CYBERWORLD: WORLD MODELS FOR SAMPLE-EFFICIENT AUTONOMOUS CYBER DEFENSE

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep reinforcement learning has become a prominent approach to autonomous cyber defense. Existing methods are predominantly model-free and consequently require extensive environment interaction. World models provide an alternative by learning predictive dynamics and optimizing policies through imagined trajectories, yielding substantial gains in sample efficiency in robotics and embodied control. Extending this paradigm to cybersecurity raises a fundamental question: what should constitute the "world" in a cyber world model? We introduce CyberWorld, a Dreamer-style world modeling framework that learns latent cyber dynamics from vector, graph, textual, and multimodal representations of the defended network. Across all four scoreable CyberWheel attack strategies, the graph-based CyberWorld variant exceeds a strategy-agnostic control after 3.6k–15.8k environment steps, compared with millions of steps required by model-free PPO. Across representation choices, graph structure provides greater robustness under topology-dependent attacks, while simpler representations remain competitive in overall performance. Among successful runs, the number of episodes required to reach the control remains approximately constant as network size increases from 15 to 100 hosts. These results establish learned cyber dynamics as a sample-efficient and scalable basis for autonomous defense, and identify world representation as a central design axis for robustness and scalability.

1 INTRODUCTION

Reinforcement learning (RL) (Sutton et al., 1998) has emerged as a prominent approach to autonomous cyber defense, where agents make sequential decisions to protect networked systems against evolving threats (Stakhanova et al., 2007; Kiely et al., 2025). In autonomous cyber defense, it is common to rely on *model-free* RL algorithms (Schulman et al., 2017; Dutta et al., 2023b; Singh et al., 2025; King et al., 2025; Kazeminajafabadi et al., 2026), directly optimizing defensive policies through repeated environment interaction. However, this prevailing paradigm is increasingly inadequate in the face of modern offensive capabilities. With frontier agentic models such as Claude Mythos (Anthropic, 2026), autonomous adversaries can increasingly reason over complex systems and exploit zero-day vulnerabilities (Zhu et al., 2026), intensifying the challenge faced by purely reactive DRL defenders. Recent evidence demonstrates this fragility. Masukawa et al. (2026) showed that even compact agentic language models (Belcak et al., 2025) can autonomously discover and exploit blind spots in state-of-the-art DRL blue agents. Separately, McFadden et al. (2026) systematized the pitfalls of RL-based cyber defense, finding that existing defenses frequently fail to account for non-stationary adversaries, inadequately address partial observability, and lack demonstrated policy convergence. Beyond these robustness issues, model-free RL can also be sample-inefficient, requiring substantial environment interaction for learning and adaptation (Egorov & Shpilman, 2022).

Meanwhile, world models have advanced rapidly in robotics, visual control, and embodied intelligence (Hafner et al., 2023; Zhang et al., 2023; Alonso et al., 2024). These methods learn predictive environment dynamics and optimize decisions through imagined trajectories (Hafner et al., 2023; Burchi & Timofte, 2025b). This separation between dynamics modeling and policy learning is particularly attractive in cyber defense because the two components change at different rates. An attacker may change its strategy from one campaign to the next, but the network protocols, services, and detection mechanisms through which both attacker and defender act, that is, the world itself, do not (Oesch et al., 2024; Kiely et al., 2025). A model of these shared dynamics therefore remains valid when the adversary changes, whereas a reactive policy fitted to one adversary must be relearned.

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

Recent work further extends world models to visual, language, and multimodal representations (Zhou et al., 2025; Lin et al., 2024; Mazzaglia et al., 2024), which suggests that cyber dynamics, too, may be learnable from heterogeneous observations and that a learned model may enable substantially more sample-efficient autonomous defense.

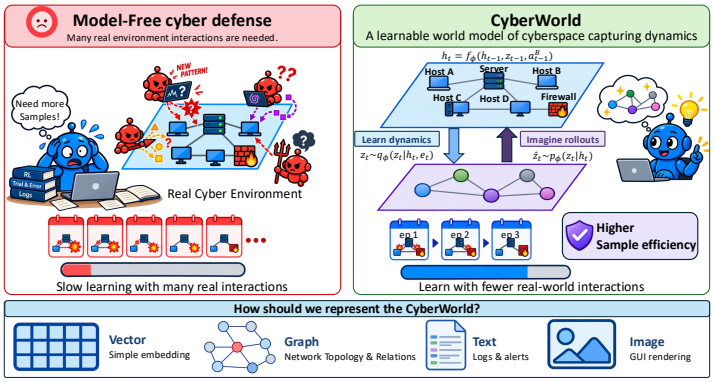


Figure 1: CyberWorld learns predictive cyber dynamics to enable more sample-efficient autonomous defense.

Transferring world models to cybersecurity, however, first requires deciding what the cyber "world" is. Unlike visual control, where observations take standardized forms such as pixels or proprioceptive signals (Hafner et al., 2023; Zhou et al., 2025), a cyber defender observes heterogeneous information, including host attributes, network topology, alerts, logs, and their temporal histories, and the suitability of a given representation may change with the scale and structure of the network. The central question of this work is therefore: **What should constitute the "world" in a world model for autonomous cyber defense?**

To investigate this question, we introduce **CyberWorld**, a Dreamer-style (Hafner et al., 2020; 2021; 2023) world model for autonomous cyber defense. CyberWorld learns latent cyber dynamics and trains defensive policies through imagined trajectories. Within a shared world-model framework, we study five representations of cyberspace: per-host vectors, host graphs, their combination, natural-language telemetry, and vector–text fusion with an added frame-reconstruction objective.

Using CyberWheel (Oesch et al., 2024), we evaluate three aspects of CyberWorld: how cyber representations affect world-model-based control, how they scale as the network grows from 15 to 100 hosts, and whether world-model-based defense can substantially improve sample efficiency over model-free PPO (Schulman et al., 2017). We also examine end-of-training reward stability to distinguish fast initial learning from stable defensive behavior.

Our experiments demonstrate a substantial reduction in environment interaction. Across all four scoreable attack strategies, the graph-based CyberWorld variant reaches the control within 3.6k–15.8k steps, while PPO requires 2.3M–3.1M steps or fails within its 3.2M-step budget. Vector and graph representations perform comparably on most attackers, and the graph representation is the most reliable on the topology-following attacker. On larger networks, successful runs require a similar number of episodes to reach the control, although the fixed interaction budget becomes restrictive at 100 hosts. End-of-training reward stability varies across representations, indicating that rapid acquisition of a defensive behavior does not by itself imply convergence.

Our contributions are threefold:

- We introduce **CyberWorld**, to the best of our knowledge, the first Dreamer-style world-model framework for autonomous cyber defense, learning predictive cyber dynamics and training defensive policies through imagined trajectories.
- We investigate which representation of the cyber world is appropriate for world modeling by comparing various representations, and further study the vector and graph representations across network scales; the results highlight the benefit of explicit graph structure when attack progression depends on network topology.
- We demonstrate that CyberWorld can achieve effective cyber defense with orders of magnitude fewer environment interactions than model-free PPO.

2 Background & Related Works

2.1 From Model-Free Reinforcement Learning to Multimodal World Models

RL formulates sequential decision making through interactions between an agent and its environment (Sutton et al., 1998). Modern deep RL has been dominated by model-free approaches such as PPO

(Schulman et al., 2017), which optimize policies directly from collected experience without explicitly learning environment dynamics. While effective, this paradigm can require substantial environment interaction, motivating renewed interest in model-based learning for improved sample efficiency. PlaNet (Hafner et al., 2019) showed that latent dynamics learned from high-dimensional observations can support planning. Building on this, the Dreamer family further established latent imagination as a practical framework for learning policies primarily inside a predictive world model (Hafner et al., 2020; 2021; 2023). Recent work has expanded this framework with Transformer-based dynamics and more accurate latent prediction (Micheli et al., 2023; Robine et al., 2023; Burchi & Timofte, 2025a). At the same time, world models have begun to move beyond primarily visual observations: Lin et al. (2024) jointly model language and vision for imagined policy learning, while Mazzaglia et al. (2024) connect multimodal foundation representations with latent environment dynamics. Structured representations have also emerged as an alternative to monolithic latent states, including causal graph formulations for explicit state reasoning (Zhou et al., 2026). These developments motivate CyberWorld's central question of how heterogeneous cyber observations, including vectors, network topology, text, and visual interfaces, should be represented within a predictive world model.

2.2 Model-Based Autonomous Cyber Defense

Model-based cyber defense has traditionally relied on environment dynamics that are manually specified or explicitly estimated rather than learned as general predictive latent models. Prior work has used transition, observation, or causal models to support belief tracking, planning, and defensive decision making (Le & Stadler, 2025; Miehling et al., 2018; Andrew et al., 2022), including domain-specific Partially Observable Markov Decision Process (POMDP) dynamics updated online from observations (Dutta et al., 2023a). In contrast, most recent DRL approaches learn defensive policies directly from interaction without explicitly modeling how the cyber environment evolves (Dutta et al., 2023b). Recent work has begun to bridge these two paradigms by learning selected aspects of the environment. Kazeminajafabadi et al. (2026), for example, infer latent attacker context to condition defensive policies while leaving the underlying environment dynamics implicit; latent dynamics have also been used for MCTS-based cyber-defense planning (Li et al., 2026). Closest to CyberWorld, Karacelebi et al. (2026) learn a graph-structured world model for networked systems with recurrent latent dynamics and imagined rollouts. Their setting, however, models non-adversarial network evolution. To the best of our knowledge, CyberWorld is the first Dreamer-style world model for autonomous cyber defense, learning predictive latent dynamics that capture the interaction between attacker behavior, defender interventions, and evolving network state.

3 CyberWorld

CyberWorld is a multimodal latent world model tailored for autonomous cyber defense, built upon the DreamerV3 framework (Hafner et al., 2023) and adapted to the discrete, sparse-reward, and relational topology of enterprise network defense. The agent constructs a predictive model of the environment from replayed interactions and trains its policy entirely inside the model's imagination, decoupling the sample complexity of behavior learning from the high computational overhead of interacting with live network simulators. Its cyber-specific components are an environment-independent intermediate representation of the defender's view, encoders for heterogeneous cyber observations (host vectors, the host graph, and telemetry text) that are fused into a single latent embedding, and modality-specific reconstruction objectives that determine what the latent state must preserve (Figure 2).

3.1 Problem Formulation

We instantiate our formulation in CyberWheel (Oesch et al., 2024), a configurable simulation environment for autonomous cyber defense. CyberWheel models enterprise networks with explicit hosts, subnets, routing, services, and firewall constraints, while scripted adversaries progress through MITRE ATT&CK- and Atomic Red Team-inspired attack stages (MITRE, 2026; Red Canary, 2026). The defender receives only detector alerts and the state of its own assets, and responds through cyber deception by deploying or removing decoy hosts along the adversary's likely path. Compared with other cyber RL environments (Microsoft Defender Research Team., 2021; Janisch et al., 2023; Kiely et al., 2025), CyberWheel is particularly well suited to controlled world-model evaluation because it combines a centralized single-defender setting with configurable attacker strategies, detector models, and network topologies. This design avoids additional confounds from multi-agent coordination and credit assignment (Lyu et al., 2021; Egorov & Shpilman, 2022; Zhang et al., 2025; Xue et al., 2026; Zhao et al., 2026), while allowing the same simulation substrate to support our representation and scalability studies.

We formalize blue-agent network defense as a POMDP $(\mathcal{S}, \mathcal{A}^B, \mathcal{O}, \mathcal{T}, \Omega, R, \gamma)$, where $\mathcal{S}$ represents the underlying global environment states (network topology, host compromise status, decoy configurations, and latent adversarial intent), $\mathcal{A}^B$ denotes the defender's action set, $\mathcal{O}$ is the observation space, $\mathcal{T}$ specifies the global transition dynamics, $\Omega(o \mid s)$ maps states to partial defender observations, $R(s, a^B)$ is the reward function, and $\gamma \in [0, 1)$ is the temporal discount factor.

