Journal of Reliable Intelligent Environments (2026) 12:14
https://doi.org/10.1007/s40860-026-00278-2

ORIGINAL ARTICLE

ACID: beta-testing active inference for active cyber-defence

Jesús F. Cevallos-Moreno[1] · Alessandra Rizzardi[1] · Sabrina Sicari[1] · Alberto Coen-Porisini[1]

Received: 3 October 2025 / Accepted: 14 September 2026
© The Author(s) 2026

Abstract
Active cyber-defence employs anticipatory techniques to proactively mitigate cyber threats. Among such techniques, the most relevant one consists in the continuous updating of cyber-threat intelligence. In this respect, there is a trade-off between the cost of intelligence assets and the benefits derived from their exploitation. While the optimisation of this trade-off has recently been tested with conventional reinforcement learning, this paper investigates the suitability of active inference-based agents for addressing the same challenge. A novel mathematical formulation for Active Inference over fully-observable Markov Decision Processes is provided in this paper, and four different off-policy bootstrapped neural implementations are given. Empirical evaluation highlights that deep active inference agents not only achieve a competitive performance with conventional deep reinforcement learning systems, but also that, crucially, some formulations of epistemic gain can indeed foster exploration and provide a discernible performance boost in terms of return maximisation. This work contributes to understanding the empirical boundaries of Active Inference's unique mechanisms in complex, real-world-inspired environments, informing future architectural design for deep active inference agents.

Keywords Active inference · Deep learning · Network intrusion detection

1 Introduction

Realistic Machine Learning (ML)-based Network Intrusion Detection (NID) faces two primary research challenges. First, it must effectively model an open-world setting where novel classes of attacks continuously attempt to penetrate the network [1]. Second, it needs to quantify and minimise the cost of acquiring training signals essential for keeping competitive detection accuracy [2].

Numerous studies addressing Zero-day network Attack (ZdA) detection focus on strategies for tackling the first challenge. These methods often employ techniques like extreme value theory[3], inductive biases from algorithmic reasoning [4], or sophisticated continual learning approaches [5]. However, most of these approaches neglect to model the significant cost associated with acquiring labelled data. For any supervised-learning-based NID, this cost is directly tied to Cyber-Threat-Intelligence (CTI) acquisition.[1]

The open-source TIGER Reinforcement Learning (RL) environment [6] was developed to address this gap. It aims to optimise the cost of acquiring training data against the benefits derived from its utilisation for defensive purposes. TIGER simulates a network environment complete with attackers and honeypots. Within this simulation, a centralised defensive agent can be trained to strategically exploit its current CTI to either accept or block traffic, or to enrich its CTI by acquiring additional training classes. Thus, TIGER effectively models an implicit meta-learning setting for CTI optimisation, where an agent must learn how to acquire CTI in order to improve its attack detection capabilities. In other words, the agent needs to *learn how to (better) learn* attack detection. Note that, although CTI acquisition *indirectly* contributes to optimising detection performance, it may turn out to be *essential* to converge to sufficiently strong and robust detection policies in the long run

✉ Sabrina Sicari
sabrina.sicari@uninsubria.it

Jesús F. Cevallos-Moreno
jf.cevallosmoreno@uninsubria.it

Alessandra Rizzardi
alessandra.rizzardi@uninsubria.it

Alberto Coen-Porisini
alberto.coenporisini@uninsubria.it

[1] Dipartimento di Scienze Teoriche ed Applicate, Università degli Studi dell'Insubria, Via O.Rossi 9, 21100 Varese, Italy

[1] Note that, for any ML-based NID system, training data inherently constitute CTI by definition, as they associate patterns of observables with specific attacks.

Published online: 30 September 2026

While standard Reinforcement Learning (RL) algorithms, such as Double-Deep Q-Network (DDQN), have been explored within TIGER, the inherent meta-learning nature of this environment raises a crucial question: is it advantageous to leverage alternative agency frameworks that are specifically designed to address the hierarchical goal-structure of meta-learning tasks?

To answer this question, a general framework is considered which has recently garnered significant interest from the deep learning research community: Active Inference (Act-Inf) [7]. Act-Inf is fundamentally a least-action principle governing agency and learning in any system owning a homeostatic goal and a generative model of its environment. This framework is grounded in the Free Energy (minimisation) Principle (FEP) [8], which postulates that systems minimise the Bayesian Surprisal of their generative model. This minimisation is achieved primarily through two 'levers': acting on the environment and calibrating the generative model itself [9]. As a non-substantive process theory, Act-Inf has been employed to explain a wide range of phenomena across diverse fields, including biology [10], psychology [11], psychiatry [12], social sciences [13], computational neuroscience [14], and, notably, artificial intelligence [15], among others [16].

