Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 23, No. 2, 2026

Safe-ARCS: Risk-Budget Shielding for Hierarchical Reinforcement Learning in Automated Cybersecurity Incident Response

Asha Singh [1, *], Benjamin Zhang [2], Caroline Young [2]

[1] Department of Computer Science and Engineering, Michigan State University, MI, USA
[2] School of Computer Science, College of Computing, Georgia Institute of Technology, GA, USA

Corresponding author: Asha Singh (Email: ashasingh1988@outlook.com)

Abstract: Automated incident response should contain malicious activity without imposing unacceptable disruption on legitimate services. We study this conflict in a reproducible six-step stochastic simulator with noisy alerts, five latent incident classes, immediate containment actions, and persistent strategic defenses. Safe-ARCS builds on two mechanisms of the ARCS framework: independently learned immediate and strategic decision streams with exponentially decayed alert urgency, and temporal-difference prioritized replay. Its new budget shield admits an action only when a verified upper envelope on induced operational cost fits the remaining episode budget. For the specified bounded-cost simulator, this filtering yields a simple zero-overspend invariant. Across five independent training seeds and 59,500 held-out evaluation episodes, Safe-ARCS achieves 86.4% attack resolution and zero budget exceedances in nominal tests, versus 88.2% resolution and 0.5% exceedances for its unshielded dual-stream counterpart. Under shifted alert distributions and weakened mitigation probabilities, the corresponding results are 77.3% versus 80.4% resolution, with zero versus 1.9% exceedances. Ablations expose component effects. A separate 16,200-episode cost-misspecification study shows that under 50% inflation of modeled intervention costs, the default shield exceeds the budget in 3.7% of shifted-test episodes, whereas a calibrated 1.5-fold reserve eliminates these violations at a 2.1-point resolution cost. The findings define an auditable, conditional safety–effectiveness trade-off rather than an estimate of operational-network performance.

Keywords: Cyber incident response; Hierarchical reinforcement learning; Action shielding; Constrained decision making; Reproducible simulation.

1. Introduction

Network monitoring and intrusion detection research has advanced substantially, but a detection score does not specify which intervention should be executed or how its operational consequences should be controlled. Survey evidence shows the diversity and evaluation limitations of intrusion detection algorithms [1], while deep intrusion detection work emphasizes data-dependent recognition performance [2]. Sequential defense systems must instead reason about intervention timing, persistent mitigation, and the possibility of isolating benign services. Surveys of reinforcement learning for cybersecurity identify sequential optimization as a promising response paradigm, together with open problems in realistic evaluation and operational constraints [3], [4].

Ren et al. proposed ARCS, with distinct immediate and strategic response streams, temporally aggregated security-event information, and prioritized experience replay [5]. Its dual-stream design separates rapid containment from persistent defense planning and provides a strong mechanistic basis for automated response. A different question arises when each emergency action consumes a finite service-disruption budget: can hierarchical decision making honor an explicit hard limit while remaining effective on uncertain attack sequences? A weighted average of effectiveness and stability, by itself, need not preclude budget overruns.

This study contributes: (i) a two-timescale tabular realization of the immediate/strategic ARCS decision modules that retains their exponentially decayed urgency and prioritized replay; (ii) an action-admissibility shield that enforces a cumulative collateral-cost invariant under a stated model envelope; and (iii) seed-wise, paired simulation experiments with fixed rules, flat Q-learning, an unshielded dual-stream policy, three component ablations, a distribution-shift test, and a controlled cost-envelope misspecification experiment. To keep counterfactual action effects observable and auditable, the experiment uses an explicitly synthetic Markov simulator rather than repurposing intrusion labels as ground-truth response outcomes.

2. Related work

2.1. Hierarchical control for cybersecurity

The options framework formalized temporally extended decisions within the semi-Markov setting [6], and MAXQ demonstrated how problem structure can decompose value learning [7]. In security, hierarchical reinforcement learning has been used to plan automated penetration tests of large networks [8]. Cyber-attack simulation using deep reinforcement learning addresses adversarial strategy development [9], while Deep VULMAN optimizes vulnerability-management decisions [10]. These applications highlight sequential planning but do not by themselves enforce an operational intervention budget in the response simulator studied here.

Adaptive anomaly detection for industrial-control environments [11] and broader surveys of cyber machine learning [12] further demonstrate why detection and automated intervention require different assessment criteria. We therefore use attack resolution and cumulative collateral

cost as separate outcomes, instead of presenting classification accuracy as the response objective.

2.2. Constraints, shielding, and learning

Safe reinforcement learning research distinguishes expected-return adjustments from explicit restrictions on exploration or execution [13]. Constrained policy optimization optimizes return subject to resource-type constraints [14]. Action shields have also been synthesized from temporal-logic specifications to prevent disallowed behavior [15], whereas risk-constrained policies may optimize tail-risk or probability constraints [16]. Reward-constrained policy optimization introduces separate timescales for multiplier and policy updates [17], and constrained Markov decision processes provide a general mathematical language for resource-limited control [18]. Our guarantee is narrower and more direct: a conservative bound on a simulator-defined scalar intervention cost enables per-action rejection, independent of how the Q values are learned.