In enterprise network defense, the transition dynamics $\mathcal{T}$ govern the coupled execution of defender countermeasures and adversarial tactics. Let $a_t^B \in \mathcal{A}^B$ denote the blue defender action, $a_t^R \in \mathcal{A}^R$ denote the concurrent red adversary action, and $c \in \mathcal{C}$ represent an adversary strategy. From the defender's perspective, the effective state transition $\mathcal{T}_c(s_{t+1} \mid s_t, a_t^B)$ factors into:

$$\mathcal{T}_c(s_{t+1} \mid s_t, a_t^B) = \sum_{a_t^R \in \mathcal{A}^R} \mathcal{T}_{\text{mech}}(s_{t+1} \mid s_t, a_t^B, a_t^R)\, \pi_c^R(a_t^R \mid b_t^R), \tag{1}$$

where $\mathcal{T}_{\text{mech}}$ captures the underlying *cyber mechanisms* (protocol semantics, service configurations, vulnerability exploits, and deception capture constraints), which constrain the attacker as much as the defender, since an adversary can exploit a host only through the protocols and services the network exposes, while π_c^R reflects the adversary's stochastic policy conditioned on their internal belief b_t^R. When an adversary shifts tactics ($c \to c'$), π_c^R changes, inducing an apparent shift in overall dynamics $\mathcal{T}_c \to \mathcal{T}_{c'}$. However, $\mathcal{T}_{\text{mech}}$ is shared across strategies under this formulation. A model-free policy does not explicitly preserve this shared transition factor, whereas a latent world model separates dynamics prediction from policy learning and can, in principle, retain mechanism information across changes in adversary strategy. We provide a formal analysis of this factorization in Appendix A, showing how adversary-strategy shifts affect the effective transition kernel and how model error decomposes into shared-mechanism and strategy-dependent components.

The defender's observation o_t is the stream of detector alerts and asset states described above, which we refer to as *telemetry*. Telemetry is incomplete and delayed and never exposes the adversary's internal state, so the observation sequence is non-Markovian. Optimal defense therefore requires a temporal belief state $b_t(s) = p(s_t = s \mid o_{\leq t}, a_{<t}^B)$, which CyberWorld approximates by a recurrent latent state $s_t = [\, h_t;\, z_t \,] \approx b_t(s)$. Under this formulation, the choice of observation representation $\psi(o_t)$ is a dynamics-modeling decision: a representation is adequate when the sequence $\psi(o_{\leq t})$ carries sufficient statistics to identify $\mathcal{T}_{\text{mech}}$ and to track b_t^R. Because attacks manifest both as localized changes in host status and as non-local traversals of the network topology, a representation must capture both local state and topological information for the transitions under $\mathcal{T}_c$ to be predictable.

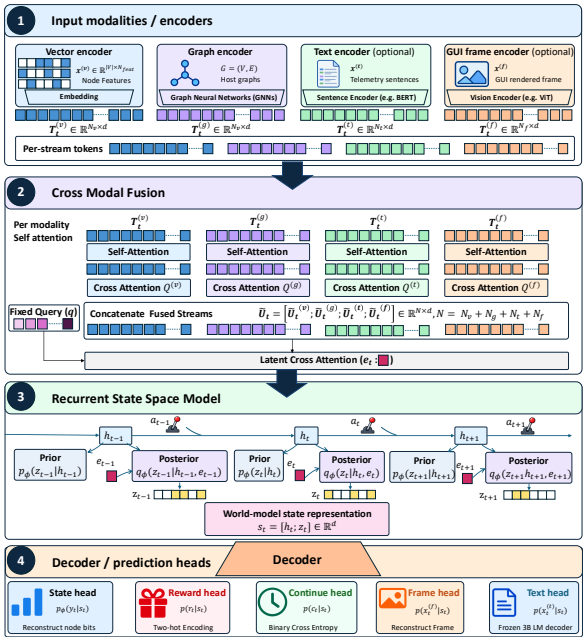


Figure 2: Architecture of CyberWorld. Multimodal telemetry and structural observations are mapped into a unified latent space via cross-attention pooling, driving an RSSM that tracks recurrent belief dynamics and trains defensive behavior purely within imagined rollouts.

3.2 World Model

The world model follows the data flow of Figure 2: each observation channel is encoded into a token sequence, the token sequences are fused into one observation embedding e_t, a Recurrent State-Space Model (RSSM) (Hafner et al., 2019; 2023) turns the sequence of embeddings and defender actions into a latent state s_t, and modality-specific heads reconstruct the observation, the reward, and the continuation flag from s_t. Let $\mathcal{M}$ denote the set of channels an arm encodes and $\mathcal{K}$ the set of

heads it trains; both are configuration, and every arm shares the same fusion, latent dynamics, head architectures, and behavior learning.

Modality Encoders. Each channel $m \in \mathcal{M}$ produces a token sequence $T_t^m \in \mathbb{R}^{L_m \times d}$ of common width d. Vector observations are encoded independently per host, while the graph encoder augments the same host features with graph-attention message passing over G (Veličković et al., 2018). Text telemetry w_t is encoded by a frozen pretrained language model (Team, 2024) and projected into the shared token space through a trainable adapter. Visual observations I_t are similarly mapped into patch-level tokens by a visual encoder and projected into the shared token space; when enabled as a prediction target, they are reconstructed by the corresponding frame head. Visual encoding is optional and unused in the reported configurations.

Observation Representations. Every channel is rendered from one environment-agnostic intermediate representation (IR) of the defender's view, which is the only object an encoder or renderer receives; ground-truth compromise labels and the adversary's position are kept in a separate annotation record used for analysis only. The IR fixes the node set before an episode starts: the n_{host} physical hosts occupy the indices given by the network configuration, and n_{decoy} reserved per-subnet slots receive decoys as they are deployed, each decoy binding to the lowest free slot of its subnet and releasing it on removal, so that the $N = n_{\text{host}} + n_{\text{decoy}}$ node indices refer to the same hosts throughout an episode and never reveal decoys that have not yet been deployed. The vector channel is the defender-visible host state

$$\mathbf{x}_t \in \{0,1\}^{N\times 5}, \qquad \mathbf{x}_t[i,:] = \big(\texttt{present}, \texttt{is_decoy}, \texttt{isolated}, \texttt{alert_now}, \texttt{alert_ever}\big)_i, \tag{2}$$

whose alert fields originate from the defender's own detectors (Stakhanova et al., 2007) and are keyed on the source host; the IR carries neither a destination host nor a technique label, since either would reveal the attacker's next target. For reconstruction, the five fields are mapped to four role bits

$$\boldsymbol{y}_t[i,:] = \big(\max(\texttt{present}, \texttt{isolated}), \texttt{is_decoy}, \texttt{alert_now}, \texttt{alert_ever}\big)_i \in \{0,1\}^4. \tag{3}$$

The encoder uses $\boldsymbol{x}_t$, while the state head reconstructs $\boldsymbol{y}_t$. The graph channel adds the sparse adjacency $G = (V, E)$ with $|V| = N$, taken from the network configuration: hosts on the same subnet are adjacent, cross-subnet edges exist where an interface is declared, and the topology is static within an episode. The text channel w_t renders the same record as prose, namely an alert summary, the defender's own action, the source host of any detected activity, the changes since the previous step, and the present hosts grouped by alert state with decoys named individually. The frame I_t draws the nodes at positions determined by the topology, adds the edges of G, and colours each present node by $\boldsymbol{y}_t$, so a frame is exactly invertible to the role bits. All channels are deterministic functions of the same IR record; every arm therefore receives the same information, and the arms differ in representation alone. Benchmark-specific constants are given in subsection 4.1.

Cross-Modal Fusion. Fusion keeps the channels as separate token streams until a final pooling step. Writing $\mathrm{Attn}(Q,K,V) = \mathrm{softmax}\big(QK^\top/\sqrt{d}\big)\,V$ for multi-head attention, each stream is first tagged with a learned channel embedding and refined by a stack of self-attention blocks in which the stream attends to itself,

$$U_t^m = \mathrm{Attn}\left(T_t^m,\ T_t^m,\ T_t^m\right), \tag{4}$$

and, when more than one channel is active, each refined stream then attends to the concatenation of the other streams,

$$\tilde{U}_t^m = \mathrm{Attn}\left(U_t^m,\ [\,U_t^{m'}\,]_{m'\neq m},\ [\,U_t^{m'}\,]_{m'\neq m}\right), \tag{5}$$

so that, for instance, a text token can attend to the node token it describes. A single learned query $q \in \mathbb{R}^d$ finally pools the concatenated fused streams into a fixed-size observation embedding (Jaegle et al., 2021),

$$e_t = \mathrm{Attn}\left(q,\ [\,\tilde{U}_t^m\,]_{m\in\mathcal{M}},\ [\,\tilde{U}_t^m\,]_{m\in\mathcal{M}}\right) \in \mathbb{R}^d, \tag{6}$$

whose size is independent of the number of hosts or text tokens. With a single active channel, the cross-attention stage is omitted and q pools U_t^m directly.

Latent Dynamics. The RSSM maintains a deterministic recurrent state h_t and a discrete stochastic state z_t; with parameters ϕ,

$$\begin{cases} \text{Recurrent model:} & h_t = f_\phi(h_{t-1}, z_{t-1}, a_{t-1}^B) \\ \text{Posterior representation:} & z_t \sim q_\phi(z_t \mid h_t, e_t) \\ \text{Prior transition dynamics:} & \hat{z}_t \sim p_\phi(\hat{z}_t \mid h_t), \end{cases} \tag{7}$$

where the posterior incorporates the current fused observation e_t, while the prior predicts the stochastic state from history alone and is used for imagined rollouts. The remaining stochastic parameterization follows DreamerV3 (Hafner et al., 2023).

The concatenated latent vector

$$s_t = [\, h_t;\, z_t \,] \tag{8}$$

constitutes the model state supplied to all decoding heads, policy actors, and value critics; s_t is the learned counterpart of the defender belief b_t and only the defender's own action conditions it.

Prediction and Reconstruction Heads. Every head reads the posterior state s_t and reconstructs a quantity of the same step t: the model is trained to explain the present from its belief, and the future enters only through the prior in Equation 7. All configurations reconstruct the four defender-visible node-state roles $\boldsymbol{y}_t$ with a shared per-node decoder, and predict reward and episode continuation. The state decoder conditions on s_t and a learned node query, so its parameters are independent of network size. The reward head uses distributional symlog two-hot parameterization (Hafner et al., 2023). Text-enabled configurations additionally reconstruct w_t through the frozen language model conditioned on a learned latent prefix, while the multimodal configuration reconstructs I_t with a convolutional decoder.

Cyber-Specific Loss Weighting. Two properties of cyber telemetry make the standard reconstruction objectives degenerate, and CyberWorld modifies them accordingly; both are departures from DreamerV3.

First, most node attributes persist between consecutive steps, allowing an unweighted objective to succeed largely by predicting no change. We therefore up-weight state elements that change:

$$\mathcal{L}_{\text{state}} = \frac{1}{4N} \sum_{i=1}^{N} \sum_{j=1}^{4} \lambda_{t,i,j}\, \mathrm{BCE}\left(\hat{y}_{t,i,j},\, y_{t,i,j}\right), \qquad \lambda_{t,i,j} = \begin{cases} \rho_x & \text{if } y_{t,i,j} \neq y_{t-1,i,j} \\ 1 & \text{otherwise.} \end{cases} \tag{9}$$

Second, the reward is dominated by rare events, namely the decoy captures and decoy-budget violations, which occur on a small fraction of steps and therefore contribute little to a uniformly weighted reward loss. CyberWorld balances the reward-bearing steps $\mathcal{R} = \{t : |r_t| \geq \tau\}$ and the remaining steps $\mathcal{R}^c$ within each batch:

$$\mathcal{L}_{\text{reward}} = \sum_t \omega_t\, \ell_t, \qquad \omega_t = \begin{cases} \frac{1}{2|\mathcal{R}|} & \text{if } t \in \mathcal{R} \\ \frac{1}{2|\mathcal{R}^c|} & \text{otherwise,} \end{cases} \tag{10}$$

where ℓ_t is the two-hot reward loss. When a batch contains one group, the mean is used.

