Accessed: 2026-09-26.

Lin Chen, Yingxu Lai, Peng Zhao, Baoshan Xie, and Yan Zhang. Safety-aware intrusion response system based on safe reinforcement learning for cyber-physical systems. *IEEE Transactions on Consumer Electronics*, 72(2):2711–2723, 2026. doi: 10.1109/TCE.2026.3674866.

Ronny Chevalier, David Plaquin, Chris Dalton, and Guillaume Hiet. Survivor: a fine-grained intrusion response and recovery approach for commodity operating systems. In *Proceedings of the 35th Annual Computer Security Applications Conference*, ACSAC '19, pp. 762–775, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450376280. doi: 10.1145/3359789.3359792. URL https://doi.org/10.1145/3359789.3359792.

Magí Dalmau-Moreno, Nestor Garcia, and Vicenç Gómez. Uncertainty-aware planning with generative world-and language-models via monte carlo tree search. In *3rd CoRL Workshop on Learning Effective Abstractions for Planning (Rolling)*.

Sherwin Doroudi, Thanassis Avgerinos, and Mor Harchol-Balter. To clean or not to clean: Malware removal strategies for servers under load. *European Journal of Operational Research*, 292(2): 596–609, 2021. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2020.10.036. URL https://www.sciencedirect.com/science/article/pii/S0377221720309127.

Tim Dudman and Martyn Bull. Towards a generalisable cyber defence agent for real-world computer networks. *arXiv preprint arXiv:2511.09114*, 2025.

Ashutosh Dutta, Ehab Al-Shaer, Samrat Chatterjee, and Qi Duan. Autonomous cyber defense against dynamic multi-strategy infrastructural ddos attacks. In *2023 IEEE Conference on Communications and Network Security (CNS)*, pp. 1–9, 2023. doi: 10.1109/CNS59707.2023.10288937.

Yiran Gao, Juntao Chen, and Tao Li. Hierarchical agentic incident response with digital-twin-validated attack inference. *arXiv preprint arXiv:2608.15016*, 2026a.

Yiran Gao, Kim Hammar, and Tao Li. In-context autonomous network incident response: An end-to-end large language model agent approach. In *Proceedings of the AAAI Symposium Series*, volume 9, pp. 246–254, 2026b.

Yiran Gao, Tao Li, and Kim Hammar. Agentic incident response through digital twin-enhanced multiscale planning. *arXiv preprint arXiv:2608.02422*, 2026c.

Kim Hammar and Tao Li. Online incident response planning under model misspecification through bayesian learning and belief quantization. In *Proceedings of the 18th ACM Workshop on Artificial Intelligence and Security*, pp. 40–51, 2025.

Kim Hammar, Neil Dhir, and Rolf Stadler. Optimal defender strategies for cage-2 using causal modeling and tree search. *arXiv preprint arXiv:2407.11070*, 2024.

Kim Hammar, Tao Li, Rolf Stadler, and Quanyan Zhu. Adaptive security response strategies through conjectural online learning. *IEEE Transactions on Information Forensics and Security*, 20:4055–4070, 2025. doi: 10.1109/TIFS.2025.3558600.

Kim Hammar, Tansu Alpcan, and Emil C. Lupu. Incident response planning using a lightweight large language model with reduced hallucination. In *Network and Distributed System Security (NDSS) Symposium*, 2026a.

Kim Hammar, Yuchao Li, Tansu Alpcan, Emil C. Lupu, and Dimitri P. Bertsekas. Adaptive network security policies via belief aggregation and rollout. *IEEE Transactions on Information Forensics and Security*, 21:6544–6559, 2026b. doi: 10.1109/TIFS.2026.3709131.