Prioritized replay, proposed by Schaul et al., samples transitions according to temporal-difference errors [19]. Deep Q-networks popularized replay-based value approximation [20], and Double Q-learning reduced value overestimation in neural Q-learning [21]. We use tabular Q-functions to isolate the effect of dual timescales and the shield without attributing deep-network gains to the experiment. Work on advanced cyber threat detection and protection [22], CVaR optimization [23], and the publicly documented UNSW-NB15 benchmark [24] informs broader evaluation practice, but no external incident records are used as action-outcome labels here. PPO [25] and neural cybersecurity approaches [26] are potential scaling directions, not benchmark implementations in this study.

3. Problem formulation

3.1. Synthetic episodic response model

Each simulator episode lasts at most $T = 6$ discrete decision steps. A hidden class z is sampled from {benign, reconnaissance, malware, exfiltration, ransomware} with probabilities (0.30, 0.22, 0.18, 0.16, 0.14). The episode also samples asset criticality $c \in \{0,1\}$ with $P(c = 1) = 0.42$ and initial attack intensity uniformly from [0.55, 1.0]. The controller does not observe z. Instead it receives a noisy suspicion score p, incident-stage indicator h, asset criticality, remaining operational budget b, exponentially aggregated severity u, and the active strategic action g. When an attack remains active, its stage may increase; successful containment terminates its damaging process, although subsequent steps continue to account for intervention choices.

There are four immediate actions: observe ($a = 0$), inspect (1), rate-limit (2), and isolate (3). Three persistent options are available at macro-decision times $t \in \{0,3\}$: no strategic change ($g = 0$), patch (1), and segment (2). Patching increases containment probabilities for malware and ransomware, whereas segmentation favors reconnaissance and exfiltration. The exact probabilities and stochastic distributions appear in the code. No task-specific real-world effect sizes are implied by these coefficients.

3.2. Reward, cost, and safety objective

The implemented environmental collateral cost is $c_t = (c_a + 1[t \bmod 3 = 0]c_g)(1 + 0.55c)m_z$, where base action costs are (0, 0.07, 0.32, 0.90), strategic costs are (0, 0.17, 0.22), and the multiplier m_z is 1.6 for benign traffic and 0.4 for attacks. The maximum-cost envelope follows immediately from $0 \le m_z \le 1.6$. Costs represent dimensionless modeled disruption rather than measured labor time or service outages.

$$C_T = \Sigma(t = 0,\ldots,T-1)\, c_t, \qquad C_T \le B = 2.6. \tag{1}$$

An episode has a budget violation when $C_T > B$. Simulated return combines containment gain G_t, continuing damage D_t and collateral cost c_t. The urgency-dependent coefficients vary the relative priorities of defense and service stability while preserving the same reward calculation for every evaluated method:

$$r_t = (0.78 + 0.22\beta_t)G_t - (1.20 + 0.20\beta_t)D_t - (1.60 - 0.45\beta_t)c_t. \tag{2}$$

A containment event contributes 6.5, or 7.5 for critical assets, to G_t; monitoring or inspecting a benign event gives 0.2. Successful response, response steps, false benign intervention, collateral expenditure, cumulative damage, and return are logged separately. We treat the budget as a hard safety quantity, not merely an additional penalty in (2).

4. Safe-arcs design

4.1. Two-timescale policy with temporal urgency

The controller adopts the two immediate/strategic decision streams and the decayed-urgency aggregation of ARCS [5], adapting them to auditable tabular state abstractions in which the urgency enters as a discretized state variable. Let e_{t-i}, $i = 0, \ldots, k-1$, be the k latest normalized severity observations and let $\lambda = 0.1$. As in ARCS, the recency-weighted severity state is the normalized exponentially weighted average:

$$u_t = [\Sigma(i=0,\ldots,k-1)\, e_{(t-i)} \exp(-\lambda i)] \,/\, [\Sigma(i=0,\ldots,k-1) \exp(-\lambda i)]. \tag{3}$$

The state is discretized into four suspicion bins, three stage values, two criticality values, three remaining-budget bins, and three urgency bins. The low-level table additionally conditions on the active strategy. Strategic decisions are updated only at steps 0 and 3; immediate actions are updated at every step. Let Q_L and Q_H denote the low-level and high-level action-value tables. Macro actions are learned using the discounted reward collected during each three-step option interval:

$$Q_H(s,g) \leftarrow Q_H(s,g) + \eta[R_{(t:t+K)} + \gamma^K \max_{g'} Q_H(s',g') - Q_H(s,g)]. \tag{4}$$

We use $\gamma = 0.94$, $\eta = 0.22$ for macro updates, and a low-level online step size of 0.20. The macro return $R_{(t:t+K)}$ is $\Sigma(j=0,\ldots,K-1)\gamma^j r_{(t+j)}$. This explicitly separates strategic persistence from immediate interventions while allowing each stream to learn from subsequent environment feedback.

4.2. Transition replay

In addition to online temporal-difference updates, a replay buffer stores up to 6,000 recent low-level transitions. At every twelfth environment step, 24 transitions are resampled with proportional probabilities $P(i) \propto (|\delta_i| + 0.001)^{0.6}$; inverse-probability weights with exponent 0.4 are normalized within the batch. A uniform-replay ablation fixes the priority distribution to uniform. This is the TD-error prioritized replay of ARCS [5], which follows [19], applied here to tabular low-level updates with the same priority exponent of 0.6.