World-Model Objective. Over replayed trajectory segments, the world model parameters ϕ are trained to minimise

$$\mathcal{L}(\phi) = \mathbb{E}_{q_\phi}\Big[\sum_t \Big(\sum_{k \in \mathcal{K}} \beta_k\, \mathcal{L}_k + \beta_{\text{dyn}}\, \mathcal{L}_{\text{dyn}} + \beta_{\text{rep}}\, \mathcal{L}_{\text{rep}} \Big) \Big], \tag{11}$$

where $\mathcal{L}_k$ denotes the active prediction and reconstruction losses, and $\mathcal{L}_{\text{dyn}}$ and $\mathcal{L}_{\text{rep}}$ are DreamerV3's dynamics and representation KL objectives (Hafner et al., 2023).

3.3 Behavior Learning and Online Training

The actor and the critic are trained entirely on trajectories imagined by the world model, and both follow DreamerV3 unchanged (Hafner et al., 2023): imagined trajectories start from posterior states of replayed segments and are rolled forward through the RSSM prior, and the actor and the distributional critic are optimized with bootstrapped λ-returns, REINFORCE, entropy regularization, and percentile-based return normalization. Training likewise follows DreamerV3's concurrent scheme. The agent collects one episode with its current policy, appends it to a growing replay buffer, and performs a fixed number of world-model and actor–critic updates on sampled segments before collecting the next episode. Training starts from randomly initialized parameters with no offline data, and every environment interaction, including the random prefill of the buffer, counts toward the reported budget; the prefill size, update ratio, and budget are given in subsection 4.1. The one cyber-specific consideration is that every declared defender action remains available at every step, so the policy can avoid the decoy-budget penalty only if the world model represents it, which is what the balanced reward objective of Equation 10 provides.

Table 1: Environment steps until the trailing-10 return first exceeds the control. All methods start from scratch. CyberWorld and GNN-PPO share the same interaction budget at each N; PPO uses its original, longer training horizon. Mean $\pm$ sd over 3 seeds; (k/n): seeds that crossed. Bold: fastest of GNN-PPO, vector, and graph with all seeds crossed. PPO: 3 seeds at $N \leq 50$, none at $N = 100$; GNN-PPO: 3 seeds, two BFSServerDowntime seeds unfinished at $N = 100$.

attacker	N	PPO	GNN-PPO	ours: vector	ours: graph
BFSServerDowntime	15	2,310,400 ± 640,000 (2/3)	**4,017 ± 821**	4,933 ± 633	5,525 ± 575 (2/3)
	25	2,549,547 ± 410,310	12,368 ± 5,398 (2/3)	10,257 ± 7,123	**5,100 ± 524**
	50	NEVER (0/3)	**2,777 ± 289**	37,570 ± 3,400 (2/3)	32,725 ± 8,245 (2/3)
	100	–	NEVER (0/1)	NEVER (0/3)	NEVER (0/3)
ServerDowntime	15	3,078,400 ± 0 (1/3)	25,875 ± 4,025 (2/3)	3,633 ± 62	**3,600 ± 41**
	25	3,274,880 ± 1,579,162	21,845 ± 0 (1/3)	**5,922 ± 1,769**	7,027 ± 2,668
	50	456,960 ± 0	**3,457 ± 924**	24,593 ± 12,038	29,580 ± 12,300
	100	–	35,175 ± 13,400 (2/3)	47,570 ± 4,690 (2/3)	16,080 ± 0 (1/3)
Exfiltration	15	2,630,400 ± 448,000 (2/3)	11,367 ± 9,182	**3,900 ± 283**	4,600 ± 41
	25	373,547 ± 102,578	14,110 ± 7,030	**4,222 ± 424**	6,290 ± 367
	50	602,027 ± 205,155	**8,330 ± 5,141**	33,745 ± 17,255 (2/3)	25,103 ± 8,951
	100	–	**9,715 ± 3,316**	28,308 ± 17,252 (2/3)	34,840 ± 10,720 (2/3)
BFSExfiltration	15	NEVER (0/3)	**9,117 ± 4,128**	31,650 ± 12,450 (2/3)	15,750 ± 12,960
	25	2,839,680 ± 615,466	23,913 ± 10,031	9,633 ± 6,351	**5,468 ± 1,627**
	50	NEVER (0/3)	14,705 ± 5,525 (2/3)	NEVER (0/3)	15,810 ± 0 (1/3)
	100	–	8,208 ± 838 (2/3)	NEVER (0/3)	17,085 ± 0 (1/3)

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Our evaluation answers three Research Questions (RQs): (**RQ1**) Can a world model substantially improve the sample efficiency of autonomous cyber defense over model-free DRL? (**RQ2**) Which representation of the cyber environment best supports world-model-based control? (**RQ3**) Does the answer survive changes in network scale? RQ1 establishes whether learning the dynamics pays off in the resource that is scarce when a defender is trained against a live or emulated network, the number of environment interactions. RQ2 determines which of the representations of section 3.2 lets the world model capture the dynamics that matter for control, and RQ3 whether that answer and the interaction saving hold as the defended network grows.

Environment. All experiments run in CyberWheel (Oesch et al., 2024) against seven scripted ART attacker strategies, four of which are scoreable (the other three have a control return of 0). RQ1 and RQ2 use the 15-host, three-subnet network; RQ3 scales the same network to $N \in \{25, 50, 100\}$ hosts with the decoy budget scaled with N. Each CyberWorld and GNN-PPO run uses the same 52,500-step interaction budget, corresponding to 1,050 episodes at N=15 (50 random prefill and 1,000 training episodes).

Representation arms. Five arms share the latent dynamics, behavior learning, and hyperparameters and differ only in encoders and reconstruction heads: *vector* (per-node MLP + self-attention), *graph* (GAT over the host graph), *vec+graph*, *text* (frozen Qwen2.5-3B over telemetry sentences), and *all* (vector + text fusion with an added frame-reconstruction head). All share one training configuration: DreamerV3's return normalisation, the balanced reward loss of Equation 10, a 1,000-step actor–critic warm-up, and cosine learning-rate decay.

Baselines. *PPO* is CyberWheel's own PPO trained from scratch under the same environment and observation but with its original substantially longer training budget, reaching up to 3.2M environment steps at N=15. *GNN-PPO* is a representation-matched baseline: the graph arm's observation and GAT encoder trained with PPO under the same online interaction budget and seeds as CyberWorld, without a world model. The strategy-agnostic `deploy_then_stop` policy serves as a fixed rule-based control, defining the minimum return threshold that a learned defender must exceed; the environment reward is unchanged throughout.

Metrics. The primary metric is the number of environment steps until the trailing-10 episode return first exceeds the control, reported as mean $\pm$ sd over 3 seeds; four text/multimodal cells use 2 seeds because the third run did not complete. Runs that never cross are reported as such. Curves show the trailing-10 return and a trailing-50 trend. PPO's crossing uses its 30-episode checkpoint evaluations, ours the online training return; no accuracy-based metric is used anywhere.

Implementation details. The world model uses a GRU with 1,024 deterministic units, a 32×32 categorical latent, encoders of width 512, and two-layer MLP heads of width 512.

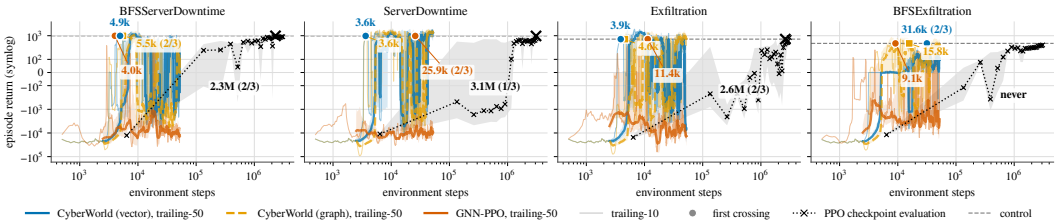


Figure 3: Return vs. environment steps for CyberWorld (vector and graph), GNN-PPO, and PPO on the four scoreable attackers at $N{=}15$ (seed mean, band = min/max over three seeds; PPO: 30-episode checkpoint evaluations). Dashed: the `deploy_then_stop` control; markers: mean first crossing, with the seed count where not all seeds crossed.

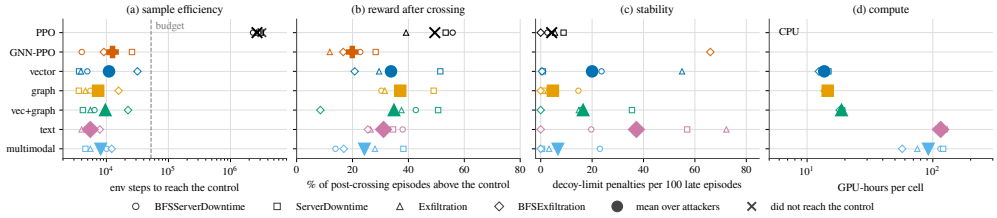


Figure 4: Representation comparison at $N{=}15$: (a) steps to control, (b) post-crossing success rate, (c) late decoy-limit penalties, and (d) GPU-hours. Small markers denote attackers, large markers their mean; $\times$: no crossing.

We optimize the world model with AdamW (learning rate $3{\times}10^{-4}$, weight decay 10^{-4}, and gradient clipping at 1.0) using batches of 16 windows of 16 steps. We set the dynamics and representation KL weights to 0.5 and 0.1, respectively, use 1 free nat and 1% uniform mixing, assign a change weight of 5 to node bits, weight the text and frame reconstruction heads by 0.3 and 20, respectively, and define reward-bearing steps by $|r_t| \geq 50$. The actor and critic use learning rate $3{\times}10^{-5}$ with a 1,000-step warm-up and cosine decay, imagination horizon 15, $\gamma = 0.997$, $\lambda = 0.95$, and entropy weight $3{\times}10^{-4}$. Each run collects a 50-episode random prefill, then one episode per iteration with 32 world-model and 32 actor–critic updates for 1,000 iterations, on a single NVIDIA A100 shared with other jobs; PPO trains on CPU with 128 parallel environments.

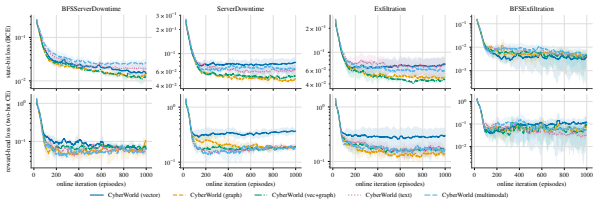


Figure 5: World-model fidelity across representations at $N{=}15$: replay-batch state-bit loss (top) and reward-head loss (bottom).

4.2 Results

Sample efficiency (RQ1). Figure 3 summarizes the primary sample-efficiency result for the vector and graph arms. On three of the four scoreable attackers, both arms reach the control after 3.6k–5.5k environment steps, whereas PPO requires 2.3M–3.1M steps, two to three orders of magnitude more; on BFSExfiltration, where no PPO seed crosses within its 3.2M-step budget, the graph arm crosses on every seed at 15,750 steps and the vector arm on two of three seeds at 31,650, while the text arm, not shown, crosses on every seed at 7,850 (exact values in Table 2; pooled profile in Figure 8). GNN-PPO, which shares the graph arm's observation and encoder, exhibits brief early crossings under the trailing-10 criterion, but these are not sustained in the longer-term trend and are followed by severe return degradation under decoy-limit penalties (Figure 4b,c). In contrast, the world-model variants maintain ServerDowntime performance with substantially fewer such penalties. These results indicate two distinct advantages: relative to PPO, world-model learning reduces the required environment interaction, while relative to a model-free learner with the same encoder, it yields greater stability after first reaching the control. These results concern environment interactions rather than computational cost, which is analyzed separately below. Appendix D shows that the main sample-efficiency conclusion is robust to the smoothing convention, while the BFSExfiltration vector crossing is sensitive to this choice.

