Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Training Versus Hiring in Security Operations Centers: Agent-Based and Reinforcement Learning Simulations of Training Duration Effects on CyberTeam Performance

Max N. Yaw
Saginaw Valley State University
Ferdinand Kpieleh
kpielefd@mail.uc.edu
University of Cincinnati
Aos Mulahuwaish
Saginaw Valley State University
Basheer Qolomany
Howard University
Jacques Bou Abdo
University of Cincinnati

Research Article

Keywords: Cyber team training, Agent-based modeling, Reinforcement learning, Team effectiveness, Security Operations Center, Workforce planning, Computational simulation

Posted Date: June 27th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9044748/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Training Versus Hiring in Security Operations Centers: Agent-Based and Reinforcement Learning Simulations of Training Duration Effects on Cyber Team Performance

Abstract

Security operations centers face critical resource allocation decisions: invest in training existing personnel or recruit additional staff? While team performance in cybersecurity depends on structure, skills, and cohesion, existing computational research has examined only static team composition without addressing how training duration affects performance trajectories. We address this gap by incorporating temporal dynamics into agent-based modeling of cyber team competitions, using reinforcement learning to simulate how teams develop strategies through sustained training. Our computational experiments ran thousands of episodes, which demonstrated three key findings. First, training duration affects performance nonlinearly through three phases: minimal improvement during early exploration (100-500 episodes, the win rate ranges from 0.00-0.15, rapid gains during skill acquisition (500-2,000 episodes, the win rate ranges from 0.53-0.57), and diminishing returns during refinement (2,000-5,000 episodes, the win rate ranges from 0.93-0.95). Second, optimal training investment occurs at 500-1,000 episodes, where marginal gains per additional episode are highest and returns decline substantially beyond 2,000 episodes. Third, trained teams achieve about 3-7× efficiency advantages in most scenarios (and higher in capped cases), with 5 trained attackers performing equivalently to roughly 15-35 untrained attackers. These findings provide quantitative evidence for cybersecurity workforce planning decisions. The nonlinear learning trajectory suggests organizations should design training programs to extend beyond exploration phases to capture high-yield improvements. The substantial efficiency advantages support training-focused workforce strategies, with a full 5,000-episode training program substituting for the recruitment of roughly 20-25 additional personnel.

Keywords: Cyber team training, Agent-based modeling, Reinforcement learning, Team effectiveness, Security Operations Center, Workforce planning, Computational simulation

1 Introduction

Cyber operations centers face a critical decision regarding resource allocation: should they invest in training existing personnel or recruit additional staff? This question has significant implications for organizational effectiveness and budget allocation, yet existing research provides limited guidance. Team performance in cybersecurity contexts depends on structure, skills, cohesion, and training (Kpieleh, Abdo, Mulahuwaish, Qolomany, & Hossain, 2025), but prior computational research has focused primarily on static team composition rather than dynamic learning processes.

Recent agent-based modeling studies have advanced the understanding of cyber team formation. (Calay, Qolomany, Mulahuwaish, Hossain, & Abdo, 2023) examined the link between team structure and performance using NetLogo simulations of red team versus blue team competitions. (Qolomany, Calay, Hossain, Mulahuwaish, & Bou Abdo, 2024) extended this work by incorporating team size, social cohesion, collective skills, and structural dynamics into a comprehensive theoretical model, which was validated through structural equation modeling. Both studies provided important insights into how team composition affects cybersecurity outcomes at a single point in time.

However, a critical gap remains: these studies measured performance in single simulation rounds without examining how teams develop capabilities through sustained training. Previous research has not addressed how training duration affects team learning curves, at what point training yields diminishing returns, or how trained small teams compare to larger untrained teams in effectiveness. These questions directly impact organizational decisions about training program design, workforce development timelines, and the relative value of training investments versus recruitment strategies.

We address this gap by incorporating temporal dynamics into the established Collaborative Cyber Team Formation (CCTF) framework (Calay et al., 2023; Qolomany et al., 2024). Our approach uses agent-based modeling with reinforcement learning (RL) to simulate how teams develop strategies through sustained training. Critically, we employ reinforcement learning not as the subject of investigation, but as a modeling tool to capture human team learning trajectories. Just as human cyber teams improve by exploring different approaches and reinforcing successful tactics through repeated practice, RL agents learn from experience by receiving rewards for effective actions. This enables us to simulate the learning process that would occur in real teams over months or years of training, compressed into thousands of computational episodes.

This modeling approach addresses a fundamental limitation of prior work: (Qolomany et al., 2024) employed stochastic attack strategies that produced random outcomes in each simulation round, making it impossible to isolate the effect of training duration. By using RL policies that adapt and improve over time, we can track performance trajectories across training phases and quantify learning curves similar to how competitive e-gaming teams develop coordinated strategies through sustained practice (Bonny, 2022). Our simulations run for up to 5,000 episodes, capturing the full arc from initial exploration through skill acquisition to performance plateau.

Our computational experiments address three research questions that carry direct implications for security operations center workforce planning:

1. How does training duration affect cyber team performance trajectories? Specifically, we investigate whether learning follows a linear progression or exhibits distinct phases, and at what point performance improvements plateau.
2. What training investment yields optimal returns? We quantify the relationship between training episodes and performance gains to identify the point of diminishing returns, enabling organizations to balance training costs against effectiveness improvements.
3. How do trained small teams compare to larger untrained teams? Here, *untrained* denotes baseline heuristic teams with fixed rules and no learning across episodes. By establishing performance equivalence thresholds, we provide evidence to inform decisions between training existing personnel versus recruiting additional staff.

These questions translate abstract concerns about team development into measurable outcomes that can guide resource allocation decisions. Organizations investing in cybersecurity teams need guidance on the duration of the training program and the relative value of skill development versus team expansion.

This study makes three contributions to understanding cyber team performance and training investment: **First, we characterize the learning trajectory of cyber teams through sustained training.** By tracking performance across training checkpoints from early exploration through skill mastery, we demonstrate that learning follows a three-phase progression: minimal improvement during initial exploration (100-500 episodes), rapid gains during skill acquisition (500-2,000 episodes), and continued but diminishing returns during refinement (2,000-5,000 episodes). This trajectory provides evidence for training program design decisions.

Second, we quantify optimal training investment through return on investment analysis. Peak training efficiency occurs at 500-1,000 episodes, where marginal gains per additional episode are highest. In episode-based units, this corresponds to approximately 6.7-8.3 percentage points of win-rate improvement per 100 episodes and is several times larger than the return observed beyond 2,000 episodes. This analysis identifies the point at which additional training episodes yield diminishing marginal returns.

Third, we establish the effectiveness-equivalence between trained and untrained teams. Computational experiments demonstrate that teams with 5 trained attackers achieve performance levels requiring 15-35 untrained attackers, a 3-7$\times$ efficiency advantage. This provides quantitative evidence for decisions between investing in training versus recruiting additional personnel.

Our findings reveal that training duration affects cyber team performance nonlinearly. Performance improvements follow a characteristic curve: slow initial exploration, mid-training gains, and eventual plateau with diminishing returns. This pattern suggests that organizations stopping training too early forfeit high-yield improvements, while extending training indefinitely yields smaller gains per additional episode.

We find that trained teams with 5 attackers achieve performance comparable to untrained teams of 25-30 attackers in favorable conditions. This efficiency advantage, up to 7$\times$ in some scenarios, provides quantitative support for training-focused workforce strategies over recruitment-focused expansion in cybersecurity teams.

The remainder of this paper proceeds as follows. Section 2 reviews related literature on cyber team performance and situates our temporal modeling approach within existing computational and theoretical frameworks. Section 3 details our agent-based modeling methodology, including the reinforcement learning implementation that enables simulation of sustained training effects. Section 4 presents results across three analyses: learning trajectories over training duration, return on investment calculations, and effectiveness comparisons between trained and untrained teams. Section 6 discusses implications for security operations centers and identifies directions for future research.

2 Related Work

Understanding cyber team performance has become increasingly important as organizations face sophisticated cyber threats that span technical and social dimensions (Mulahuwaish et al., 2025). A systematic review of 137 studies identified communication, coordination, and leadership as core predictors of team effectiveness across cybersecurity, software engineering, and e-gaming domains (Kpieleh et al., 2025). However, traditional team effectiveness models (Hackman, 2002; Ilgen, Hollenbeck, Johnson, & Jundt, 2005; Kozlowski & Ilgen, 2006; Mathieu, Gallagher, Domingo, & Klock, 2019; Salas, Sims, & Burke, 2005; Tuckman, 1965; Woolley, Chabris, Pentland, Hashmi, & Malone, 2010) require adaptation for cyber contexts due to distributed work, temporal pressure, and adversarial threats (Burke, Stagl, Salas, Pierce, & Kendall, 2006; Kpieleh et al., 2025; Shah, Ganesan, Jajodia, Cam, & Hutchinson, 2023), with performance-based team formation frameworks (Shah et al., 2023) and research on security operations centers (Balázs & Csaba, 2017; Tilbury & Flowerday, 2024) demonstrating the need for specialized approaches addressing technical skills, threat dynamics, and socio-technical coordination challenges.

Despite growing research on cyber team effectiveness, existing studies have not examined how training duration affects the performance trajectories of cyber teams. Research has addressed temporal aspects of team learning (Santos, Uitdewilligen, & Passos, 2015). Systematic evidence confirms that team training improves performance across healthcare (Fung et al., 2015; Hughes et al., 2016; Weaver, Dy, & Rosen, 2014) and military domains (Johnston et al., 2019), with empirical studies demonstrating measurable improvements from training and role specialization in cybersecurity competitions (La Fleur, Hoffman, Gibson, & Buchler, 2021), questions regarding optimal training duration, learning curve characteristics, and return on training investment remain unexplored in cybersecurity team development.

Two computational studies established foundations for modeling cyber team dynamics through agent-based simulation, though both focus on static configurations. (Calay et al., 2023) introduced the Collaborative Cyber Team Formation (CCTF) framework, using NetLogo simulations of red team versus blue team competitions to investigate correlations between team structure and performance. The framework established roles aligned with EC-Council's ethical hacking methodology (EC-Council, 2024): attacker scouts and exploiters, defender detectors and interceptors. (Qolomany

et al., 2024) extended this work by building a comprehensive theoretical model incorporating team size, skills, collaboration, and social cohesion, validating it through Structural Equation Modeling against computational simulation results. While defining and measuring cyber team performance presents methodological challenges across contexts (Dobson, 2019; GEOFFREY & KATHLEEN, 2025), both studies measured performance at single time points without examining how teams develop through sustained training, leaving questions about training duration, learning curves, and performance trajectories unexplored.

Agent-based modeling provides a computational approach for investigating cyber team dynamics under controlled conditions (Dasgupta & Khazanchi, 2019; Zhao, Chong, & Li, 2023). The Cyber-FIT framework (Dobson, 2022) demonstrates another application to cyber warfare simulation, with validation efforts showing correspondence between agent-based simulations and military cyber team behaviors (Dobson & Carley, 2017, 2020, 2021). These approaches enable precise variable control to isolate the effects of team structure and coordination mechanisms. However, existing agent-based models focus on static configurations, measuring performance at single time points rather than tracking team development processes over time (Kozlowski & Ilgen, 2006). Research on team effectiveness has established that teams function through multiphase episodic processes (Marks, Mathieu, & Zaccaro, 2001), performing activities simultaneously and sequentially across temporal episodes rather than as static entities.

Reinforcement learning provides a mechanism for modeling how teams develop strategies through repeated practice. Unlike supervised learning, RL enables agents to learn from experience by receiving rewards for effective actions. This learning trajectory (initial exploration, gradual strategy identification, eventual plateau) mirrors human team development patterns observed in competitive e-gaming, military training, and incident response (Bonny, 2022). We employ RL not as the subject of investigation, but as a modeling tool to simulate human team learning processes. By running thousands of training episodes, we capture learning curves that would develop in real teams over months or years, compressed into computationally tractable simulations.

Critical gaps remain regarding temporal factors in cyber team performance: How does sustained training affect team learning trajectories? At what point does additional training yield diminishing returns? How do trained small teams compare to larger untrained teams? These questions have direct implications for organizational resource allocation between training existing personnel and recruiting additional staff. Single-round simulations cannot address these questions, which require modeling team development over hundreds or thousands of training iterations. We address these gaps by incorporating temporal dynamics into the CCTF framework (Calay et al., 2023; Qolomany et al., 2024) through reinforcement learning agents that develop strategies over extended training duration, providing evidence for organizational decisions that would be prohibitively expensive or time-consuming to obtain through longitudinal human subject experiments.

