



Research article

Causal-intervention and uncertainty-aware resilience enhancement for multi-agent autonomous systems

Kexin Wang¹, Wanlong Qi⁴, Tian Tang² and Dingrui Xue^{3,*}

¹ Northwest Institute of Mechanical and Electrical Engineering, Xianyang 712099, Shaanxi, China

² School of Computer Science and Technology, Tsinghua University, Beijing 100084, China

³ School of Information Engineering, Chang'an University, Xi'an 710064, Shaanxi, China

⁴ School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, Shaanxi, China

* **Correspondence:** Email: drxue@chd.edu.cn.

Abstract: Multi-agent autonomous systems must maintain mission performance in the presence of node failures and structural disruption. Because recovery operations alter functional dependencies, representations learned from nominal graphs can become inadequate for the states encountered during reconfiguration. We present causal-intervention and uncertainty-aware resilience enhancement (CURE), a two-phase framework that combines structural interventions specified by the simulator with uncertainty-aware multiobjective value decomposition. In Phase I, a shared encoder that models relational structure is trained on nominal graphs and graphs containing node replacements or deletions, using predefined functional metapaths. The interventions modify node availability and functional assignments within the simulator. Regenerated path targets, together with shared supervision over relations, expose the encoder to their structural consequences. In Phase II, the encoder is frozen, and recurrent local value functions shared across node types are learned with a monotonic value mixer under a budget of one operation per decision. An ensemble of Gaussian predictors provides estimates of aleatoric variance and disagreement among ensemble members. Analytical deficit weights distribute emphasis among sustained performance, rapid recovery, and steady-state performance, whereas predictive uncertainty parameterizes a log-sum-exp reward aggregation that increasingly prioritizes lower-performing objectives as confidence declines. On a heterogeneous swarm simulator, CURE achieves greater resilience than six reinforcement-learning baselines across eight attack scenarios at 50% damage. Additional experiments characterize performance for damage ratios from 30% to 80%, recovery speed, and variation across random topologies.

Keywords: Markov decision process; autonomous unmanned swarm; reinforcement learning; heterogeneous networks; intervention-aligned representation learning; uncertainty quantification

Mathematics Subject Classification: 90C40, 68T07, 93E35, 90B25

1. Introduction

Multi-agent autonomous systems support defense, surveillance, and critical infrastructure operations because they can be deployed at scale, tolerate the failure of individual nodes, and degrade gracefully [1, 2]. The distributed architecture that provides these advantages also creates channels through which failures can propagate along functional dependencies. Consequently, the compromise of a single node can trigger cascading degradation in mission performance [3, 4]. This vulnerability is especially consequential in unmanned systems: Cooperative reconnaissance missions must reconfigure formations when positioning or communication services are denied, and integrated cyber and electronic warfare can simultaneously degrade sensing, decision, and engagement layers [5]. Such missions are organized around kill chains, in which functional sequences such as *sense–decide–influence–target* translate perception into operational effects [6, 7]. Mission performance therefore depends on whether intact functional chains remain available rather than on the number of surviving nodes alone.

Recovery therefore requires more than replacing a missing node with another node of the same type. When a decision node fails, the sensing and influence nodes that previously depended on it may no longer form an admissible functional chain. A substitute is useful only if its relations support a valid continuation, whereas deleting the failed node may terminate some paths while leaving others intact. These changes modify the structural information on which subsequent recovery decisions depend. Training solely on nominal graphs provides limited supervision for this distinction, motivating representations learned from both the original graph and feasible replacement and deletion edits.

Resilience engineering provides the conceptual foundation for sustaining such systems under disruption. Early work characterized resilience as the capacity of a system to absorb perturbations while preserving its core functions [8]. Later formulations distinguished resilience from both reliability and stability and extended the concept to include absorption, adaptation, and restoration capabilities [9–11]. A broad range of metrics now quantifies these capabilities through the trajectory of system performance over time [12, 13]. Within systems-of-systems research, such metrics have guided the design of reconfigurable architectures and recovery strategies [14].

Traditional approaches operationalize resilience through static redundancy, mathematical programming, or metric-based heuristics. Predeployed backups and deterministic failover paths are provably optimal only when threat models are known a priori, and they remain vulnerable to resource-exhaustion and path-bypassing attacks [15, 16]. Formulations based on genetic algorithms and mixed-integer optimization can produce cooperative reconfiguration plans, yet they address an offline, one-shot pairing problem and therefore cannot track rapidly evolving graph states [17]. Heuristics based on centrality or connectivity are computationally efficient and interpretable, but they capture topological salience rather than functional semantics. As a result, repair effort can be misaligned with the continuity of the kill chain [5, 18].

Deep reinforcement learning has improved the adaptability of recovery policies. Methods that augment actor–critic algorithms with graph encoders, variants based on proximal policy optimization, and graph-based reinforcement-learning strategies with prioritized recovery learn reactive reconfiguration under partial observability [19–22]. Hierarchical and multiscale frameworks identify critical components and rebalance resilience objectives over time [23, 24], and methods guided by metapaths align node representations with kill-chain semantics before policy learning [4, 25, 26]. Two limitations remain. First, node representations are usually induced from observational data alone. A

representation that predicts observed correlations need not remain useful under the *graph edits* performed by a recovery policy: Replacing a failed node or deleting its incident edges changes the available functional route [27, 28]. Metapath encoders, in particular, are trained on nominal routes and are not resupervised once a replacement or deletion changes which continuations remain admissible, so their representations need not follow the reconfigured graph. Second, reward design remains largely static. Multiobjective frameworks fix preference vectors, tune them exhaustively, or learn an auxiliary weighting policy, and they rarely condition the aggregation of competing resilience objectives on predictive uncertainty [23, 29].

Uncertainty-aware and robust decision-making offers a potential remedy. Ensemble and probabilistic world models estimate aleatoric and epistemic uncertainty and have improved model-based reinforcement learning [30–34]. Scalable world models such as TD-MPC2 extend this approach to continuous control at scale [35], and recent surveys consolidate the resulting design space [36]. Robust Markov decision processes and risk-sensitive objectives derive conservative policies whose performance degrades gracefully under model misspecification [37, 38], whereas soft-robust aggregation provides a smooth interpolation between optimistic and worst-case value estimates [39]. In multi-agent settings, robust and risk-sensitive value factorization can reduce fragility to environmental shifts [40–42]. Recent work has also established sample-efficient robust multi-agent reinforcement learning under environmental uncertainty, including algorithms designed to overcome the multiagency curse [43, 44]. These principles, however, have not been integrated with intervention-aware representation learning for multi-agent systems in which resilience is mission critical.

CURE addresses these representation and reward-design requirements in two phases. Phase I combines prediction on edited metapaths, contrastive learning among nodes of matching types, and a shared objective for relation scoring. This exposes the encoder to the structural changes associated with failure and replacement before policy learning begins. In Phase II, the encoder is kept fixed while a Gaussian prediction ensemble, analytical objective weights, and recurrent value decomposition are learned jointly. The ensemble supplies a confidence signal before each decision; the objective weights track deficits in sustained performance, rapid recovery, and steady-state performance; and the reward aggregation adjusts the emphasis placed on lower-performing objectives. Local value networks shared across node types support distributed action proposals, and a system-level arbiter enforces a budget of one recovery operation. The central contribution is the coupling of edit-aware representations with uncertainty-sensitive, multiobjective recovery within a common graph-based framework.

The remainder of this manuscript is organized as follows. Section 2 reviews resilience engineering, heterogeneous graph representation learning, intervention-aligned representation learning, and predictive uncertainty. Section 3 defines the dynamic heterogeneous system, the feasible recovery-action budget, and the resilience objectives. Section 4 presents the two phases of CURE and their optimization details. Section 5 describes the simulator, evaluation protocol, comparative experiments, and component analyses. Section 6 summarizes the findings and identifies directions for future work.

2. Related work

2.1. Resilience of systems-of-systems

Research on resilience has progressed from qualitative definitions [8, 9, 11] to quantitative metrics that track performance trajectories [12, 13] and ultimately to design-time optimization for engineered

systems of systems [10, 14]. Within this progression, static redundancy allocation and mixed redundancy strategies optimize component provisioning under known failure modes [15, 16]. Cooperative reconfiguration formulates recovery as a node-pairing problem solved with genetic algorithms [17], whereas greedy procedures optimize kill-chain coverage for unmanned combat systems [7]. Complementary approaches use heuristics based on degree, betweenness, or cut sets to identify critical nodes and initiate rerouting or backup activation [5, 18]. Other studies use game-theoretic formulations to distribute reconfiguration effort among self-interested agents [45] or targeted disintegration analyses to expose how attackers exploit heterogeneous unmanned aerial vehicle (UAV)-swarm structure, thereby motivating structure-aware defenses [46]. Although interpretable, these methods are predominantly offline or greedy: They presuppose known threat models, neglect the long-horizon consequences of current repairs, and cannot adapt to cascades affecting multiple points. CURE retains the kill-chain terminology of this literature but replaces offline search with online learning guided by priors over functional structure.

2.2. *Heterogeneous graph representation learning*

Graph neural networks learn permutation-invariant inductive representations through neighborhood aggregation [47, 48], and their expressive power has been characterized using the Weisfeiler–Lehman hierarchy [49]. Heterogeneous information networks extend this framework by distinguishing node and relation types. Metapath-guided methods aggregate information along semantically meaningful paths to preserve cross-type semantics [50], whereas recent contrastive variants reweight negative samples according to metapath context [51]. Surveys document the broad utility of these models in transportation, scheduling, and anomaly detection [52]. Within resilience engineering, graph encoders have been integrated with policy optimization to identify failure modes and synthesize recovery strategies [19, 20, 22]. CURE builds on this machinery but explicitly targets a property not guaranteed by generic neighborhood aggregation: consistency of the learned representation under interventions that replace or delete nodes.

2.3. *Intervention-aligned representation learning*

Causal learning uses interventions to study how changes to a mechanism alter its consequences [53, 54]. Representation-learning methods can exploit intervention information to recover latent factors under explicit assumptions [55–57]. An alternative line of work uses known structural mechanisms to generate training data. For example, counterfactual data augmentation propagates controlled changes through a dynamics model [27]. CURE adopts this latter perspective. Within the simulator, replacement and deletion operations modify functional assignments and node availability, and the path-generation rule is re-evaluated after each intervention. Accordingly, the causal interpretation is restricted to these controlled simulator mechanisms, and the learned embeddings represent their consequences for recovery.

In heterogeneous network recovery, meta-path-guided causal inference (MPGCI) [4] combines sampling guided by metapaths, feature alignment based on directed acyclic graphs, and actor–critic policy optimization. Meta-path-driven reinforcement learning (MPD-RL) [25] develops embeddings driven by metapaths, hierarchical fusion with dynamic gating, and metapath-aware double-dueling Q-learning. Meta-path-guided policy distillation for resilient coordination (MPGPD-RC) [26] transfers the action distributions produced by a proximal policy optimization (PPO) teacher, together with

trajectory-level structural information, to a student policy. However, these methods rely primarily on nominal metapath structures or observed trajectories and do not explicitly regenerate supervision after node-replacement or deletion interventions, potentially limiting their ability to represent the functional consequences of recovery-induced graph changes.

2.4. Predictive world models and uncertainty in reinforcement learning

Contrastive predictive coding learns sequence representations by predicting future latent states against negative samples [58]. Probabilistic dynamics models support sample-efficient planning [30], ensemble disagreement can stabilize policy gradients [31], and model-based policy optimization weights predictions according to their estimated accuracy [32]. Dreamer-style agents have shown that learned world models can master diverse control tasks [33]. Scalable models such as TD-MPC2 extend this approach to continuous control at scale [35], and recent surveys consolidate the design space [36]. Aleatoric and epistemic uncertainty are commonly separated through predictive variance and ensemble disagreement, respectively [34], and uncertainty-modulated policies have been applied in cooperative multi-agent reinforcement learning [42]. CURE adopts a deliberately lightweight Gaussian ensemble, consisting of a shared recurrent backbone with independently masked prediction heads, to separate the two uncertainty components without introducing a full variational state-space model.