Tobias Hürten, Johannes F. Loevenich, Florian Spelter, Erik Adler, Johannes Braun, Linnet Moxon, Yann Gourlet, Thomas Lefeuvre, and Roberto Rigolin F. Lopes. Hierarchical multi-agent reinforcement learning for autonomous cyber defense in coalition networks. In *MILCOM 2024 - 2024 IEEE Military Communications Conference (MILCOM)*, pp. 176–181, 2024. doi: 10.1109/MILCOM61039.2024.10773689.

Yu Li, Sizhe Tang, Rongqian Chen, Fei Xu Yu, Guangyu Jiang, Mahdi Imani, Nathaniel D Bastian, and Tian Lan. Acdzero: Mcts agent for mastering automated cyber defense. *arXiv preprint arXiv:2601.02196*, 2026.

Xihuan Lin, Jie Zhang, Gelei Deng, Tianzhe Liu, Tianwei Zhang, Qing Guo, and Riqing Chen. Ircopilot: Automated incident response with large language models. *arXiv preprint arXiv:2505.20945*, 2025.

Johannes Loevenich, Erik Adler, Tobias Hürten, and Roberto Rigolin F. Lopes. Design and evaluation of an autonomous cyber defence agent using drl and an augmented llm. *Computer Networks*, 262:111162, 2025. ISSN 1389-1286. doi: https://doi.org/10.1016/j.comnet.2025.111162. URL https://www.sciencedirect.com/science/article/pii/S1389128625001306.

Tom Purves, Konstantinos G. Kyriakopoulos, Siân Jenkins, Iain Phillips, and Tim Dudman. Causally aware reinforcement learning agents for autonomous cyber defence. *Knowledge-Based Systems*, 304:112521, 2024. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2024.112521. URL https://www.sciencedirect.com/science/article/pii/S0950705124011559.

Shaochen Ren, Jianian Jin, Guanchong Niu, and Yang Liu. Arcs: Adaptive reinforcement learning framework for automated cybersecurity incident response strategy optimization. *Applied Sciences*, 15(2):951, 2025.

Nitasha Sahani and Chen-Ching Liu. Model-based detection of coordinated attacks (dca) in distribution systems. *IEEE Open Access Journal of Power and Energy*, 11:558–570, 2024. doi: 10.1109/OAJPE.2024.3489477.

Daniel Schlette, Philip Empl, Marco Caselli, Thomas Schreck, and Günther Pernul. Do you play it by the books? a study on incident response playbooks and influencing factors. In *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 3625–3643, 2024. doi: 10.1109/SP54263.2024.00060.

Avi Shaked, Yulia Cherdantseva, Pete Burnap, and Peter Maynard. Operations-informed incident response playbooks. *Computers Security*, 134:103454, 2023. ISSN 0167-4048. doi: https://doi.org/10.1016/j.cose.2023.103454. URL https://www.sciencedirect.com/science/article/pii/S0167404823003644.

Fan Shen, Levi Perigo, and James H. Curry. Sr2apt: A detection and strategic alert response model against multistage apt attacks. *Security and Communication Networks*, 2023(1):6802359, 2023. doi: https://doi.org/10.1155/2023/6802359. URL https://onlinelibrary.wiley.com/doi/abs/10.1155/2023/6802359.

Richard Smith, Helge Janicke, Ying He, Fenia Ferra, and Adham Albakri. The agile incident response for industrial control systems (air4ics) framework. *Computers Security*, 109:102398, 2021. ISSN 0167-4048. doi: https://doi.org/10.1016/j.cose.2021.102398. URL https://www.sciencedirect.com/science/article/pii/S0167404821002224.

Stefan Radic Webster. Risk sensitive model-based reinforcement learning using uncertainty guided planning. In *The Fortieth Annual Conference on Neural Information Processing Systems*, 2021.

Xue Yan, Yan Song, Xidong Feng, Mengyue Yang, Haifeng Zhang, Haitham Bou Ammar, and Jun Wang. Efficient reinforcement learning with large language model priors. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 48691–48715, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/797be96e4481c3fe5d675c1ba5352969-Paper-Conference.pdf.