The decision hierarchy is deliberately lightweight. Macro-

state and low-state tables share suspicion, stage, criticality, remaining budget and urgency, while only the immediate table includes the selected persistent strategy. This prevents the low-level intervention policy from silently treating patching and segmentation as the same action. Greedy execution and training-time exploration are identical except for epsilon-greedy exploration, which decreases from approximately 0.37 toward its 0.045 floor over the training schedule. A fixed environment horizon ensures unresolved threats receive no artificially extended mitigation opportunities at evaluation time.

The low-level online update is $Q_L(s,a) \leftarrow Q_L(s,a) + 0.20[r + 0.94 \max Q_L(s',a') - Q_L(s,a)]$, with the terminal bootstrap term omitted at step six. Replay adds weighted low-level updates of size 0.06 per sampled transition. The high-level learner stores complete option returns, avoiding an incorrect one-step bootstrapping target for three-step strategic commitments. The discrete abstraction is not intended to replace recurrent neural representations at large scale; rather, it provides an inspectable implementation of the shared hierarchical architecture.

4.3. Bounded-cost action shield

At each macro decision the shield first rejects strategic options whose largest possible commitment cost exceeds the remaining budget. It then filters immediate actions using a worst-case upper envelope $U(a,g,c) = 1.6(1 + 0.55c)[c_a + I_{macro} c_g]$. Among admissible immediate actions, the low-level policy selects the action with maximal Q_L; exploration is also restricted to this same set. Observe with no new strategy has zero collateral cost and is always available.

$$A_t = \{a : U(a,g,c) \le B - \Sigma(\tau<t)c_\tau\}; \qquad a^*_t = \operatorname{argmax}(a \in A_t) Q_L(s_t,a). \tag{5}$$

Proposition 1 (bounded-cost feasibility). Suppose the simulator starts at $C_0 = 0$, its actual per-step cost satisfies $0 \le c_t \le U(a_t,g_t,c)$, and actions are selected from (5), including the macro-strategy feasibility check. Then $C_t \le B$ for all $t \le T$. Proof: the result holds at $t = 0$. If $C_t \le B$, admissibility implies $c_t \le U \le B - C_t$; hence $C_{(t+1)} = C_t + c_t \le B$. Induction completes the argument. The claim depends on correctness of the stated cost envelope and does not imply feasibility under unmodeled deployment costs.

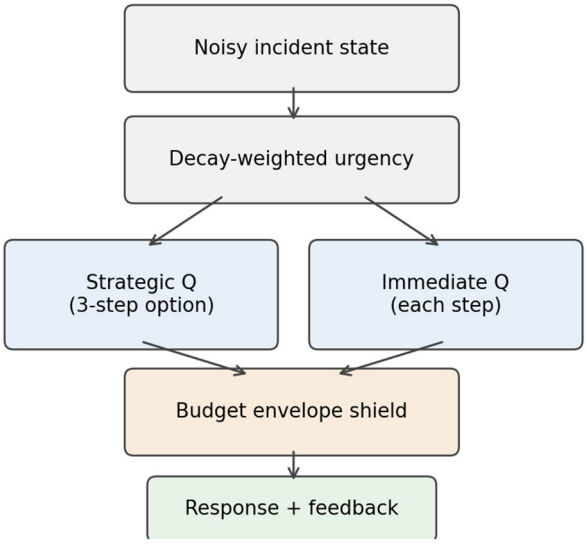


Fig. 1 Executed dual-timescale decision path and cost-envelope filter.

5. Experimental protocol

5.1. Conditions, data generation, and reproducibility

All samples are simulator-generated, not downloaded intrusion incidents. The nominal suspicion observation uses Beta(2,5) for benign events and Beta(5,2) for malicious events; corresponding severities use Beta(1.9,4.1) and Beta(4,2). Upon containment, suspicion and severity are reduced. Each method trains for 1,700 episodes independently under five seeds (0–4). Each trained model is evaluated without exploration on 850 new episodes for each of two conditions per seed. Seven methods × two evaluation conditions × five seeds × 850 episodes yield 59,500 recorded evaluation episodes. Training data are regenerated from seeded pseudorandom streams, and evaluation episode identities are paired across policies.

The shifted condition increases the benign suspicion value by 0.17, decreases malicious suspicion by 0.12, modifies the class mixture to (0.34, 0.14, 0.16, 0.20, 0.16), and reduces nonzero mitigation probabilities by 15%. It is a designed stress test rather than an empirically estimated domain shift. The Python implementation, environment transitions, configurations, random-seed formulas, seed-specific results, and aggregate computations accompany the manuscript.

5.2. Comparators and metrics

The methods are: Rules (fixed suspicion thresholds with strategy selection); Flat-Q (no strategic stream); Dual-Q (unshielded hierarchical adaptation); Safe-ARCS (shielded hierarchical policy); Safe-flat (safety without strategic learning); No urgency (safe hierarchical learner with constant urgency feature); and No PER (safe hierarchical learner with uniform replay). The same simulator, training episodes, evaluation episodes, and reward definition are used for all learned methods. The Safe-flat, No urgency and No PER variants therefore measure the contribution of each ARCS component inside the shielded controller.