3. Problem formulation

We represent the heterogeneous multi-agent autonomous system at time t by the dynamic graph

$$\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t, \mathcal{T}, \mathcal{R}, \mathbf{m}_t), \quad (3.1)$$

where $\mathcal{V} = \{v_1, \dots, v_N\}$ is the fixed node index set; $\mathcal{E}_t$ is the time-varying relation set; $\mathcal{T} = \{S, D, I, T\}$ is the set of sensor, decision, influence, and target node types; and $\mathcal{R}$ denotes the functional relation types. The binary variable $m_{i,t} \in \{0, 1\}$ records whether node v_i is operational.

Building on prior knowledge and functional-chain configurations established in previous studies [17, 20, 21], we prescribe four representative kill-chain metapaths to characterize the heterogeneous dependencies among sensing, decision, influence, and target nodes. A generic metapath P_p is defined as

$$P_p = \left(T_1 \xrightarrow{R_1} T_2 \xrightarrow{R_2} \dots \xrightarrow{R_{L_p-1}} T_{L_p} \right), \quad P_p \in \mathcal{P}, \quad (3.2)$$

where $T_l \in \mathcal{T}$ denotes the node type at the l th functional position, $R_l \in \mathcal{R}$ denotes the corresponding functional relation, and L_p is the number of positions in P_p . The adopted metapath set is therefore

$$\begin{aligned} \mathcal{P} = \bigg\{ & P_1 : S \rightarrow D \rightarrow I \rightarrow T \quad (\text{SDIT}), \\ & P_2 : S \rightarrow S \rightarrow D \rightarrow I \rightarrow T \quad (\text{SSDIT}), \\ & P_3 : S \rightarrow D \rightarrow S \rightarrow D \rightarrow I \rightarrow T \quad (\text{SDSDIT}), \\ & P_4 : S \rightarrow D \rightarrow D \rightarrow I \quad (\text{SDDI}) \bigg\}. \end{aligned} \quad (3.3)$$

These metapaths encode functional priors derived from domain knowledge and define the path-prediction targets used for the prescribed graph edits.

A graph encoder f_θ maps the current graph to a fixed-dimensional matrix of node representations

$$\mathbf{X}_t = [\mathbf{h}_{1,t}, \dots, \mathbf{h}_{N,t}]^\top = f_\theta(\mathcal{G}_t) \in \mathbb{R}^{N \times d}, \quad (3.4)$$

where $\mathbf{h}_{i,t} \in \mathbb{R}^d$ represents the structural and functional state of node v_i . The representation of every inactive node is masked before being supplied to the decision network.

Each node acts as an agent with the following local action candidates:

$$\mathcal{A}_i = \{\text{noop}\} \cup \{\text{replace}(j)\}_{j=1}^N \cup \{\text{cooperate}(j)\}_{j=1}^N. \quad (3.5)$$

Replacement assigns a compatible substitute to a failed function, whereas cooperation temporarily shares the capability of a source node. Let M^{type} , M^{rel} , M^{avail} , and M^{cap} denote binary feasibility masks. The joint feasibility mask is then

$$\begin{aligned} M_{i,t}(a) &= M_{i,t}^{\text{type}}(a)M_{i,t}^{\text{rel}}(a)M_{i,t}^{\text{avail}}(a)M_{i,t}^{\text{cap}}(a), \\ \mathcal{A}_{i,t}^{\text{valid}} &= \{\text{noop}\} \cup \{a \in \mathcal{A}_i \setminus \{\text{noop}\} : M_{i,t}(a) = 1\}, \\ \mathcal{A}_t^{\text{sys}} &= \left\{ \mathbf{a} \in \prod_i \mathcal{A}_{i,t}^{\text{valid}} : \sum_i \mathbf{1}[a_i \neq \text{noop}] \leq 1 \right\}. \end{aligned} \quad (3.6)$$

The execution arbiter retains the highest-valued admissible non-noop proposal and maps every other component to noop.

The environment evolves according to the graph-level transition

$$\mathcal{G}_{t+1} = \mathcal{F}(\mathcal{G}_t, \mathbf{a}_t, \boldsymbol{\varepsilon}_t). \quad (3.7)$$

Here, $\boldsymbol{\varepsilon}_t$ denotes external disruption. Let $x_{i,j,t}$ and $y_{i,j,t}$ indicate active replacement and cooperation assignments from source j to destination i , respectively. Their resource constraints are

$$\begin{aligned} x_{i,j,t}, y_{i,j,t} &\in \{0, 1\}, \quad \sum_i x_{i,j,t} \leq B_j^{\text{rep}}, \quad \sum_i y_{i,j,t} \leq 1, \\ \Delta t_{\text{rep}} &= \Delta t_{\text{coop}} = 1, \quad \ell_{\text{rep}}(t) = \min(t_{\text{fail}}, T) - t, \quad \ell_{\text{coop}} = 1. \end{aligned} \quad (3.8)$$

Here, B_j^{rep} is the finite inventory of spare capabilities, Δt is the operation duration, and ℓ is the assignment lifetime. The variable t_{fail} denotes the next relevant failure. Replacement reserves the source slot for ℓ_{rep} , whereas cooperation preserves the source's primary function.

The corresponding relation sets therefore satisfy

$$\begin{aligned} \mathcal{E}_{t+1} &= (\mathcal{E}_t \setminus [\mathcal{E}_t^{\text{coop}} \cup \mathcal{E}_t^{\text{fail}}]) \cup \Delta \mathcal{E}_t^{\text{rep}}(\mathbf{a}_t) \cup \Delta \mathcal{E}_t^{\text{coop}}(\mathbf{a}_t), \\ \mathbf{X}_{t+1} &= f_\theta(\mathcal{G}_{t+1}). \end{aligned} \quad (3.9)$$

The sets $\mathcal{E}_t^{\text{coop}}$ and $\mathcal{E}_t^{\text{fail}}$ contain expiring support relations and relations removed by disruption, respectively. The $\Delta \mathcal{E}$ terms contain restored and newly established support relations.

Let $f_i \geq 0$ denote the observed mission performance, and let $f_b > 0$ denote its nominal baseline. The normalized performance is $\tilde{f}_i = \text{clip}(f_i/f_b, 0, 1)$. Resilience is represented by three complementary

subrewards:

$$\begin{aligned}
 r_{\text{SP}}(t) &= \bar{f}_t, \\
 r_{\text{RR}}(t) &= \mathbf{1}_{t \geq t_m} \exp[-\lambda_r(t - t_m)] \text{clip}\left(\frac{\bar{f}_t - \bar{f}_{t-1}}{\Delta_{\max}}, 0, 1\right), \\
 r_{\text{SSP}}(t) &= \mathbf{1}_{t \geq t_s} \text{clip}\left(\mu_t^W - \lambda_v(\sigma_t^W)^2, 0, 1\right), \\
 \mu_t^W &= \frac{1}{W} \sum_{\tau=t-W+1}^t \bar{f}_\tau, \quad (\sigma_t^W)^2 = \frac{1}{W} \sum_{\tau=t-W+1}^t (\bar{f}_\tau - \mu_t^W)^2.
 \end{aligned} \tag{3.10}$$

Here, t_m denotes the most recent attack onset, $\Delta_{\max} > 0$ normalizes the largest meaningful one-step recovery, and t_s is the first time at which performance remains above a prescribed recovery threshold for a fixed number of consecutive steps.

The normalized multiobjective reward vector is

$$\mathbf{r}_t^{\text{sub}} = [r_{\text{SP}}(t+1), r_{\text{RR}}(t+1), r_{\text{SSP}}(t+1)]^\top \in [0, 1]^3. \tag{3.11}$$

The functions $r_j(t)$ are indexed by observations. The reward assigned to decision t is evaluated from the resulting observation at $t+1$.

The learning objective is to obtain a decentralized joint policy that maximizes the expected discounted robust return over the T decision intervals:

$$\pi^* = \arg \max_{\pi} \mathbb{E}_{\pi, \mathcal{F}} \left[\sum_{t=0}^{T-1} \gamma^t R_t^{\text{rob}} \right], \quad 0 < \gamma < 1, \tag{3.12}$$

where the uncertainty-aware scalar reward R_t^{rob} is defined in Section 4.2.

4. Method

Figure 1 summarizes CURE. In Phase I, the model learns intervention-consistent node representations from heterogeneous autonomous-system graphs and prescribed kill-chain metapaths by combining relation-compatible structural interventions, future-state prediction, contrastive learning among nodes of matching types, and relation-consistency regularization. In Phase II, the pretrained graph encoder is frozen. Its node embeddings are pooled into a type-aware system state, and an ensemble Gaussian world model estimates predictive uncertainty to regulate the aggregation of system-performance, rapid-recovery, and steady-state-performance rewards. These uncertainty-aware rewards drive type-shared recurrent local Q -networks and a monotonic QMIX mixer, which learn cooperative recovery policies under centralized training and decentralized execution.

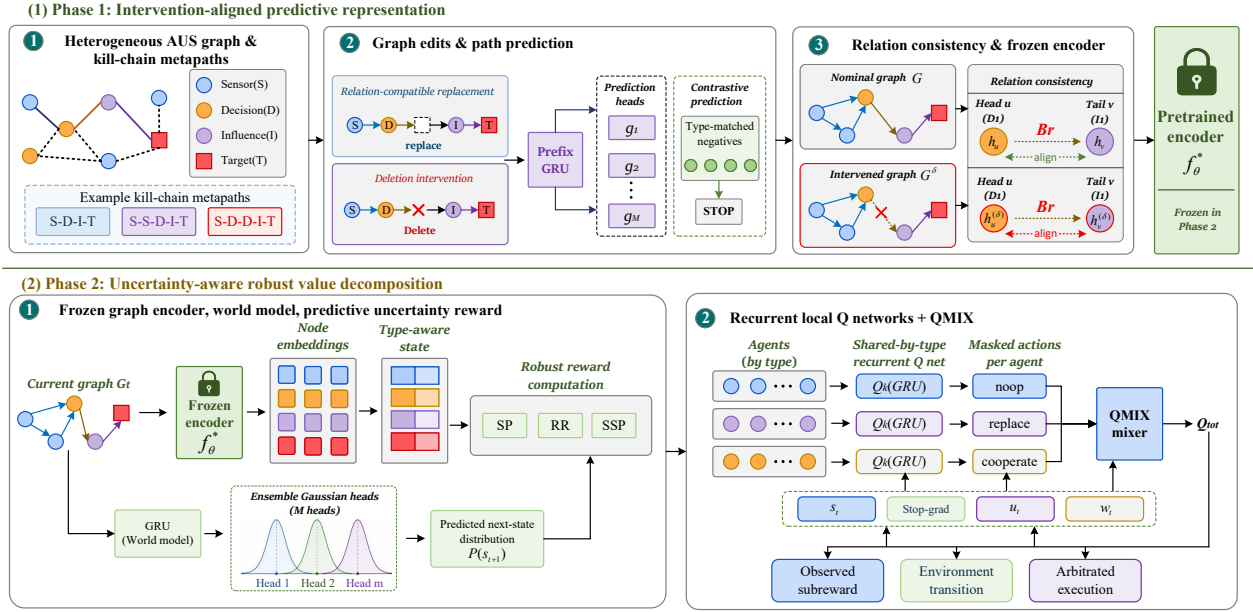


Figure 1. Overview of CURE. Phase I learns functional representations from nominal and edited paths. Phase II combines a frozen encoder, predictive uncertainty, and recurrent value decomposition. Local action proposals are subject to system-level arbitration under the one-operation budget.

4.1. Phase I: Intervention-aligned predictive representation

For an intervention environment $\delta \in \mathcal{D} = \{\text{obs}, \text{rep}, \text{del}\}$, let $\mathcal{G}^{(\delta)}$ denote the nominal graph or the graph augmented by a replacement or deletion edit. A shared two-layer relation-aware graph encoder produces

$$\mathbf{H}^{(\delta)} = f_\theta(\mathcal{G}^{(\delta)}) = [\mathbf{h}_1^{(\delta)}, \dots, \mathbf{h}_N^{(\delta)}]^\top. \quad (4.1)$$

The parameters θ are shared across environments. Each edit changes both the graph and its prediction targets, whereas the relation-scoring operators remain common.