Representation comparison (RQ2). On crossing speed, the three numeric arms are indistinguishable within seed variation on three of the four attackers (Figure 4(a)), despite clear differences in world-

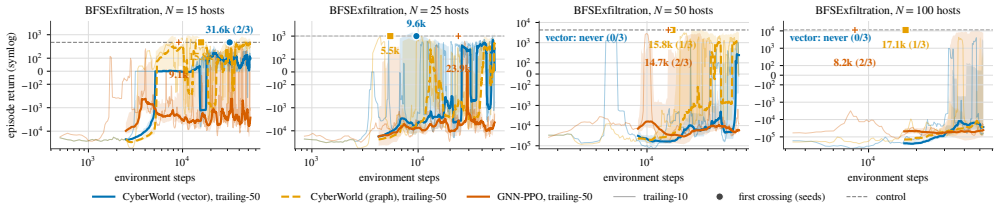


Figure 6: Scaling with network size for the vector, graph, and GNN-PPO arms. Curves show trailing-50 means with min/max bands; thin lines show trailing-10.

model fidelity. The graph arm achieves the lowest state-bit and reward-head losses on the attackers that separate the representations, whereas elevated vector reward loss coincides with higher decoy-limit penalties (Figure 4(c), Figure 5). Richer representations provide selective benefits: the text arm crosses on every scoreable attacker, on all but one of its seeds, and is fastest on BFSExfiltration, whereas the multimodal arm crosses on every seed of the two non-BFS attackers but on a single seed of each BFS attacker. BFSExfiltration provides the clearest structural distinction among the three numeric representations: because its target order follows the network topology, only the graph arm crosses on every seed at N=15 (Figure 6). Zero-control attackers are omitted (Appendix E).

Scalability (RQ3). Among successful runs, the number of episodes required to reach the control remains approximately constant as the network grows, while environment steps increase with episode length (Figure 7). Scalability becomes representation-dependent under BFSExfiltration: vector and graph both reach the control at $N \leq 25$, whereas among the world-model arms only the graph arm does so at N=50 and N=100 (Figure 6). At N=100, the fixed interaction budget covers relatively few episodes, making crossing frequency increasingly budget-limited. GNN-PPO's trailing-10 return also crosses at these scales, but its trailing-50 trend does not remain above the control once decoy-limit penalties accumulate. Thus, simple representations suffice for most attackers, while explicit graph structure becomes increasingly important when attack progression depends on network topology. See Appendix F.

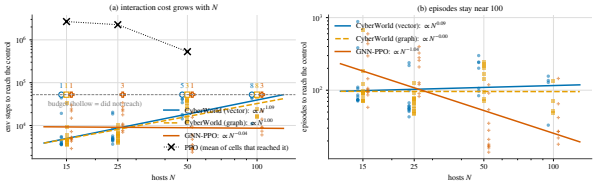


Figure 7: (a) Environment steps and (b) episodes to the control vs. N for vector, graph and GNN-PPO: small markers are runs that crossed, lines are power-law fits over them, hollow markers at the budget line count runs that did not cross, and $\times$ is PPO's mean where it crossed.

Discussion and Limitations. CyberWorld reaches effective defense with two to three orders of magnitude fewer environment interactions than PPO, exhibits greater post-crossing stability than a representation-matched model-free baseline, and, among successful runs, requires an approximately constant number of episodes to reach the control as the network grows from 15 to 100 hosts. Among the evaluated representations, the host graph provides the most consistent performance under topology-dependent attacks. These results are subject to three main limitations. First, interaction efficiency does not imply equivalent wall-clock efficiency, particularly for text and multimodal variants that incur substantially higher inference cost. Second, our evaluation considers an RSSM-based latent dynamics model and does not compare alternative world-model backbones such as transformers or diffusion models. Third, all experiments are conducted in CyberWheel with scripted adversaries, so broader validation across environments and adversary models remains necessary.

5 Conclusion

We introduced **CyberWorld**, a world-model-based framework for autonomous cyber defense that learns predictive latent dynamics of the defended network and optimizes defensive policies through imagined trajectories. Our results show that explicit dynamics modeling can substantially reduce environment interaction while maintaining effective defensive behavior across adversary strategies and network scales, and that representation choice materially affects this capability when attack progression depends on network topology. Future work should extend CyberWorld to alternative latent dynamics architectures, more efficient semantic representations, additional cyber-defense environments, emulated networks, and learned adversaries, with the broader goal of developing transferable cyber world models that capture reusable network dynamics across heterogeneous operational settings.

AI USE STATEMENT

Generative AI tools were used as supporting tools during the development of this work. The core research idea, the formulation of RQ1–RQ3, the experimental design, and the scientific interpretation of the results were developed by the authors. Claude Code was used to assist with software implementation, debugging, experiment automation, and the creation and refinement of some visualizations. Generative AI tools were also used to assist with drafting, editing, and improving the clarity of portions of the manuscript, while the intended arguments, claims, and scientific content were determined and reviewed by the authors. ChatGPT image generation was used to create some of the illustrative icons appearing in Figures 1 and 2. All AI-assisted code, text, figures, and experimental outputs were reviewed and validated by the authors. The authors take full responsibility for the final content of the paper and its associated artifacts.

ETHICS STATEMENT

This work studies autonomous cyber defense in a simulated environment. All experiments are conducted within CyberWheel using scripted adversary strategies and do not involve human subjects, private user data, or interactions with production systems. The proposed methods are intended to improve the sample efficiency and robustness of defensive agents. As with other research on autonomous cyber systems, the underlying modeling and simulation techniques may have dual-use potential. However, this work does not introduce new exploits, vulnerability-discovery procedures, or operational attack techniques. We report the methodology and experimental results for the purpose of reproducible research on defensive learning systems. The authors have reviewed and followed the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the environment interface, intermediate representation, model architecture, training procedure, baselines, evaluation metrics, and hyperparameters in Sections 3 and 4. Appendix A gives the formal analysis of the problem formulation with proofs, Appendix C reports the exact sample-efficiency measurements, Appendix D examines sensitivity to the smoothing convention, Appendix E provides complete learning curves and post-crossing behavior, Appendix F reports scaling and computational costs, and Appendix G visualizes open-loop imagination of the learned model. To further support reproducibility, we provide an anonymous repository (https://anonymous.4open.science/r/CyberWorld-768D/README.md) containing the implementation, experiment configurations, and evaluation code.

REFERENCES

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. *Advances in Neural Information Processing Systems*, 37:58757–58791, 2024.

Alex Andrew, Sam Spillard, Joshua Collyer, and Neil Dhir. Developing optimal causal cyber-defence agents via cyber security simulation. *arXiv preprint arXiv:2207.12355*, 2022.

Anthropic. Claude mythos preview. `https://red.anthropic.com/2026/mythos-preview/`, 2026. [Accessed 18-09-2026].

Karl Johan Åström. Optimal control of markov processes with incomplete state information i. *Journal of mathematical analysis and applications*, 10:174–205, 1965.

Peter Belcak, Greg Heinrich, Shizhe Diao, Yonggan Fu, Xin Dong, Saurav Muralidharan, Yingyan Celine Lin, and Pavlo Molchanov. Small language models are the future of agentic ai. *arXiv preprint arXiv:2506.02153*, 2025.

Maxime Burchi and Radu Timofte. Accurate and efficient world modeling with masked latent transformers. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp,

Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 5894–5912. PMLR, 13–19 Jul 2025a. URL https://proceedings.mlr.press/v267/burchi25a.html.

Maxime Burchi and Radu Timofte. Learning transformer-based world models with contrastive predictive coding. In *International Conference on Learning Representations*, volume 2025, pp. 6949–6970, 2025b.

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

Ashutosh Dutta, Samrat Chatterjee, Arnab Bhattacharya, and Mahantesh Halappanavar. Deep reinforcement learning for cyber system defense under dynamic adversarial uncertainties. *arXiv preprint arXiv:2302.01595*, 2023b.

Vladimir Egorov and Alexei Shpilman. Scalable multi-agent model-based reinforcement learning. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '22, pp. 381–390, Richland, SC, 2022. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450392136.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pp. 2555–2565. PMLR, 2019.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=0oabwyZbOu.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pp. 4651–4664. PMLR, 2021.

Jaromír Janisch, Tomáš Pevný, and Viliam Lisý. Nasimemu: Network attack simulator & emulator for training agents generalizing to novel scenarios. In *Computer Security. ESORICS 2023 International Workshops: CPS4CIP, ADIoT, SecAssure, WASP, TAURIN, PriST-AI, and SECAI, The Hague, The Netherlands, September 25–29, 2023, Revised Selected Papers, Part II*, pp. 589–608, Berlin, Heidelberg, 2023. Springer-Verlag. ISBN 978-3-031-54128-5. doi: 10.1007/978-3-031-54129-2_35. URL https://doi.org/10.1007/978-3-031-54129-2_35.

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

Can Karacelebi, Yusuf Talha Sahin, Elif Surer, and Ertan Onur. Learning ad hoc network dynamics via graph-structured world models. *arXiv preprint arXiv:2604.14811*, 2026.

Armita Kazeminajafabadi, Amirhossein Ravari, Nathaniel D Bastian, Tian Lan, and Mahdi Imani. Posterior-conditioned multi-agent reinforcement learning for adaptive cyber defense. *Artificial Intelligence*, pp. 104606, 2026.

Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49(2):209–232, 2002.

Mitchell Kiely, Metin Ahiskali, Etienne Borde, Benjamin Bowman, David Bowman, Dirk Van Bruggen, KC Cowan, Prithviraj Dasgupta, Erich Devendorf, Ben Edwards, et al. Exploring the efficacy of multi-agent reinforcement learning for autonomous cyber defence: A cage challenge 4 perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 28907–28913, 2025.

Isaiah J King, Benjamin Bowman, and H Howie Huang. Automated cyber defense with generalizable graph-based reinforcement learning agents. *arXiv preprint arXiv:2509.16151*, 2025.

Duc Huy Le and Rolf Stadler. Learning optimal defender strategies for cage-2 using a pomdp model. In *2025 21st International Conference on Network and Service Management (CNSM)*, pp. 1–9. IEEE, 2025.

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

Jessy Lin, Yuqing Du, Olivia Watkins, Danijar Hafner, Pieter Abbeel, Dan Klein, and Anca Dragan. Learning to model the world with language. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=7dP6Yq9Uwv.

Xueguang Lyu, Yuchen Xiao, Brett Daley, and Christopher Amato. Contrasting centralized and decentralized critics in multi-agent reinforcement learning. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '21, pp. 844–852, Richland, SC, 2021. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450383073.

Ryozo Masukawa, Ian Bryant, Armita Kazeminajafabadi, Sanggeon Yun, Hyunwoo Oh, SungHeon Jeong, Nathaniel D Bastian, Mahdi Imani, and Mohsen Imani. Trident: How to break deep reinforcement learning cyber defenses (agentic). *arXiv preprint arXiv:2608.04317*, 2026.

Pietro Mazzaglia, Tim Verbelen, Bart Dhoedt, Aaron Courville, and Sai Rajeswar. GenRL: Multimodal-foundation world models for generalization in embodied agents. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=za9Jx8yqUA.

Shae McFadden, Myles Foley, Elizabeth Bates, Ilias Tsingenopoulos, Sanyam Vyas, Vasilios Mavroudis, Chris Hicks, and Fabio Pierazzi. SoK: The pitfalls of deep reinforcement learning for cybersecurity. In *35th USENIX Security Symposium (USENIX Security 26)*, Baltimore, MD, August 2026. USENIX Association. URL https://www.usenix.org/conference/usenixsecurity26/presentation/mcfadden.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=vhFu1Acb0xb.