Attack resolution is the fraction of malicious episodes neutralized by the final step. Budget exceedance is the fraction of all episodes whose realized cumulative cost exceeds 2.6. False intervention is the fraction of benign episodes in which rate limiting or isolation occurs at least once. Containment steps average the first successful containment step, assigning 7 to unresolved attacks; simulated cumulative damage and return average across all episodes. These metrics distinguish protection, legitimate-traffic disruption, and the agent objective.

Table I. Simulator and training configuration

Quantity	Value
Horizon; option interval	6 steps; 3 steps
Actions; strategic modes	4 immediate; 3 persistent
Per-episode cost budget	2.6 normalized units
Training; test per seed	1,700; 850 × 2 conditions
Random seeds	0, 1, 2, 3, 4
Q discount; TD priority	0.94; exponent 0.6
Replay buffer; mini-batch	6,000; 24
Operational evidence	synthetic stochastic simulator

The model assumes that a defender can obtain a noisy attack suspicion estimate and an event severity estimate at each step. In the present simulator those are generated directly from specified Beta distributions; they are not fitted probability scores from an intrusion classifier. This separation matters because classifying the hidden attack label using its known simulator generation process would give an optimistic upper bound. Similarly, the class label is available only to the environment transition and metric computation. Every policy action is chosen from the same observable tuple, with no access to the hidden threat family, the future attack progression or a random-number generator state.

6. Results

6.1. Overall performance

Tables II and III report means of the five independent per-seed estimates. The main safety outcome is consistent across both environmental conditions: Safe-ARCS produced no budget exceedances in 4,250 nominal and 4,250 shifted evaluation episodes. The unshielded Dual-Q adaptation exceeded the budget in 0.52% and 1.88% of these episodes, respectively. Because the shield is model-based, its zero exceedance rate is guaranteed by Proposition 1 under the prescribed maximum-cost multiplier; the experimental check confirms the implementation follows that model.

Table II. Nominal test results (5-seed means)

Method	Res. %	Exceed %	False %	Return
Rules	86.2	0.3	19.9	1.63
Flat-Q	80.9	0.7	10.0	0.98
Dual-Q	88.2	0.5	11.8	1.77
Safe-ARCS	86.4	0.0	19.2	1.68
Safe-flat	76.7	0.0	13.8	0.73
No urgency	86.7	0.0	20.0	1.63
No PER	85.5	0.0	13.8	1.56

Safe-ARCS resolved 86.4% of nominal attacks, compared with 88.2% for Dual-Q and 80.9% for Flat-Q. This reflects the cost of rejecting some high-impact interventions, as well as the benefit of strategic actions relative to flat learning. The nominal paired seed difference (Safe-ARCS minus Dual-Q) in resolution was −1.83 percentage points with a two-sided 95% Student-t interval of [−4.16, 0.50] points. The nominal decrease in budget exceedance was 0.52 points (paired interval [0.12, 0.91] for Dual-Q minus Safe-ARCS). Thus, the budget-safety effect is clear under the modeled envelope; a superiority claim for unrestricted response effectiveness would be unwarranted.

Table III. Shifted test results (5-seed means)

Method	Res. %	Exceed %	False %	Return
Rules	62.4	3.2	69.9	-0.95
Flat-Q	72.2	1.3	38.9	-0.09
Dual-Q	80.4	1.9	35.6	0.64
Safe-ARCS	77.3	0.0	43.8	0.56
Safe-flat	69.3	0.0	39.3	-0.22
No urgency	77.9	0.0	59.7	0.35
No PER	79.4	0.0	36.4	0.61

With the alert-distribution shift, Safe-ARCS resolved 77.3% of malicious incidents compared with 80.4% for Dual-Q; both performed above the fixed rule policy at 62.4%. The shield again preserved zero exceedances, whereas Dual-Q incurred 1.9% exceedances. The paired Safe-ARCS–Dual-Q resolution difference was −3.10 percentage points (95% interval [−5.26, −0.95]); the violation difference was −1.88 points (95% interval [−2.60, −1.17]). Operationally, the shift creates a measurable false-intervention burden: 43.8% of benign episodes for Safe-ARCS compared with 35.6% for Dual-Q. The hard budget bounds cumulative severity but does not guarantee that every individual response decision is appropriate.

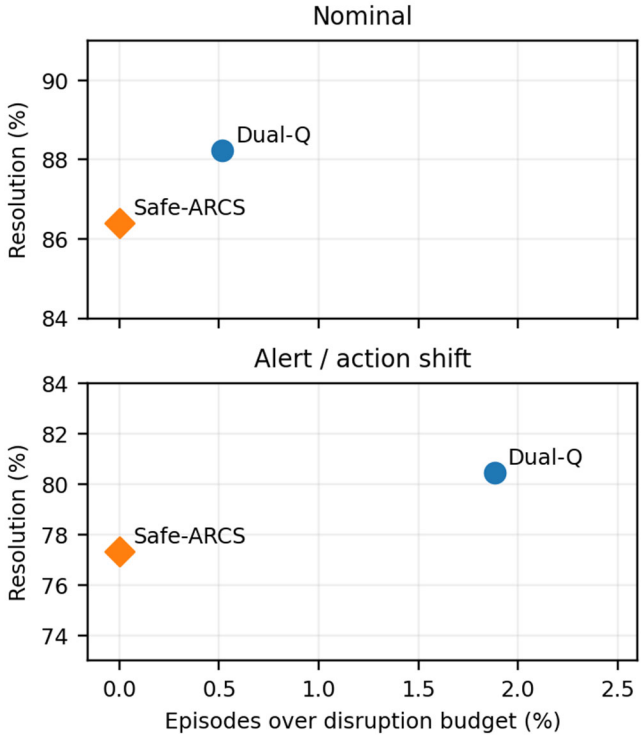


Fig. 2 Attack resolution versus modeled budget exceedance. Points are 5-seed means.