For each nominal path $I_p^{(\text{obs})} = (v_1, \dots, v_{L_p})$, we sample $j \sim \text{Uniform}\{1, \dots, L_p - 1\}$ and construct one admissible replacement example and one deletion example. With the availability and capability indicator $c_j(u)$, the corresponding operators are

$$\begin{aligned} C_j^{\text{rep}} &= \{u \neq v_j : \tau(u) = \tau(v_j), c_j(u) = 1, (v_{j-1}, R_{j-1}, u) \in \mathcal{E}\}, \\ \mathcal{G}^{(\text{rep})} &= \mathcal{I}_{\text{rep}}(\mathcal{G}^{(\text{obs})}, v_j, v), \quad v \sim \text{Uniform}(C_j^{\text{rep}}), \\ \mathcal{G}^{(\text{del})} &= \mathcal{I}_{\text{del}}(\mathcal{G}^{(\text{obs})}, v_j), \\ \tilde{\mathbf{h}}_j^{(\text{del})} &= \mathbf{e}_{\text{del}} + \text{Mean}\{\mathbf{h}_u^{(\text{del})} : u \in \mathcal{N}(v_j), m_u = 1\}. \end{aligned} \quad (4.2)$$

The predecessor constraint is omitted when $j = 1$, and an empty candidate set is resampled. The deletion aggregate uses pre-edit neighbors that remain available after the edit, with $\text{Mean}(\emptyset) = \mathbf{0}$. Deletion therefore represents node failure during augmentation.

The path suffix is reconstructed in the edited graph. The first unavailable continuation is replaced by STOP and included in $\tilde{L}_p$, and all later horizons are masked.

For a fixed context $C = (\mathcal{G}^{(\text{obs})}, I_p^{(\text{obs})}, j)$, let $E = e$ specify an admissible edit, and let U denote the randomness used to sample the suffix. The augmentation has the following structural form [54, 27]:

$$\begin{aligned} \mathcal{G}' &:= I_E(\mathcal{G}^{(\text{obs})}), & Y &:= \Phi_P(\mathcal{G}', C, U), \\ \mathcal{G}^e &:= I_e(\mathcal{G}^{(\text{obs})}), & Y^e &:= \Phi_P(\mathcal{G}^e, C, U), \end{aligned} \quad (4.3)$$

$$\Pr_{\mathcal{M}}(Y = y \mid \text{do}(E = e), C) = \int \mathbf{1}[\Phi_P(\mathcal{G}^e, C, u) = y] dP(u \mid C).$$

Here, Φ_P is the common path-continuation rule, including STOP, and $\mathcal{M}$ denotes the specified simulator. Setting $E = e$ changes availability or assignment while leaving Φ_P and the sampling law unchanged. The resulting Y^e supplies the interventional prediction targets used below.

Because the kill-chain paths contain only a small number of ordered functional positions, a single-layer gated recurrent unit (GRU) provides a compact autoregressive sequence representation:

$$\mathbf{c}_{p,t}^{(\delta)} = \text{GRU}_{\phi}(\tilde{\mathbf{h}}_{p,1}^{(\delta)}, \dots, \tilde{\mathbf{h}}_{p,t}^{(\delta)}) \in \mathbb{R}^{d_c}. \quad (4.4)$$

The GRU receives positions only up to t , thereby preventing future-path information from entering the context.

An independent prediction head is used for each horizon:

$$\tilde{\mathbf{h}}_{p,t+k}^{(\delta)} = g_k(\mathbf{c}_{p,t}^{(\delta)}), \quad 1 \leq k \leq \min(K, \tilde{L}_p - t). \quad (4.5)$$

Each prediction head maps the recurrent context into the node-embedding space.

Let $y_{p,t,k}^{(\delta)} = \tilde{v}_{t+k}$ denote the future node at the corresponding horizon in the factual or edited path. Negative samples are drawn from nodes with the same type as the positive target. For a terminated path, the positive target is the STOP token, whereas the negative set contains valid nodes of the expected functional type. The STOP token is excluded from ordinary node negatives. Only valid (p, t, k, δ) tuples are counted in N_p , and a tuple is not duplicated when a terminated path is padded.

The contrastive loss is

$$\ell_{\text{pred}}^{(\delta)}(p, t, k) = -\log \frac{\exp\left(\text{sim}\left(\tilde{\mathbf{h}}_{p,t+k}^{(\delta)}, \mathbf{h}_{y_{p,t,k}^{(\delta)}}^{(\delta)}\right) / \tau_p\right)}{\exp\left(\text{sim}\left(\tilde{\mathbf{h}}_{p,t+k}^{(\delta)}, \mathbf{h}_{y_{p,t,k}^{(\delta)}}^{(\delta)}\right) / \tau_p\right) + \sum_{u \in \mathcal{N}_{p,t,k}^-} \exp\left(\text{sim}\left(\tilde{\mathbf{h}}_{p,t+k}^{(\delta)}, \mathbf{h}_u^{(\delta)}\right) / \tau_p\right)}, \quad (4.6)$$

where $\text{sim}(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \mathbf{y} / (\|\mathbf{x}\|_2 \|\mathbf{y}\|_2)$. Matching the negative samples to the target type prevents the model from solving the prediction task using node-type differences alone.

For each relation r , a shared linear operator $\mathbf{B}_r$ represents the associated functional transformation:

$$s_r^{(\delta)}(u, v) = -\left\| \mathbf{B}_r \mathbf{h}_u^{(\delta)} - \mathbf{h}_v^{(\delta)} \right\|_2^2. \quad (4.7)$$

Using the same relation operator in every intervention environment yields the relation-consistency loss

$$\mathcal{L}_{\text{rel}} = -\frac{1}{N_E} \sum_{\delta \in \mathcal{D}} \sum_{(u,r,v) \in \mathcal{E}^{(\delta)}} \log \frac{\exp\left(s_r^{(\delta)}(u, v) / \tau_r\right)}{\exp\left(s_r^{(\delta)}(u, v) / \tau_r\right) + \sum_{v^- \in \mathcal{N}_r^-} \exp\left(s_r^{(\delta)}(u, v^-) / \tau_r\right)}. \quad (4.8)$$

Here, v^- has the same type as v , and N_E counts positive relation instances. The shared operator $\mathbf{B}_r$ enforces relation consistency by applying the same edge-classification task in every environment.

The complete Phase I objective is

$$\mathcal{L}_I = \frac{1}{N_P} \sum_{\delta, p, t, k} \lambda_\delta \ell_{\text{pred}}^{(\delta)}(p, t, k) + \lambda_{\text{rel}} \mathcal{L}_{\text{rel}}. \quad (4.9)$$

The summation contains the N_P valid prediction tuples. The coefficients λ_δ balance the intervention environments, whereas λ_{rel} controls the contribution of relation supervision.

Phase I updates $\Theta_I = \{\theta, \phi, \{g_k\}, \{\mathbf{B}_r\}, \mathbf{e}_{\text{del}}, \mathbf{h}_{\text{STOP}}\}$. Phase II fixes $\theta = \theta^*$, thereby preserving the learned representation while the world model and value networks adapt.

In effect, the intervention module edits the graph, and the prediction and relation losses encode the consequences of that edit in the shared representation, so the encoder handed to Phase II is therefore aligned with the graph states that recovery creates.

4.2. Phase II: Uncertainty-aware robust value decomposition

For each functional type $q \in \mathcal{T}$, operational node representations are pooled separately to form the global state

$$\mathbf{s}_t = \left[\left\{ \frac{\sum_{i:\tau(i)=q} m_{i,t} \mathbf{h}_{i,t}}{\max(1, \sum_{i:\tau(i)=q} m_{i,t})} \right\}_{q \in \mathcal{T}} ; \left\{ \frac{\sum_{i:\tau(i)=q} m_{i,t}}{|\mathcal{V}^q|} \right\}_{q \in \mathcal{T}} ; \bar{f}_t \right]. \quad (4.10)$$

The state has dimension $d_s = 4d + 5$ and retains a separate representation for each functional type.

Before the recurrent system state is updated, the preceding joint action is compressed through a shared action embedding:

$$\begin{aligned} \bar{\mathbf{a}}_{t-1} &= \frac{1}{N} \sum_{i=1}^N E_a(a_{i,t-1}), \\ \mathbf{z}_t &= \text{GRU}_\psi(\mathbf{z}_{t-1}, [\mathbf{s}_t; \bar{\mathbf{a}}_{t-1}]). \end{aligned} \quad (4.11)$$

To distinguish environmental randomness from model uncertainty without introducing a full variational state-space model, CURE uses M lightweight Gaussian prediction heads:

$$p_m(\mathbf{s}_{t+1} | \mathbf{z}_t, \bar{\mathbf{a}}_t) = \mathcal{N}(\boldsymbol{\mu}_{t+1|t}^{(m)}, \text{diag}[\boldsymbol{\sigma}_{t+1|t}^{2(m)}]). \quad (4.12)$$

The heads share a recurrent backbone. For sampled segment q and prediction head m ,

$$b_q^{(m)} \sim \text{Bernoulli}(0.8), \quad b_b^{(m)} = b_{q(b)}^{(m)}, \quad (\sigma_{b,\ell}^{(m)})^2 \geq 10^{-6}, \quad (4.13)$$

where $q(b)$ maps a learning transition to its segment. The burn-in prefix reconstructs hidden states and is excluded from the loss. A head with no active target contributes zero.

The world model is trained using the Gaussian negative log-likelihood:

$$\mathcal{L}_{\text{wm}} = \frac{1}{N_{\text{mask}}} \sum_{m=1}^M \sum_{b=1}^B \sum_{\ell=1}^{d_s} b_b^{(m)} \left[\frac{(s'_{b,\ell} - \mu_{b,\ell}^{(m)})^2}{2(\sigma_{b,\ell}^{(m)})^2} + \frac{1}{2} \log(\sigma_{b,\ell}^{(m)})^2 \right], \quad (4.14)$$

where $b_b^{(m)}$ is the bootstrap mask, $N_{\text{mask}} = \max\{1, d_s \sum_{m,b} b_b^{(m)}\}$, and a positive variance floor is imposed during implementation.

At time $t - 1$, the slowly updated target world model predicts the distribution of $\mathbf{s}_t$. Before selecting the action at time t , CURE summarizes confidence in this prediction as

$$\begin{aligned} u_t^{\text{ale}} &= \frac{1}{Md_s} \sum_{m=1}^M \sum_{\ell=1}^{d_s} (\sigma_{t|t-1,\ell}^{(m)})^2, \\ u_t^{\text{epi}} &= \frac{1}{Md_s} \sum_{m=1}^M \|\boldsymbol{\mu}_{t|t-1}^{(m)} - \bar{\boldsymbol{\mu}}_{t|t-1}\|_2^2, \\ \bar{\boldsymbol{\mu}}_{t|t-1} &= \frac{1}{M} \sum_{m=1}^M \boldsymbol{\mu}_{t|t-1}^{(m)}. \end{aligned} \quad (4.15)$$

The first term is the mean predicted variance, whereas the second measures disagreement among the independently trained heads.

The bounded uncertainty signal is

$$\widehat{u}_t = \frac{\lambda_a u_t^{\text{ale}} + \lambda_e u_t^{\text{epi}}}{\lambda_a u_t^{\text{ale}} + \lambda_e u_t^{\text{epi}} + c_u} \in [0, 1). \quad (4.16)$$

The scale c_u is estimated from warm-up data. The initial values are

$$\mathbf{z}_{-1} = \mathbf{0}, \quad \mathbf{a}_{-1} = \mathbf{noop}, \quad \widehat{u}_0 = \frac{1}{2}, \quad \chi_0 = \frac{1}{2}(\chi_{\min} + \chi_{\max}). \quad (4.17)$$

For $t \geq 1$, $\widehat{u}_t$ summarizes the prediction emitted at $t - 1$ and is shared by all candidate actions.

The objective weights are determined analytically rather than produced by an additional trainable policy. Let $\bar{r}_{j,t}^-$ denote the predecision moving average of objective j :

$$\begin{aligned} g_t &= \mathbf{1}_{t \geq t_a}, \\ \bar{r}_{j,0}^- &= \xi_j, \\ \bar{r}_{j,t+1}^- &= (1 - g_t) \bar{r}_{j,t}^- + g_t [(1 - \rho_r) \bar{r}_{j,t}^- + \rho_r r_{j,t}], \\ d_{j,t} &= [\xi_j - \bar{r}_{j,t}^-]_+. \end{aligned} \quad (4.18)$$

Here, ξ_j denotes the desired level of objective j , t_a is the attack onset, and g_t is the attack gate. Before the attack, $g_t = 0$ and the deficit statistics remain at their neutral initial values. Once the gate opens, the moving averages are updated from the observed subrewards:

$$d_{j,0} = 0, \quad \omega_{j,0} = \frac{1}{3}, \quad j \in \{\text{SP}, \text{RR}, \text{SSP}\}. \quad (4.19)$$

At decision t , the weights depend only on rewards available through the preceding completed transition. Consequently, an attack at t_a affects the weights at the first decision following the attack transition.