The peculiar characteristic of Act-Inf, that renders it particularly suitable for meta-learning problems like the one presented in TIGER, is its explicit pursuit of epistemic gain, or equivalently, the minimisation of uncertainty. This epistemic goal is pursued in Act-Inf concurrently with the maximisation of a pragmatic goal, which is the sole focus of conventional RL agents. Therefore, this paper investigates the efficacy of adopting the Active Inference paradigm for the Network Intrusion Detection and Cyber-Threat Intelligence lifecycle-optimisation problem within the challenging TIGER environment. To address this, our work presents the following **main contributions**:

- A rigorous formulation for Active Inference in fully observable Markov Decision Processes is provided. While Act-Inf is conventionally modelled within Partially Observable Markov Decision Processes, a novel mathematical derivation for its application in any fully observable settings is presented, which is essential for environments like TIGER.
- Four distinct off-policy bootstrapped neural implementations of Deep Active Inference (DAI) agents are developed and rigorously evaluated to offer a meticulous analysis and comparative evaluation of diverse modelling strategies concerning the Free Energy-related functionals.
- Through extensive empirical evaluation in the TIGER environment, useful insights are provided into the practical utility of DAI agents in high-dimensional cyber-defence. Our experiments show that, while the pragmatic component of DAI effectively aligns with conventional reward-based learning, the inclusion of epistemic gain terms does yield a performance advantage over conventional Deep Reinforcement Learning (DRL) agents.

Overall, our results contribute to understanding the empirical boundaries and nuanced effects of Active Inference's unique components in cyber-defensive environments. To the best of the authors' knowledge, this paper represents the first exploration of applying the Free Energy Principle in the domain of active cyber defence.

Paper Structure. The remainder of this paper is organised as follows. Section 2 reviews related work and introduces preliminary concepts. Section 3 presents an original mathematical framework for Active Inference in fully observable Markov Decision Processes (MDPs). Section 4 discusses the specific neural implementation details of such a mathematical framework. Section 5 revisits the TIGER environment to establish a suitable experimental setting for testing Active Inference agents. Section 6 documents and discusses the experimental results. Finally, Sect. 7 provides conclusive remarks and outlines future work.

2 Related works

This work intersects with three main research areas: open-set network intrusion detection, deep active inference, and cyber-defence testbeds.

2.1 Open-set network intrusion detection

Anomaly detection is the de facto standard strategy for ZdA detection [17]. This coupling arises from the inherently unsupervised nature of both problems. However, within network intrusion detection, a significant challenge that anomaly-oriented approaches leave unaddressed is the accurate classification of detected anomalies as either benign or malicious traffic. To address this, various techniques have been employed to augment anomaly detection with inferences regarding the benign or malicious nature of anomalies, even in unsupervised settings. These include ensemble techniques [18], confidence-based methods [19], approaches leveraging algorithmic inductive biases [4, 20], and other methods related to continual learning [21, 22].

ZdA detection can also be framed as anomaly detection under distribution shifts. Specifically, it is equivalent to closed-set traffic classification when the class-specific patterns are dynamic over time [23]. Beyond traditional periodical retraining [24, 25], this problem has been addressed through various advanced techniques, including metric-

based meta learning [26, 27], weight-consolidation schemes [23, 28], self-supervised pre-training [29], and memory-replaying [30], among others [5]. This work builds upon the metric-based meta-learning setting introduced in [26], extending its collective anomaly detection capabilities with two active features: online augmentation of the supervisory signal set and the implementation of mitigation actions against anomalous traffic.

2.2 Deep active inference

Active inference agents, inherently equipped with a generative model, are typically trained within Partially Observable Markov Decision Processes (POMDPs). However, adaptations of the Act-Inf framework for fully observable Markov Decision Processes (MDPs) have also been explored [31, 32]. Early algebraic implementations of Act-Inf-based agents for POMDPs were limited to discrete and low-dimensional state and action spaces [33]. More recent deep learning implementations of Act-Inf leverage search algorithms, like Monte Carlo Tree Search (MCTS) [34] or behavioural cloning techniques [35] to extend its application to continuous state spaces [36]. Our work adopts the temporal difference setting from [31] to mitigate the computational burden associated with training Act-Inf agents in a *de novo* learning context [37].

2.3 Cyber-defence environments

A key prerequisite for researchers contributing to ML-based network intrusion detection is access to robust testbeds [38]. Lots of existing cyber-defence environments employ simplified settings, often using scalar features to represent the defensive and attacking strengths of network nodes [39, 40]. More realistic environments, conversely, leverage trace-driven simulation for training reinforcement learning agents [41–44]. Among these more realistic environments is [6], which adds an action/perception interface specifically designed to model CTI management. This work investigates whether and how agents based in Act-Inf can interact with this interface to more effectively manage CTI compared to conventional RL schemes.