6.2. Ablations and temporal response

The hierarchy ablation is substantial and confirms the value of the ARCS dual-stream design [5]. Removing strategic learning while retaining the shield lowered nominal resolution from 86.4% to 76.7% and shifted resolution from 77.3% to 69.3%. Mean capped containment steps also increased from 3.01 to 3.56 under nominal conditions and from 3.55 to 4.04 under shift. These results support the utility of persistent patching or segmentation choices in the explicitly constructed attack transition model. They do not establish transferability to enterprise workflows without further calibration.

Removing the urgency feature yields 86.7% nominal resolution, close to 86.4%, but raises mean false-intervention rates in the shifted condition from 43.8% to 59.7%; simulated shifted return drops from 0.56 to 0.35. This pattern is consistent with urgency state conditioning assisting with alert ambiguity. Replacing prioritized replay with uniform replay yields 85.5% nominal and 79.4% shifted resolution, compared with 86.4% and 77.3% for Safe-ARCS: replay prioritization is not uniformly beneficial in this small tabular regime. Reporting this negative ablation avoids attributing the

safety result to a component that is not essential to its invariant.

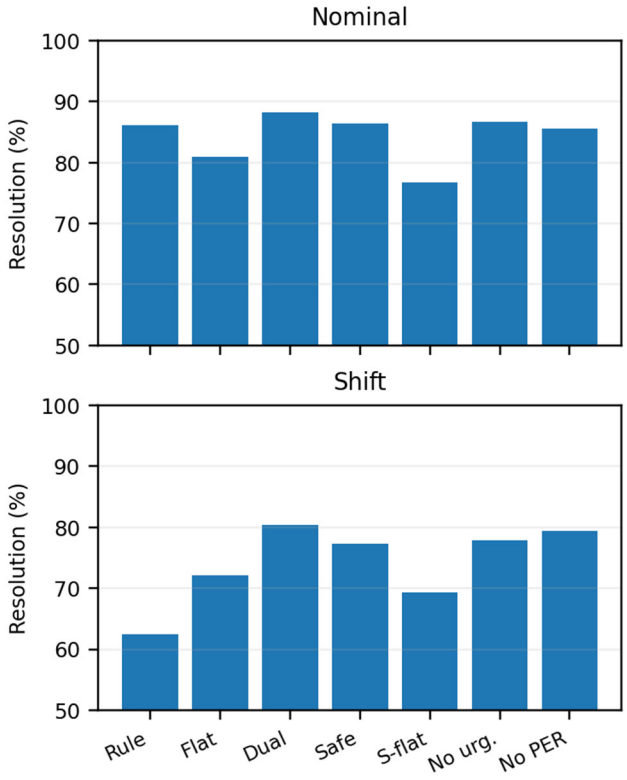


Fig. 3 Resolution across all seven policies and both test conditions.

6.3. Attack-family heterogeneity

To examine whether the aggregate results are driven by one highly represented threat, Table IV stratifies the shifted-condition results using the simulator's unobserved class (available to the evaluator, not the agent). The segmentation option was designed to support reconnaissance and exfiltration, whereas patching supports malware and ransomware. The stratified outcomes therefore act as an internal validation of the option-dependent transition logic. They cannot validate external attack taxonomies or real security product effectiveness.

Table IV. Shifted attack-class resolution (%)

Simulated attack	Rules	Dual-Q	Safe-ARCS
Reconnaissance	80.3	88.5	88.2
Malware	59.5	83.0	78.3
Exfiltration	60.2	76.2	72.5
Ransomware	52.2	76.5	73.2

Reconnaissance is the easiest shifted attack for both learned hierarchies (88.5% for Dual-Q; 88.2% for Safe-ARCS). Exfiltration is more challenging (76.2% and 72.5%). The highest shield-related shifted resolution gap is observed for malware (4.7 percentage points), consistent with isolating a compromised critical service sometimes exceeding the conservative remaining budget. These are conditional empirical averages over simulated episodes, not independent real-world attack trials; random variation and category frequencies should be considered before inferring fine-grained ordering.

A second diagnostic is the realized residual damage rather than successful resolution alone. Under nominal conditions the mean simulated damage is 1.17 for Dual-Q and 1.24 for Safe-ARCS; under shift it is 1.43 and 1.53, respectively. The shield also lowers mean collateral expenditure from 0.70 to 0.61 nominally and from 0.90 to 0.75 under shift. Hence the lower episode budget violation rate is accompanied by lower average cost, rather than achieved solely by reclassifying an outcome metric. The same results expose the residual coverage penalty, preserving a balanced assessment of the control trade-off.