The normalized objective weights are

$$\omega_{j,t} = \epsilon_\omega + (1 - 3\epsilon_\omega) \frac{\exp(d_{j,t}/\tau_\omega)}{\sum_{l=1}^3 \exp(d_{l,t}/\tau_\omega)}, \quad \sum_{j=1}^3 \omega_{j,t} = 1. \quad (4.20)$$

A larger deficit receives greater weight. The parameter τ_ω controls the sensitivity of this allocation, and $\epsilon_\omega > 0$ preserves a nonzero contribution from every objective.

The uncertainty signal controls a positive risk temperature:

$$\chi_t = \chi_{\min} + (\chi_{\max} - \chi_{\min})(1 - \widehat{u}_t), \quad 0 < \chi_{\min} < \chi_{\max}. \quad (4.21)$$

Higher predictive uncertainty decreases χ_t , thereby strengthening the influence of low subrewards in the aggregation below. Lower uncertainty permits a smoother compromise among the objectives.

After the current joint action has been executed and $\mathbf{r}_t^{\text{sub}}$ observed, the scalar team reward is

$$R_t^{\text{rob}} = -\chi_t \log \left[\sum_{j=1}^3 \omega_{j,t} \exp \left(-\frac{r_{j,t}}{\chi_t} \right) \right]. \quad (4.22)$$

For positive weights,

$$\lim_{\chi \rightarrow 0^+} R^{\text{rob}}(\chi) = \min_j r_j, \quad \lim_{\chi \rightarrow \infty} R^{\text{rob}}(\chi) = \sum_j \omega_j r_j. \quad (4.23)$$

Thus, deficits allocate relative emphasis, whereas uncertainty controls how strongly low-performing objectives influence the reward.

Increasing a single weight preselects the objective to emphasize, whereas lowering the temperature acts on whichever objective currently performs worst; at low χ_t , a high subreward cannot compensate for a low one.

The local observation available to agent i is

$$\begin{aligned} \mathbf{o}_{i,t} &= [\mathbf{h}_{i,t}; m_{i,t}; \widehat{u}_t; \omega_t], \\ \mathbf{g}_{i,t} &= \text{GRU}_{\eta_{\tau(i)}}(\mathbf{g}_{i,t-1}, [\mathbf{o}_{i,t}; a_{i,t-1}]), \\ Q_i(\tau_{i,t}, a_i) &= Q_{\eta_{\tau(i)}}(\mathbf{g}_{i,t}, a_i). \end{aligned} \quad (4.24)$$

Agents of the same functional type share network parameters. Before either greedy or exploratory selection, invalid actions are assigned a sufficiently negative value.

During centralized training, individual action values are combined by a mixing network:

$$\begin{aligned} Q_{\text{tot}} &= f_{\text{mix}}(Q_1, \dots, Q_N; \mathbf{x}_t), \\ \mathbf{x}_t &= [\mathbf{s}_t; \text{stopgrad}(\mathbf{z}_t); \widehat{u}_t; \omega_t], \\ \frac{\partial Q_{\text{tot}}}{\partial Q_i} &\geq 0. \end{aligned} \quad (4.25)$$

Non-negative mixing weights enforce monotonicity. Under the coupled budget in Eq (3.6), local greedy proposals must still be resolved by the execution arbiter.

Using double Q-learning, we define the temporal-difference target as

$$y_t = R_t^{\text{rob}} + \gamma(1 - d_t)Q_{\text{tot}}^-(\mathbf{x}_{t+1}, \mathbf{a}_{t+1}^\star), \quad \mathbf{a}_{t+1}^\star = \mathfrak{A}_{t+1} \left(\left\{ \arg \max_{a_i \in \mathcal{A}_{i,t+1}^{\text{valid}}} Q_i(\tau_{i,t+1}, a_i) \right\}_{i=1}^N \right). \quad (4.26)$$

Here, d_t indicates termination, and $\mathfrak{A}_{t+1}$ applies the execution arbiter to the valid greedy proposals and their scores. The superscript $\star$ denotes the selected feasible bootstrap action.

The value-decomposition loss is therefore

$$\mathcal{L}_{\text{TD}} = \mathbb{E}_{\mathcal{D}} \left[(Q_{\text{tot}}(\mathbf{x}_t, \mathbf{a}_t) - y_t)^2 \right]. \quad (4.27)$$

Because both the world model and the local value models are recurrent, the replay buffer stores complete episodes and samples contiguous segments with a burn-in prefix to reconstruct hidden states.

Let Θ_{wm} denote the online recurrent world model and its prediction heads and Θ_{wm}^- its slowly updated target copy, and let Θ_Q denote the online recurrent local value networks and mixing network, with target copy Θ_Q^- . The online and target parameters are updated according to

$$\Theta_{\text{wm}}^- \leftarrow \rho_{\text{wm}} \Theta_{\text{wm}} + (1 - \rho_{\text{wm}}) \Theta_{\text{wm}}^-, \quad \Theta_Q^- \leftarrow \rho_Q \Theta_Q + (1 - \rho_Q) \Theta_Q^-. \quad (4.28)$$

The superscript $-$ denotes a delayed target copy. The target world model supplies the uncertainty signal used during interaction.

The two Phase II losses use disjoint gradient paths:

$$\nabla_{\Theta_{\text{wm}}} \mathcal{L}_{\text{TD}} = \mathbf{0}, \quad \nabla_{\Theta_Q} \mathcal{L}_{\text{wm}} = \mathbf{0}, \quad \nabla_{\theta^*} (\mathcal{L}_{\text{wm}} + \mathcal{L}_{\text{TD}}) = \mathbf{0}. \quad (4.29)$$

The replay buffer stores detached uncertainty estimates, objective weights, and rewards. Consequently, temporal-difference optimization updates only the value networks, whereas likelihood optimization updates only the world model.

The time-consistent execution order at decision time t is

$$\mathcal{G}_t \rightarrow \mathbf{X}_t \rightarrow \mathbf{s}_t \rightarrow (\mathbf{z}_t, \widehat{\mathbf{u}}_t) \rightarrow \omega_t \rightarrow \mathbf{a}_t \rightarrow \mathcal{G}_{t+1} \rightarrow \mathbf{r}_t^{\text{sub}} \rightarrow R_t^{\text{rob}}. \quad (4.30)$$

At decision time t , the policy uses the current graph, the uncertainty predicted at time $t - 1$, and objective statistics accumulated through time $t - 1$. Neither the current reward nor the next state is available before the current action is selected.

In short, the frozen encoder produces the state, the ensemble sets the confidence that determines the risk temperature, and the deficits and temperature together form the scalar reward that trains the cooperative value networks.

5. Experiments

5.1. Experimental environment and settings

Swarm topology. The autonomous unmanned swarm is modeled as a directed heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T}, \mathcal{R})$ with four functional node types, $\mathcal{T} = \{S, D, I, T\}$. Sensor nodes (S) acquire measurements, decision nodes (D) fuse observations and issue directives, influence nodes (I) execute those directives, and target nodes (T) represent mission objects whose engagement determines performance. Table 1 summarizes the demonstration system. Its scale matches that of the environments used by recent baselines, thereby preserving compatibility with their published configurations, and Figure 2 presents one realized topology. The seven relation types encode the closed-loop functional semantics introduced in Section 3. Specifically, $T \rightarrow S$ denotes target sensing, $S \rightarrow S$ lateral fusion, $S \rightarrow D$ and $D \rightarrow S$ the exchange between perception and decision processes, $D \rightarrow D$ command consolidation, $D \rightarrow I$ command dissemination, and $I \rightarrow T$ engagement.

The benchmark follows the functional graph abstraction adopted in research on unmanned weapon system-of-systems (UWSoS) reconfiguration [4, 17, 25]. All methods are evaluated within the same simulator and under identical topology, attack realizations, feasibility masks, and recovery budgets.

Table 1. Basic properties of the simulated autonomous unmanned swarm.

Node type		Relation type	
Type	Number	Type	Number
S (sensor)	71	$T \rightarrow S$	71
D (decision)	45	$S \rightarrow S$	40
I (influence)	60	$S \rightarrow D$	45
T (target)	10	$D \rightarrow S$	42
Total nodes $ \mathcal{V} $	186	$D \rightarrow D$	39
Total edges $ \mathcal{E} $	339	$D \rightarrow I$	41
		$I \rightarrow T$	61

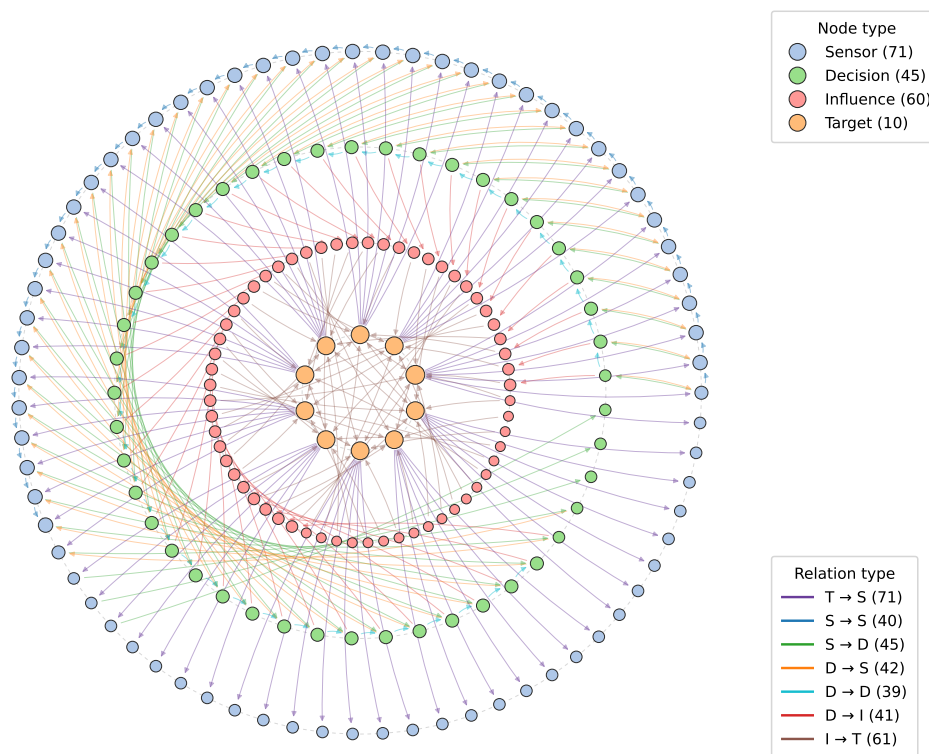


Figure 2. Realized swarm topology used in the experiments. Nodes are arranged in four concentric functional layers, ordered from the outside inward as sensor, decision, influence, and target. The seven relation types are shown in distinct colors, and node size is proportional to in-degree.

Mission performance. Let $n_{v,t}$ denote the number of operational metapath instances of type P_1 ($S \rightarrow D \rightarrow I \rightarrow T$) that reach target $v \in \mathcal{V}^T$ at time t . The raw mission performance is defined as the

mean coverage of targets by at least one operational path:

$$f_t = \frac{1}{|\mathcal{V}^T|} \sum_{v \in \mathcal{V}^T} \min(1, n_{v,t}), \quad \bar{f}_t = \text{clip}(f_t/f_b, 0, 1), \quad (5.1)$$

where f_b denotes the nominal coverage of the intact topology so that $\bar{f}_t = 1$ before disruption. Replacing a failed node restores its functional edges, whereas cooperation temporarily augments the effective coverage of the supported node. Both effects are represented in $\mathcal{G}_{t+1}$ and therefore contribute to f_{t+1} .