The following section describes our agent-based modeling methodology, including the network simulation environment, baseline heuristic agents representing untrained

teams, reinforcement learning agents representing trained teams, training protocols, and an evaluation framework.

3 Methodology

Agent-based modeling has been employed to simulate cyber team competitions and measure how training duration affects performance. Our approach implemented two types of teams: heuristic baselines, which represent untrained personnel following simple rules, and reinforcement learning agents representing trained teams that develop strategies through repeated practice. By running thousands of simulated engagements, we track learning curves, quantify return on training investment, and establish performance equivalences between trained and untrained configurations.

This section proceeds as follows. We first describe the network simulation environment that provides the competitive arena (Section 3.1). We then detail the heuristic baseline agents that serve as our untrained control condition (Section 3.2) and the reinforcement learning agents that model trained teams (Section 3.3). Next, we explain our training protocol for capturing learning over time (Section 3.4), the evaluation framework for measuring performance (Section 3.5), and our statistical analysis methods (Section 3.6).

All simulations use NetLogo, an established agent-based modeling platform designed for simulating complex systems with multiple interacting entities (Wilensky, 1999). NetLogo provides the computational infrastructure for creating agents, networks, and iterative dynamics required for our experiments.

For readability, the full terminology table is provided in Appendix Table A1. In this paper, *baseline*, *heuristic*, and *untrained* refer to the same fixed-rule, non-learning condition.

3.1 Network Simulation Environment

The simulation environment models a cyber-attack and defense scenario, where attacker teams attempt to compromise a target network while defender teams work to prevent intrusions. This environment provides a competitive arena for measuring team performance under controlled conditions.

3.1.1 Network Topology and Router States

We generate scale-free network topologies using Barabási-Albert preferential attachment, creating 60-node router networks that exhibit realistic degree distributions similar to an organization's IT infrastructure. Each router maintains three Boolean state flags: *online* (operational status), *vulnerable* (exploitable by attackers), and *infected* (compromised by attackers). Attackers begin connecting to low-degree peripheral routers and aim to compromise the designated central high-degree router, while defenders protect the network from the center outward. The simulation uses breadth-first search (BFS) for network reachability calculations, supporting routing decisions, layout computations, and post-defense cleanup operations. When forwarding tables are enabled, packets use next-hop routing; otherwise, paths fall back to BFS shortest-path

routing. After successful defensive actions, a BFS sweep resets unreachable routers to clean states, preventing stranded infections.

3.1.2 Agent Roles and Actions

Four operational roles mirror EC-Council's ethical hacking methodology (EC-Council, 2024): attacker scouts perform reconnaissance, attacker exploiters compromise vulnerable systems, defender detectors observe and orient to threats, and defender interceptors execute defensive countermeasures. This role structure reflects realistic cyber operation team compositions. Agents act through packets, mobile action carriers that traverse the network. Each packet encodes its originating agent, destination router, and action type (ATTACK, DEFEND, SCOUT, or DIAGNOSE). Packets route using forwarding tables when available; otherwise, the simulation computes BFS routes to destinations.

3.1.3 Temporal Dynamics

Time advances in discrete ticks $t = 0, 1, 2, \ldots$. Each agent has an action delay d and a next-action tick. Agents become eligible to act when $t \geq$ *next-action tick*; after acting, they schedule their next action at $t + d$. This turn-based system prevents simultaneous actions and creates realistic temporal sequencing. In our experiments, attacker exploiters act every $d_{\text{attacker}} = 10$ ticks, reflecting the time required for exploitation attempts. Attacker scouts act every $d_{\text{scout}} = 1$ tick, representing rapid reconnaissance operations. Defender roles remain stationed at the central router and dispatch actions from that position.

3.1.4 Episode Termination

Episodes end under two conditions: (1) attackers successfully compromise the designated central router (attacker victory), or (2) the simulation reaches the tick cutoff $T_{\max} = 1000$ without central compromise (defender victory by timeout). This 1000-tick limit provides sufficient time for strategic approaches to unfold while maintaining computational feasibility across thousands of training episodes.

3.2 Baseline Agents: Untrained Heuristic Control

Baseline agents implement hand-crafted heuristic policies to represent cyber teams operating without sustained training. These agents follow simple condition-action rules based on immediate observations rather than learned strategic patterns. By comparing baseline performance against reinforcement learning policies, we isolate the effect of training duration on team effectiveness.

3.2.1 Heuristic Design Principles

Baseline policies operate within the tick-gated action windows described in Section 3.1. They maintain awareness tables updated by scout observations and resolve action ties through uniform random selection. This design creates competent but non-adaptive opponents that execute reasonable strategies without learning from experience.

3.2.2 Attacker Exploiter Policy

At each eligible action window, an attacker exploiter inspects its currently attached router r and executes a two-phase decision: **Phase 1: Movement decision.** If r is offline or not vulnerable, the agent probabilistically decides whether to relocate (controlled by parameter p_{move}). When relocating, the agent prefers to move to known vulnerable routers from the attacker awareness table A^{att}; if none are known, it selects a random peripheral router. **Phase 2: Attack action.** The agent sends an ATTACK packet toward the first non-infected router on the shortest path from r to the central router. If the center is already infected, the target defaults to a randomly selected clean router where a path exists. After selecting actions, the agent creates the appropriate packet and advances its next-action tick by d_{attacker}.

3.2.3 Attacker Scout Policy

Scouts operate on the faster action schedule ($d_{\text{scout}} = 1$), continuously gathering intelligence. At each action window, a scout selects a peripheral target uniformly from candidates and dispatches a SCOUT packet. Upon packet arrival, the scout updates the team's attacker awareness table A^{att} with the target router's vulnerability and infection status.

3.2.4 Defender Policies

Defender interceptors remain attached to the central router and consult the defender awareness table A^{def} at each action window. When A^{def} contains identified threats, interceptors prioritize infected routers over merely vulnerable ones and sample uniformly among ties. They dispatch DEFEND packets to chosen targets, then schedule their next action. Defender detectors probe routers selected by a degree-staleness weighting function with distance penalties, favoring high-degree routers near the center that haven't been checked recently. DIAGNOSE packets report infection and vulnerability status when a skill-scaled random draw exceeds detection thresholds, updating A^{def} with configured report delays. This heuristic baseline design creates teams that execute sensible strategies but cannot adapt to opponent behaviors or learn from repeated engagements, characteristics we contrasted with trained RL policies.

3.3 Reinforcement Learning Agents: Modeling Trained Teams

We implemented reinforcement learning (RL) agents not to investigate whether RL improves robot team coordination, but to model how human cyber teams develop strategies through sustained training. RL provides a principled framework for simulating the learning process that occurs when teams train together repeatedly: early exploration of different approaches, gradual identification of effective tactics, and eventual convergence to refined strategies.

3.3.1 RL as a Training Simulation Tool

Just as human cyber teams improve through practice by trying different approaches and reinforcing successful ones, RL agents learn from experience by receiving rewards

for effective actions. This parallel enables us to compress months or years of human team training into thousands of simulated episodes, capturing the learning trajectory without requiring longitudinal human subject experiments. The key insight is that RL's learning curve (its initial poor performance during exploration, rapid improvement during skill acquisition, and eventual plateau) mirrors human team development patterns observed in competitive e-gaming and military training (Bonny, 2022). By tracking RL performance across training episodes, we simulate how training duration affects team effectiveness.

3.3.2 Rainbow DQN Architecture

Our implementation employed Rainbow Deep Q-Network (DQN), which combines multiple proven RL components: n-step returns for multi-step credit assignment, prioritized experience replay with importance sampling for efficient learning from past experiences, and periodic target network updates for training stability. We include distributional, double, dueling, and noisy network extensions that collectively improve learning efficiency and final performance.

Both attacker exploiters and attacker scouts are implemented as Rainbow DQN agents. All attackers share a policy through weight sharing: all scouts share a separate policy. This weight sharing mirrors how human teams develop shared tactical knowledge and supports evaluating fixed formations without retraining separate per-agent models.

Full layer-by-layer architecture details are provided in Appendix Table A2. Complete optimization, replay, exploration, and target-update hyperparameters are reported in Appendix Table A3.

3.3.3 State Representation and Action Space

Agents observe the network state through a fixed-order feature vector combining per-router information with team and global context. Per-router features include perceived state (clean/vulnerable/infected), hop distance from current location, network degree, distance to center, periphery status, infected neighbor count, and recency counters tracking this agent's recent moves, attacks, and scout reports to each router. Team context features capture teammate positioning (counts on periphery, near center, on vulnerable/compromised routers) and recent teammate actions (e.g., recent sends to vulnerable routers). Global features include normalized episode time $t/T_{\max}$, network size, and fractions of routers in each state category. The action space consists of $2N + 1$ discrete actions for N routers: N `move` actions (relocate to router i), N `send` actions (dispatch attack packet to router i), and one `stay` action (idle). We compute legality masks at each decision point to restrict actions to valid choices, for example, preventing attacks on routers that haven't been scouted yet (enforcing reconnaissance-before-exploitation).

3.3.4 Reward Structure

The per-step reward combines objective progress with behavioral shaping:

$$r_t = r_{\text{tick}} + r_{\text{action-shaping}} + r_{\text{packet-event}} + r_{\text{terminal/team}}.$$

The largest positive rewards are tied to successful infection and central-router compromise, while shaping terms discourage oscillation/redundant actions and encourage timely reconnaissance and coordinated exploitation. Coefficients were tuned through pilot runs to preserve the primary objective while stabilizing learning behavior across seeds. Full reward definitions, coefficients, and trigger conditions are provided in Appendix Table A4.

3.3.5 Training and Evaluation Modes

During training, policies use Noisy Linear exploration plus the staged ε schedule in Appendix Table A3. For the paper-reported checkpoints, training uses the initial-team regime capped at four total agents per side (attacker plus scout roles). Environment conditions are sampled according to the curriculum in Section 3.4. For evaluation, we freeze checkpoints, disable learning updates, and set `eval-epsilon` to the experiment value (for paper-reported checkpoint results: $\varepsilon = 0$). We then evaluate fixed formations RL(5/3) and RL(3/5) on the standardized 12-cell grid to measure how the learned policies transfer beyond the smaller training team regime.

3.3.6 Training Stability Diagnostics

To document training stability and support reproducibility, the implementation logs and validates several diagnostics:

- **Per-episode loss logs:** cumulative running loss and update count are written each episode to csv files.
- **Episode-level training log:** episode logs record win flag, episode length, rewards, rolling win rate, sampled environment parameters, and seed.
- **Q-value stability indicator:** on training updates, expected Q values from distributional logits are checked for finite values before target-action selection; non-finite values trigger an immediate runtime failure.
- **Numerical stability guards:** explicit finite-value assertions are applied to scaled states, rewards, Q values, projected target distributions, log-probabilities, and loss; invalid values raise runtime errors.
- **Gradient controls:** NaN gradients are zeroed and gradients are clipped to norm 10.0 before optimizer steps.

Learning curves and rolling-win trajectories derived from these logs are reported in the Results section to show convergence behavior and phase transitions.

3.3.7 Code Availability and Algorithm Specification

The replication artifact for this study includes the NetLogo model, Python agent implementations embedded in the model, checkpoint manifests, and analysis scripts.

These files are sufficient to reproduce training, checkpoint evaluation, and all reported tables/figures.

Core training/evaluation workflow:

1. Initialize NetLogo environment and Python bridge; build Rainbow attacker and scout agents.
2. For each training episode $e = 1, \ldots, 5000$: sample curriculum environment under the initial training-team regime (total agents per side capped at 4), reset simulation, advance one RL episode counter, and run until attacker win or tick cutoff.
3. During each episode: collect transitions from action and packet events, build up to 6-step returns, and append to prioritized replay.
4. Apply gradient updates on schedule (attacker every 10 ticks, scout every 4 ticks) with distributional double-DQN target projection and PER reweighting.
5. Save checkpoints every 50 episodes and at training end.
6. For each milestone checkpoint (100, 500, 1,000, 2,000, 5,000): reload frozen policies, set evaluation ε, disable learning, and run the full seeded 12-cell evaluation grid with 10 replicates per cell and team configuration.

3.4 Training Duration Protocol

Training duration is our primary independent variable. Each training episode represents a complete simulated engagement from environment initialization to either attacker success or the tick cutoff ($T_{\max} = 1000$). We train for up to 5,000 episodes to capture the full learning trajectory from initial exploration through skill acquisition to performance plateau.

3.4.1 Episode-Based Reporting

Our primary and only training-duration unit is **episodes**. We report all learning and investment conclusions in episode space (100, 500, 1,000, 2,000, 5,000 checkpoints), which is platform-independent and directly tied to the simulation protocol.