3 A mathematical framework for active inference on fully-observable MDPs

While the standard formulation of the Active Inference framework has been widely studied, it has also faced several critiques and alternative reformulations within the research community. Beyond its implicit assumptions [37, 45], a key challenge for exporting Act-Inf to RL settings is the lack of a standard formulation for fully observable settings. To address this, this section provides a rigorous and explicit exposition of active inference specifically tailored for fully observable MDPs. Our formulation is structurally analogous to the standard POMDP setting in [7], ensuring theoretical consistency.

3.1 The fully-observable Markov decision process

Act-Inf conceptualises agency and learning as an inference problem. This framework posits an agent owning two key characteristics: first, it is equipped with a generative model of its environment; second, it employs an action policy to influence that environment. In a fully observable MDP, the agent's perceptive-active model can be formally expressed as $\mathcal{Q}(o_\tau, o_{0:\tau-1}, a_{0:\tau-1})$. The objective of this model is to become isomorphic to the joint distribution of the environmental stochastic process and its associated target policy, denoted as $P(o_\tau, o_{0:\tau-1}, a_{0:\tau-1})$. This learning objective is captured by the following relation:

$$\underbrace{\mathcal{Q}(o_\tau, \underline{o}, \underline{a})}_{\text{(learning) model}} \leftarrow \underbrace{P(o_\tau, \underline{o}, \underline{a})}_{\text{(visible) process}} \tag{1}$$

Here, $\underline{o}$ denotes the sequence of past observations $o_{0:\tau-1}$, and $\underline{a}$ represents the sequence of past actions, $a_{0:\tau-1}$. Crucially, both the model and the process can be factorised by leveraging the Markovian assumption. This implies that the observation at time τ, denoted as o_τ, depends solely on the previous observation, $o_{\tau-1}$, and the preceding action, $a_{\tau-1}$:

$$\mathcal{Q}(o_\tau, \underline{o}, \underline{a}) := P(o_0) \prod_{t=1}^{\tau} \underbrace{\mathcal{Q}(o_t|a_{t-1}, o_{t-1})}_{\text{perception model}} \underbrace{\mathcal{Q}(a_{t-1}|o_{t-1})}_{\text{learned action policy}} \tag{2}$$

$$P(o_\tau, \underline{o}, \underline{a}) := P(o_0) \prod_{t=1}^{\tau} \underbrace{P(o_t|a_{t-1}, o_{t-1})}_{\text{transition process}} \underbrace{P(a_{t-1}|o_{t-1})}_{\text{normative action policy}} \tag{3}$$

In this work, the generative model refers to the agent's internal probabilistic model of the environment's transition dynamics together with its own action selection policy. This model is implemented as a parametric model learned via deep neural networks and is continuously refined online during agent–environment interaction. At the start of training, the generative model is randomly initialized and therefore only weakly accurate. It is then progressively improved through gradient-based updates driven by the agent's learning objectives, as detailed in Sect. 4. Because the environment is fully observable and the transition dynamics are relatively stable, a reasonably accurate generative model can be learned concur-

rently with the policy. Variants of the proposed agents differ in how strongly they rely on this learned model (see Sect. 4).

3.2 The variational free energy

The primary goal of the Act-Inf framework is to invert the perceptive-active model in (2) to infer the posterior distribution $P(\underline{o}|o_\tau, \underline{a})$. Since this exact inference is often intractable, it is approximated by modelling a variational posterior distribution $\mathcal{Q}_{o_\tau}(\underline{o}|\underline{a})$ such that $\mathcal{Q}_{o_\tau}(\underline{o}|\underline{a}) \approx P(\underline{o}|o_\tau, \underline{a})$. Here, the action sequence $\underline{a}$ is treated as given, and the variational posterior captures the distribution over past observations conditioned on it after updating on the current observation o_τ. For ease of notation, the conditioning subscript o_τ is omitted in the remainder of this section whenever it is clear from context. Within this framework, the Variational Free Energy (VFE), denoted as $\mathcal{F}$, serves as a computationally tractable upper bound on the model's *surprisal* $-\log P(o_\tau|\underline{a})$. It is defined as the KL divergence between the variational posterior and the joint distribution of past and current observations given actions:

$$\mathcal{F}[\mathcal{Q}, P] := D_{\mathrm{KL}}\left[\, \mathcal{Q}(\underline{o}|\underline{a}) \,\middle||\, P(o_\tau, \underline{o}|\underline{a}) \right] \tag{4}$$