Attack scenarios and damage intensities. Following protocols established in recent resilience studies [23, 21], we consider eight attack scenarios that disable subsets of the operational layers S , D , and I at prescribed intensities. Specifically, SA disables 50% of the sensors, DA disables 50% of the decision nodes, and IA disables 50% of the influence nodes. SDA disables 25% of the sensors and 25% of the decision nodes, whereas SIA disables 25% of the sensors and 25% of the influence nodes. DIA disables 25% of the decision and influence nodes. MA distributes the same total damage budget evenly across all three layers, corresponding to 16.7% per layer, whereas RA disables 50% of all nodes uniformly at random. Each attack is injected instantaneously at step $t_a = 10$ in an episode of length $T = 100$. Because the policy may act from step 0, recovery actions unfold online throughout the episode. For robustness analyses, the damage intensity is varied over $\{30\%, 40\%, 50\%, 60\%, 70\%, 80\%\}$ while preserving the per-layer proportions of each scenario.

5.2. Evaluation metrics

The primary metric is the integral resilience score, defined as the time-averaged normalized mission performance over the post-attack horizon:

$$R = \frac{1}{T - t_a + 1} \sum_{t=t_a}^T \bar{f}_t. \quad (5.2)$$

This quantity is the discrete mean of the $T - t_a + 1$ post-attack performance observations, including the terminal observation. The trajectory-based measure is related to the performance-area metrics used in resilience assessment [9, 12, 13]. An episode contains T decision intervals, indexed by $0, \dots, T - 1$, and $T + 1$ state observations, indexed by $0, \dots, T$. The learning return in Eq (3.12) therefore contains T rewards, whereas Eq (5.2) evaluates the observed performance trajectory through its terminal state. The terminal observation contributes to evaluation but generates no additional action or reward.

We additionally report three trajectory-level indicators: the *recovery time* $\tau_{\text{rec}} = \min\{t \geq t_a : \bar{f}_t \geq 0.75 \text{ for } \tau = t, \dots, t + 4\}$, the *absorption capacity* $A = \min_{t_a \leq t \leq T} \bar{f}_t$, and the *recovery level* $L = \bar{f}_T$. Each configuration is evaluated over ten random seeds. We report the mean and standard deviation and compare CURE with the strongest baseline in each scenario using a two-sided Wilcoxon signed-rank test across seeds, with Holm–Bonferroni correction at significance level 0.05 [59]. Wherever possible, each learning method uses its published hyperparameters, and all methods are exposed to identical attack realizations to ensure paired comparisons [60].

5.3. Implementation and training protocol

Table 2 lists the principal hyperparameters. Phase I uses AdamW for at most 550 epochs, with early stopping after 10 epochs without improvement and gradient clipping at 1.0. Each sampled path provides one nominal example and one example for each admissible edit environment, as described in Section 4.1. Phase II uses 600,000 joint environment steps, corresponding to 6000 episodes of 100 decisions. A 30,000-step random-policy warm-up precedes Phase II learning. The uncertainty scale c_u is set to the 90th percentile of mixed uncertainty on the warm-up data so that a mixed uncertainty equal to this scale maps to $\widehat{u}_t = 0.5$. Training uses a single NVIDIA RTX 4090 GPU. In each ablation, all remaining hyperparameters are unchanged after the specified component is removed.

Table 2. Principal hyperparameters of CURE.

Component	Setting
Graph encoder	Two-layer relation-aware GCN, $d = 128$
Phase I prediction	GRU, $d_c = 64$, $K = 2$ heads, $\tau_p = \tau_r = 0.1$, and 128 type-matched negatives
Phase I optimizer	AdamW, lr 3×10^{-4} , weight decay 10^{-4} , up to 550 epochs
World model	Shared GRU, hidden size 128, $M = 5$ Gaussian heads, and bootstrap inclusion $p = 0.8$
Uncertainty and risk	$\lambda_a = \lambda_e = 1.0$, $\chi_{\min} = 0.05$, $\chi_{\max} = 1.0$, and c_u at the 90th percentile of warm-up data
Objective deficits	$\xi = (0.80, 0.60, 0.80)$, $\rho_r = 0.05$, $\tau_\omega = 0.10$, and $\epsilon_\omega = 0.05$
Subreward parameters	$W = 10$, $\lambda_v = 1.0$, $\lambda_r = 0.05$, $\Delta_{\max} = 0.2$, and threshold 0.75 over 5 steps
Local Q -networks	Type-shared GRU, hidden size 128, and mixer hypernetwork $64 \rightarrow 32$
Value learning	Adam with lr 5×10^{-4} , Double Q-learning, $\gamma = 0.95$, and ϵ -greedy $1.0 \rightarrow 0.05$ over 50k steps
Replay and targets	Episode buffer 10^5 , segments of length 60 with burn-in 20, batch size 128, and $\rho_{\text{wm}} = \rho_Q = 0.005$

5.4. Baselines

We compare CURE with six reinforcement-learning baselines. Hierarchical reinforcement learning with kill-chain-informed multiobjective optimization (*HRL-KCIMOO*) [23] combines kill-chain-informed pretraining with a hierarchical multiobjective controller operating over multi-agent deep deterministic policy gradient (MADDPG) agents. Graph diffusion reinforcement recovery (*GDRR*) [24] uses multiscale diffusion embeddings together with dependency-aware hierarchical PPO. Collaboration action resilience contribution index (CARCI)-guided proximal policy optimization (*CGPPO*) [19] combines a graph convolutional network (GCN) encoder with PPO, whereas ensemble graph convolutional network (*EGCN*) [20] stacks attention-enhanced GCN layers with an actor–critic policy. The extended comparison includes MPGCI [4] and MPD-RL [25], which represent two complementary metapath approaches: feature alignment with actor–critic optimization and gated representation fusion with value-based learning.

Each method, including MPGCI and MPD-RL, receives the same nominal 600,000-step policy-learning budget, measured in joint environment interactions, and the same hardware allocation while retaining its own representation module, reward construction, and published learning algorithm.

5.5. Comparative results

Table 3 compares seven methods across eight attack scenarios at 50% damage. CURE achieves the highest resilience in every scenario. Averaged across scenarios, it attains 0.6859, compared with 0.6611

for MPGCI and 0.6601 for MPD-RL, corresponding to relative gains of 3.74% and 3.90%, respectively. The advantage persists across distinct functional disruptions: Under IA, CURE reaches 0.521 versus 0.489 for the strongest baseline, MPGCI, an increase of 6.54%. Under DA, it reaches 0.588 versus 0.571 for MPD-RL. These additional comparisons support the value of combining edit-aware representation learning with uncertainty-sensitive recovery. Relative to EGCN and CGPPO, the average absolute improvement is 0.043. Relative to GDRR and HRL-KCIMOO, it is 0.032.

Table 3. Resilience score R (mean over 10 seeds) under eight attack scenarios at 50% damage. The best value in each scenario is shown in bold. The dagger denotes the best-performing method.

Method	SA	DA	IA	SDA	SIA	DIA	MA	RA
EGCN	0.701	0.529	0.467	0.689	0.639	0.523	0.800	0.795
CGPPO	0.709	0.538	0.461	0.682	0.637	0.527	0.799	0.794
GDRR	0.719	0.544	0.471	0.692	0.649	0.539	0.812	0.803
HRL-KCIMOO	0.721	0.552	0.474	0.694	0.646	0.543	0.810	0.795
MPGCI	0.727	0.563	0.489	0.710	0.643	0.538	0.812	0.807
MPD-RL	0.721	0.571	0.482	0.715	0.651	0.529	0.809	0.803
CURE [†]	0.752	0.588	0.521	0.731	0.669	0.571	0.831	0.824

Table 4. Trajectory-level indicators at 50% damage: Recovery time τ_{rec} (steps), absorption A , recovery level L , and volatility σ of $\hat{f}_t$ over the post-attack horizon.

Method	Scenario	τ_{rec}	A	L	σ
CURE	DA	21.6	0.412	0.782	0.031
	IA	24.3	0.337	0.749	0.036
	SIA	22.8	0.355	0.768	0.033
HRL-KCIMOO	DA	26.5	0.391	0.758	0.043
	IA	29.8	0.311	0.719	0.049
	SIA	28.1	0.338	0.744	0.046
GDRR	DA	27.4	0.385	0.749	0.044
	IA	30.6	0.303	0.709	0.050
	SIA	28.9	0.331	0.736	0.047
CGPPO	DA	26.8	0.389	0.755	0.042
	IA	30.2	0.308	0.715	0.048
	SIA	28.4	0.336	0.741	0.045
EGCN	DA	27.0	0.386	0.752	0.043
	IA	30.5	0.305	0.711	0.049
	SIA	28.6	0.332	0.737	0.046
MPGCI	DA	26.8	0.389	0.755	0.0433
	IA	30.1	0.308	0.716	0.0493
	SIA	28.4	0.336	0.741	0.0463
MPD-RL	DA	27.1	0.387	0.752	0.0437
	IA	30.3	0.306	0.712	0.0497
	SIA	28.6	0.333	0.739	0.0467

Table 4 confirms this pattern using trajectory-level indicators for the three most adversarial scenarios. CURE restores performance above the 0.75 threshold 4.9–5.5 steps earlier than HRL-KCIMOO and 5.4–6.2 steps earlier than EGCN. It also retains a higher post-strike minimum, indicating greater absorption capacity, and finishes at a higher level with lower volatility.

Because these indicators summarize the entire post-attack trajectory, the moderate differences in R correspond to operationally meaningful gains: earlier recovery above the 0.75 threshold, a higher minimum performance during the strike, and steadier mission output at the end of the horizon, with the margin widening as damage intensifies.

5.6. Convergence and sample efficiency

Figure 3 reports cumulative reward curves for CURE and the original four baselines over the complete 6000-episode Phase II budget under all attack scenarios at 50% damage. Solid lines denote means over ten seeds, and shaded regions indicate one standard deviation. The converged plateaus reproduce the corresponding per-scenario scores in Table 3, from which three observations emerge.

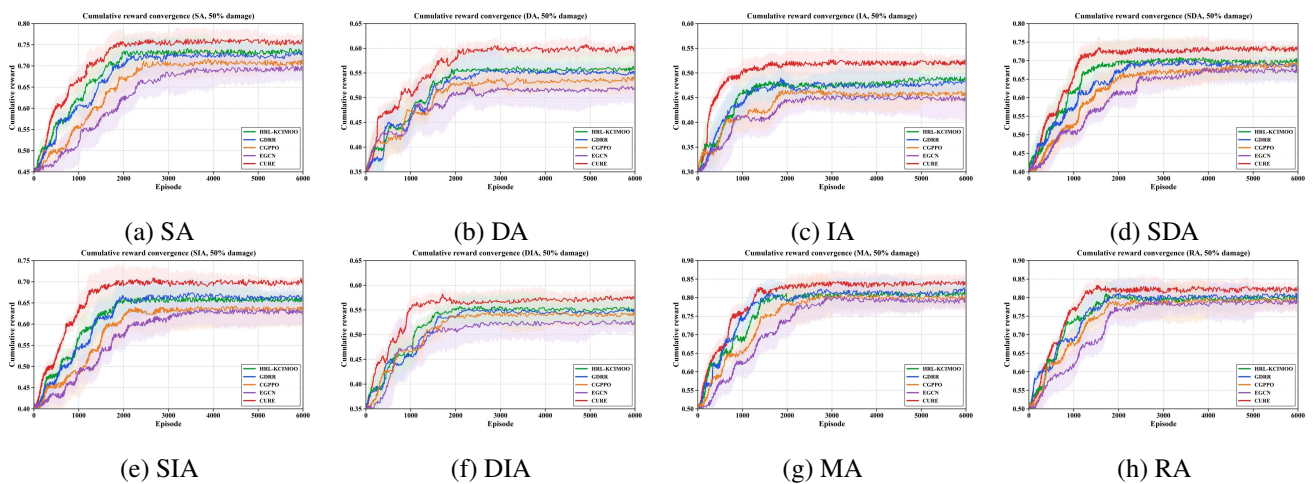


Figure 3. Cumulative reward during training for the five methods over 6000 episodes under eight attack scenarios at 50% damage: (a) SA, (b) DA, (c) IA, (d) SDA, (e) SIA, (f) DIA, (g) MA, and (h) RA. Solid lines denote means over ten seeds, and shaded regions indicate one standard deviation. Episode 0 corresponds to the no-recovery performance level of each scenario. CURE additionally benefits from Phase I pretraining.