3.4.2 Checkpoint Evaluation Strategy

We adopt two complementary approaches to measuring learning over time: **Training-log learning curves:** We recompute rolling win rates directly from raw training episodes using a 100-episode sliding window:

$$\hat{W}_t = \frac{1}{100} \sum_{i=t-99}^{t} \text{win}_i$$

Training log mean the per episode metrics recorded directly during RL optimization process, such as win outcomes, related training signals,and environment parameters. The logger field rolling win rate percentage uses a 20 episode window for runtime monitoring and is not used for the published learning curve statistic. This provides high temporal resolution of on-policy learning dynamics, including exploration noise and phase transitions. **Checkpoint evaluations:** We save policy checkpoints every

50 episodes during training, then select milestone checkpoints at 100, 500, 1,000, 2,000, and 5,000 episodes. Each frozen checkpoint undergoes standardized evaluation (described in Section 3.5) with identical defender configurations and matched random seeds; the evaluation sweep resets the deterministic run counter at the start of each checkpoint run so seed assignments are aligned across milestones. This approach separates training dynamics from evaluation performance, providing cleaner measurements of learning progress.

3.4.3 Curriculum Phasing

Our training curriculum introduces progressively challenging opponents with phase boundaries at episodes 200 and 750:

- **Phase 1 (episodes 1-200):** Forced very-easy sampling ($\mathrm{VD}, \mathrm{ID} \in \{90, 100\}$ and $\mathrm{RV}, \mathrm{RT} \in \{80, 100\}$)
- **Phase 2 (episodes 201-750):** Mostly easy sampling with occasional very-easy carryover (approximately 85% easy, 15% very easy). Easy draws use $\mathrm{VD}, \mathrm{ID} \in \{50, 70, 90\}$ and RV/RT follow a decaying schedule.
- **Phase 3 (episodes 751-5,000):** Probabilistic mixture (approximately 70% mixed/harder settings, 15% easy, 15% very easy)

After episode 750, mixed/harder settings are sampled from $\mathrm{VD}, \mathrm{ID} \in \{25, 75\}$ and $\mathrm{RV}, \mathrm{RT} \in \{1, 4, 9, 13\}$, with remaining episodes sampled from easy or very-easy branches according to the curriculum probabilities.

3.4.4 Return on Training Investment Calculation

We compute return on training investment primarily in episode space as win-rate improvement per episode across fixed episode blocks. For checkpoint-based ROI, we use adjacent milestone checkpoints (100→500, 500→1,000, 1,000→2,000, 2,000→5,000):

$$\mathrm{ROI}_{\mathrm{interval}} = \frac{\hat{W}_{\mathrm{end}} - \hat{W}_{\mathrm{start}}}{\mathrm{episodes}_{\mathrm{interval}}} \times 100$$

For training-log ROI, we compute gains across 200-episode blocks and apply short moving-average smoothing over adjacent bins for visualization. For readability, ROI plots report gain rates as percentage points per 100 episodes.

3.5 Evaluation Protocol

Evaluation measures frozen policy performance under standardized conditions to ensure controlled comparisons between training durations, team sizes, and baseline heuristics.

3.5.1 Environment Grid

We construct a 12-cell environment grid by crossing network vulnerability parameters with defender configurations (Table 1). All environments use 60-router BA topologies with:

- **Router vulnerability (RV):** $\{3\%, 7\%, 10\%\}$ - percentage of routers initially vulnerable
- **Router threat (RT):** 5% (fixed) - baseline infection probability
- **Vulnerability detection (VD):** 15% (fixed) - defender detection rate for vulnerable routers
- **Infection detection (ID):** $\{3\%, 5\%\}$ - defender detection rate for infected routers

Table 1 Environment grid used for evaluation.

Env ID	Defenders	Routers	RV%	RT%	VD%	ID%
E01	3 Det / 5 Int	60	3	5	15	3
E02	3 Det / 5 Int	60	3	5	15	5
E03	3 Det / 5 Int	60	7	5	15	3
E04	3 Det / 5 Int	60	7	5	15	5
E05	3 Det / 5 Int	60	10	5	15	3
E06	3 Det / 5 Int	60	10	5	15	5
E07	5 Det / 3 Int	60	3	5	15	3
E08	5 Det / 3 Int	60	3	5	15	5
E09	5 Det / 3 Int	60	7	5	15	3
E10	5 Det / 3 Int	60	7	5	15	5
E11	5 Det / 3 Int	60	10	5	15	3
E12	5 Det / 3 Int	60	10	5	15	5

These ranges span easy to challenging defensive scenarios based on pilot studies, showing that they avoid floor and ceiling effects, and both teams have opportunities to succeed across conditions.

Defender configurations employ two role splits: (5 detectors / 3 interceptors) emphasizing observation, and (3 detectors / 5 interceptors) emphasizing response. Crossing 3 RV levels $\times$ 2 ID levels $\times$ 2 defender splits yields 12 distinct environment cells.

3.5.2 Team Configurations

For RL teams, we evaluate two formations: RL(5/3) with 5 attackers and 3 scouts, and RL(3/5) with 3 attackers and 5 scouts. These configurations test different role balances within the same total team size. For baseline teams, we fix scout count at 3 and vary attacker count across $A \in \{5, 10, 15, 20, 25, 30, 35\}$. This range enables dose-response analysis and establishes the baseline team sizes needed to match RL performance.

3.5.3 Episode Execution

For each environment cell and team configuration, we run $S = 10$ independent episodes using distinct fixed random seeds. Episodes use frozen policies with learning disabled ($\varepsilon = 0$), ensuring reproducible measurements. Episodes terminate at the central router compromise or tick cutoff $T_{\max} = 1000$. Agents follow the action scheduling from

Section 3.1: attacker exploiters step every 10 ticks, scouts every 1 tick. We log binary outcome (win/loss), ticks-to-win for successful episodes, and configuration identifiers for each run.

3.5.4 Performance Metrics

Our primary metric is **win rate**: the proportion of episodes ending in attacker victory (central router compromised). This measures strategic effectiveness, whether teams accomplish their objective. As a secondary metric, we compute **time-to-win** (mean ticks until victory) conditioned on successful episodes only. This measures tactical efficiency, given success, addressing how quickly winning strategies are completed.

3.6 Statistical Analysis

All uncertainty quantification uses nonparametric episode-level bootstrap resampling to avoid distributional assumptions.

3.6.1 Win Rate Estimation

For each configuration, we compute win rates over E episodes. To estimate uncertainty, we draw B bootstrap resamples of size E with replacement using a fixed random seed (NumPy `default_rng`, seed = 12345). For each resample $b = 1, \ldots, B$, we compute the resample mean $\hat{p}^{(b)}$. In the bootstrap-based tables and figures, point estimates are bootstrap means and confidence intervals (95% level) use equal-tailed percentile intervals $[\mathrm{q}_{2.5}, \mathrm{q}_{97.5}]$ from the bootstrap distribution $\{\hat{p}^{(b)}\}$. We use $B = 2,000$ resamples for win-rate estimation. For configurations at boundaries (all wins or all losses), bootstrap intervals degenerate to single values. In these cases, we report Wilson score binomial confidence intervals at 95% level using the empirical proportion as the point estimate. For near-boundary rates ($\hat{p} < 0.05$ or $\hat{p} > 0.95$) in aggregated analyses, we apply bias-corrected and accelerated (BCa) intervals when appropriate.

3.6.2 Dose-Response Curves

We model baseline win rate as a function of attacker count n using isotonic regression (Pool-Adjacent-Violators Algorithm), which enforces monotonicity: win rates should not decrease as teams add more attackers. This produces piecewise-linear fits between adjacent tested values. For method-level aggregates (pooling across environment cells), we fit weighted isotonic regression with weights equal to episode counts at each n. For cell-specific EBTS calculations, we use unweighted isotonic fits on each cell's observed n grid.

3.6.3 Equivalent Baseline Team Size (EBTS)

For each configuration and RL team size A_{RL}, we estimate the smallest baseline attacker count $\hat{N}$ whose win rate matches or exceeds the RL win rate. This establishes performance equivalence. We compute EBTS through a joint episode-level bootstrap over RL and all baseline attacker counts. For each bootstrap replicate b:

1. Compute RL win rate $\hat{p}_{\text{RL}}^{(b)}$ from resampled RL episodes
2. Compute baseline win rates $\{\hat{p}_{\text{base}}^{(b)}(n)\}$ for all tested $n \in \{5, 10, \ldots, 35\}$
3. Fit unweighted isotonic regression $\tilde{p}_{\text{base}}^{(b)}(\cdot)$ over baseline n values
4. Invert via piecewise linear interpolation:

$$\hat{N}^{(b)} = \inf\{x \geq 0 : \tilde{p}_{\text{base}}^{(b)}(x) \geq \hat{p}_{\text{RL}}^{(b)}\}$$

If $\hat{p}_{\text{RL}}^{(b)}$ exceeds the maximum baseline win rate within the tested range, we set $\hat{N}^{(b)} = \infty$ (indicating the baseline cannot reach RL performance with ≤ 35 attackers). We report EBTS point estimates as bootstrap means with 95% percentile confidence intervals over $\{\hat{N}^{(b)}\}$ using $B = 1,000$ resamples. The efficiency multiplier $k = \hat{N}/A_{\text{RL}}$ indicates how many baseline attackers are required per RL attacker. In figures, infinite $\hat{N}$ values are marked "beyond range" and visually capped at $1.5 \times N_{\max}$. In tables, we print "> 35" for these cases.

3.6.4 Time-to-Win Analysis

We compute mean ticks-to-win over successful episodes only (win = 1) using the same bootstrap procedure. Methods with fewer than 5 wins in aggregated visualizations are shown with reduced opacity to reflect a higher uncertainty estimate, even though we report all estimates without exclusions.

4 Results

Our computational experiments address three research questions about cyber team training and composition. This section presents evidence demonstrating that (1) training duration substantially affects performance through a three-phase learning trajectory, (2) optimal training investment occurs at 500-1,000 episodes with diminishing returns thereafter, and (3) trained small teams achieve performance equivalent to untrained teams 3-7 times larger.

We organize results as follows. Section 4.1 establishes the baseline dose-response relationship and RL reference performance. Section 4.2 compares aggregate win rates across methods. Section 4.3 examines tactical efficiency through time-to-win analysis. Section 4.4 addresses Research Question 1 by tracking learning trajectories over training duration. Section 4.5 addresses Research Question 2 through return on investment calculations. Section 4.6 addresses Research Question 3 by quantifying the effectiveness advantage of trained teams. All analyses use the evaluation framework and statistical methods described in Sections 3.5 and 3.6.

Unless otherwise noted, RL results use the 5,000-episode checkpoint representing fully trained policies. Sections 4.4 and 4.5 examine intermediate checkpoints (100, 500, 1,000, 2,000 episodes) to characterize learning progression. All experiments use 60-router networks across the 12-cell environment grid (Table 1) with two defender configurations: 5 detectors / 3 interceptors and 3 detectors / 5 interceptors.

Throughout this section, *baseline*, *heuristic*, and *untrained* refer to the same fixed-rule, non-learning policy class.

4.1 Baseline Dose-Response and RL Reference Performance

We first establish how baseline team performance scales with team size, providing context for interpreting trained team efficiency. Figure 1 shows baseline win rates as a function of attacker count (with fixed 3 scouts), overlaid with RL(5/3) performance at the 5,000-episode checkpoint.

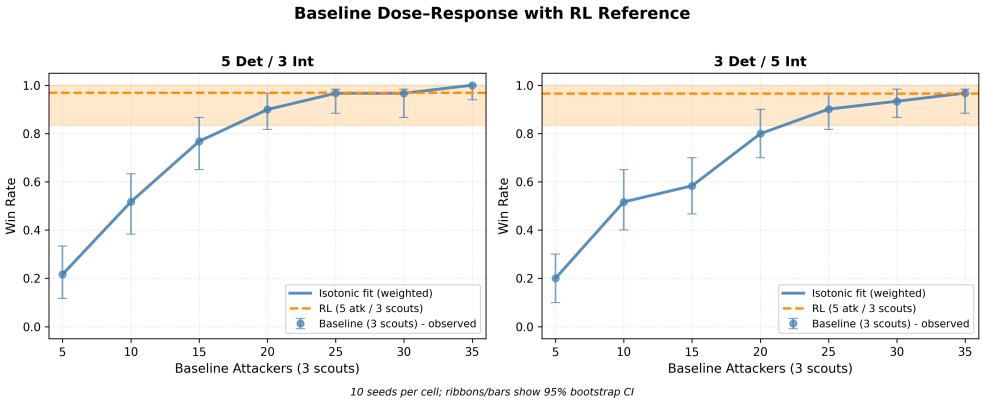


Fig. 1 Baseline dose-response with RL reference. Blue circles show observed baseline win rates (3 scouts, varying attackers) with 95% bootstrap confidence intervals; the solid blue curve is weighted isotonic regression enforcing monotonicity. Orange band shows RL(5/3) mean win rate with 95% bootstrap CI. Left panel: 5 Det / 3 Int defenders. Right panel: 3 Det / 5 Int defenders. The baseline requires approximately 25-30 attackers to match RL(5/3) performance with only 5 attackers, a 5-6× efficiency advantage.