Here, $D_{\mathrm{KL}}[\, Q||\, P]$ denotes the Kullback–Leibler (KL) divergence between two distributions Q and P, defined as $\mathbb{E}_Q[\log Q - \log P]$. Thus, the previous expression can be decomposed as:

$$\begin{aligned} \mathcal{F}[\mathcal{Q}, P] &= \mathbb{E}_{\mathcal{Q}(\underline{o}|\underline{a})}\left[\log \mathcal{Q}(\underline{o}|\underline{a}) - \log P(o_\tau, \underline{o}|\underline{a})\right] \\ &= \underbrace{D_{\mathrm{KL}}\left[\, \mathcal{Q}(\underline{o}|\underline{a}) \,\middle||\, P(\underline{o}|\underline{a}) \right]}_{\text{complexity}} \\ &\quad - \underbrace{\mathbb{E}_{\mathcal{Q}(\underline{o}|\underline{a})}\left[\log P(o_\tau|\underline{o}, \underline{a})\right]}_{\text{accuracy}} \end{aligned} \tag{5}$$

Minimising the VFE thus requires balancing two competing objectives. The *accuracy* term, $\mathbb{E}_{\mathcal{Q}(\underline{o}|\underline{a})}[\log P(o_\tau|\underline{o}, \underline{a})]$, reflects the least action principle [8]: the variational posterior should assign high probability to observation sequences that best explain the current observation o_τ under the generative model. The *complexity* term, $D_{\mathrm{KL}}\left[\, \mathcal{Q}(\underline{o}|\underline{a}) \,\middle||\, P(\underline{o}|\underline{a}) \right]$, aligns with the maximum entropy principle [46]: it penalises the variational posterior for deviating from the prior $P(\underline{o}|\underline{a})$, thereby regularising the perceptive model and preventing overfitting. Together, these two terms define the canonical accuracy-complexity trade-off of variational inference, which the agent resolves by optimising the perceptive model $\mathcal{Q}(\underline{o}|\underline{a})$.

The perceptive model can be optimised through minimising the VFE functional as defined in (4). This minimisation is achieved by adjusting the perceptive model $\mathcal{Q}(\underline{o}|\underline{a})$ so as to simultaneously maximise accuracy and minimise complexity. The action policy, $\mathcal{Q}(\underline{a}|\underline{o})$, is optimised separately through an expectation of the VFE functional over a path of predicted actions and observations, which is called the Expected Free Energy.

3.3 The expected free energy

While the accuracy and complexity terms in (5) allow the perceptive model to be optimised through VFE minimisation, the action policy requires a separate objective. In the Act-Inf framework, the normative or target policy is implicitly defined by the Expected Free Energy (EFE), and is denoted as $\mathcal{G}$. The EFE is a functional representing the future expectation of the VFE in (4) over a predicted path of future actions and their corresponding observations.

It is important to note that the VFE in (4) is a functional of the model ($\mathcal{Q}$) and process (P) distributions over a sequence of past actions ($\underline{a}$) and observations ($\underline{o}$). In contrast, the EFE extends this concept into the future. An expectation of the VFE over h future actions and observations implicitly considers what *will become the 'past' h* observations and actions from the perspective of future time steps. Specifically, for a planning horizon of h, the EFE is defined over a sequence of h future actions, $\overline{a} := a_{\tau:\tau+h}$, and their corresponding observations, $\overline{o} = o_{\tau:\tau+h}$. The conceptual relationship between the EFE and the expectation of the VFE over $\overline{a}$ and $\overline{o}$ is expressed as:

$$\mathcal{G}[\mathcal{Q}, C] \triangleq \mathbb{E}_{\mathcal{Q}(\tilde{o}|\bar{a})}\{\mathcal{F}[\mathcal{Q}, P]\} \tag{6}$$

Here, the $\triangleq$ symbol indicates an isomorphic or equivalent structural relationship, rather than a strict equality. It is crucial to note that the EFE is a functional of the agent's model ($\mathcal{Q}$) and of prior preferences over future observations, here denoted by the conditioning variable C. Accordingly, $P(\tilde{o}|C)$ represents a distribution of preferred future observations and is distinct from the actual environment process P defined in (3).

Let $\tilde{o} = o_{\tau+1:\tau+h+1}$ denote the sequence of the future observations for each time step $t \in [\tau, \tau+h]$ within the planning horizon. The expectation in (6) can then be expanded and simplified as follows:

$$\begin{aligned} &\mathbb{E}_{\mathcal{Q}(\tilde{o}|\bar{a})}\{\mathcal{F}[\mathcal{Q}, P]\} \\ &= \mathbb{E}_{\mathcal{Q}(\bar{o}|\bar{a})}\{D_{\mathrm{KL}}\left[\, \mathcal{Q}(\bar{o}|\tilde{o}, \bar{a}) || \, P(\tilde{o}, \bar{o}|\bar{a}) \right]\} \\ &= \mathbb{E}_{\mathcal{Q}(\tilde{o}, \bar{o}|\bar{a})}[\log \mathcal{Q}(\bar{o}|\tilde{o}, \bar{a}) - \log P(\bar{o}|\bar{a}) - \log P(\tilde{o}|\bar{o}, \bar{a})] \\ &\triangleq \mathbb{E}_{\mathcal{Q}(\tilde{o}, \bar{o}|\bar{a})}[\log \mathcal{Q}(\bar{o}|\tilde{o}, \bar{a}) - \log \mathcal{Q}(\bar{o}|\bar{a}) - \log P(\tilde{o}|C)] \\ &= \underbrace{-\mathbb{E}_{\mathcal{Q}(\tilde{o}|\bar{a})}\{D_{\mathrm{KL}}\left[\, \mathcal{Q}(\bar{o}|\tilde{o}, \bar{a}) || \, \mathcal{Q}(\bar{o}|\bar{a}) \right]\}}_{\text{expected epistemic gain}} \end{aligned}$$

$$
\begin{aligned}
&\underbrace{-\mathbb{E}_{\mathcal{Q}(\tilde{o}|\bar{a})}[log\, P(\tilde{o}|C)]}_{\text{expected pragmatic gain}} \\
:=&\mathcal{G}[\mathcal{Q}, \mathcal{P}] \qquad (7)
\end{aligned}
$$

Two crucial assumptions are made in the fourth line of this derivation (marked with $\triangleq$):

- The log-likelihood of the future trajectory under the process, $log\, P(\tilde{o}|\bar{a})$, is replaced by the corresponding log-likelihood under the agent's model, $log\, \mathcal{Q}(\tilde{o}|\bar{a})$. This substitution is vital within the Act-Inf framework as it induces exploration, ensuring the EFE, as defined in the last line of (7), promotes information-seeking behaviour.
- The true distribution of future observations, $P(\tilde{o}|\bar{o}, \bar{a})$, is replaced by a predefined distribution of preferred observations, $\mathcal{P}(\tilde{o})$. This $\mathcal{P}(\tilde{o})$ encodes the perceptive goals of the agent, independent of its actions, and must be specified a priori for the EFE to be practically maximised. In the context of classical reinforcement learning, $\mathcal{P}(\tilde{o})$ can be understood as encoding a distribution of desired rewards in terms of observations [47], thereby guiding the agent's primary goal-directed behaviour when maximising the *expected pragmatic gain* term in (7).

The *expected epistemic gain* term in (7) is *perceptive by nature*: it measures the expected KL divergence between the approximate posterior updated after conditioning on the candidate future observation $\tilde{o}$, and the baseline approximate posterior $\mathcal{Q}(\tilde{o}|\bar{a})$. It thus quantifies the *expected reduction in uncertainty* (or 'disambiguation') of the perceptive model induced by anticipating future observations. A greater disambiguation implies a more informed transition model and promotes exploration. The *expected pragmatic gain* term is *active by nature*: conditioned on the action sequence $\bar{a}$, it encodes the agent's preference for future observations $\tilde{o}$ and drives goal-directed behaviour.

Relation to Reinforcement Learning and Prior Preferences. A potential point of tension arises when comparing the Active Inference formulation in (7) with standard reinforcement learning. In the EFE, the pragmatic term depends on a predefined distribution of preferred observations $\mathcal{Q}(\tilde{o}|C)$, which is specified a priori. This may appear at odds with the minimal-assumption perspective of many RL approaches, where the agent is instead driven by an externally provided scalar reward and does not explicitly maintain prior beliefs over desirable outcomes. This conceptual difference is well known in the literature on Active Inference and RL, where goals are recast as prior beliefs about preferred sensory consequences rather than as externally imposed returns [48–50]. In the present work, however, the practical instantiation is deliberately closer to the RL setting: in the one-step EFE approximations of our DAI agents, the pragmatic component is directly implemented through the environment reward r_t, as shown in (11) and (13). Therefore, the pragmatic gain remains aligned with the conventional objective of return maximisation, while the distinctive contribution of our formulation lies in the epistemic gain term—which is perceptive by nature and adds an intrinsic uncertainty-reducing drive to policy evaluation [51]. This framing helps clarify why the empirical gains observed in the LION environment should not be attributed to an alternative reward specification, but rather to the epistemic component that shapes exploration and CTI acquisition. More broadly, although classical Active Inference formulations assume fixed prior preferences, these preferences can also be learned from data, demonstrations, or interaction [50, 52]. In applied settings such as active cyber-defence, where desirable outcomes are comparatively well specified, encoding such preferences is not necessarily burdensome; nevertheless, extending the TIGER framework with online preference learning remains an important direction for future work.