First, CURE is the best-performing method in all eight scenarios and remains above the strongest baseline from the first episode through the end of training. This initial advantage is inherited from Phase I: The frozen action-aligned encoder provides functional representations from the outset, allowing CURE to begin above the no-recovery performance level of each scenario, whereas the four baselines start at that level and must construct their representations online.

Second, CURE converges fastest and exhibits the smallest fluctuations. It reaches 95% of its final reward after approximately 1900 episodes. Its one-standard-deviation band is also the narrowest throughout training: The baseline bands widen during the steep learning transient and contract only near the plateau, whereas CURE's band contracts monotonically, reflecting the stabilizing effect of the uncertainty-gated robust reward across seeds.

Third, the baselines separate into two groups. HRL-KCIMO, GDRR, and CGPPO behave similarly under the mixed and random attacks: Their plateaus span 0.799–0.812 in MA and 0.794–0.803 in RA, leaving CURE a clear margin at 0.831 and 0.824, respectively. EGCN is the slowest learner in every scenario because its graph encoder is trained online from scratch without pretraining. The functional

structure required for recovery must therefore be reconstructed from reward alone, and the method reaches the 95% point only after 3350 episodes.

5.7. Ablation study

Table 5 removes one component at a time. Phase I pretraining yields the largest average improvement: Removing it reduces the score from 0.686 to 0.663. The corresponding decreases under DA and IA are 0.022 and 0.029, respectively, consistent with the importance of multihop structure when substitutes are selected. Removing the replacement and deletion environments reduces the average by 0.010, removing predictive uncertainty reduces it by 0.007, and using uniform objective weights reduces it by 0.009. These results indicate complementary contributions: Edited-graph pretraining improves the functional representation, whereas predictive uncertainty and deficit weights improve reward aggregation.

Table 5. Ablation study at 50% damage (mean resilience over 10 seeds). “w/o” denotes the full CURE with the indicated component removed. Best per scenario in bold.

Variant	SA	DA	IA	SDA	SIA	DIA	MA	RA	Avg
CURE (full)	0.752	0.588	0.521	0.731	0.669	0.571	0.831	0.824	0.686
w/o Phase I pretraining	0.722	0.566	0.492	0.704	0.648	0.553	0.812	0.804	0.663
w/o rep/del interventions	0.742	0.578	0.509	0.722	0.659	0.560	0.824	0.816	0.676
w/o world-model uncertainty	0.744	0.581	0.514	0.724	0.661	0.564	0.825	0.817	0.679
w/o deficit-driven weights	0.742	0.579	0.511	0.721	0.659	0.562	0.824	0.815	0.677

5.8. Robustness to attack intensity and topology

Damage intensity. Figure 4 and the measured entries in Table 6 evaluate robustness over damage ratios from 30% to 80%. In the full sweep involving CURE and the original four baselines, CURE’s advantage over HRL-KCIMOO grows from 0.010 (1.1%) at 30% damage to 0.032 (4.9%) at 50% and 0.044 (17.5%) at 80%. At 50% damage, the extended comparison also includes MPGCI and MPD-RL, whose scenario-averaged scores are 0.661 and 0.660, respectively, whereas CURE reaches 0.686. The widening margin across the full sweep is consistent with the complementary roles of edit-aware embeddings and uncertainty-sensitive recovery under increasingly severe structural disruption.

Table 6. Average resilience score across the eight scenarios under increasing damage ratios.

Method	30%	40%	50%	60%	70%	80%
CURE	0.912	0.841	0.686	0.542	0.411	0.296
HRL-KCIMOO	0.902	0.822	0.654	0.508	0.371	0.252
GDRR	0.900	0.817	0.654	0.505	0.367	0.249
CGPPO	0.896	0.812	0.643	0.492	0.352	0.232
EGCN	0.895	0.810	0.643	0.490	0.349	0.228
MPGCI	0.904	0.813	0.661	0.509	0.363	0.256
MPD-RL	0.901	0.819	0.660	0.506	0.372	0.247

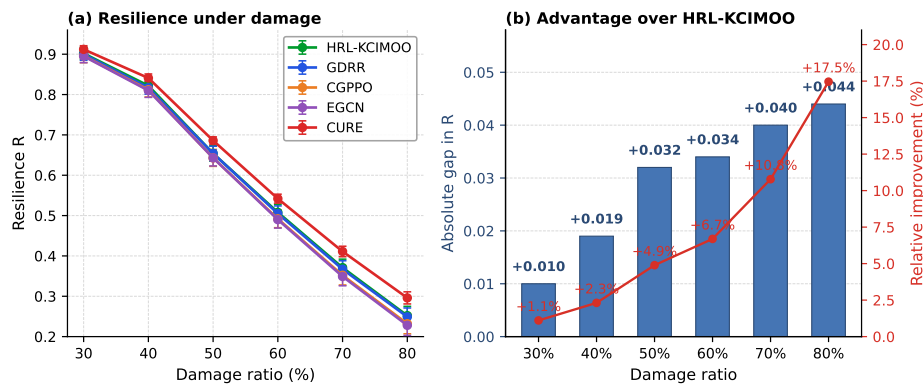


Figure 4. Robustness to damage intensity for CURE and the original four baselines. (a) Resilience score as a function of damage ratio, averaged over eight attack scenarios and ten seeds. Shaded regions indicate one standard deviation. (b) Advantage of CURE over HRL-KCIMOO. Bars show the absolute gap, and the line referenced to the right axis shows the relative improvement.

Topological generalization. To test whether the representation survives structural distribution shift rather than only variation in damage, we generate 100 random topologies by varying the node counts within $S \in [40, 100]$, $D \in [25, 60]$, $I \in [30, 80]$, and $T \in [5, 15]$, and we sample relations subject to the type constraints in Table 1. We then apply 50% damage under all eight scenarios. Figure 5 shows the resulting distribution of R . CURE attains the highest mean (0.679), the smallest standard deviation (0.008), and the strongest worst-case behavior: Its 5th percentile (0.665) exceeds those of HRL-KCIMOO (0.639) and EGCN (0.627) by 0.026 and 0.038, respectively. The baselines exhibit wider spreads and heavier left tails (standard deviations of 0.009–0.011), consistent with policies that overfit the connectivity patterns encountered during online interaction.

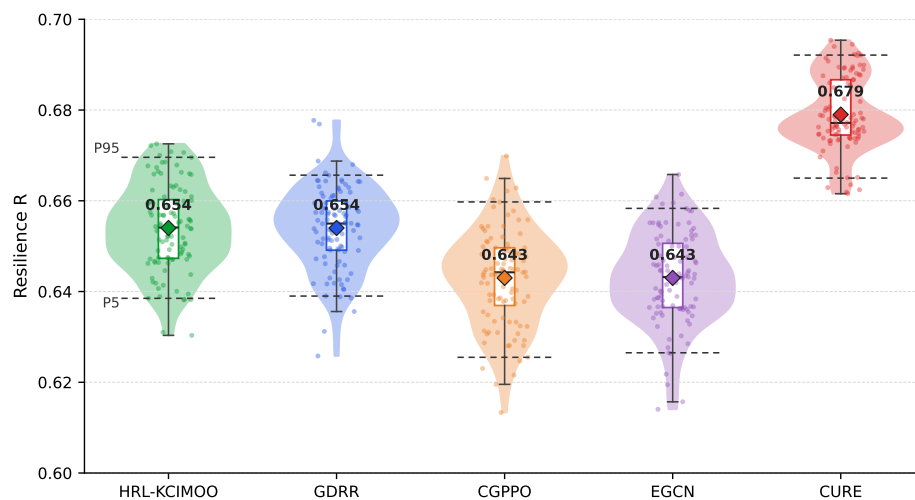


Figure 5. Distribution of the resilience score across 100 randomly generated topologies under 50% damage, averaged over the eight attack scenarios. Violins summarize each distribution, boxes mark quartiles, jittered points are individual topologies, and dashed lines mark the 5th and 95th percentiles.

5.9. Sensitivity analysis

Metapath configurations. Table 7 evaluates the pretraining metapath set. The primary chain P_1 alone recovers 98% of the full-model average (0.672 versus 0.686). The largest marginal gains arise from adding P_3 , which captures sensing–decision feedback, and P_4 , which captures collaborative decision consolidation. Both are particularly relevant under DA and SIA, where decision-layer redundancy is important. The gains begin to saturate once P_1 , P_2 , and P_3 are combined, indicating that the four priors are informative but partially redundant.

Table 7. Resilience score under different Phase I metapath configurations at 50% damage. Best per scenario in bold.

Metapath set	SA	DA	IA	SDA	SIA	DIA	MA	RA	Avg
P_1 only	0.735	0.574	0.507	0.716	0.655	0.558	0.820	0.813	0.672
P_1, P_2	0.739	0.578	0.511	0.720	0.659	0.562	0.823	0.816	0.676
P_1, P_3	0.745	0.582	0.516	0.725	0.663	0.567	0.827	0.819	0.681
P_1, P_4	0.743	0.581	0.514	0.724	0.662	0.565	0.826	0.818	0.679
P_1, P_2, P_3	0.748	0.585	0.518	0.728	0.666	0.568	0.829	0.821	0.683
P_1, P_2, P_3, P_4 (full)	0.752	0.588	0.521	0.731	0.669	0.571	0.831	0.824	0.686

Uncertainty and reward hyperparameters. Table 8 summarizes one-dimensional parameter sweeps. The ensemble is effective with $M = 3$ heads and reaches a plateau at $M = 5$. A single head removes the epistemic signal and reduces the score by 0.010, whereas $M = 10$ provides no further gain. The risk-temperature range also matters: An overly narrow range, with upper bound 0.2, overconstrains the optimistic regime (0.679), whereas increasing the conservative floor to $\chi_{\min} = 0.5$ likewise degrades performance (0.678). Equal weighting of aleatoric and epistemic uncertainty is preferred, and a two-step prediction horizon is sufficient. The deficit rate ρ_r is relatively insensitive within $[0.02, 0.10]$. Only highly reactive tracking with $\rho_r = 0.2$ becomes harmful. All results are averaged over eight scenarios at 50% damage and indicate robustness to moderate hyperparameter perturbations.

Table 8. Hyperparameter sensitivity (average R over eight scenarios at 50% damage). The adopted setting is in bold.

Parameter	Values	Avg R
Ensemble heads M	{1, 3, 5, 8, 10}	{0.676, 0.682, 0.686, 0.686, 0.684}
Risk bounds $[\chi_{\min}, \chi_{\max}]$	{[0.05, 0.2], [0.05, 1.0], [0.2, 1.0], [0.5, 1.0]}	{0.679, 0.686, 0.683, 0.678}
Uncertainty balance $\lambda_a:\lambda_e$	{0:1, 1:2, 1:1, 2:1, 1:0}	{0.680, 0.683, 0.686, 0.684, 0.682}
Prediction horizon K	{1, 2, 3, 4}	{0.679, 0.686, 0.685, 0.681}
Deficit rate ρ_r	{0.02, 0.05, 0.10, 0.20}	{0.683, 0.686, 0.684, 0.678}

5.10. Behavior of the uncertainty signal and robust reward

Figure 6 follows one evaluation episode under DIA. Before the strike, the world model is confident ($\widehat{u}_t \approx 0.06$), the risk temperature remains close to $\chi_{\max}$, and the attack-gated objective weights retain their neutral value (1/3, 1/3, 1/3). At the attack, performance collapses, and $\widehat{u}_t$ rises to approximately 0.78 within two steps. The temperature χ_t moves toward $\chi_{\min}$, and the objective weights begin adapting at the

first decision following the attack transition. During early recovery, the steady-state objective receives the largest deficit-driven weight because its observed reward remains below the desired level. As recovery proceeds, the rapid-recovery weight increases whenever its moving average remains below target. Across all evaluation episodes, the Pearson correlation between $\hat{u}_t$ and the absolute performance change $|\hat{f}_t - \hat{f}_{t-1}|$ is 0.71, indicating that the bounded uncertainty signal identifies the regime shifts for which conservative aggregation is intended. The two components also behave differently: The epistemic term dominates immediately after the strike, when the policy encounters fragmented states rarely observed by the world model, whereas the aleatoric term dominates the stochastic fluctuations of the recovery transient.