The isotonic fits (solid blue curves) capture the expected pattern: baseline win rates increase monotonically as teams add more attackers. However, the rate of improvement diminishes; adding the 30th attacker provides less marginal benefit than adding the 10th. This nonlinearity reflects coordination overhead and diminishing returns from numerical superiority alone.

The RL(5/3) reference bands (orange) reveal a striking efficiency advantage. Under the 5 Det / 3 Int configuration (left panel), RL(5/3) achieves approximately 93% win rate. The baseline requires roughly 25 attackers to reach comparable performance, a 5× advantage. Under the 3 Det / 5 Int configuration (right panel), RL(5/3) maintains approximately 95% win rate, matching baselines with approximately 30 attackers, a 6× advantage.

This pattern appears consistently across both defender configurations, indicating that training effects are robust to defensive team composition. The weighted isotonic fits account for varying episode counts at each baseline size, ensuring reliable trend estimation.

Table 2 provides detailed win rates for all method-environment combinations, enabling examination of specific scenarios. RL(5/3) achieves win rates above 0.85 in 11 of 12 environment cells, while baselines require substantially larger teams to reach

comparable performance. Readers interested in how specific vulnerability rates (RV) or detection capabilities (ID) affect performance can consult this table for granular comparisons.

Table 2 Per-cell win rates (RL vs. Baseline). Left: RL formations by environment ID (Table 1). Right: Baseline with 3 scouts, split across attacker counts (top: 5/10/15; bottom: 20/25/30/35). Entries are bootstrap means [95% CI] over seeds; Wilson at boundaries.

RL win rates (by Env ID)

E01–E06

Env ID	RL(5a/3s)	RL(3a/5s)
E01	0.802 [0.500–1.000]	0.807 [0.500–1.000]
E02	1.000 [0.722–1.000]	0.496 [0.200–0.800]
E03	1.000 [0.722–1.000]	0.602 [0.300–0.900]
E04	0.900 [0.700–1.000]	0.706 [0.400–1.000]
E05	0.896 [0.700–1.000]	0.905 [0.700–1.000]
E06	1.000 [0.722–1.000]	0.606 [0.300–0.900]

E07–E12

Env ID	RL(5a/3s)	RL(3a/5s)
E07	1.000 [0.722–1.000]	0.905 [0.700–1.000]
E08	1.000 [0.722–1.000]	0.902 [0.700–1.000]
E09	0.904 [0.700–1.000]	0.794 [0.500–1.000]
E10	0.899 [0.700–1.000]	0.605 [0.300–0.900]
E11	1.000 [0.722–1.000]	0.796 [0.500–1.000]
E12	0.895 [0.700–1.000]	0.896 [0.700–1.000]

Baseline win rates (3 scouts)

A = 5, 10, 15

Env ID	A=5	A=10	A=15
E01	0.000 [0.000–0.278]	0.602 [0.300–0.900]	0.403 [0.100–0.700]
E02	0.396 [0.100–0.700]	0.306 [0.000–0.600]	0.699 [0.400–1.000]
E03	0.199 [0.000–0.500]	0.607 [0.300–0.900]	0.804 [0.598–1.000]
E04	0.303 [0.100–0.600]	0.595 [0.300–0.900]	0.798 [0.500–1.000]
E05	0.000 [0.000–0.278]	0.500 [0.200–0.800]	0.399 [0.100–0.700]
E06	0.310 [0.100–0.600]	0.506 [0.200–0.800]	0.405 [0.100–0.700]
E07	0.196 [0.000–0.500]	0.297 [0.000–0.600]	0.696 [0.400–1.000]
E08	0.196 [0.000–0.500]	0.696 [0.400–1.000]	0.695 [0.400–0.900]
E09	0.101 [0.000–0.300]	0.600 [0.300–0.900]	0.605 [0.300–0.900]
E10	0.098 [0.000–0.300]	0.398 [0.100–0.700]	0.899 [0.700–1.000]
E11	0.392 [0.100–0.700]	0.494 [0.200–0.800]	0.902 [0.700–1.000]
E12	0.299 [0.000–0.600]	0.597 [0.300–0.900]	0.802 [0.500–1.000]

A = 20, 25, 30, 35

Env ID	A=20	A=25	A=30	A=35
E01	0.392 [0.100–0.700]	0.900 [0.700–1.000]	0.899 [0.700–1.000]	1.000 [0.722–1.000]
E02	0.709 [0.400–1.000]	0.804 [0.500–1.000]	0.702 [0.400–1.000]	0.801 [0.600–1.000]
E03	0.896 [0.700–1.000]	1.000 [0.722–1.000]	1.000 [0.722–1.000]	1.000 [0.722–1.000]
E04	1.000 [0.722–1.000]	0.902 [0.700–1.000]	1.000 [0.722–1.000]	1.000 [0.722–1.000]
E05	0.899 [0.700–1.000]	0.904 [0.700–1.000]	1.000 [0.722–1.000]	1.000 [0.722–1.000]
E06	0.899 [0.700–1.000]	0.895 [0.700–1.000]	1.000 [0.722–1.000]	1.000 [0.722–1.000]
E07	0.899 [0.700–1.000]	1.000 [0.722–1.000]	0.899 [0.700–1.000]	1.000 [0.722–1.000]
E08	0.601 [0.300–0.900]	0.902 [0.700–1.000]	0.904 [0.700–1.000]	1.000 [0.722–1.000]
E09	1.000 [0.722–1.000]	1.000 [0.722–1.000]	1.000 [0.722–1.000]	1.000 [0.722–1.000]
E10	1.000 [0.722–1.000]	0.899 [0.700–1.000]	1.000 [0.722–1.000]	1.000 [0.722–1.000]
E11	1.000 [0.722–1.000]	1.000 [0.722–1.000]	1.000 [0.722–1.000]	1.000 [0.722–1.000]
E12	0.902 [0.700–1.000]	1.000 [0.722–1.000]	1.000 [0.722–1.000]	1.000 [0.722–1.000]

Note. Baseline columns fix 3 scouts. RL columns denote attacker/scout counts. Seeds aggregated per cell; CIs via nonparametric bootstrap (see 3.6).

4.2 Aggregate Win Rate Comparisons

Figure 2 summarizes win-rate distributions across team configurations, aggregating over environment cells within each defender split. This view complements the dose-response curves by showing performance variability across conditions.

RL(5/3) demonstrates consistently high performance across both defender configurations, achieving aggregate win rates of approximately 0.95. In contrast, RL(3/5), which reverses the attacker-scout ratio, shows greater variability, particularly under the 3 Det / 5 Int configuration, where the mean win rate drops to approximately 0.68. This suggests that the 5-attacker / 3-scout formation better balances exploitation and reconnaissance for the networks and objectives in our experiments.

Baseline performance improves steadily with team size but remains below RL(5/3) levels until reaching 25-30 attackers. At $A = 5$ (matching RL attacker count), baselines achieve only 0.20-0.22 win rates—far below the 0.93-0.95 achieved by trained

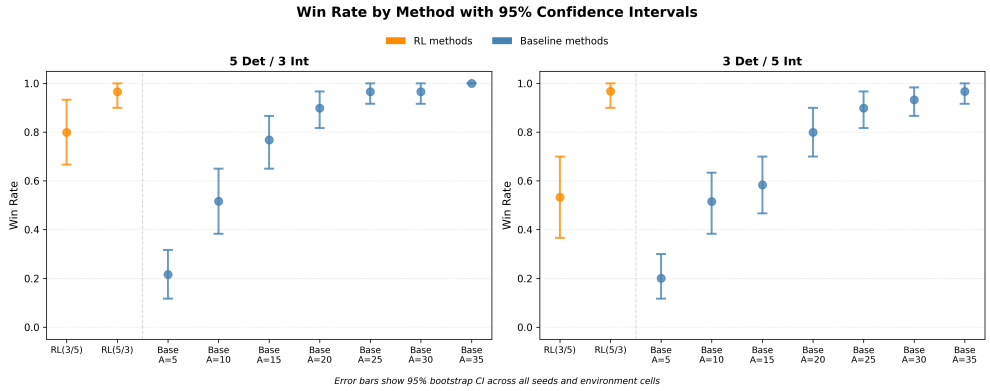


Fig. 2 Win rate by method with 95% confidence intervals. Points show mean win rates aggregated over environment cells and random seeds within each defender configuration. Error bars represent 95% bootstrap confidence intervals. Baseline teams used 3 scouts with varying attacker counts A. Left panel: 5 Det / 3 Int. Right panel: 3 Det / 5 Int. RL(5/3) achieves consistently high performance ($\approx$0.95) while baselines require $A \geq$ 25-30 to approach these levels.

RL(5/3) teams. This gap narrows as baselines add personnel, but substantial team-size differences are required to close it.

These aggregated patterns motivate the subsequent EBTS analysis (Section 4.6), which quantifies exactly how many baseline attackers are needed to match each RL configuration across specific environment cells.

4.3 Time-to-Win: Tactical Speed Conditional on Success

Beyond win probability, we examine how quickly successful attacks are completed. This time-to-win analysis can address tactical efficiency given success, distinguishing between strategic effectiveness (winning) and operational speed (winning quickly).

Figure 3 reveals an interesting trade-off. Baseline teams generally become faster as attacker count increases: under 5 Det / 3 Int (left panel), baselines average roughly 370-400 ticks at A = 5-10 and drop to about 200 ticks by $A = 35$. The overall trend is downward (with some non-monotonic fluctuations), consistent with larger teams applying more parallel pressure on the network.

RL teams show comparatively stable completion times. RL(5/3) averages approximately 403 ticks under 5 Det / 3 Int and 438 ticks under 3 Det / 5 Int. RL(3/5) is somewhat slower at 474-591 ticks. Large baseline teams ($A \geq 30$) finish successful attacks faster than either RL configuration.

However, this analysis conditions on success, answering "how quickly do wins finish?" rather than "how often do they happen?" The baseline's speed advantage at large A must be weighed against the finding from Section 4.1 that achieving comparable win *rates* requires 25-30 baseline attackers versus only 5 RL attackers.

In organizational terms, a 35-person baseline team completes successful attacks in approximately 200-250 ticks, while a 5-person trained RL team requires approximately 400 ticks. However, the baseline requires 5-7$\times$ more personnel to maintain a similar

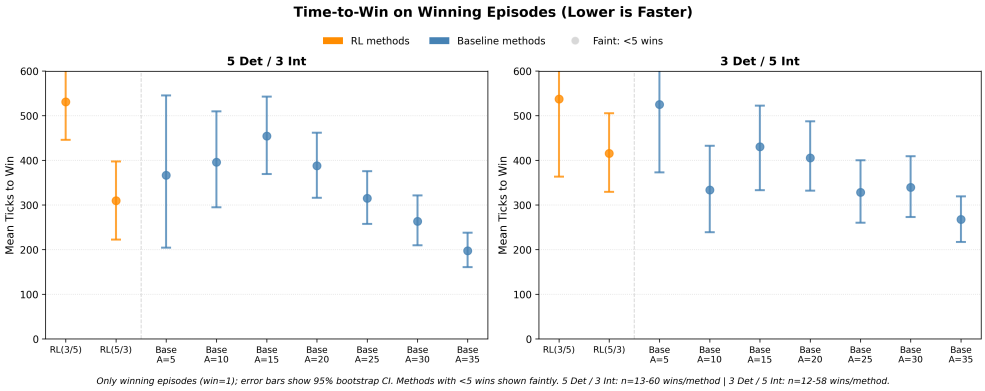


Fig. 3 Time-to-win on winning episodes (lower is faster). Points show mean ticks-to-win computed over successful episodes only; error bars are 95% bootstrap confidence intervals. Left panel: 5 Det / 3 Int. Right panel: 3 Det / 5 Int. Methods with fewer than 5 wins are rendered faintly to indicate estimated uncertainty. Large baseline teams complete successful attacks faster than RL teams, but this is conditional on success; RL achieves comparable win *rates* with far fewer attackers.

success probability. This represents a classic efficiency-versus-speed trade-off: trained teams achieve comparable effectiveness with far fewer people, even though brute-force numerical superiority can accelerate operations, provided that the operation is successful.