3.4 Defining the EFE-driven normative policy

Finally, the Act-Inf framework posits that, at any given instant t, and with a specific planning horizon h, the normative action policy is derived by minimising the EFE, as defined in (7):

$$
P(a_t|o_t) := \arg\min_{a_t} \mathcal{G}[\mathcal{Q}, C] \qquad (8)
$$

Utilising this model, the agent's objective, as expressed in (1), is achieved by shaping its perceptive model and its policy, both defined in (2), to minimise the VFE and EFE functionals. These are respectively defined in (5) and (7). Crucially, the EFE definition inherently provides the agent with a balancing mechanism between exploration and exploitation: minimising the *sum of the negatives* of the expected epistemic and pragmatic gains effectively prompts the agent to maximise both such gains. This formulation thus helps prevent convergence to suboptimal or even deceptive policies, while also mitigating exploration regimes that hinder policy convergence.

Most Act-Inf implementations typically adopt an actor-critic setting, where the agent's policy is optimised through a dynamic criterion analogous to (8). Notably, within the fully observable MDP presented here, the perceptive model can be implemented using any Markov-Kernel regression machinery capable of learning the environment's transition dynamics.

4 Neural implementation

Four distinct agent baselines are considered in this paper, together with the DAI baseline defined in the previous section. The baselines should be understood as information-theoretic extensions or ablations around the DAI agent. The DDQN baseline removes the epistemic term entirely and therefore serves as a purely pragmatic reference.

An important commonality of our agents is the absence of an actor-critic setting. Even when a policy model is present, it is not directly employed for action selection. Instead, its utility is confined to the computation of auxiliary intrinsic-value targets. This design choice was made to facilitate a fair comparison between the DAI agents and the DDQN baseline. Actor-critic schemes, such as the one described in [31], and even some off-policy policy-gradient schemes, were observed to struggle with convergence in the LION environment. Finally, all agents rely on a fully bootstrapped estimate of their corresponding intrinsic action-value functional, consistently employing a planning horizon of $h = 1$.

On the planning horizon. A planning horizon of $h = 1$ may appear to forgo the multi-step epistemic planning that is often considered the most distinctive feature of Active Inference [33]. However, this concern conflates two distinct notions of horizon. The first is the *explicit* EFE approximation depth, which is fixed at $h = 1$ in our implementation. The second is the *effective* temporal scope of the value function, which is jointly determined by the discount factor γ and the bootstrapped critic in (10). Since $\mathbf{G}_\psi$ is trained via temporal difference learning with a periodically updated target network $\hat{\mathbf{G}}_{\hat{\psi}}$, the fixed-point of this update integrates expected EFE values over an *infinite* future horizon—precisely as a standard value-based agent integrates infinite-horizon returns through single-step Bellman backups. In this sense, though not equivalent to explicit belief-tree search [33], which marginalises over chains of future observations to propagate uncertainty step by step, long-horizon epistemic value is carried implicitly by the bootstrapped critic.

What $h = 1$ genuinely constrains is the ability to perform explicit *belief-tree search*: marginalising over sequences of future observations to compute anticipated reductions in uncertainty step by step [33]. This form of planning is computationally prohibitive in the continuous, high-dimensional state space of LION, and is precisely the limitation that motivates the temporal-difference approximation adopted from [31]. Moreover, explicit multi-step belief propagation is substantially less critical in the fully observable MDP setting than in POMDPs, because the agent already has direct access to the environmental state at each time step, obviating the need to maintain and propagate latent belief distributions across time. We therefore argue that $h = 1$ is a well-motivated and principled design choice for the present experimental context. Extending the explicit planning depth remains an interesting direction for future work in CTI settings with larger action and state spaces.

Regarding their neural architecture, the proposed agents are equipped with the following modules:

1. A critic network $\mathbf{G}_\psi(a_t, o_t)$, parametrised by ψ, responsible for estimating the intrinsic action value associated with each action a_t given the current observation o_t. For the DAI baseline, this value corresponds to the one-step EFE approximation; for the remaining variants, it corresponds to the particular hybrid objective used by each agent.
2. A policy network $\mathbf{Q}_\phi(a_t|o_t)$, parametrised by ϕ, which outputs a distribution over possible actions a_t given the current observation o_t.
3. A transition network $\mathbf{Q}_\theta(o_t|o_{t-1}, a_{t-1})$, parametrised by θ, which predicts the next observation o_t given the previous observation o_{t-1} and action taken a_{t-1}. This network effectively implements the perceptive model discussed in Sect. 3.