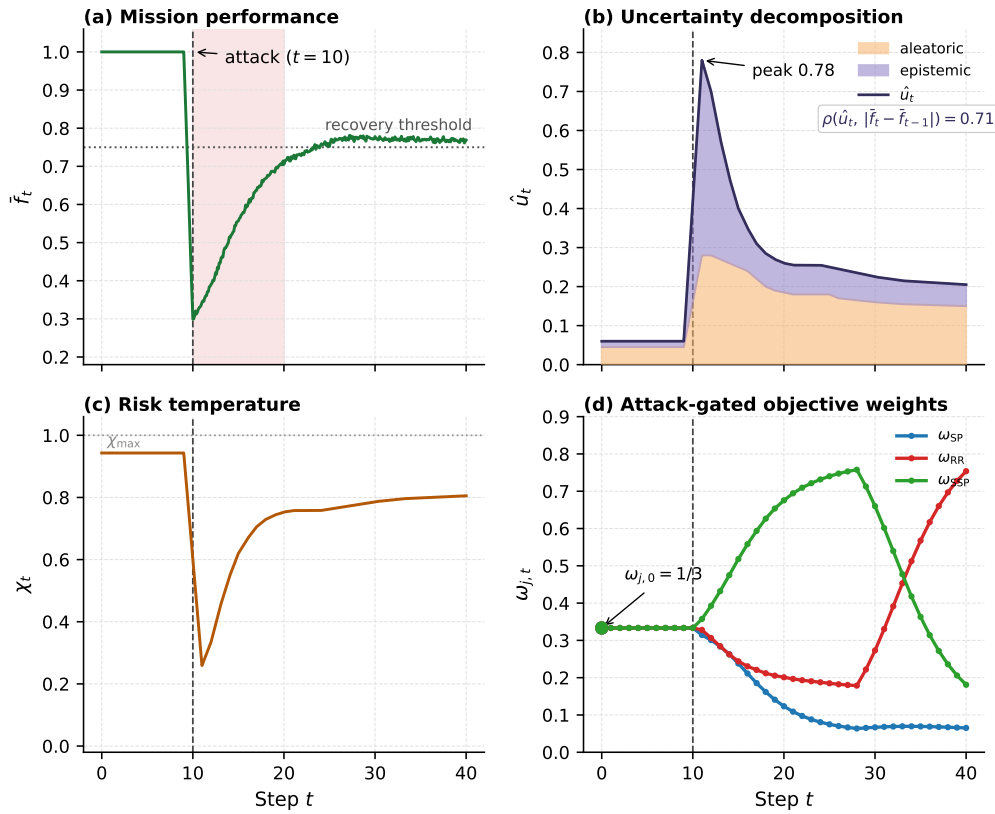


Figure 6. Decision-interval behavior of CURE in one DIA episode at 50% damage. (a) Normalized mission performance $\hat{f}_t$. The dashed line marks the recovery threshold 0.75, and the shaded band spans the ten steps following the attack at Step 10. (b) Bounded uncertainty $\hat{u}_t$ with its aleatoric and epistemic components. (c) Risk temperature χ_t . (d) Objective weights $\omega_{j,t}$. The weights remain uniform before the attack gate opens and adapt after the attack transition.

5.11. Encoder adaptation and predictive-uncertainty validation

To assess the contribution of freezing the pretrained encoder, we compare three Phase II settings: a frozen encoder, end-to-end fine-tuning, and updating only the final graph-convolution layer. All variants use the same Phase I checkpoint, Phase II budget, feasibility masks, and ten seeds. Table 9 reports resilience, across-seed standard deviation, and the number of episodes required to reach 95% of each variant's final score.

Table 9. Comparison of encoder-update strategies at 50% damage (mean over eight scenarios and ten seeds).

Encoder during Phase II	Mean R	Std.	Episodes to 95%
Frozen (adopted)	0.686	0.008	1900
Fine-tuned end-to-end	0.679	0.014	2300
Final layer updated	0.683	0.011	2100

The frozen encoder achieves the highest mean resilience (0.686), the lowest across-seed standard deviation (0.008), and the fastest convergence (1900 episodes). Relative to end-to-end fine-tuning, it improves resilience by 0.007, reduces the standard deviation by 42.9%, and requires 17.4% fewer episodes to reach the 95% threshold. Updating only the final layer produces intermediate results on all three measures. These findings support retaining the pretrained functional representation during policy learning: The adopted strategy combines higher recovery performance with greater training stability and faster convergence.

Uncertainty and prediction error. Table 10 stratifies held-out one-step predictions by bounded uncertainty. Each estimate emitted at $t - 1$ is paired with the error in predicting s_t , normalized by the 90th percentile of the warm-up prediction errors.

Table 10. Predictive-uncertainty validation on held-out transitions. The normalized error is the one-step ℓ_2 prediction error divided by the 90th percentile of warm-up prediction errors.

Uncertainty bin	Mean $\widehat{u}_t$	Mean normalized error	Samples
[0.0, 0.2)	0.11	0.08	1,240
[0.2, 0.4)	0.29	0.17	1,118
[0.4, 0.6)	0.49	0.29	986
[0.6, 0.8)	0.69	0.43	734
[0.8, 1.0)	0.86	0.61	402
Spearman correlation (uncertainty, error)			0.73

The mean normalized prediction error increases monotonically across the five uncertainty bins, from 0.08 to 0.61, and the sample-level Spearman correlation reaches 0.73. This result indicates that ensemble uncertainty reliably ranks one-step predictive error. In the component ablation, incorporating this signal into reward aggregation increases mean resilience from 0.679 to 0.686.

5.12. Case study: Recovery under a sensor–influence attack

Table 11 records the first twelve actions of CURE under SIA together with cumulative resilience and the per-step gain. The policy begins with six consecutive replacements concentrated on the compromised sensing and influence layers, each producing a substantial gain, and subsequently interleaves cooperative actions whose immediate gains are smaller but which build redundancy. Replacements targeting the attacked layers are selected before cross-layer repairs, indicating that the action-aligned representations preserve layer identity and substitution compatibility.

Table 11. Recovery log under SIA at 50% damage. R_t is the cumulative resilience through step t , and ΔR_t is the per-step gain. The action (rep, i , j) replaces node i with node j , whereas (coop, i , j) allocates node j 's capability to support node i .

Step	Action	R_t	ΔR_t
0	—	0.429	—
1	(rep, 3I, 7I)	0.451	+0.022
2	(rep, 12S, 19S)	0.473	+0.022
3	(rep, 8I, 15I)	0.494	+0.021
4	(rep, 21S, 28S)	0.514	+0.020
5	(rep, 26I, 31I)	0.533	+0.019
6	(rep, 17S, 24S)	0.550	+0.017
7	(coop, 9I, 13I)	0.558	+0.008
8	(coop, 33S, 41S)	0.565	+0.007
9	(coop, 38I, 44I)	0.570	+0.005
10	(rep, 45I, 52I)	0.577	+0.007
11	(coop, 27S, 36S)	0.579	+0.002
12	(coop, 47I, 54I)	0.580	+0.001

6. Conclusions

CURE combines representation learning on edited functional graphs with uncertainty-aware multiobjective recovery. Phase I learns a shared encoder through path prediction and relation supervision, whereas Phase II integrates the frozen representation, ensemble predictive confidence, analytical deficit weights, and recurrent value decomposition. The framework outperforms six baselines across eight attack scenarios at 50% damage, with average relative improvements of 3.74% and 3.90% over MPGCI and MPD-RL, respectively. Comparisons with the original four baselines further demonstrate its resilience over damage ratios from 30% to 80%. The ablation and encoder-update studies support the adopted design within the evaluated graph benchmark, and the held-out uncertainty–error analysis supports the use of predictive confidence in reward aggregation. Future work will examine repeated attacks, delayed observations, communication failures, explicit physical resource dynamics, and encoder adaptation under larger distribution shifts, together with action-conditional uncertainty and distributional calibration.

Use of generative AI tools declaration

The authors used GPT-5.6 (OpenAI) for English-language polishing throughout the manuscript to improve clarity and readability. All AI-assisted edits were reviewed and revised by the authors. The authors take full responsibility for the content, interpretation, and conclusions of this work.

Author contributions

Kexin Wang and Wanlong Qi contributed equally to this work. Kexin Wang: Conceptualization of this study, methodology, writing—original draft. Wanlong Qi: Methodology, software, visualization,

writing—original draft. Tian Tang: Visualization, supervision. Dingrui Xue: Conceptualization of this study, methodology, software, supervision.

Acknowledgments

This research was supported by the Ph.D. Intelligent Innovation Foundation Project, China (201-CXCY-A01-08-17-03), the Key Research and Development Program of Shaanxi Province (2024GX-YBXM-010), and the National Science Foundation of China (61972302).

Conflict of interest

The authors declare no conflict of interest.

References

1. J. Dahmann, Current landscape of system of systems engineering, *Proc. IEEE Int. Conf. Syst. Syst. Eng.*, 2024, 1–9. <https://doi.org/10.1109/sose62659.2024.10620929>
2. B. Clark, D. Patt, H. Schramm, Mosaic warfare: exploiting artificial intelligence and autonomous systems to implement decision-centric operations, *Center for Strategic and Budgetary Assessments*, 2020. https://csbaonline.org/uploads/documents/Mosaic_Warfare.pdf.
3. Z. Chen, S. Yin, L. Li, W. Cui, D. Hong, Resilience metric and dynamic assessment of unmanned system-of-systems considering cooperative reconfiguration strategies, *IEEE Trans. Reliab.*, **74** (2025), 2942–2954. <https://doi.org/10.1109/TR.2024.3438810>
4. K. Wang, D. Xue, Y. Gou, W. Qi, B. Li, J. Liu, et al., Meta-path-guided causal inference for hierarchical feature alignment and policy optimization in enhancing resilience of UWSoS, *J. Supercomput.*, **81** (2025), 358. <https://doi.org/10.1007/s11227-024-06848-6>
5. M. Teng, C. Gao, Z. Wang, X. Li, A communication-based identification of critical drones in malicious drone swarm networks, *Complex Intell. Syst.*, **10** (2024), 3197–3211. <https://doi.org/10.1007/s40747-023-01316-9>
6. E. M. Hutchins, M. J. Cloppert, R. M. Amin, *Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains*, Leading Issues in Information Warfare Security Research, 2011.
7. Y. Zhong, H. Li, Q. Sun, Z. Huang, Y. Zhang, A kill chain optimization method for improving the resilience of unmanned combat system-of-systems, *Chaos Solitons Fractals*, **181** (2024), 114685. <https://doi.org/10.1016/j.chaos.2024.114685>
8. C. S. Holling, Resilience and stability of ecological systems, *Annu. Rev. Ecol. Syst.*, **4** (1973), 1–23. <https://doi.org/10.1146/annurev.es.04.110173.000245>
9. M. Bruneau, S. E. Chang, R. T. Eguchi, G. C. Lee, T. D. O'Rourke, A. M. Reinhorn, et al., A framework to quantitatively assess and enhance the seismic resilience of communities, *Earthq. Spectra*, **19** (2003), 733–752. <https://doi.org/10.1193/1.1623497>