4.4 Learning Trajectories Over Training Duration

Research Question 1: How does training duration affect cyber team performance trajectories? We address this question by tracking performance across training checkpoints at 100, 500, 1,000, 2,000, and 5,000 episodes. Figure 4 shows the checkpoint evaluation learning curve for RL(5/3), with each point representing frozen policy performance measured under standardized evaluation conditions.

The learning trajectory exhibits three distinct phases: **Phase 1: Exploration (episodes 100-500, checkpoint evaluation view).** Win rates remain low: 0.00 at 100 episodes for both defender splits, rising only to about 0.07 (3 Det / 5 Int) and 0.15 (5 Det / 3 Int) by 500 episodes. During this phase, agents discover basic mechanics (movement, attack timing, and scout coordination), but robust strategic sequences are not yet stable. The curriculum provides easier opponents during this period (phases 1-2 per Section 3.4), but policies still struggle to win consistently.

Phase 2: Skill acquisition (episodes 500-2,000). Performance increases sharply. At 1,000 episodes, both defender splits reach about 0.48 win rate; by 2,000 episodes, they rise to about 0.53 (3 Det / 5 Int) and 0.57 (5 Det / 3 Int). This phase captures the transition from exploratory behavior to consistently effective execution, including adaptation after the curriculum hardening at episode 750.

Phase 3: Late-stage refinement with diminishing per-episode returns (episodes 2,000-5,000). Performance continues improving to high absolute levels: about 0.93 (3 Det / 5 Int) and 0.95 (5 Det / 3 Int) by 5,000 episodes. Although the

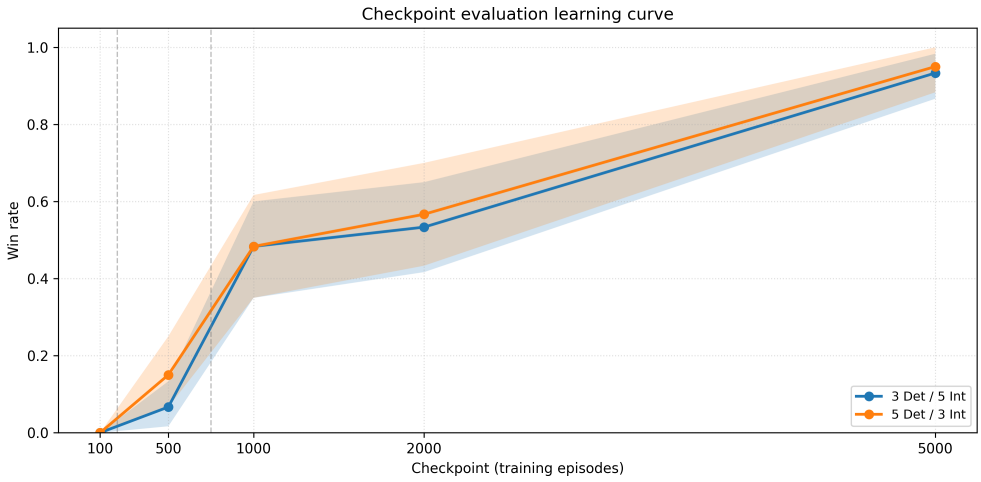


Fig. 4 Checkpoint learning curves over training duration. Win rates for frozen policies were evaluated at 100, 500, 1,000, 2,000, and 5,000 episode checkpoints on the fixed 12-cell benchmark grid (Section 3.5) with learning disabled ($\varepsilon = 0$). Error bars show 95% bootstrap confidence intervals. Dashed vertical lines mark curriculum phase boundaries at episodes 200 and 750. Learning exhibits three distinct phases: near-zero exploration (100-500), rapid improvement (500-2,000), and late-stage refinement with lower marginal gains per episode (2,000-5,000).

absolute gain from 2,000 to 5,000 is large, the gain rate per episode is lower than in the 500-2,000 interval, matching the ROI analysis in Section 4.5.

Figure 5 provides a complementary view using the rolling win rate from the training log. This on-policy curve is noisier and can be higher early because it reflects easier early curriculum settings and live exploration during training; the dip near episode 750 aligns with curriculum hardening. We use this curve to describe training dynamics, while Figure 4 provides the standardized checkpoint evaluation view.

These learning trajectories directly answer Research Question 1: *training duration substantially affects performance, following a characteristic three-phase progression.* Organizations stopping training before 500 episodes capture limited gains (checkpoint win rates near 0.00-0.15). The critical improvement window spans 500-2,000 episodes, where teams transition from exploration to effective execution. Performance continues improving through 5,000 episodes, with lower marginal gain per episode after 2,000.

Figure 6 connects training duration to baseline parity by overlaying dose-response curves with RL references at different checkpoints. As training progresses from 100 to 2,000 episodes, the RL reference band shifts upward and the baseline parity point moves towards the right, illustrating that longer training yields greater efficiency advantages relative to untrained teams.

4.5 Optimal Training Investment

Research Question 2: What training investment yields optimal returns? We computed return on training investment (ROI) in episode space as win-rate gain

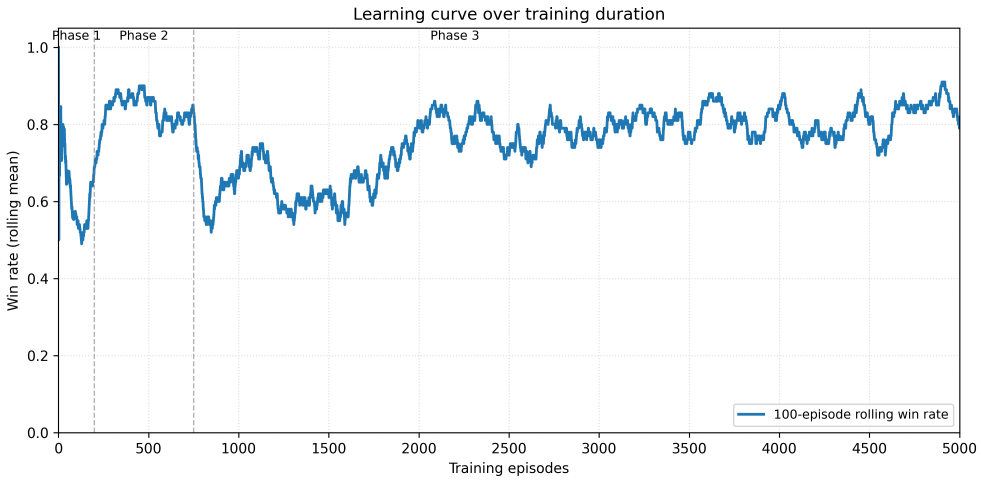


Fig. 5 Training-log learning curve with rolling 100-episode window. Win rate computed from raw on-policy training episodes under the curriculum and exploration settings. Dashed lines mark curriculum phase boundaries. The dip at episode 750 reflects curriculum hardening; subsequent recovery demonstrates policy adaptation to more challenging opponents. Absolute levels are not directly comparable to Figure 4, which uses fixed benchmark evaluation with frozen policies.

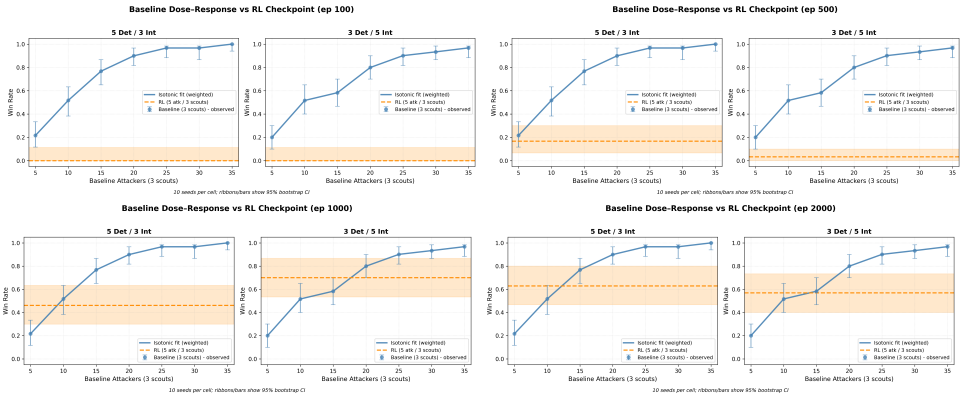


Fig. 6 Baseline dose-response overlays by training checkpoint. Each panel shows baseline performance (blue) versus RL(5/3) at the indicated training episode. At 100 episodes (top-left), RL barely exceeds baseline. By 2,000 episodes (bottom right), substantial separation emerges—RL outperforms baselines with fewer than 20 attackers.

rate across adjacent checkpoint intervals. Figure 7 reports checkpoint ROI directly in episode-based units (percentage points per 100 episodes).

ROI is highest during the 500-1,000 episode interval. This corresponds to approximately 6.7-8.3 percentage points of win-rate gain per 100 episodes, depending on defender configuration. This peak occurs within the skill-acquisition phase identified

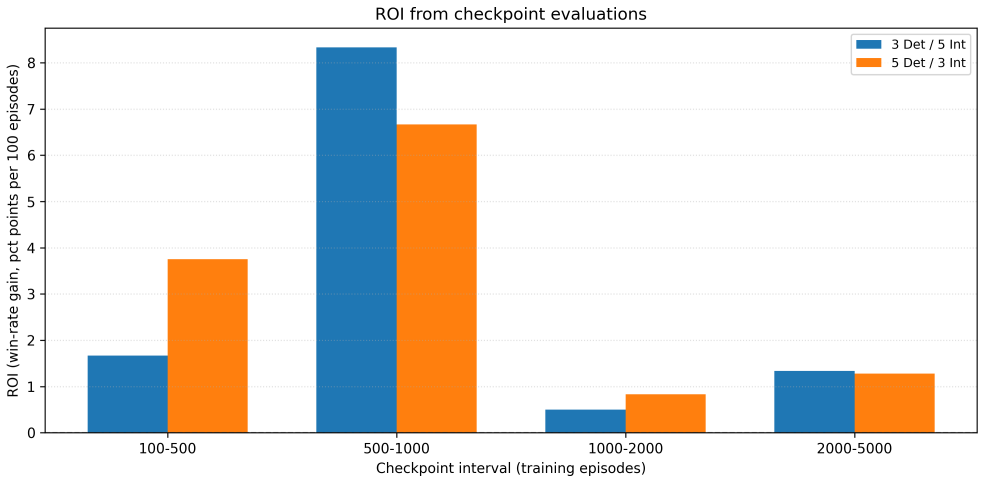


Fig. 7 Return on training investment by interval. Bars show gain rate between adjacent checkpoints (100→500, 500→1,000, 1,000→2,000, 2,000→5,000), grouped by defender configuration. Values are displayed as win-rate percentage-point gain per 100 episodes. The zero line indicates no marginal improvement. Peak ROI occurs at 500-1,000 episodes; returns drop sharply thereafter but remain positive through 5,000 episodes.

in Section 4.4, after policies completed initial exploration but before reaching the performance plateau. Returns decline in subsequent intervals:

- **1,000-2,000 episodes:** ROI drops to approximately 0.50-0.83 percentage points per 100 episodes, an 88-94% reduction from peak.
- **2,000-5,000 episodes:** ROI remains low at approximately 1.28-1.33 percentage points per 100 episodes, still about 81-84% below peak.

Despite diminishing returns, ROI remains positive through 5,000 episodes. Performance continues to improve in the plateau phase, just at slower rates than during skill acquisition. Figure 8 presents the same ROI concept computed from training-log rolling win rates in 200-episode bins. This higher-resolution view reveals greater variability due to on-policy exploration and curriculum transitions. The spike around episode 200 captures the curriculum's first phase transition; the dip at episode 750 reflects curriculum hardening. After episode 750, ROI oscillates around lower levels with occasional rebounds, supporting the checkpoint-based finding of diminishing returns after 1,000 episodes.

Answer to Research Question 2: Optimal training investment occurs at 500-1,000 episodes, where marginal gains per additional episode are highest. At this point, ROI is approximately 6.7-8.3 percentage points of win-rate improvement per 100 episodes. This represents the sweet spot between completing exploration and reaching the performance plateau. For organizations with limited training budgets:

- **Minimum threshold:** Train for at least 500 episodes to exit the exploration phase. Stopping earlier forfeits the high-yield improvement period.