While all proposed DAI agents incorporate a critic network, the presence of a policy or transition network varies across implementations. Specifically, the agent taxonomy considered in this paper is the following:

1. **DAI-PERCEPTIVE (DAI-P)**: This is the canonical Active Inference agent. Its critic target is exactly the two-term EFE approximation from Sect. 3: a perceptive epistemic term implemented through the transition-model KL, plus a pragmatic term implemented through the environment reward. Accordingly, DAI- P is the MDP analogue of the standard POMDP Active Inference agent and does not require a policy model.
2. **DAI-ACTIVE (DAI-A)**: This implementation augments the pragmatic term with an *active* mutual-information bonus defined over actions. Importantly, this bonus is *not* derived from the VFE/EFE decomposition in Sect. 3; rather, it is introduced as an information-theoretic extension designed to test whether action-disambiguation incentives are empirically useful in this domain. Its computation requires both a policy model and a transition model.
3. **DAI-SURROGATE ACTIVE (DAI-SA)**: To simplify the setting for DAI- A, a surrogate approximation of the same active information-theoretic bonus is introduced, using only the policy model. This variant should likewise be interpreted as a non-canonical extension rather than as a pure Active Inference agent.
4. **DAI-FULL (DAI-F)**: This agent is best understood as a hybrid obtained by adding the active mutual-information bonus of DAI- A to the canonical DAI- P objective. There-

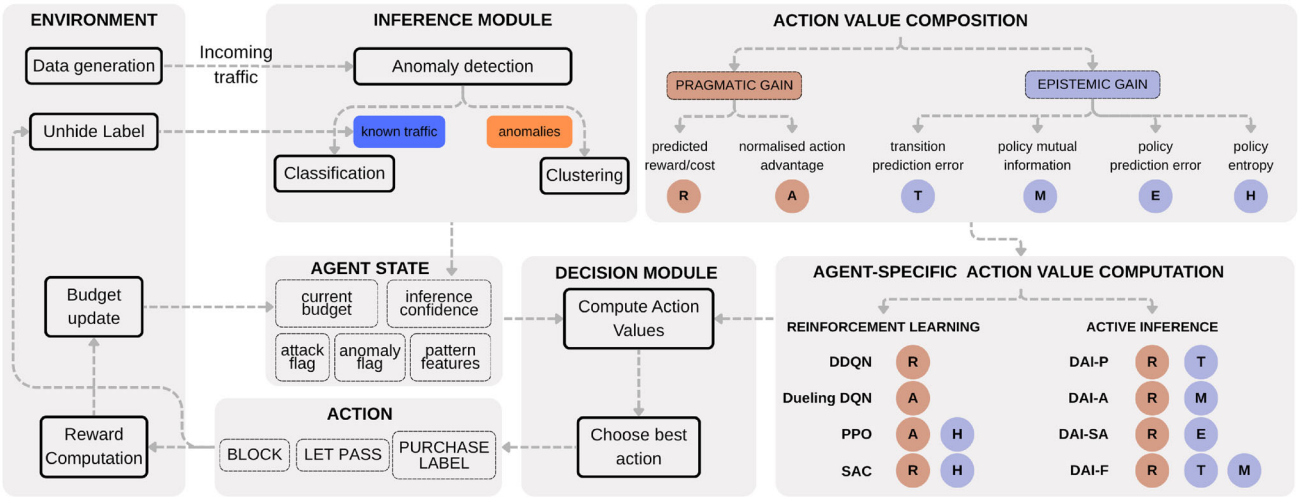


Fig. 1 This work presents a proof-of-concept framework for testing Deep Active Inference (DAI) agents on a synthetic Cyber-defensive game, where Cyber Threat Intelligence acquisition is operationalised as data labelling on a continual-learning setting for a deep-learning-based network intrusion detection system (inference module). The agent (decision module) implements four different DAI formulations and four different Deep Reinforcement Learning baselines

fore, if the non-canonical active bonus is removed, DAI- F collapses to DAI- P. We nevertheless retain it as a separate experimental condition in order to test whether the active bonus provides any incremental utility on top of the canonical formulation.

The main characteristics of the implemented DAI agents are schematised in Fig. 1.

4.1 The target policy

As explained in Sect. 3, the Act-Inf paradigm defines a normative or target policy $P(a_t|o_t)$ by minimising the EFE functional. For the hybrid baselines, this functional is the corresponding augmented objective. To implement such a policy in a unified way across all algorithms, the critic network $\mathbf{G}_\psi$ is used to score actions. Following [31], a Boltzmann sampling scheme is then employed across all DAI implementations and in the DDQN baseline. This sampling scheme is known to aid exploration. Specifically, the action selection employed in the agents is expressed as:

$$P(a_t|o_t) = \sigma[-\zeta \mathbf{G}_\psi(o_t, a_t)] \tag{9}$$

Here, ζ is a precision hyperparameter, $\sigma[\bullet]$ denotes the Softmax function, and $\mathbf{G}_\psi(o_t, a_t)$ is assumed to approximate the intrinsic action value associated with action a_t given observation o_t. With the normative policy implemented in this manner, the parameters of the critic, transition, and policy networks are optimised differently across the various implementations, as elaborated below.