10. S. Hosseini, K. Barker, J. E. Ramirez-Marquez, A review of definitions and measures of system resilience, *Reliab. Eng. Syst. Saf.*, **145** (2016), 47–61. <https://doi.org/10.1016/j.ress.2015.08.006>
11. D. D. Woods, Four concepts for resilience and the implications for the future of resilience engineering, *Reliab. Eng. Syst. Saf.*, **141** (2015), 5–9. <https://doi.org/10.1016/j.ress.2015.03.018>
12. H. T. Tran, M. Balchanos, J. C. Domercant, D. N. Mavris, A framework for the quantitative assessment of performance-based system resilience, *Reliab. Eng. Syst. Saf.*, **158** (2017), 73–84. <https://doi.org/10.1016/j.ress.2016.10.014>
13. C. Nan, G. Sansavini, A quantitative method for assessing resilience of interdependent infrastructures, *Reliab. Eng. Syst. Saf.*, **157** (2017), 35–53. <https://doi.org/10.1016/j.ress.2016.08.013>
14. P. Uday, K. Marais, Designing resilient systems-of-systems: A survey of metrics, methods, and challenges, *Syst. Eng.*, **18** (2015), 491–510. <https://doi.org/10.1002/sys.21325>
15. A. Peiravi, M. Nourelfath, M. K. Zanjani, Universal redundancy strategy for system reliability optimization, *Reliab. Eng. Syst. Saf.*, **225** (2022), 108576. <https://doi.org/10.1016/j.ress.2022.108576>
16. G. Levitin, L. Xing, Y. Dai, Optimizing partial component activation policy in multi-attempt missions, *Reliab. Eng. Syst. Saf.*, **235** (2023), 109251. <https://doi.org/10.1016/j.ress.2023.109251>
17. Q. Sun, H. Li, Y. Wang, Y. Zhang, Multi-swarm-based cooperative reconfiguration model for resilient unmanned weapon system-of-systems, *Reliab. Eng. Syst. Saf.*, **222** (2022), 108426. <https://doi.org/10.1016/j.ress.2022.108426>
18. X. Wu, D. Sheldon, S. Zilberstein, Optimizing resilience in large scale networks, *Proc. AAAI Conf. Artif. Intell.*, 2016. <https://doi.org/10.1609/aaai.v30i1.9911>
19. Q. Sun, H. Li, Y. Zhong, K. Ren and Y. Zhang, Deep reinforcement learning-based resilience enhancement strategy of unmanned weapon system-of-systems under inevitable interferences, *Reliab. Eng. Syst. Saf.*, **242** (2024), 109749. <https://doi.org/10.1016/j.ress.2023.109749>
20. K. Wang, Y. Gou, D. Xue, J. Liu, W. Qi, G. Hou, et al., Resilience augmentation in unmanned weapon systems via multi-layer attention graph convolutional neural networks, *Comput. Mater. Contin.*, **80** (2024), 2941–2962. <https://doi.org/10.32604/cmc.2024.052893>
21. D. Xue, Y. Lin, S. Wei, Z. Zhang, W. Qi, J. Liu, et al., Leveraging hierarchical temporal importance sampling and adaptive noise modulation to enhance resilience in multi-agent task execution systems, *Neurocomputing*, **637** (2025), 130134. <https://doi.org/10.1016/j.neucom.2025.130134>
22. Y. Peng, J. Wu, T. Duan, Y. Liu, Z. Zhang, J. Zhang, Prioritized recovery strategy for robust UAV swarm communication via graph reinforcement learning, *IEEE Internet Things J.*, **12** (2025), 23891–23904. <https://doi.org/10.1109/jiot.2025.3553919>
23. Y. Gou, S. Wei, K. Xu, J. Liu, K. Li, B. Li, et al., Hierarchical reinforcement learning with kill chain-informed multi-objective optimization to enhance resilience in autonomous unmanned swarm, *Neural Netw.*, **195** (2026), 108255. <https://doi.org/10.1016/j.neunet.2025.108255>

24. K. Li, Y. Lin, X. Su, A. Liu, J. Liu, W. Qi, et al., Unified multi-agent recovery framework via multi-scale diffusion and dependency-aware hierarchical PPO for resilience enhancement, *J. Big Data*, **12** (2025), 226. <https://doi.org/10.1186/s40537-025-01285-5>
25. Z. Han, S. Wei, D. Xue, Y. Lin, K. Li, J. Liu, et al., MPD-RL: Meta-path-driven reinforcement learning for enhancing resilience in unmanned weapon system-of-systems, *J. Supercomput.*, **81** (2025), 951. <https://doi.org/10.1007/s11227-025-07435-z>
26. X. Han, H. Wang, Q. Jia, Y. Gou, B. Li, J. Liu, et al., Meta-path guided policy distillation for resilient coordination in autonomous unmanned swarm, *PLoS One*, **20** (2025), e0339675. <https://doi.org/10.1371/journal.pone.0339675>
27. S. Pitis, E. Creager, A. Garg, Counterfactual data augmentation using locally factored dynamics, *Adv. Neural Inf. Process. Syst.*, **33** (2020), 3976–3990. https://proceedings.neurips.cc/paper_files/paper/2020/hash/294e09f267683c7ddc6cc5134a7e68a8-Abstract.html
28. J. Mitrovic, B. McWilliams, J. Walker, L. Buesing, C. Blundell, Representation learning via invariant causal mechanisms, *Int. Conf. Learn. Represent.*, 2021. <https://openreview.net/forum?id=9p2ekP904Rs>
29. C. F. Hayes, R. Rădulescu, E. Bargiacchi, J. Källström, M. Macfarlane, M. Reymond, et al., A practical guide to multi-objective reinforcement learning and planning, *Auton. Agents Multi-Agent Syst.*, **36** (2022), 26. <https://doi.org/10.1007/s10458-022-09552-y>
30. K. Chua, R. Calandra, R. McAllister, S. Levine, Deep reinforcement learning in a handful of trials using probabilistic dynamics models, *Adv. Neural Inf. Process. Syst.*, 2018. https://proceedings.neurips.cc/paper_files/paper/2018/hash/3de568f8597b94bda53149c7d7f5958c-Abstract.html
31. T. Kurutach, I. Clavera, Y. Duan, A. Tamar, P. Abbeel, Model-ensemble trust-region policy optimization, *Int. Conf. Learn. Represent.*, 2018. <https://openreview.net/forum?id=SJJinbWRZ>
32. M. Janner, J. Fu, M. Zhang, S. Levine, When to trust your model: Model-based policy optimization, *Adv. Neural Inf. Process. Syst.*, 2019. Available from: https://proceedings.neurips.cc/paper_files/paper/2019/hash/5faf461eff3099671ad63c6f3f094f7f-Abstract.html.
33. D. Hafner, J. Pasukonis, J. Ba, T. Lillicrap, Mastering diverse control tasks through world models, *Nature*, **640** (2025), 647–653. <https://doi.org/10.1038/s41586-025-08744-2>
34. A. Kendall, Y. Gal, What uncertainties do we need in Bayesian deep learning for computer vision? *Adv. Neural Inf. Process. Syst.*, 2017. Available from: https://proceedings.neurips.cc/paper_files/paper/2017/hash/2650d6089a6d640c5e85b2b88265dc2b-Abstract.html.
35. N. Hansen, H. Su, X. Wang, TD-MPC2: Scalable, robust world models for continuous control, *Int. Conf. Learn. Represent.*, 2024. <https://openreview.net/forum?id=Oxh5CstDJU>
36. J. Ding, Y. Zhang, Y. Shang, Y. Zhang, Z. Zong, J. Feng, et al., Understanding world or predicting future? A comprehensive survey of world models, *ACM Comput. Surv.*, **58** (2025), 1–38. <https://doi.org/10.1145/3746449>
37. G. N. Iyengar, Robust dynamic programming, *Math. Oper. Res.*, **30** (2005), 257–280. <https://doi.org/10.1287/moor.1040.0129>

38. Y. Chow, A. Tamar, S. Mannor, M. Pavone, Risk-sensitive and robust decision-making: a CVaR optimization approach, *Adv. Neural Inf. Process. Syst.*, 2015. Available from: https://proceedings.neurips.cc/paper_files/paper/2015/hash/64223ccf70bbb65a3a4aceac37e21016-Abstract.html.
39. E. Derman, D. J. Mankowitz, T. A. Mann, S. Mannor, Soft-robust actor-critic policy-gradient, *Conf. Uncertainty Artif. Intell.*, 2018. <https://www.auai.org/uai2018/proceedings/papers/70.pdf>.
40. K. Zhang, T. Sun, Y. Tao, S. Genc, S. Mallya, T. Başar, Robust multi-agent reinforcement learning with model uncertainty, *Adv. Neural Inf. Process. Syst.*, 2020. Available from: https://proceedings.neurips.cc/paper_files/paper/2020/hash/774412967f19ea61d448977ad9749078-Abstract.html.
41. A. Bukharin, Y. Li, Y. Yu, Q. Zhang, Z. Chen, S. Zuo, et al., Robust multi-agent reinforcement learning via adversarial regularization: theoretical foundation and stable algorithms, *Adv. neural inf. process. syst.*, 2023. <https://doi.org/10.52202/075280-2979>
42. X. Zhao, J. Liu, F. Wu, X. Zhang, G. Wang, Uncertainty modified policy for multi-agent reinforcement learning, *Appl. Intell.*, **54** (2024), 12020–12034. <https://doi.org/10.1007/s10489-024-05811-5>
43. L. Shi, E. Mazumdar, Y. Chi, A. Wierman, Sample-efficient robust multi-agent reinforcement learning in the face of environmental uncertainty, *Proc. Mach. Learn. Res.*, 2024. <https://proceedings.mlr.press/v235/shi24d.html>
44. L. Shi, J. Gai, E. Mazumdar, Y. Chi, A. Wierman, Breaking the curse of multiagency in robust multi-agent reinforcement learning, *Proc. Mach. Learn. Res.*, **267** (2025), 54904–54918. <https://proceedings.mlr.press/v267/shi25c.html>
45. C. Wu, H. Deng, H. Wu, C. Tu, Enhancing resilience of unmanned autonomous swarms through game theory-based cooperative reconfiguration, *Reliab. Eng. Syst. Saf.*, **260** (2025), 110951. <https://doi.org/10.1016/j.res.2025.110951>
46. Y. Wang, J. Cui, C. Zhai, ADHGR: An optimized adaptive disintegration strategy targeting heterogeneous UAV swarms based on graph reinforcement learning, *Reliab. Eng. Syst. Saf.*, **265** (2025), 111453. <https://doi.org/10.1016/j.res.2025.111453>
47. T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, *Int. Conf. Learn. Represent.*, 2017. <https://openreview.net/forum?id=SJU4ayYgl>
48. P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, *Int. Conf. Learn. Represent.*, 2018. <https://openreview.net/forum?id=rJXMpikCZ>
49. K. Xu, W. Hu, J. Leskovec, S. Jegelka, How powerful are graph neural networks?, *Int. Conf. Learn. Represent.*, 2019. <https://openreview.net/forum?id=ryGs6iA5Km>
50. X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, Heterogeneous graph attention network, *Proc. Int. World Wide Web Conf.*, 2019. <https://doi.org/10.1145/3308558.3313562>
51. J. Yu, Q. Ge, X. Li, A. Zhou, Heterogeneous graph contrastive learning with meta-path contexts and adaptively weighted negative samples, *IEEE Trans. Knowl. Data Eng.*, **36** (2024), 5181–5193. <https://doi.org/10.1109/tkde.2024.3377431>

52. B. Khemani, S. Patil, K. Kotecha, S. Tanwar, A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions, *J. Big Data*, **11** (2024), 18. <https://doi.org/10.1186/s40537-023-00876-4>
53. J. Pearl, *Causality: Models, Reasoning, and inference*, Cambridge University Press, 2009. <https://doi.org/10.1017/CBO9780511803161>
54. J. Pearl, Causal inference in statistics: An overview, *Statist. Surv.*, **3** (2009), 96–146. <https://doi.org/10.1214/09-SS057>
55. B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, et al., Toward causal representation learning, *Proc. IEEE*, **109** (2021), 612–634. <https://doi.org/10.1109/jproc.2021.3058954>
56. J. Brehmer, P. de Haan, P. Lippe, T. Cohen, Weakly supervised causal representation learning, *Adv. Neural. Inf. Process Syst.*, **35** (2022), 38319–38331. <https://doi.org/10.52202/068431-2776>
57. K. Ahuja, D. Mahajan, Y. Wang, Y. Bengio, Interventional causal representation learning, *Int. Conf. Mach. Learn.*, **22** (2023), 372–407. <https://proceedings.mlr.press/v202/ahuja23a.html>
58. A. van den Oord, Y. Li, O. Vinyals, *Representation learning with contrastive predictive coding*, 2018. <https://doi.org/10.48550/arXiv.1807.03748>
59. J. Demšar, Statistical comparisons of classifiers over multiple data sets, *J. Mach. Learn. Res.*, **7** (2006), 1–30. <https://jmlr.org/papers/v7/demsar06a.html>
60. P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, D. Meger, Deep reinforcement learning that matters, *Proc. AAAI Conf. Artif. Intell.*, 2018. <https://doi.org/10.1609/aaai.v32i1.11694>



AIMS Press

©2026 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0>)