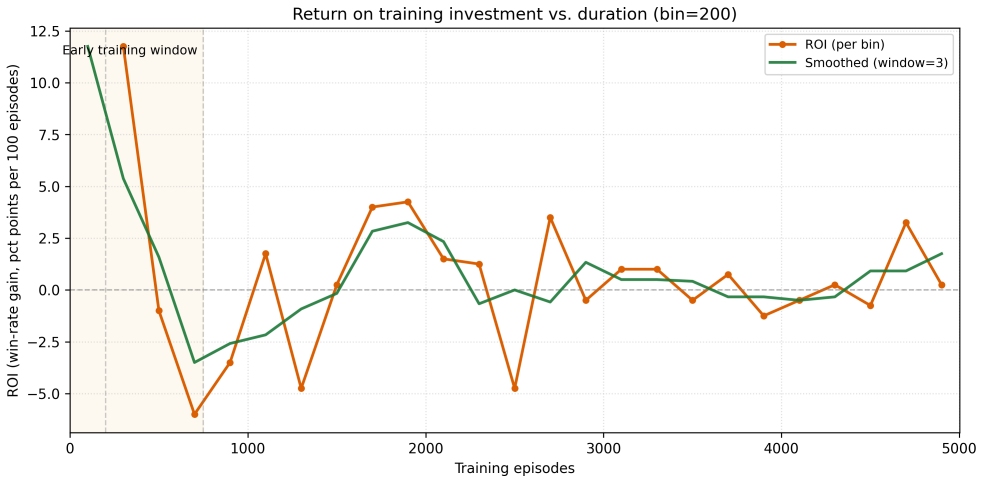


Fig. 8 Training-log ROI with 200-episode bins and smoothing. ROI computed from rolling win-rate series and smoothed across adjacent bins. The shaded region marks the easier early curriculum (episodes 1-750). High variability reflects on-policy exploration dynamics; the general declining trend aligns with checkpoint-based analysis.

- **Recommended target:** Train for 1,000 episodes to capture peak ROI while avoiding the steepest part of the diminishing-returns curve.
- **Extended training:** Training to 2,000 episodes provides additional gains, but at low ROI (about 0.8-1.4 points/episode).
- **Sustained training:** Organizations with ample resources can continue training to 5,000 episodes for maximum absolute performance, though the incremental benefit per hour is small (0.8-1.3 points/episode).

4.6 Trained Versus Untrained Team Effectiveness

Research Question 3: How do trained small teams compare to larger untrained teams? We quantify this comparison through Equivalent Baseline Team Size (EBTS): the number of baseline attackers $\hat{N}$ required to match a trained RL team's win rate. Figure 9 shows $\hat{N}$ for each environment cell and RL configuration.

RL(5/3) teams demonstrate substantial equivalence advantages. Across environment cells, $\hat{N}$ typically ranges from 15 to 35 baseline attackers. In several cells (marked with upward triangles), $\hat{N}$ exceeds 35, our maximum tested baseline size, indicating that baselines cannot match RL(5/3) performance within the evaluated range. RL(3/5) shows smaller but still notable advantages. Most cells require $\hat{N}$ = 10-25 baseline attackers, with fewer instances exceeding the tested range. This aligns with the aggregate win-rate analysis (Section 4.2), which shows that RL(3/5) achieves lower absolute performance than RL(5/3). Figure 10 translates these equivalences into efficiency multipliers $k = \hat{N}/A_{\mathrm{RL}}$, representing how many baseline attackers are needed per trained RL attacker. Values $k > 1$ indicate RL advantages.

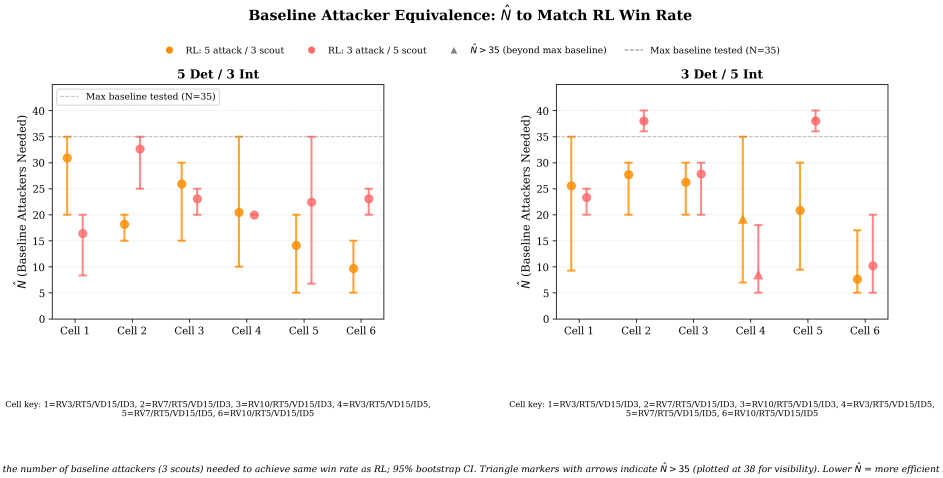


Fig. 9 Baseline attackers needed to match RL win rate ($\hat{N}$). Points show $\hat{N}$ with 95% bootstrap confidence intervals for each environment cell. Upward triangles indicate $\hat{N} > 35$ (beyond maximum tested baseline size). RL(5/3) typically requires 15-35 baseline attackers to match its performance; several cells exceed our tested range. RL(3/5) requires fewer baseline attackers on average but still demonstrates efficiency advantages.

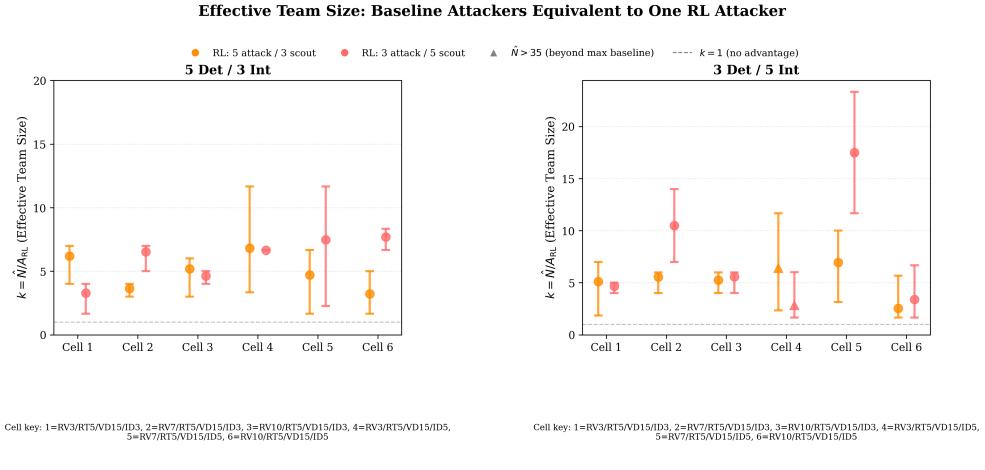


Fig. 10 Efficiency multiplier $k = \hat{N}/A_{\mathbf{RL}}$: baseline attackers per RL attacker. Error bars show 95% bootstrap confidence intervals. Dashed horizontal line marks $k = 1$ (no advantage). Upward triangles indicate $\hat{N}$ beyond the maximum tested baseline ($A = 35$), where k is computed using $N_{\max} = 35$ as a lower bound. Most cells show $k \approx$ 3-7, meaning each trained RL attacker performs equivalently to 3-7 untrained baseline attackers.

Across environment cells, efficiency multipliers typically range from $k \approx 3$ to $k \approx 8$. For RL(5/3):

- **Typical advantage:** Most cells exhibit k = 3-6, meaning each of the 5 trained attackers performs equivalently to 3-6 untrained baseline attackers.
- **Strong advantage:** Several cells reach k = 7-8, particularly in challenging defensive scenarios.
- **Beyond range:** A few cells exceed our tested baseline capacity ($k > 7$ based on $\hat{N} > 35$), suggesting even larger advantages.

While RL(3/5) has lower absolute win rates, its EBTS parity factors are generally comparable to RL (5/3) (k = 3-6), with broader observed range up to about 9 in some cells, reflecting the formation's reduced aggregate win rate while still demonstrating training advantages over heuristic baselines. Table 3 provides detailed EBTS calculations for selected environment cells, showing both $\hat{N}$ and k with confidence intervals alongside RL and baseline win rates.

Table 3 Equivalence Summary: RL Performance and Baseline-Equivalent Team Size

Defenders	Routers	RV%	RT%	VD%	ID%	RL Form	RL Win Rate	$\hat{N}$	k
3 Det / 5 Int	60	3	5	15	3	RL(3/5)	0.807 [0.500-1.000]	25.5 [20.0-35.0]	8.50 [6.67-11.67]
3 Det / 5 Int	60	3	5	15	3	RL(5/3)	0.802 [0.500-1.000]	24.9 [18.0-35.0]	4.97 [3.60-7.00]
3 Det / 5 Int	60	3	5	15	5	RL(3/5)	0.496 [0.200-0.800]	12.0 [5.0-32.0]	4.01 [1.67-10.67]
3 Det / 5 Int	60	3	5	15	5	RL(5/3)	1.000 [0.722-1.000]	>35	>7.0
3 Det / 5 Int	60	7	5	15	3	RL(3/5)	0.602 [0.300-0.900]	10.9 [5.0-21.7]	3.65 [1.67-7.22]
3 Det / 5 Int	60	7	5	15	3	RL(5/3)	1.000 [0.722-1.000]	23.3 [20.0-25.0]	4.66 [4.00-5.00]
3 Det / 5 Int	60	7	5	15	5	RL(3/5)	0.706 [0.400-1.000]	12.8 [5.0-20.0]	4.26 [1.67-6.67]
3 Det / 5 Int	60	7	5	15	5	RL(5/3)	0.900 [0.700-1.000]	19.5 [9.4-30.0]	3.90 [1.88-6.00]
3 Det / 5 Int	60	10	5	15	3	RL(3/5)	0.905 [0.700-1.000]	23.2 [16.2-30.0]	7.72 [5.42-10.00]
3 Det / 5 Int	60	10	5	15	3	RL(5/3)	0.896 [0.700-1.000]	22.8 [17.1-30.0]	4.57 [3.43-6.00]
3 Det / 5 Int	60	10	5	15	5	RL(3/5)	0.606 [0.300-0.900]	15.1 [5.0-25.0]	5.02 [1.67-8.34]
3 Det / 5 Int	60	10	5	15	5	RL(5/3)	1.000 [0.722-1.000]	27.8 [20.0-30.0]	5.56 [4.00-6.00]
5 Det / 3 Int	60	3	5	15	3	RL(3/5)	0.905 [0.700-1.000]	23.5 [13.6-35.0]	7.82 [4.54-11.67]
5 Det / 3 Int	60	3	5	15	3	RL(5/3)	1.000 [0.722-1.000]	30.9 [20.0-35.0]	6.19 [4.00-7.00]
5 Det / 3 Int	60	3	5	15	5	RL(3/5)	0.902 [0.700-1.000]	26.5 [9.5-35.0]	8.82 [3.18-11.67]
5 Det / 3 Int	60	3	5	15	5	RL(5/3)	1.000 [0.722-1.000]	32.6 [25.0-35.0]	6.53 [5.00-7.00]
5 Det / 3 Int	60	7	5	15	3	RL(3/5)	0.794 [0.500-1.000]	16.2 [9.2-20.0]	5.41 [3.06-6.67]
5 Det / 3 Int	60	7	5	15	3	RL(5/3)	0.904 [0.700-1.000]	18.2 [10.0-20.0]	3.63 [2.00-4.00]
5 Det / 3 Int	60	7	5	15	5	RL(3/5)	0.605 [0.300-0.900]	12.1 [8.6-15.0]	4.02 [2.86-5.00]
5 Det / 3 Int	60	7	5	15	5	RL(5/3)	0.899 [0.700-1.000]	16.2 [13.0-20.0]	3.23 [2.60-4.00]
5 Det / 3 Int	60	10	5	15	3	RL(3/5)	0.796 [0.500-1.000]	13.3 [5.0-15.0]	4.45 [1.67-5.00]
5 Det / 3 Int	60	10	5	15	3	RL(5/3)	1.000 [0.722-1.000]	18.2 [15.0-20.0]	3.64 [3.00-4.00]
5 Det / 3 Int	60	10	5	15	5	RL(3/5)	0.896 [0.700-1.000]	18.9 [11.2-25.0]	6.29 [3.75-8.33]
5 Det / 3 Int	60	10	5	15	5	RL(5/3)	0.895 [0.700-1.000]	18.7 [8.8-25.0]	3.74 [1.75-5.00]

Bootstrapped mean with 95% confidence intervals across 10 seeds per environment cell. $\hat{N}$ is the baseline attacker count matching RL win rate; $k = \hat{N}/A_{\text{RL}}$ is the effective team size multiplier. > 35 indicates baseline never reaches RL performance within tested range.