Microsoft Defender Research Team. Cyberbattlesim, 2021. Created by Christian Seifert, Michael Betser, William Blum, James Bono, Kate Farris, Emily Goren, Justin Grana, Kristian Holsheimer, Brandon Marken, Joshua Neil, Nicole Nichols, Jugal Parikh, Haoran Wei.

Erik Miehling, Mohammad Rasouli, and Demosthenis Teneketzis. A pomdp approach to the dynamic defense of large-scale cyber networks. *IEEE Transactions on Information Forensics and Security*, 13(10):2490–2505, 2018. doi: 10.1109/TIFS.2018.2819967.

MITRE. Mitre att&ck. https://attack.mitre.org/, 2026. [Accessed 20-04-2026].

Sean Oesch, Amul Chaulagain, Brian Weber, Matthew Dixson, Amir Sadovnik, Benjamin Roberson, Cory Watson, and Phillipe Austria. Towards a high fidelity training environment for autonomous cyber defense agents. In *Proceedings of the 17th Cyber Security Experimentation and Test Workshop*, CSET '24, pp. 91–99, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400709579. doi: 10.1145/3675741.3675752. URL https://doi.org/10.1145/3675741.3675752.

Red Canary. Atomic red team. https://www.atomicredteam.io/, 2026. [Accessed 20-04-2026].

Jan Robine, Marc Höftmann, Tobias Uelwer, and Stefan Harmeling. Transformer-based world models are happy with 100k interactions. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=TdBaDGCpjly.

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

Aditya Vikram Singh, Ethan Rathbun, Emma Graham, Lisa Oakley, Simona Boboila, Peter Chin, and Alina Oprea. Hierarchical multi-agent reinforcement learning for cyber network defense. *Reinforcement Learning Journal*, 6:790–810, 2025.

Natalia Stakhanova, Samik Basu, and Johnny Wong. A taxonomy of intrusion response systems. *Int. J. Inf. Comput. Secur.*, 1(1/2):169–184, January 2007. ISSN 1744-1765. doi: 10.1504/IJICS.2007.012248. URL https://doi.org/10.1504/IJICS.2007.012248.

Richard S Sutton, Andrew G Barto, and Andrew Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

Di Xue, Jing Jiang, Shaowei Zhang, Wenhao Guo, Lei Yuan, Zongzhang Zhang, and Yang Yu. Learning disentangled multi-agent world model for decentralized control. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=nYyfpPubnW.

Hongxin Zhang, Zeyuan Wang, Qiushi Lyu, Zheyuan Zhang, Sunli Chen, Tianmin Shu, Behzad Dariush, Kwonjoon Lee, Yilun Du, and Chuang Gan. COMBO: Compositional world models for embodied multi-agent cooperation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=YXRyYkb1im.

Weipu Zhang, Gang Wang, Jian Sun, Yetian Yuan, and Gao Huang. Storm: Efficient stochastic transformer based world models for reinforcement learning. *Advances in Neural Information Processing Systems*, 36:27147–27166, 2023.

Zijie Zhao, Honglei Guo, Shengqian Chen, Kaixuan Xu, Bo Jiang, Yuanheng Zhu, and Dongbin Zhao. Empowering multi-robot cooperation via sequential world models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=IvUM6UwYCJ.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=D5RNACOZEI.

Xuanru Zhou, Jiachen Lian, Henry Hong, Xinyi Yang, and Gopala Anumanchipalli. Speech world model: Causal state–action planning with explicit reasoning for speech. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=YGUKPGO182.

Yuxuan Zhu, Antony Kellermann, Akul Gupta, Philip Li, Richard Fang, Rohan Bindu, and Daniel Kang. Teams of LLM agents can exploit zero-day vulnerabilities. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23–35, Rabat, Morocco, March 2026. Association for Computational Linguistics. ISBN 979-8-89176-380-7. doi: 10.18653/v1/2026.eacl-long.2. URL https://aclanthology.org/2026.eacl-long.2/.

APPENDIX

A THEORETICAL ANALYSIS OF THE PROBLEM FORMULATION

A.1 MECHANISM INVARIANCE AND TRANSFER ACROSS ADVERSARY STRATEGIES

Equation (1) decomposes the defender-facing transition law into a strategy-dependent adversary policy π_c^R and a shared mechanism kernel $\mathcal{T}_{\text{mech}}$. We formalize three consequences of this factorization. First, changes in the effective transition law are bounded by changes in the adversary strategy. Second, under partial observability, the filtering posterior provides a sufficient statistic for defender control, motivating the use of a recurrent latent state. Third, model error under a new adversary can be decomposed into mechanism-model error and strategy-estimation error. Proofs are provided in Appendix B.

Setup. Consider a family of POMDPs $\{\mathcal{M}_c\}_{c\in\mathcal{C}}$,

$$\mathcal{M}_c = (\mathcal{S}, \mathcal{A}^B, \mathcal{O}, \mathcal{T}_c, \Omega, R, \gamma),$$

that share all components except the effective transition law $\mathcal{T}_c$, which factors according to Eq. (1). Rewards are bounded as $|R(s, a^B)| \le R_{\max}$, with $R_{\max} = 5{,}000$ in CyberWheel, and $\gamma \in [0, 1)$. We write $\mathrm{TV}(\mu, \nu)$ for total-variation distance and $V^\pi_{\mathcal{M}}$ for the value of policy π in model $\mathcal{M}$, with the initial-state distribution suppressed from the notation.

Assumption 1 (Closed information state)**.** *The global state $s \in \mathcal{S}$ includes the adversary's internal belief b^R ("latent adversarial intent" in Section 3.1), such that $\pi_c^R(a^R \mid b^R)$ is a measurable function of s and Eq. (1) defines a Markov kernel on $\mathcal{S}$ for every c.*

Assumption 2 (Shared mechanisms)**.** *$\mathcal{T}_{\text{mech}}$, Ω, and R are identical across $c \in \mathcal{C}$; adversary strategies differ only through π_c^R.*

Assumption 2 formalizes the intended separation between strategy and environment mechanisms: protocol semantics, action effects, exploit preconditions, and decoy behavior are assumed to remain unchanged when the adversary strategy changes. Assumption 1 is a state augmentation used to express the coupled defender–adversary process as a Markov process.

Sensitivity of the effective dynamics to strategy shift.

Lemma 1 (Strategy-shift bound)**.** *Under Assumptions 1–2, for every $s \in \mathcal{S}$, $a^B \in \mathcal{A}^B$, and $c, c' \in \mathcal{C}$,*

$$\mathrm{TV}\big(\mathcal{T}_c(\cdot \mid s, a^B), \mathcal{T}_{c'}(\cdot \mid s, a^B)\big) \le \mathrm{TV}\big(\pi_c^R(\cdot \mid b^R(s)), \pi_{c'}^R(\cdot \mid b^R(s))\big)\,.$$

Lemma 1 shows that, under the shared-mechanism factorization, a change in adversary strategy cannot induce a larger total-variation change in the effective transition kernel than the corresponding change in the adversary action distribution. Thus, strategy variation affects the defender-facing dynamics through a constrained component of the transition model, while $\mathcal{T}_{\text{mech}}$ remains shared by assumption.

Belief-state sufficiency under partial observability.

Proposition 1 (Belief-state sufficiency)**.** *For every c, the filtering posterior*

$$b_t \doteq p(s_t \mid o_{\le t}, a^B_{<t})$$

is a sufficient statistic of the interaction history. In particular, there exists an optimal defender policy of the form $\pi^\star(a_t^B \mid b_t)$, and the belief state evolves according to a Bayes operator

$$b_{t+1} = U(b_t, a_t^B, o_{t+1}),$$

whose dependence on c enters through $\mathcal{T}_c$, and hence through π_c^R under Assumption 2 (Åström, 1965; Kaelbling et al., 1998).

Proposition 1 provides a control-theoretic interpretation of the recurrent latent state $s_t = [h_t; z_t]$. An exact belief state is generally unavailable, and CyberWorld does not require the learned latent state

to recover b_t exactly. Rather, the RSSM can be viewed as learning a history-dependent predictive statistic from which the actor and value functions are computed. Its prior $p_\phi(z_t \mid h_t)$ and posterior $q_\phi(z_t \mid h_t, e_t)$ are respectively analogous to the prediction and observation-update stages of Bayesian filtering. This interpretation motivates RQ2: different observation representations may preserve different amounts of information relevant to constructing such a predictive latent state.

Error decomposition under adversary shift. The following decomposition is an analytical abstraction of the transition factorization in Equation (1); CyberWorld does not explicitly parameterize $\widehat{\mathcal{T}}_{\mathrm{mech}}$ and $\widehat{\pi}^R_c$ as separate learned modules. To characterize the potential benefit of shared mechanisms, suppose an idealized factored predictive model has obtained a mechanism model $\widehat{\mathcal{T}}_{\mathrm{mech}}$ from data generated under one or more adversary strategies and, for a possibly new adversary c', an estimate $\widehat{\pi}^R_{c'}$ of its strategy. Define the corresponding imagined dynamics as

$$\widehat{\mathcal{T}}_{c'}(s' \mid s, a^B) = \sum_{a^R} \widehat{\mathcal{T}}_{\mathrm{mech}}(s' \mid s, a^B, a^R)\widehat{\pi}^R_{c'}(a^R \mid b^R(s)),$$

with error radii

$$\varepsilon_{\mathrm{mech}} \doteq \sup_{s, a^B, a^R} \mathrm{TV}\Big(\mathcal{T}_{\mathrm{mech}}(\cdot \mid s, a^B, a^R), \widehat{\mathcal{T}}_{\mathrm{mech}}(\cdot \mid s, a^B, a^R)\Big),$$

and

$$\varepsilon_{\mathrm{strat}}(c') \doteq \sup_{s} \mathrm{TV}\big(\pi^R_{c'}(\cdot \mid b^R(s)), \widehat{\pi}^R_{c'}(\cdot \mid b^R(s))\big).$$

Theorem 1 (Decomposed simulation bound)**.** *Under Assumptions 1–2, for every stationary defender policy π and every $c' \in \mathcal{C}$,*

$$\left|V^\pi_{\mathcal{M}_{c'}} - V^\pi_{\widehat{\mathcal{M}}_{c'}}\right| \le \frac{2\gamma R_{\max}}{(1-\gamma)^2}\left(\varepsilon_{\mathrm{mech}} + \varepsilon_{\mathrm{strat}}(c')\right).$$

Consequently, if $\widehat{\pi} \in \arg\max_\pi V^\pi_{\widehat{\mathcal{M}}_{c'}}$, then

$$V^{\pi^\star}_{\mathcal{M}_{c'}} - V^{\widehat{\pi}}_{\mathcal{M}_{c'}} \le \frac{4\gamma R_{\max}}{(1-\gamma)^2}\left(\varepsilon_{\mathrm{mech}} + \varepsilon_{\mathrm{strat}}(c')\right).$$

Theorem 1 separates two sources of error. The term $\varepsilon_{\mathrm{mech}}$ measures error in the component assumed to be shared across adversary strategies, whereas $\varepsilon_{\mathrm{strat}}(c')$ measures error associated with the particular adversary c'. Consequently, improvements to $\widehat{\mathcal{T}}_{\mathrm{mech}}$ can, under Assumption 2, benefit predictions under multiple adversary strategies. Adaptation to c' nevertheless remains dependent on how accurately its strategy can be represented or inferred.

Corollary 1 (Residual error under a previously learned mechanism)**.** *Fix a target suboptimality ϵ and suppose*

$$\varepsilon_{\mathrm{mech}} \le \frac{(1-\gamma)^2}{8\gamma R_{\max}}\epsilon.$$

Then the bound in Theorem 1 guarantees suboptimality at most ϵ whenever

$$\varepsilon_{\mathrm{strat}}(c') \le \frac{(1-\gamma)^2}{8\gamma R_{\max}}\epsilon.$$

Thus, once the mechanism error is below the stated threshold, the bound requires no further reduction in $\varepsilon_{\mathrm{mech}}$ for this target accuracy; the remaining contribution is $\varepsilon_{\mathrm{strat}}(c')$.