6.4. Transfer to alternate operational budgets

An additional sensitivity experiment tests whether the bounded-cost result depends on the selected budget B = 2.6. For this diagnostic, the same five trained policies from each algorithm are recreated at B = 2.6 and then evaluated without parameter updates at new cost allowances $B \in \{1.4, 2.6, 3.8\}$. Only the evaluation environment budget and associated shield admissibility computation are changed. The same 850 episode seeds per condition and training seed are reused at each allowance. This is budget-transfer testing rather than a new training comparison; a policy trained specifically for each alternative allowance might produce different resolution rates.

Table V. Budget transfer without retraining

Budget	Policy	Shift Res. %	Shift Exceed %
1.4	Dual-Q	51.5	4.4
1.4	Safe-ARCS	44.5	0.0
2.6	Dual-Q	80.4	1.9
2.6	Safe-ARCS	77.3	0.0
3.8	Dual-Q	83.0	0.5
3.8	Safe-ARCS	81.2	0.0

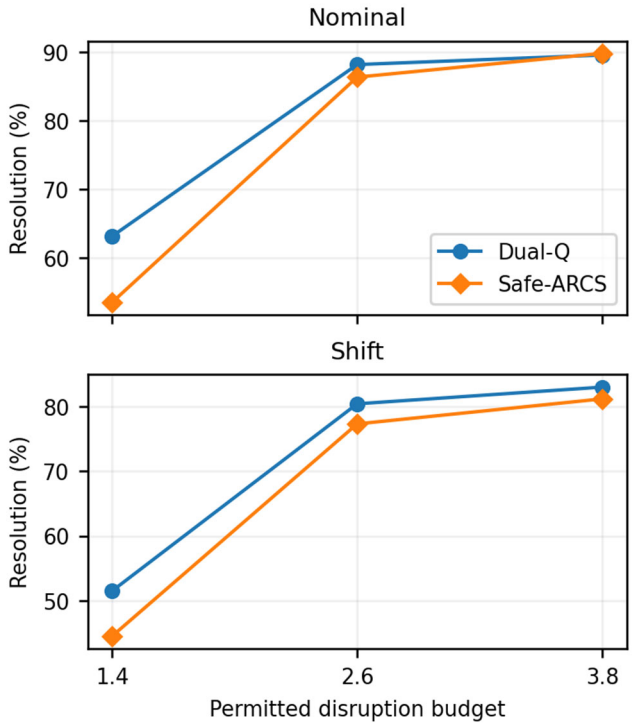


Fig. 4 Resolution after changing the simulated safety budget without retraining either policy.

At the stringent B = 1.4 budget, shifted-condition resolution is 44.5% for Safe-ARCS and 51.5% for Dual-Q; the unshielded policy incurs a 4.4% violation rate and the

shielded policy incurs none. At the original B = 2.6, shifted resolution is 77.3% versus 80.4%, with 0% versus 1.9% violations. When B increases to 3.8, Safe-ARCS recovers to 81.2% shifted resolution while Dual-Q reaches 83.0%; the latter retains a 0.5% violation rate. The nominal resolution rates likewise rise from 53.6% to 89.9% for Safe-ARCS between B = 1.4 and B = 3.8, compared with 63.2% to 89.6% for Dual-Q. These results reveal a nonlinear safety–response trade-off: excessive conservatism at tight budgets can substantially limit mitigation, while additional budget permits the same learned strategic controller to act more effectively.

The zero-violation result at all three budgets arises from a fixed, correct cost envelope. To test the failure mode of an underestimated envelope rather than simply mentioning it as a limitation, we next alter the cost-generation process while holding the learned policy fixed.

6.5. Stress testing the cost-envelope assumption

Let the actual cost become $c_t(\kappa) = \kappa c_t$ while the shield assumes an upper cost of $1.6\rho(c_a + I_{macro}\, c_g)(1 + 0.55c)$, where κ is actual cost inflation and ρ is the safety-reserve multiplier. Proposition 1 extends whenever $\rho \geq \kappa$; otherwise the originally certified bound is no longer conservative. Only collateral costs and the shield admissibility threshold change; attack probabilities and the learned Q values remain fixed within each environment condition. In a separate 3-seed follow-up, each policy is trained for 1,700 nominal episodes and then evaluated over 300 fixed episodes for each combination of $\kappa \in \{1.0, 1.25, 1.5\}$, $\rho \in \{1.0, 1.25, 1.5\}$ and nominal/shifted alert conditions. This yields 16,200 individually recorded episodes; identical trained tables are reused at every reserve value for each seed, so the reserve comparisons are paired and retraining is not confounded with shielding.