Answer to Research Question 3: Trained teams with 5 attackers achieve performance levels requiring 15-35 untrained attackers—a 3-7× efficiency advantage in most scenarios. This provides quantitative evidence for organizational decisions between training investments and recruitment strategies.

In practical terms:

- A trained RL team of 5 attackers + 3 scouts (8 total personnel) performs comparably to an untrained baseline team of 25-30 attackers + 3 scouts (28-33 total personnel).
- Organizations can either recruit 20-25 additional personnel for basic heuristic strategies, or invest in a full 5,000-episode training program to achieve equivalent effectiveness with existing staff.
- The choice depends on training costs versus hiring costs, but our simulations demonstrate that sustained training provides substantial manpower efficiency gains.

4.7 Summary of Findings

Our computational experiments provide three key results: **1. Training duration affects performance nonlinearly through distinct phases (Research Question 1).** Learning follows a three-phase trajectory: minimal gains during early exploration (100-500 episodes, win rates near 0.00-0.15), rapid improvement during skill acquisition (500-2,000 episodes, rising to about 0.53-0.57), and continued gains with lower per-episode returns during late-stage refinement (2,000-5,000 episodes, reaching about 0.93-0.95). Organizations stopping training before 500 episodes forfeit high-yield improvements; extending training beyond 2,000 episodes yields progressively smaller marginal benefits per episode.

2. Optimal training investment occurs at 500-1,000 episodes (Research Question 2). Return on investment peaks in the 500-1,000 episode interval at approximately 6.7-8.3 percentage points of win-rate gain per 100 episodes. Returns then drop by about 88-94% in the 1,000-2,000 interval and remain low (about 81-84% below peak) in 2,000-5,000, though performance continues improving throughout. Organizations should target at least 1,000 episodes to capture peak efficiency gains while avoiding the steepest part of the diminishing returns curve.

3. Trained teams achieve 3-7× efficiency advantages over untrained teams (Research Question 3). Teams with 5 trained attackers perform equivalently to baseline teams with 15-35 untrained attackers across environment conditions. This efficiency advantage provides quantitative support for training-focused workforce strategies: a full 5,000-episode training program for a team of 5 attackers yields performance comparable to recruiting 20-25 additional personnel operating on basic heuristics. Together, these findings demonstrate that sustained training substantially improves cyber team effectiveness, with the highest returns occurring in the 500-2,000 episode window and sustained benefits through 5,000 episodes. The magnitude of training effects enables small trained teams to outperform much larger untrained teams, suggesting that organizations should prioritize skill development investments alongside or ahead of workforce expansion strategies.

5 Discussion

This study extends the Collaborative Cyber Team Formation framework to examine temporal factors in cyber team performance. Building on (Calay et al., 2023)'s team structure analysis, (Kpieleh et al., 2025), and (Qolomany et al., 2024)'s comprehensive theoretical model, this paper adds training duration as a critical performance factor. The computational model demonstrates that training duration influences team effectiveness in cyber competitions.

5.1 Key Findings and Organizational Implications

Our results reveal three distinct patterns with direct organizational implications. First, learning follows a three-phase trajectory: minimal gains during early exploration, rapid improvement during skill acquisition, and continued but diminishing gains during refinement. This pattern suggests organizations should plan training programs that extend beyond initial exploration phases to capture high-yield improvement periods.

Second, return on investment analysis identifies optimal training windows. Peak efficiency occurs mid-training, with substantially higher performance gains compared to early or late training phases. Organizations that terminate training too early miss this high-efficiency period, while those that train indefinitely beyond peak ROI experience diminishing marginal returns.

Third, trained teams achieve performance levels requiring substantially larger untrained teams. This efficiency advantage has direct implications for workforce development strategies. Organizations face a choice between deploying larger teams using basic heuristic approaches or investing in training to develop adaptive strategies in smaller teams. The computational model provides evidence that training investment can substitute for team size, with the efficiency advantage becoming substantial after teams complete mid-training phases.

5.2 Theoretical Contributions

These findings extend the frameworks developed by (Calay et al., 2023) and (Qolomany et al., 2024) by demonstrating that temporal factors, accumulated training duration, operate through distinct mechanisms compared to structural factors. Team structure, size, and initial skills represent starting conditions that influence baseline performance levels. Training duration represents a developmental process that shapes performance trajectories over time. The interaction between these factor types creates performance dynamics that single-round simulations cannot capture.

The use of reinforcement learning as a modeling tool enables the simulation of strategy development that occurs when teams train together repeatedly, providing insights into learning patterns that would require months or years to observe in real operational teams. This methodological contribution distinguishes our approach from prior work that treated team capabilities as static. By incorporating temporal dynamics into agent-based modeling, we demonstrate a computational approach for investigating questions that would be impractical to address through longitudinal human subject experiments.

5.3 Limitations and Validity Considerations

The study has several limitations that suggest directions for future research. First, the model simulates cyber competition scenarios with defined attack and defense protocols. Real-world cybersecurity operations involve greater complexity, ambiguity, and stress. Validating these findings with operational security teams would strengthen confidence in their applicability beyond competition contexts.

Second, the learning curves observed in this study are partially shaped by curriculum design. The training protocol used a progressive curriculum with phase boundaries at episodes 200 and 750, transitioning from very easy to easy to mixed/hard difficulty regimes. Performance shows a temporary decline at the curriculum transition (750 episodes) before recovering. Future research should examine learning trajectories under fixed-difficulty training to isolate team development patterns independent of curriculum effects. This would clarify whether the observed performance patterns reflect intrinsic learning dynamics or are artifacts of the training protocol design.

Third, the model focuses on team-level factors without examining individual learning rates or skill development. Future research should examine the influence of individual variations in learning capacity on team-level performance trajectories. Some team members may reach their skill ceiling earlier than others, creating asymmetries that influence overall team performance.

Fourth, this study did not explicitly test the impact of team consistency: whether teams that train with the same members consistently outperform teams with frequent member changes. While our training protocol maintained fixed agent roles throughout training, we did not compare this approach with teams with shuffled compositions. Future work should directly test consistency effects by training multiple teams with varying levels of member stability and comparing their learning trajectories.

Fifth, the model examines training in isolation from other organizational factors. In practice, team performance depends on management support, organizational culture, resource availability, and work-life balance. Future research should examine how these contextual factors moderate the relationship between training duration and performance.

Sixth, the model uses fixed adversary strategies represented by the heuristic defender policies. Future work should examine how training effectiveness changes when adversaries also learn and adapt their approaches over time, creating an arms race dynamic between attackers and defenders.

5.4 Future Research Directions

Several promising research directions emerge from this work. First, investigating mixed experience levels within teams would address questions about optimal composition. How many trained members does a team need to accelerate learning for untrained members? Can one highly trained agent compensate for multiple minimally trained agents, or do teams need a minimum ratio of trained to untrained personnel? The model can simulate these scenarios to identify optimal composition strategies.

Second, examining transfer learning across different network topologies would test whether training in one environment prepares teams for different contexts. If teams

train on simple networks, does their performance transfer to complex networks? Does each network configuration require topology-specific training? Understanding transfer effects would help organizations design efficient training programs that prepare teams for diverse operational scenarios.

Third, investigating purple team training, where teams practice both offensive and defensive roles, would address questions about cross-functional skill development. Does training in both roles enhance defensive performance by providing insight into attacker strategies? Or does it create confusion by requiring teams to switch between opposing mindsets? Computational modeling can test these hypotheses before organizations invest in purple team programs.

Fourth, exploring automated training systems that adapt to team performance would leverage the model's insights about return on investment patterns. If the model can predict when teams are in high-efficiency learning phases versus diminishing-returns phases, training systems could automatically adjust difficulty levels, introduce new challenges, or shift focus areas. This adaptive approach could enhance training efficiency by maximizing time spent in high ROI training regimes.

Fifth, examining the economic implications of training investment would connect the technical findings to organizational decision-making. What is the monetary value of improving the win rate by a specific percentage? How much should organizations spend on training to achieve target performance levels? Developing economic models that translate performance gains into financial returns would help justify training investments to organizational leadership.

6 Conclusion

Security operations centers face critical resource allocation decisions when building cyber defense capabilities: invest in training existing personnel or recruit additional staff? This study provides computational evidence to inform these decisions by quantifying the effects of training duration on team performance and comparing trained teams to larger untrained teams in terms of effectiveness.

Our agent-based modeling experiments demonstrate three key findings. First, training duration substantially affects cyber team performance through a characteristic three-phase learning trajectory: with teams progressing from exploration, skill acquisition, and then to a performance plateau. Second, optimal training investment occurs during mid-training phases where return on investment peaks before diminishing returns set in. Third, trained teams achieve efficiency advantages of 3-7$\times$ over untrained teams, with small trained groups matching or exceeding the performance of substantially larger untrained groups.

These findings have direct implications for organizational practice. The nonlinear relationship between training duration and performance suggests that abbreviated training programs, which terminate during exploration phases, capture minimal benefits, while extended programs continuing into plateau phases experience diminishing marginal returns. Organizations should design training programs that reach mid-training efficiency windows to maximize return on investment. The substantial efficiency advantages observed for trained teams, where five trained attackers

match the effectiveness of 25-30 untrained attackers, provide quantitative support for training-focused workforce strategies over recruitment-focused expansion.

Beyond immediate practical applications, this work demonstrates a methodological contribution: reinforcement learning can serve as a modeling tool to simulate human team learning trajectories that would require months or years to observe through longitudinal studies. By compressing extensive team development processes into computationally tractable simulations, this approach enables the investigation of workforce planning questions that human subject experiments cannot practically address. The computational framework we establish can be extended to incorporate additional temporal dynamics, skill heterogeneity, team stability, multi-scenario transfer, and adaptive adversaries, creating opportunities for evidence-based workforce planning research across cybersecurity and other high-performance team domains.

The central insight of this research is that temporal factors in team development operate through fundamentally different mechanisms than structural factors. Team size and composition establish baseline capabilities, yet sustained training shapes performance trajectories over time. Organizations that recognize this distinction can make more informed decisions about where to invest resources: expanding workforce for immediate capacity or developing expertise for long-term effectiveness. Our computational evidence suggests the efficiency gains from sustained training warrant serious consideration in organizational workforce planning, even when accounting for the costs and complexities of implementing structured training programs. As cybersecurity threats continue to evolve in sophistication, the ability to develop highly effective small teams through targeted training may prove more sustainable than attempting to match adversary capabilities through workforce expansion alone.

Funding Statement

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Appendix A Terminology and RL Reproducibility

This appendix provides terminology definitions and full RL implementation details referenced in Section 3.

A.1 Terminology Definitions

Table A1 defines the terms used consistently throughout the manuscript. To prevent ambiguity, we use the following terminology consistently throughout the paper.

Table A1 Terminology used in this study.

Term	Definition in this paper
Training episode	One complete computational simulation run from initialization to attacker win or tick cutoff ($T_{\max} = 1000$).
Training duration	Number of RL training episodes completed (e.g., 100, 500, 1,000, 2,000, 5,000).
RL training	Computational policy optimization of Rainbow DQN parameters across episodes.
Human-team training (analogy)	Conceptual interpretation of RL learning phases as a proxy for human team development; this is an analogy, not a calibrated real-time equivalence.
Baseline / heuristic / untrained	Same condition in this paper: fixed hand-crafted rules with no parameter updates and no cross-episode learning from experience.
Checkpoint evaluation	Standardized offline evaluation of frozen policies on the fixed benchmark grid with learning disabled ($\varepsilon = 0$).
Training-log curve	On-policy learning signal computed from training episodes under curriculum and exploration settings.

A.2 Rainbow DQN Architecture

Table A2 lists the attacker/scout network architecture used in all reported runs.

Table A2 Neural network architecture used by both Rainbow agents.