Notably, the bound contains no explicit dependence on the number of hosts N. This does not imply that adaptation is independent of network size: larger networks may make state estimation, belief tracking, function approximation, or data collection more difficult. Rather, under the stated factorization, network size does not introduce an additional term into the error decomposition itself. The approximately stable episodes-to-control observed from 15 to 100 hosts in RQ3 are empirically consistent with this interpretation, while not constituting a direct consequence of the bound.

Remark 1 (Sensitivity of a fixed policy to strategy shift)**.** *For a fixed policy π_c, the same simulation argument gives*

$$\left|V^{\pi_c}_{\mathcal{M}_c} - V^{\pi_c}_{\mathcal{M}_{c'}}\right| \le \frac{2\gamma R_{\max}}{(1-\gamma)^2} \sup_s \mathrm{TV}\big(\pi^R_c(\cdot \mid b^R(s)), \pi^R_{c'}(\cdot \mid b^R(s))\big)\,.$$

This bound quantifies the sensitivity of a fixed policy to an adversary shift. It does not, by itself, establish the interaction complexity required to adapt a model-free agent to c'. The distinction relevant to Theorem 1 is therefore analytical: under the factorization of Equation (1), mechanism information constitutes a shared component that can in principle remain useful when the strategy-dependent component changes.

B Proofs for Section A

B.1 Proof of Lemma 1

Fix s, a^B and abbreviate $b = b^R(s)$ and

$$K_{a^R}(\cdot) = \mathcal{T}_{\mathrm{mech}}(\cdot \mid s, a^B, a^R).$$

By Eq. (1), for any measurable $E \subseteq \mathcal{S}$,

$$\mathcal{T}_c(E \mid s, a^B) - \mathcal{T}_{c'}(E \mid s, a^B) = \sum_{a^R} K_{a^R}(E)\left[\pi^R_c(a^R \mid b) - \pi^R_{c'}(a^R \mid b)\right].$$

Since $0 \le K_{a^R}(E) \le 1$, taking the supremum over E and using

$$\mathrm{TV}(\mu, \nu) = \sup_E |\mu(E) - \nu(E)|$$

together with

$$\sup_E \left|\sum_i w_i K_i(E)\right| \le \frac{1}{2}\sum_i |w_i|$$

for signed weights satisfying $\sum_i w_i = 0$ yields

$$\mathrm{TV}\big(\mathcal{T}_c(\cdot \mid s, a^B), \mathcal{T}_{c'}(\cdot \mid s, a^B)\big) \le \frac{1}{2}\sum_{a^R} \left|\pi^R_c(a^R \mid b) - \pi^R_{c'}(a^R \mid b)\right|.$$

The right-hand side is $\mathrm{TV}(\pi^R_c(\cdot \mid b), \pi^R_{c'}(\cdot \mid b))$, which proves the result. □

B.2 Proof of Proposition 1

Sufficiency of the filtering posterior and the existence of an optimal belief-measurable policy are standard results for POMDPs. The belief update is

$$b_{t+1}(s') \propto \Omega(o_{t+1} \mid s') \sum_s \mathcal{T}_c(s' \mid s, a^B_t) b_t(s).$$

Under Assumption 2, the dependence on c enters only through $\mathcal{T}_c$, which by Eq. (1) depends on the adversary strategy π^R_c. Hence b_t is a sufficient statistic for optimal control and evolves according to a Bayes operator of the stated form. □

The proposition concerns the exact filtering posterior of the POMDP. The RSSM interpretation in Appendix A is approximate: CyberWorld does not assume that $s_t = [h_t; z_t]$ recovers b_t, nor do the RSSM losses establish such recovery. Rather, $p_\phi(z_t \mid h_t)$ and $q_\phi(z_t \mid h_t, e_t)$ play roles analogous to predictive and observation-conditioned latent distributions, respectively, providing a learned recurrent representation on which the policy is conditioned.

B.3 Proof of Theorem 1

Step 1: One-step model error. Introduce the hybrid kernel

$$\widetilde{\mathcal{T}}(s' \mid s, a^B) = \sum_{a^R} \widehat{\mathcal{T}}_{\text{mech}}(s' \mid s, a^B, a^R)\pi^R_{c'}(a^R \mid b^R(s)),$$

which combines the learned mechanism model with the true adversary strategy. By the triangle inequality,

$$\mathrm{TV}\Big(\mathcal{T}_{c'}, \widehat{\mathcal{T}}_{c'}\Big) \le \mathrm{TV}\Big(\mathcal{T}_{c'}, \widetilde{\mathcal{T}}\Big) + \mathrm{TV}\Big(\widetilde{\mathcal{T}}, \widehat{\mathcal{T}}_{c'}\Big).$$

The first term is bounded by $\varepsilon_{\text{mech}}$ because it averages mechanism-model deviations under $\pi^R_{c'}(\cdot \mid b^R(s))$. The second term is bounded by $\varepsilon_{\text{strat}}(c')$ by the same mixture-contraction argument as Lemma 1. Therefore,

$$\mathrm{TV}\Big(\mathcal{T}_{c'}(\cdot \mid s, a^B), \widehat{\mathcal{T}}_{c'}(\cdot \mid s, a^B)\Big) \le \varepsilon_{\text{mech}} + \varepsilon_{\text{strat}}(c').$$

Step 2: Simulation bound. Let

$$\varepsilon \doteq \varepsilon_{\text{mech}} + \varepsilon_{\text{strat}}(c').$$

For any stationary policy π, $\|V^\pi\|_\infty \le R_{\max}/(1-\gamma)$. A standard simulation-lemma argument (Kearns & Singh, 2002) gives

$$\left|V^\pi_{\mathcal{M}_{c'}} - V^\pi_{\widehat{\mathcal{M}}_{c'}}\right| \le \sum_{t\ge 0} \gamma^{t+1}\, 2\varepsilon\, \frac{R_{\max}}{1-\gamma}.$$

Summing the geometric series yields

$$\left|V^\pi_{\mathcal{M}_{c'}} - V^\pi_{\widehat{\mathcal{M}}_{c'}}\right| \le \frac{2\gamma R_{\max}}{(1-\gamma)^2}\varepsilon.$$

Step 3: Policy suboptimality. Let

$$\widehat{\pi} \in \arg\max_\pi V^\pi_{\widehat{\mathcal{M}}_{c'}}.$$

Then

$$\begin{aligned} V^{\pi^\star}_{\mathcal{M}_{c'}} - V^{\widehat{\pi}}_{\mathcal{M}_{c'}} = &\left(V^{\pi^\star}_{\mathcal{M}_{c'}} - V^{\pi^\star}_{\widehat{\mathcal{M}}_{c'}}\right) \\ &+ \left(V^{\pi^\star}_{\widehat{\mathcal{M}}_{c'}} - V^{\widehat{\pi}}_{\widehat{\mathcal{M}}_{c'}}\right) \\ &+ \left(V^{\widehat{\pi}}_{\widehat{\mathcal{M}}_{c'}} - V^{\widehat{\pi}}_{\mathcal{M}_{c'}}\right). \end{aligned}$$

The middle term is non-positive by optimality of $\widehat{\pi}$ in $\widehat{\mathcal{M}}_{c'}$, and the first and third terms are each bounded by Step 2. Hence

$$V^{\pi^\star}_{\mathcal{M}_{c'}} - V^{\widehat{\pi}}_{\mathcal{M}_{c'}} \le \frac{4\gamma R_{\max}}{(1-\gamma)^2}\left(\varepsilon_{\text{mech}} + \varepsilon_{\text{strat}}(c')\right).$$

□

B.4 Proof of Remark 1

Fix a policy π_c and compare its value under $\mathcal{M}_c$ and $\mathcal{M}_{c'}$. Applying the simulation argument from Theorem 1 together with Lemma 1 gives

$$\left|V^{\pi_c}_{\mathcal{M}_c} - V^{\pi_c}_{\mathcal{M}_{c'}}\right| \le \frac{2\gamma R_{\max}}{(1-\gamma)^2} \sup_s \mathrm{TV}\big(\pi^R_c(\cdot \mid b^R(s)), \pi^R_{c'}(\cdot \mid b^R(s))\big).$$

This establishes sensitivity of a fixed policy to a change in adversary strategy. The inequality does not imply a particular sample complexity for retraining or adaptation under c'. Its role here is to contrast the fixed-policy sensitivity bound with the explicit mechanism–strategy decomposition available in Theorem 1. □

C Crossing Statistics and Performance Profiles at $N{=}15$

This appendix provides the measurements behind RQ1 in Section 4.2, in particular the two-to-three-orders-of-magnitude gap to PPO and the first-crossing comparison with GNN-PPO. Figure 3 reports, for each attacker, the environment step at which each method first exceeds the strategy-agnostic control. Table 2 lists the exact values with seed counts, and Figure 8 aggregates them across attackers as a performance profile: for each method, the fraction of all (attacker, seed) runs that have reached the control by environment step τ. A profile that rises earlier and higher denotes a more sample-efficient defender, and a run that never crosses remains censored at its budget.

Table 2: Environment steps to first reach the `deploy_then_stop` control, 15 hosts, all methods from scratch. Mean ± sd over 3 seeds; four text/multimodal cells use 2 seeds because the third run did not complete; (k/n): seeds that crossed when not all did. Bold: lowest mean in the column. PPO: raw 30-episode checkpoint evaluations, 3 seeds. GNN-PPO: the graph arm's observation and GAT encoder trained with PPO instead of the world model, online training return, 3 seeds. DFSImpact, BruteForce and Inactive are omitted: their control is 0, so no crossing is defined (all arms end at a final return of 0 there).

	BFSServerDowntime	ServerDowntime	Exfiltration	BFSExfiltration
PPO (from scratch)	2,310,400 ± 640,000 (2/3)	3,078,400 (1/3)	2,630,400 ± 448,000 (2/3)	NEVER (0/3)
GNN-PPO (from scratch)	**4,017 ± 821**	25,875 ± 4,025 (2/3)	11,367 ± 9,182	9,117 ± 4,128
ours: vector	4,933 ± 633	3,633 ± 62	**3,900 ± 283**	31,650 ± 12,450 (2/3)
ours: graph	5,525 ± 575 (2/3)	**3,600 ± 41**	4,600 ± 41	15,750 ± 12,960
ours: vec+graph	6,433 ± 1,462	4,183 ± 554	5,533 ± 1,897	22,325 ± 12,525 (2/3)
ours: text	4,900 ± 750 (2/3)	5,283 ± 1,109	4,000 ± 400	**7,850 ± 1,203**
ours: multimodal	10,100 ± 0 (1/3)	4,600 ± 100	5,525 ± 1,125	12,200 ± 0 (1/2)

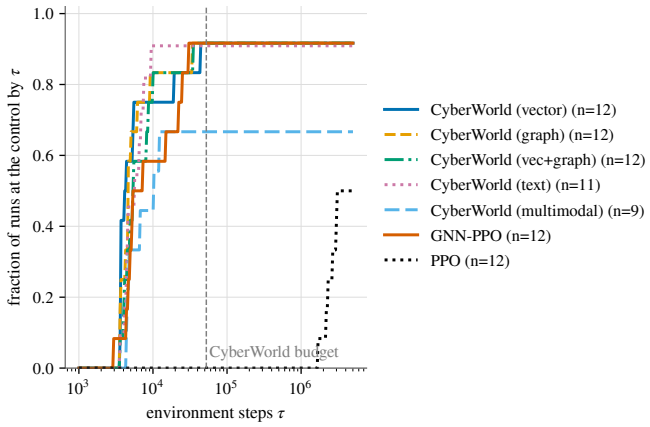


Figure 8: Performance profile at $N{=}15$: fraction of (attacker, seed) runs that have reached the control by environment step τ, pooled over the four scoreable attackers. Runs that never cross are censored at their budget. Dashed vertical line: the 52.5k-step budget of the world-model runs.