Table VI. Cost-model stress, shifted condition

Actual κ	Reserve ρ	Resolution %	Over budget %
1.25	1.00	75.6	1.6
1.25	1.25	74.9	0.0
1.25	1.50	73.7	0.0
1.50	1.00	74.4	3.7
1.50	1.25	73.0	2.1
1.50	1.50	72.3	0.0

At $\kappa = 1.25$, the nominally calibrated shield ($\rho = 1.0$) exceeds the budget in 1.6% of shifted episodes, whereas a matching reserve ($\rho = 1.25$) reduces the observed exceedance rate to zero and shifts resolution from 75.6% to 74.9%. At the stronger $\kappa = 1.5$, the under-reserved controller violates the budget in 3.7% of shifted episodes. The matching $\rho = 1.5$ envelope eliminates observed violations and reduces shifted resolution from 74.4% to 72.3%; $\rho = 1.25$ is insufficient (2.1% violations). The nominal-condition results show the same qualitative pattern (0.8% versus zero violations at $\kappa = 1.5$, $\rho = 1.0$ versus $\rho = 1.5$). At $\kappa = 1.0$ every tested reserve is conservative and produces zero violations. These results identify the operational boundary of the guarantee instead of treating an assumed envelope as an automatically learned fact.

The 3-seed stress test is intentionally smaller than the five-seed primary benchmark, so fine-grained resolution differences are exploratory. The costs are scaled by synthetic multipliers rather than empirical outage measurements; the robust reserve represents assumed uncertainty bounds, not confidence-calibrated uncertainty learned from deployment telemetry. No claim is made about exceeding or meeting a real service-level objective.

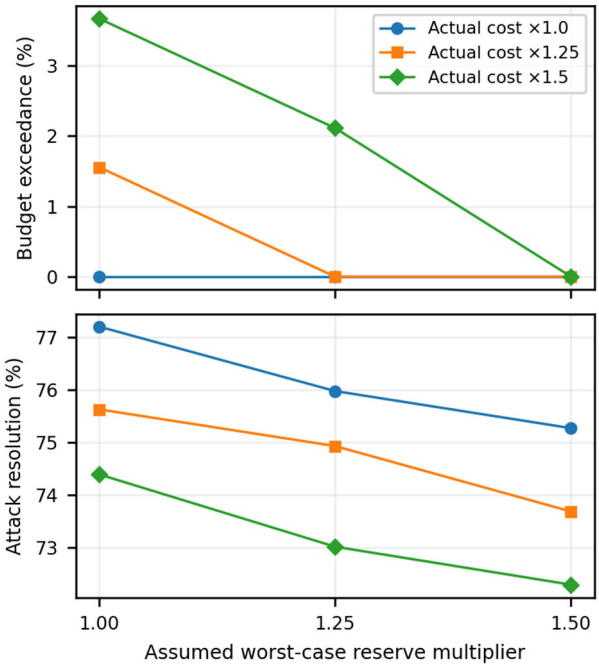


Fig. 5 Shifted-test cost inflation: budget violations and response effectiveness as the reserve multiplier increases; three independent training seeds.

6.6. Reproducibility and uncertainty

The primary uncertainty unit is the independent learner seed, not individual episode draws within the same trained policy. Means and standard deviations are calculated over five per-seed estimates. For paired comparisons we compute a t interval with four degrees of freedom from the five matched differences; intervals are descriptive with limited power and are not corrected for multiple comparisons. The evaluation has no live deployment, causal incident logs, wall-clock response-latency instrumentation, security analyst adjudication, or independent action-effect calibration. As a result, the numerical percentages are simulator behavior, not estimates of industry incident-handling performance.

7. Discussion

The results distinguish a conditional system-level guarantee from policy optimality. A hard upper envelope is stronger than an average penalty when the requirement is to respect the simulated disruption budget, but only if the envelope covers actual realized costs. The misspecification study demonstrates the logical converse empirically: optimistic cost bounds can admit interventions that exceed the budget, whereas adding a reserve large enough to cover the injected cost inflation restores feasibility. Conservative rejection reduces possible mitigation actions, producing an explicit containment penalty. Operational use would require defensible upper estimates of service outages, cumulative dependencies, and changing business criticality rather than borrowing the simulator cost coefficients.

The environment deliberately couples strategy and mitigation probability to make the two-timescale design

testable, and the resulting effect sizes depend on those design choices. Real incidents feature delayed and confounded rewards, dependent attack campaigns, partial observability of assets, escalation to humans, and rapidly changing mitigations. These limitations motivate evaluation with cyber ranges and human-supervised incident playbooks. Future extensions should model uncertain cost envelopes, enforce probabilistic rather than absolute bounds when costs are unbounded, and train richer belief encoders only after obtaining suitable longitudinal response-outcome evidence.

8. Conclusion

Safe-ARCS combines the ARCS dual response streams, decay-weighted urgency and prioritized replay with an explicit cost-envelope shield. Across 59,500 primary held-out synthetic episodes, the correctly bounded shield incurred no budget violations and traded 1.8 and 3.1 percentage points of nominal and shifted attack resolution for feasibility relative to unshielded Dual-Q. A separate 16,200-episode cost inflation experiment demonstrates that the guarantee fails when the envelope is optimistic but is restored by a matching conservative reserve, at further resolution cost. These findings delimit the safety–effectiveness trade-off in an auditable model; validation on calibrated response costs and operational incident trajectories remains necessary.