Component	Structure	Notes
Input state size	Attacker: $13 + 9N + 14$; Scout: $13 + 9N + 16$	N = number of routers (60 in all reported experiments)
Action space	$2N + 1$ discrete actions	N move actions, N send actions, 1 stay action
Trunk	Linear(state_dim, 128) → ReLU → Linear(128, 128) → LayerNorm(128) → ReLU	Shared feature extractor
Value stream	NoisyLinear(128, 128) → ReLU → NoisyLinear(128, 51)	Distributional value logits
Advantage stream	NoisyLinear(128, 128) → ReLU → NoisyLinear(128, action_dim × 51)	Distributional advantage logits
Distributional support	51 atoms over $[v_{\min}, v_{\max}] = [-100, 200]$	Categorical projection update
Noisy layers	Factorized NoisyLinear, std_init = 0.5	Exploration in value and advantage heads

A.3 Rainbow DQN Hyperparameters

Table A3 provides the complete training configuration for reproducibility.

Table A3 Complete Rainbow DQN hyperparameters for reproducibility.

Setting	Attacker policy	Scout policy	Implementation detail
Optimizer	Adam	Adam	Learning rate $= 2 \times 10^{-4}$
Discount factor γ	0.99	0.97	Applied in n-step return construction
Replay capacity	50,000	50,000	FIFO replacement
Batch size	256	256	PER sampling each gradient step
PER α	0.6	0.6	Priority exponent
PER β (initial)	0.4	0.4	Importance-sampling correction
PER β increment/episode	0.001	0.004	Annealed by `next_episode()`
Priority ϵ	10^{-6}	10^{-6}	Added to absolute per-sample loss
Priority cap	1e6	none	Attacker priority clipping for stability
n-step horizon	up to 6	up to 6	Flush when $n \geq 6$ or terminal
Distributional atoms	51	51	Support $[-100, 200]$
Target sync cadence	200 updates; 500 after replay $>$ 10,000	200 updates; 500 after replay $>$ 10,000	Soft update every cadence trigger
Soft update rate τ	0.005	0.005	Applied to target parameters
Gradient handling	NaN-to-zero + clip-norm 10.0	NaN-to-zero + clip-norm 10.0	Before optimizer step
State normalization	StandardScaler	StandardScaler	Fitted after 1,500 buffered states
Exploration schedule (training)	same	same	Episodes < 10: $\varepsilon = 1.0$; 10–509: $\varepsilon = 0.05$; ≥ 510: $\varepsilon = 0.01$
Policy update frequency	every 10 ticks	every 4 ticks	Triggered in NetLogo training loop
Checkpointing cadence	every 50 episodes + final save	every 50 episodes + final save	Main checkpoint plus timestamped backup
Training team regime (paper checkpoints)	up to 4 agents/-side	up to 4 agents/-side	Initial-training setting used for paper checkpoints
Training role splits (paper checkpoints)	sampled within 4-agent cap	sampled within 4-agent cap	Representative attacker/scout split includes 3 attackers + 1 scout; total capped at 4 per side during training

A.4 Reward Coefficients and Conditions

Table A4 lists the complete reward specification used in training for attacker and scout policies, including coefficient values, event conditions, and terminal bonuses. These values are the exact implementation settings used in the reported experiments.

Table A4 Complete reward function coefficients and trigger conditions used during RL training.

Term	Coefficient/value	Condition
Tick cost (attackers and scouts)	-0.01 per tick	Applied every tick to all attacker-side agents
Idle streak penalty	$-0.5 \times$ stay_streak	Stay action, both attacker and scout policies
Repeat move penalty	$-1.5 \times$ repeat_count	If same move target appears at least twice in recent window
Backtrack penalty	-2	Immediate A→B→A move pattern
Move-heavy window penalty	$-0.2 \times (\text{moves}_6 - 3)$	If more than 3 moves in last 6 actions
Inward move bonus	$+2$	Move to a router closer to center (both roles)
Oscillation penalty	-4	Revisiting same router within 3 ticks (both roles)
Move action cost	-0.2	Charged on move execution (both roles)
Scout send: unknown-state bonus	$+3$	Scout sends to router marked "unchecked"
Scout send: duplicate send penalty	$-2 \times$ sent_count	Multiple scout sends to same target in current tick
Scout send: first-send bonus	$+2$	First send to target in current tick
Scout send: center-neighbor bonus	$+3$	Target adjacent to center router
Scout send: recency penalty	-2	Same target sent within last 3 ticks
Scout freshness shaping	$+\left(1 - \frac{\text{last_rep}}{\tau_f}\right)$ or $-\left(2 + \frac{\text{overshoot}}{\tau_f}\right)$	Uses freshness threshold τ_f
Scout infection-neighbor bonus	$+6$	Target adjacent to recently infected router and currently unchecked
Attacker send: stale-awareness penalty	-2	Last scout report older than freshness threshold
Attacker send: fresh-awareness bonus	$+\left(1 - \frac{\text{last_rep}}{\tau_f}\right)$	If report age within freshness threshold
Attacker send: vulnerable target bonus	$+4$	Target marked vulnerable in shared awareness
Attacker send: clean target penalty	-2	Target marked clean in shared awareness
Attacker send: center-neighbor bonus	$+2$	Target adjacent to center router
Attacker send: recent-infection-neighbor bonus	$+8$	Target adjacent to recent infection and marked vulnerable
Failed attack penalty (invalid target)	-5 or -3	-5 if offline/already infected; -3 if not vulnerable
Successful infection reward	$+25$	Router successfully infected
Center-compromise bonus	additional $+25$	If infected router is designated center
Scout assist bonus	+scout_infection_assist_bonus (default 2)	Scout reported same router shortly before infection
Scout terminal coverage bonus	$+30 \times$ coverage_fraction	Added at episode end for scout packet transitions
Team terminal bonus (attackers)	$+20$ win / -20 loss	Terminal synthetic transition
Team terminal bonus (scouts)	$+10$ win / -10 loss	Terminal synthetic transition

References

Balázs, P.H., & Csaba, K. (2017). Prerequisites of virtual teamwork in security operations centers: knowledge, skills, abilities and other characteristics. *AARMS–Academic and Applied Research in Military and Public Management Science*, *16*(3), 73–92,

Bonny, J.W. (2022). Using collective metrics to assess team dynamics and performance in esports. *International Journal of Gaming and Computer-Mediated Simulations (IJGCMS)*, *14*(1), 1–13,

Burke, C.S., Stagl, K.C., Salas, E., Pierce, L., Kendall, D. (2006). Understanding team adaptation: A conceptual analysis and model. *Journal of Applied Psychology*, *91*(6), 1189,

Calay, T.J., Qolomany, B., Mulahuwaish, A., Hossain, L., Abdo, J.B. (2023). Cctfv1: Computational modeling of cyber team formation strategies. *International conference on social computing, behavioral-cultural modeling and prediction and behavior representation in modeling and simulation* (pp. 199–208).

Dasgupta, P., & Khazanchi, D. (2019). A unified approach integrating human shared mental models with intelligent autonomous team formation for crisis management. *Iscram* (pp. 1012–1020).

Dobson, G.B. (2019). The measures of cyber team performance. *International Conference on Social Computing, Behavioral-Cultural Modeling, & Prediction and Behavior Representation in Modeling and Simulation (SBP-BRiMS)*, ,

Dobson, G.B. (2022). *Cyber-forces, interactions, terrain: An agent-based framework for simulating cyber team performance* (Unpublished doctoral dissertation). Carnegie Mellon University.

Dobson, G.B., & Carley, K.M. (2017). Cyber-fit: an agent-based modelling approach to simulating cyber warfare. *International conference on social computing, behavioral-cultural modeling and prediction and behavior representation in modeling and simulation* (pp. 139–148).

Dobson, G.B., & Carley, K.M. (2020). Towards agent validation of a military cyber team performance simulation. *International conference on social computing, behavioral-cultural modeling and prediction and behavior representation in modeling and simulation* (pp. 182–191).

Dobson, G.B., & Carley, K.M. (2021). Cyber-fit agent-based simulation framework version 4. *Center for the Computational Analysis of Social and Organizational Systems*, ,

EC-Council (2024). *What is ethical hacking?* Available online: https://www.eccouncil.org/cybersecurity/what-is-ethical-hacking/. ((accessed on 18 February 2026))

Fung, L., Boet, S., Bould, M.D., Qosa, H., Perrier, L., Tricco, A., ... Reeves, S. (2015). Impact of crisis resource management simulation-based training for interprofessional and interdisciplinary teams: a systematic review. *Journal of interprofessional care*, *29*(5), 433–444,

GEOFFREY, D., & KATHLEEN, C. (2025). Did the cyber team win? *International conference on cyber warfare and security* (Vol. 20, pp. 62–70).

Hackman, J.R. (2002). *Leading teams: Setting the stage for great performances.* Boston, MA: Harvard Business Press.

Hughes, A.M., Gregory, M.E., Joseph, D.L., Sonesh, S.C., Marlow, S.L., Lacerenza, C.N., ... Salas, E. (2016). Saving lives: A meta-analysis of team training in healthcare. *Journal of applied psychology*, *101*(9), 1266,

Ilgen, D.R., Hollenbeck, J.R., Johnson, M., Jundt, D. (2005). Teams in organizations: From input-process-output models to imoi models. *Annual Review of Psychology*, *56*, 517–543,

Johnston, J.H., Phillips, H.L., Milham, L.M., Riddle, D.L., Townsend, L.N., DeCostanza, A.H., ... Fitzhugh, S.M. (2019). A team training field research study: extending a theory of team development. *Frontiers in Psychology*, *10*, 1480,

Kozlowski, S.W.J., & Ilgen, D.R. (2006). Enhancing the effectiveness of work groups and teams. *Psychological Science in the Public Interest*, *7*(3), 77–124,

Kpieleh, F., Abdo, J.B., Mulahuwaish, A., Qolomany, B., Hossain, L. (2025). *Cyber team performance: A systematic literature review.* Authorea.

La Fleur, C., Hoffman, B., Gibson, C.B., Buchler, N. (2021). Team performance in a series of regional and national us cybersecurity defense competitions: Generalizable effects of training and functional role specialization. *Computers & security*, *104*, 102229,

Marks, M.A., Mathieu, J.E., Zaccaro, S.J. (2001). A temporally based framework and taxonomy of team processes. *Academy of Management Review*, *26*(3), 356–376,

Mathieu, J.E., Gallagher, P.T., Domingo, M.A., Klock, E.A. (2019). Embracing complexity: Reviewing the past decade of team effectiveness research. *Annual Review of Organizational Psychology and Organizational Behavior*, *6*(1), 17–46,

Mulahuwaish, A., Qolomany, B., Gyorick, K., Abdo, J.B., Aledhari, M., Qadir, J., ... Al-Fuqaha, A. (2025). A survey of social cybersecurity: Techniques for attack detection, evaluations, challenges, and future prospects. *Computers in Human Behavior Reports*, *18*, 100668,

Qolomany, B., Calay, T.J., Hossain, L., Mulahuwaish, A., Bou Abdo, J. (2024). Cctfv2: Modeling cyber competitions. *Entropy*, *26*(5), 384, https://doi.org/10.3390/e26050384

Salas, E., Sims, D.E., Burke, C.S. (2005). Is there a "big five" in teamwork? *Small Group Research*, *36*(5), 555–599,

Santos, C.M., Uitdewilligen, S., Passos, A.M. (2015). A temporal common ground for learning: The moderating effect of shared mental models on the relation between team learning behaviours and performance improvement. *European Journal of Work and Organizational Psychology*, *24*(5), 710–725, https://doi.org/10.1080/1359432x.2015.1049158

Shah, A., Ganesan, R., Jajodia, S., Cam, H., Hutchinson, S. (2023). A novel team formation framework based on performance in a cybersecurity operations center. *IEEE Transactions on Services Computing*, *16*(4), 2359–2371,

Tilbury, J., & Flowerday, S. (2024). Humans and automation: Augmenting security operation centers. *Journal of Cybersecurity and Privacy*, *4*(3), 388–409,

Tuckman, B.W. (1965). Developmental sequence in small groups. *Psychological Bulletin*, *63*(6), 384–399,

Weaver, S.J., Dy, S.M., Rosen, M.A. (2014). Team-training in healthcare: a narrative synthesis of the literature. *BMJ quality & safety*, *23*(5), 359–372,

Wilensky, U. (1999). *Netlogo* (Tech. Rep.). Evanston, IL: Center for Connected Learning and Computer-Based Modeling, Northwestern University. Retrieved from http://ccl.northwestern.edu/netlogo/

Woolley, A.W., Chabris, C.F., Pentland, A., Hashmi, N., Malone, T.W. (2010). Evidence for a collective intelligence factor in the performance of human groups. *science*, *330*(6004), 686–688,

Zhao, N., Chong, H.-Y., Li, Q. (2023). Agent-based modelling of helping behaviour diffusion in project teams as an evolutionary process. *Journal of Simulation*, *17*(3), 279–296, https://doi.org/10.1080/17477778.2021.1997342