The three numeric representations reach the control in 92% of their runs within the 52.5k-step budget, and the text representation does so in all but one of its runs. In contrast, PPO's profile remains at zero until approximately 2M steps and reaches only 5 of 12 runs (42%) by the end of its 3.2M-step budget. The multimodal representation plateaus lowest among the world-model arms because only one seed crosses on each of the two BFS attackers. GNN-PPO, the representation-matched baseline trained in the same loop, also reaches 92%, and its profile rises at similar steps to those of the numeric world-model arms; relative to GNN-PPO, the clearest advantage of the world model therefore lies in post-crossing retention rather than initial crossing speed, as Table 3 shows. PPO is particularly seed-sensitive on ServerDowntime: only one of its three seeds ever exceeds the control, whereas every vector and graph seed does so within 4k steps.

D Sensitivity of the Crossing Criterion to the Smoothing Window

This appendix examines how the primary trailing-10 crossing metric changes under alternative smoothing windows and tests whether the main sample-efficiency conclusion depends on this convention. The first crossing is defined on the trailing-10 episode return. Because the environment contains a rare $-5{,}000$ penalty, a short window makes the return curves volatile and could in principle shift the crossing. Figure 9a shows one trajectory (vector, BFSServerDowntime, seed 1) as raw returns and at trailing-10, trailing-50, and trailing-100 resolution; Figure 9b recomputes the first-crossing step of the vector arm on every attacker under windows $W \in \{10, 25, 50, 100\}$, as mean $\pm$ sd over seeds, annotated with the number of seeds that still cross whenever it falls below three.

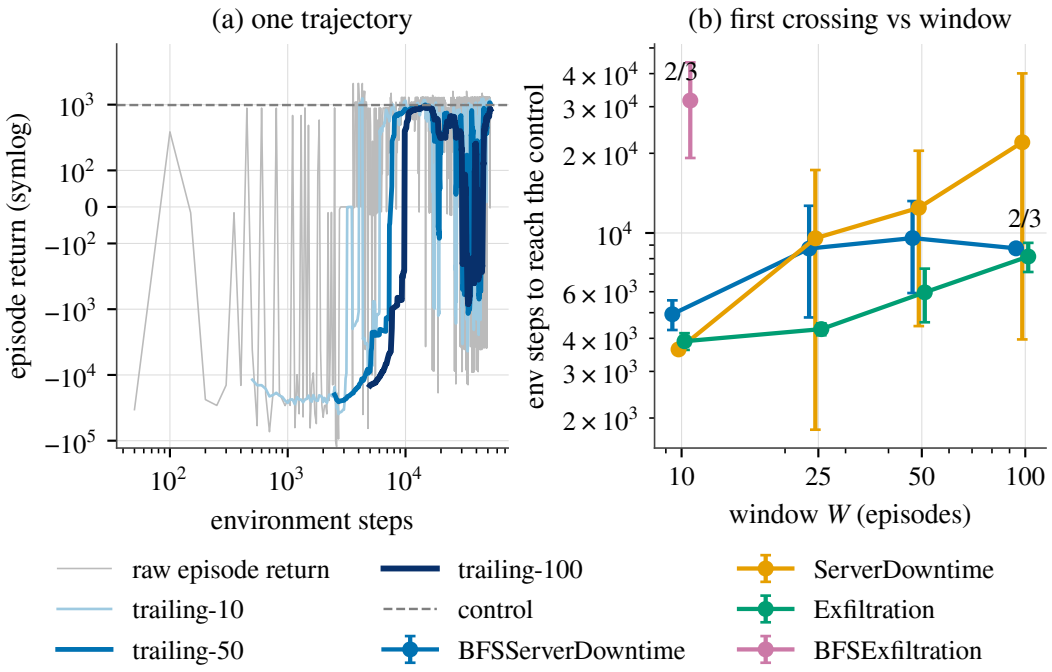


Figure 9: Smoothing-window sensitivity. (a) One trajectory at four temporal resolutions. (b) First-crossing step of the vector arm vs. the window W used to define it (mean $\pm$ sd over 3 seeds; $k/3$ where fewer seeds cross).

In these runs, widening the window generally delays the crossing, but by at most a factor of two on ServerDowntime, Exfiltration, and BFSServerDowntime between $W{=}10$ and $W{=}50$, against a 10^2–10^3 gap to PPO. On BFSExfiltration, where the vector arm settles close to the control after crossing, only $W{=}10$ registers a crossing, and only for two of three seeds; wider windows do not, which indicates that the behavior is acquired but not consistently held. The headline conclusion is therefore independent of the smoothing convention, whereas the BFSExfiltration result depends on it, and the main text reports that cell together with its seed count.

E Per-Representation Learning Dynamics and Post-Crossing Retention

This appendix expands the representation comparison of RQ2 in Section 4.2: it provides the complete learning curves summarized in Figure 4 and Figure 5, and the post-crossing statistics on which the retention argument of RQ1 (the contrast between the world-model arms and GNN-PPO) rests. Figure 4 summarizes each representation by its crossing step per attacker. This section provides the underlying evidence: Figure 10 shows the complete learning curve of every representation on every scoreable attacker, Figure 11 presents the same crossing steps as a lookup grid, and Table 3 reports what happens after the crossing, namely the share of post-crossing episodes above the control, the

median return of the last 200 training episodes, and the rate of decoy-limit penalties in those episodes, for every method including PPO and GNN-PPO.

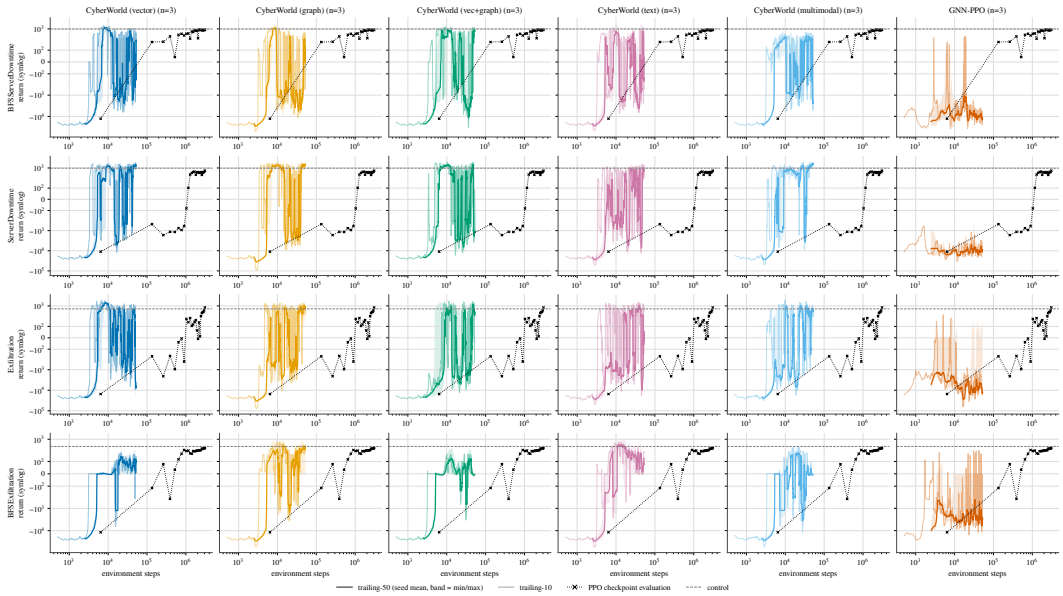


Figure 10: Learning curves of the five representations and GNN-PPO (columns) on the four scoreable attackers (rows) at $N{=}15$. Thick: trailing-50 seed mean with min/max band; thin: trailing-10; black $\times$: PPO checkpoint evaluations; dashed: control.

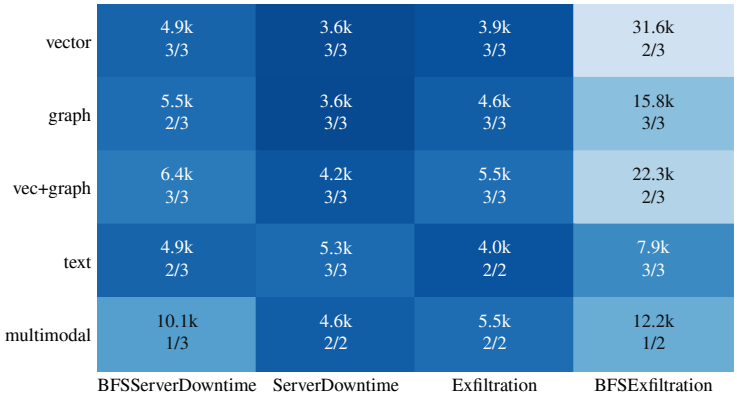


Figure 11: Environment steps to first reach the control per representation and attacker at $N{=}15$ (darker = fewer; k/n = seeds that reached it; grey = no seed reached it within the 52.5k-step budget).

The crossing is the same event for every arm: a transition from the random-policy return of about -10^4 to the level of the control within a few hundred episodes. The arms differ in retention, where the comparison with PPO is mixed; Table 3 accordingly carries no bold entries, since the two budgets differ by a factor of 60. On ServerDowntime the vector and graph arms hold the control in about half of their post-crossing episodes, with a late median above it and fewer than two penalties per 100 episodes, which is on par with PPO after 3M steps. On Exfiltration the vector and text arms incur the decoy-limit penalty in more than half of their late episodes, whereas the graph and multimodal arms do not. On BFSExfiltration the vec+graph, text, and multimodal arms settle at the do-nothing return of 0 after a brief excursion above it, which corresponds to a collapsed rather than a stable policy. The multimodal arm holds the control with the fewest penalties on ServerDowntime and Exfiltration, at the cost of crossing later or on fewer seeds than the numeric arms. GNN-PPO is the clearest instance of a crossing that does not constitute a defence: its late median return is $-3{,}700$

Table 3: Retention after the first crossing, 15 hosts. Each cell: share of post-crossing episodes above the control / median return of the last 200 training episodes / decoy-limit penalties per 100 of those episodes. CyberWorld statistics are taken at the end of its 52.5k-step budget with the stochastic training policy; PPO's are its 30-episode checkpoint evaluations after its own crossing and its last three checkpoints, i.e. after up to 3.2M steps. The two columns therefore measure retention under very different budgets and are not bolded. Seeds are pooled; four text and multimodal cells have two seeds.

	BFSServerDowntime	ServerDowntime	Exfiltration	BFSExfiltration
PPO	56% / 899 / 2.6	53% / 886 / 8.9	39% / 290 / 5.6	– / −1 / 0.0
GNN-PPO	23% / −3,700 / 237.2	28% / −3,700 / 220.8	12% / 0 / 167.8	17% / 0 / 66.0
CyberWorld (vector)	33% / 888 / 23.7	51% / 1,192 / 0.8	30% / 285 / 55.0	21% / 0 / 0.5
CyberWorld (graph)	30% / 388 / 14.7	49% / 1,181 / 1.7	32% / 190 / 2.7	37% / −4 / 0.0
CyberWorld (vec+graph)	43% / 1,092 / 15.7	51% / 1,281 / 35.5	38% / 582 / 14.8	8% / −3 / 0.0
CyberWorld (text)	38% / 886 / 19.7	35% / 900 / 57.0	26% / 192 / 72.2	26% / −2 / 0.0
CyberWorld (multimodal)	14% / 490 / 23.0	38% / 1,186 / 0.5	28% / 89 / 3.2	17% / −8 / 0.0

on both ServerDowntime attackers and 0 on both Exfiltration attackers, with 66 to 237 decoy-limit penalties per 100 late episodes, that is, the policy that crossed continues to deploy beyond the cap.

F Scaling Analysis: Crossing Statistics, Learning Curves, and Computational Cost for $N \in \{25, 50, 100\}$

This appendix provides the evidence behind RQ3 in Section 4.2: the exact values summarized in Figure 7 and Figure 6, every learning curve at $N \in \{25, 50, 100\}$, and the computational cost that the limitations in the Summary paragraph and in Section 4.2 refer to. Figure 7 fits the cost of adaptation against N over the runs that crossed. Table 1 in the main text gives the crossing step per attacker and size with seed counts, including PPO at every size at which it was trained. Figure 12 reports the cost of each representation as N grows, in terms of peak GPU memory, training seconds per environment step, and the size of the intermediate representation, and Figure 13 shows every learning curve for every attacker and size, including the three attackers whose control is 0 and for which no crossing is defined.