References

[1] Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials, 18(2), 1153-1176. https://doi.org/10.1109/COMST.2015.2494502

[2] Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications, 50, Article 102419. https://doi.org/10.1016/j.jisa.2019.102419

[3] Nguyen, T. T., & Reddi, V. J. (2023). Deep reinforcement learning for cyber security. IEEE Transactions on Neural Networks and Learning Systems, 34(8), 3779-3795. https://doi.org/10.1109/TNNLS.2021.3121870

[4] Adawadkar, A. M. K., & Kulkarni, N. (2022). Cyber-security and reinforcement learning: A brief survey. Engineering Applications of Artificial Intelligence, 114, Article 105116. https://doi.org/10.1016/j.engappai.2022.105116

[5] Ren, S., Jin, J., Niu, G., & Liu, Y. (2025). ARCS: Adaptive reinforcement learning framework for automated cybersecurity incident response strategy optimization. Applied Sciences, 15(2), Article 951. https://doi.org/10.3390/app15020951

[6] Sutton, R. S., Precup, D., & Singh, S. (1999). Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. Artificial Intelligence, 112(1-2), 181-211. https://doi.org/10.1016/S0004-3702(99)00052-1

[7] Dietterich, T. G. (2000). Hierarchical reinforcement learning with the MAXQ value function decomposition. Journal of Artificial Intelligence Research, 13, 227-303. https://doi.org/10.1613/JAIR.639

[8] Ghanem, M. C., Chen, T. M., & Nepomuceno, E. G. (2023). Hierarchical reinforcement learning for efficient and effective automated penetration testing of large networks. Journal of Intelligent Information Systems, 60, 281-303. https://doi.org/10.1007/s10844-022-00738-0

[9] Oh, S. H., Kim, J., Nah, J. H., & Park, J. (2024). Employing deep reinforcement learning to cyber-attack simulation for enhancing cybersecurity. Electronics, 13(3), Article 555. https://doi.org/10.3390/electronics13030555

[10] Hore, S., Shah, A., & Bastian, N. D. (2023). Deep VULMAN: A deep reinforcement learning-enabled cyber vulnerability management framework. Expert Systems with Applications, 221, Article 119734. https://doi.org/10.1016/j.eswa.2023.119734

[11] Vávra, J., Hromada, M., Lukáš, L., & Dworzecki, J. (2021). Adaptive anomaly detection system based on machine learning algorithms in an industrial control environment. International Journal of Critical Infrastructure Protection, 34, Article 100446. https://doi.org/10.1016/j.ijcip.2021.100446

[12] Shaukat, K., Luo, S., Varadharajan, V., Hameed, I. A., & Xu, M. (2020). A survey on machine learning techniques for cyber security in the last decade. IEEE Access, 8, 222310-222354. https://doi.org/10.1109/ACCESS.2020.3041951

[13] García, J., & Fernández, F. (2015). A comprehensive survey on safe reinforcement learning. Journal of Machine Learning Research, 16(42), 1437-1480.

[14] Achiam, J., Held, D., Tamar, A., & Abbeel, P. (2017). Constrained policy optimization. In Proceedings of the 34th International Conference on Machine Learning (ICML), PMLR 70 (pp. 22-31). PMLR.

[15] Alshiekh, M., Bloem, R., Ehlers, R., Könighofer, B., Niekum, S., & Topcu, U. (2018). Safe reinforcement learning via shielding. Proceedings of the AAAI Conference on Artificial Intelligence, 32(1), 2669-2678. https://doi.org/10.1609/aaai.v32i1.11797

[16] Chow, Y., Ghavamzadeh, M., Janson, L., & Pavone, M. (2018). Risk-constrained reinforcement learning with percentile risk criteria. Journal of Machine Learning Research, 18(167), 1-51.

[17] Tessler, C., Mankowitz, D. J., & Mannor, S. (2019). Reward constrained policy optimization. In Proceedings of the International Conference on Learning Representations (ICLR).

[18] Altman, E. (1999). Constrained Markov decision processes. Chapman & Hall/CRC.

[19] Schaul, T., Quan, J., Antonoglou, I., & Silver, D. (2016). Prioritized experience replay. In Proceedings of the International Conference on Learning Representations (ICLR).

[20] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529-533. https://doi.org/10.1038/nature14236

[21] van Hasselt, H., Guez, A., & Silver, D. (2016). Deep reinforcement learning with double Q-learning. Proceedings of the AAAI Conference on Artificial Intelligence, 30(1), 2094-2100. https://doi.org/10.1609/aaai.v30i1.10295

[22] Sewak, M., Sahay, S. K., & Rathore, H. (2023). Deep reinforcement learning in the advanced cybersecurity threat detection and protection. Information Systems Frontiers, 25, 589-611. https://doi.org/10.1007/s10796-022-10333-x

[23] Tamar, A., Glassner, Y., & Mannor, S. (2015). Optimizing the CVaR via sampling. Proceedings of the AAAI Conference on Artificial Intelligence, 29(1), 2993-2999. https://doi.org/10.1609/aaai.v29i1.9561

[24] Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems. In 2015 Military Communications and Information Systems Conference (MilCIS) (pp. 1-6). IEEE. https://doi.org/10.1109/MilCIS.2015.7348942

[25] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv, arXiv:1707.06347. https://doi.org/10.48550/arXiv.1707.06347

[26] Sarker, I. H. (2021). Deep cybersecurity: A comprehensive overview from neural network and deep learning perspective. SN Computer Science, 2, Article 154. https://doi.org/10.1007/s42979-021-00535-6