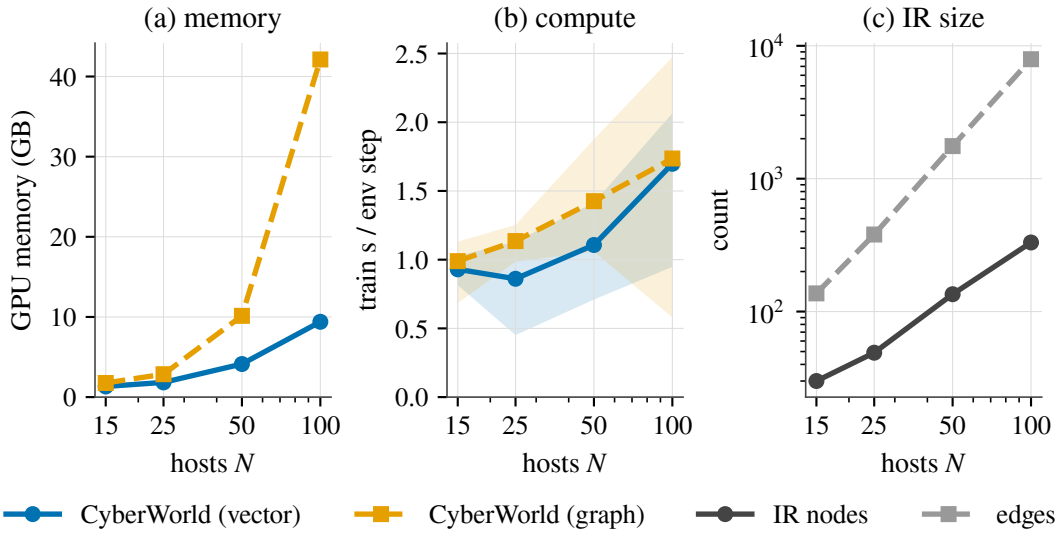


Figure 12: Compute scaling of the two numeric representations. (a) Peak GPU memory and (b) training seconds per environment step (median over runs, band = min/max); (c) nodes and edges of the intermediate representation vs. N.

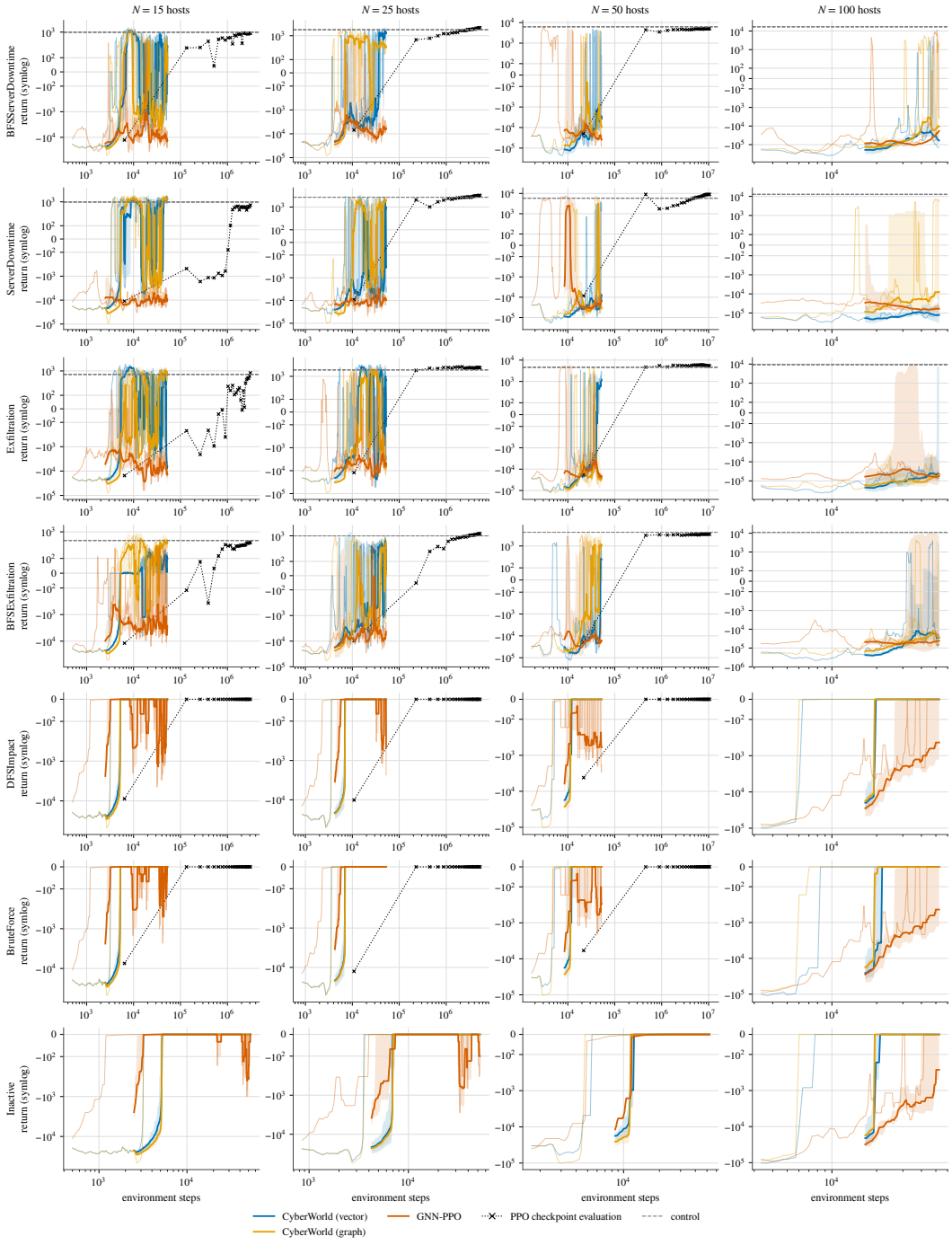


Figure 13: Return vs. environment steps for every attacker (rows) and network size (columns): CyberWorld vector, graph, and GNN-PPO (trailing-50 seed mean with min/max band, thin = trailing-10), PPO checkpoint evaluations, and the control.

At $N{=}25$ the curves resemble those at $N{=}15$ with a longer prefill. At $N{=}50$ the crossing moves to 16–38k steps and the seed spread around the control widens in absolute return, because the control itself is 5–6$\times$ larger. At $N{=}100$ the 52.5k-step budget covers only 157 episodes, the curves are still rising when the budget ends, and the wide seed bands indicate that crossing becomes highly seed-dependent; the power-law fit in Figure 7 uses only the runs that did cross and is therefore a statement about the cost of the successful runs rather than a guarantee of success. The graph representation's advantage on the topology-following attacker is obtained at the cost of memory, 43 GB at $N{=}100$ against 10 GB for the vector representation, at similar seconds per environment step. GNN-PPO's curves in Figure 13 exhibit the pattern of Table 1 at every size, an early excursion above the control followed by a return that settles far below it, reaching $-9{,}000$ to $-11{,}000$ on the ServerDowntime attackers at $N{=}50$.

G Open-Loop Imagination of the Learned World Model

Figure 14 visualizes open-loop trajectories imagined by the learned world model. The model consumes the host representation and adjacency defined in Section 3.2. The $192{\times}256$ semantic canvas shown here is an exactly invertible rendering of the four reconstruction-role bits: hosts are drawn as circles, decoys as squares, and node colour encodes the observable role state. Because the rendering is invertible, decoded node states can be repainted onto the same canvas as the observed states, making the two rows directly comparable.

The top row shows a held-out BFSExfiltration episode. The graph arm receives observations for the first four displayed steps. After the fourth observed step, the latent state is rolled forward open-loop using the true defender actions but without further environment observations, and each imagined latent state is decoded and repainted. Hosts whose imagined state differs from the observed state are marked with red rings, while steps on which the adversary touched a decoy are marked with yellow borders.

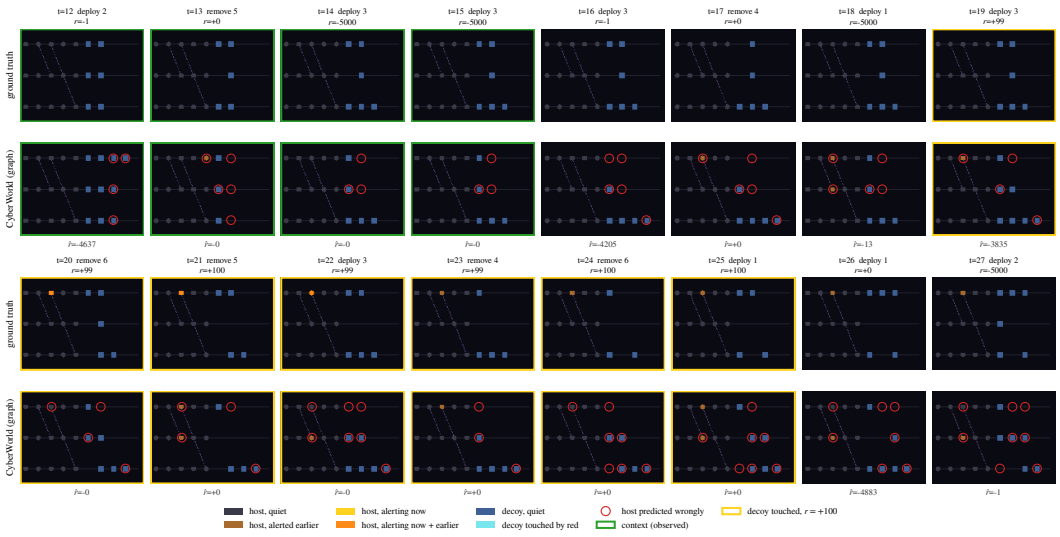


Figure 14: Open-loop imagination of the graph arm on a held-out BFSExfiltration episode. Top: the observed canvas at each step, annotated with the defender action that led to the step and the true reward. Bottom: the decoded model state, obtained from the posterior during the four observed steps (green border) and from the RSSM prior under the true defender actions thereafter, with the predicted reward shown below each canvas. Red rings mark hosts whose imagined state differs from the observed state; yellow borders mark steps on which the adversary touched a decoy.

The topology is fixed by construction throughout the visualization, while the model approximately preserves the observable host-state layer during the open-loop rollout. Its errors are concentrated in two aspects of the decoded state. First, the model can misplace deployed decoys among reserved slots within the same subnet. These slots are implementation-level bindings that are only indirectly

reflected in the observation, so an imagined rollout may assign a deployed decoy to a neighbouring slot while preserving the broader subnet-level structure. Second, the model can fail to anticipate new alert activity. Once the rollout becomes fully open-loop, the prior receives no new observation indicating that adversarial activity has reached another host, and the decoded state can therefore continue to predict that host as quiet after the observed trajectory begins to alert there.

The reward predictions expose the same limitation. Decoy-touch events are sparse and depend on latent adversarial progress that is only partially revealed through subsequent alerts. During the open-loop rollout, the predicted reward often remains near zero at these events, and large negative rewards can be temporally misaligned with the observed decoy-limit penalties. Thus, the figure should not be interpreted as demonstrating pixel-perfect or event-perfect long-horizon prediction. Instead, it shows that the learned latent dynamics preserve substantial coarse host-state structure while becoming less reliable for events that depend on unobserved adversarial progress and precise reward timing.

Despite these open-loop errors, the learned policy reaches the control in the corresponding experiments. This suggests that precise long-horizon reconstruction of every latent event is not necessary for acquiring the defensive behavior observed in our evaluation, although improving prediction of adversary-dependent events remains an important direction for future work.