4.2 Optimising the critic network

In all the baselines implementations proposed herein, the critic network is optimised using Temporal Difference (TD) learning. Specifically, a ground-truth intrinsic value, $\mathcal{G}(o_t, a_t)$, is estimated via one-step bootstrapping:

$$\mathcal{G}(o_t, a_t) \approx \bar{\mathcal{G}}(o_t, a_t) + \gamma \cdot \hat{\mathbf{G}}_{\hat{\psi}}(o_{t+1}, a_{t+1}) \tag{10}$$

where $\gamma \in [0, 1]$ is a *discount-factor* hyperparameter for future values, and $\hat{\mathbf{G}}_{\hat{\psi}}$ denotes a target network. The target network is a periodically updated copy of the main critic network, $\mathbf{G}_\psi$, which is used to stabilise the training process. Specifically, the critic module is trained by minimising the Mean-Squared Error (MSE) between the target value $\mathcal{G}(o_t, \cdot)$ and the critic estimate $\mathbf{G}_\psi(o_t, \cdot)$ across all possible actions:

$$\psi^* = \arg\min_\psi \left\| \mathcal{G}(o_t, \cdot) - \mathbf{G}_\psi(o_t, \cdot) \right\|_2^2$$

To compute the one-step bootstrapped target $\bar{\mathcal{G}}(o_t, a_t)$, a one-step approximation of the relevant objective is employed. It is precisely in this approximation that the different implementations vary:

1. **DAI-P**: In this variant, the bootstrapped target $\bar{\mathcal{G}}(o_t, a_t)$ is the canonical one-step approximation of the EFE derived in Sect. 3, namely the sum of the pragmatic gain and the perceptive epistemic gain:

$$\bar{\mathcal{G}}_{\text{DAI-P}}(o_t, a_t) := \underbrace{-D_{\text{KL}}\left[\mathcal{Q}(o_t) \,||\, \mathcal{Q}(o_t \mid o_{t-1}, a_{t-1})\right]}_{\text{epistemic gain}} \underbrace{-\ r_t}_{\text{pragmatic gain}} \tag{11}$$

where r_t is the reward obtained by the agent at time step t. In our implementation, this reward directly instantiates the one-step pragmatic gain discussed in (7). The first term of (11), conversely, is the one-step perceptive KL that implements the epistemic gain term in (7) : it compares a posterior-side variational target, informed by the realised next observation, against the prior-side predictive belief produced by the transition model.

In practice, the posterior-side term is instantiated from a one-step sample o_t directly obtained from the environment at time step t. Rather than interpreting $\mathcal{Q}(o_t)$ as a free-standing learned posterior distribution, we treat it as the observation-informed target distribution against which the baseline approximate posterior is trained. Mathematically, o_t is modelled as a spherical Gaussian centred at the realised observation with unit variance:

$$\mathcal{Q}(o_t) \triangleq \mathcal{N}(o_t, \mathbf{I}) \rightarrow o_t \sim \mathcal{Q}(o_t)$$

Conversely, the distribution $\mathcal{Q}(o_t|o_{t-1}, a_{t-1})$ represents the model's prior over the next observation o_t. This prior is also assumed to be a spherical Gaussian, even though, in practice, it is directly implemented through the output of the transition network without explicit sampling:

$$\begin{aligned}&\mathcal{Q}(o_t|o_{t-1}, a_{t-1})\\ &\quad\triangleq \mathcal{N}(\mathbf{Q}_\theta(o_t|o_{t-1}, a_{t-1}), \mathbf{I})\\ &\quad\rightarrow o_t \sim \mathbf{Q}_\theta(o_t|o_{t-1}, a_{t-1})\end{aligned}$$

Given the assumption of Gaussian forms for both $\mathcal{Q}(o_t)$ and $\mathcal{Q}(o_t|o_{t-1}, a_{t-1})$, the epistemic term in (11) can be computed analytically as:

$$\begin{aligned}&D_{\text{KL}}\left[\ \mathcal{Q}(o_t)||\ \mathcal{Q}(o_t|o_{t-1}, a_{t-1})\right]\\ &\quad= D_{\text{KL}}\left[\mathcal{N}(o_t, \mathbf{I})||\mathcal{N}(\mathbf{Q}_\theta(o_t|o_{t-1}, a_{t-1}), \mathbf{I})\right]\\ &\quad\approx \frac{1}{2}\left\|o_t - \mathbf{Q}_\theta(o_t|o_{t-1}, a_{t-1})\right\|_2^2\end{aligned} \tag{12}$$

2. **DAI-A**: Conversely, the DAI- A scheme formulates $\bar{\mathcal{G}}(o_t, a_t)$ in terms of a non-canonical active information bonus together with the pragmatic gain:

$$\bar{\mathcal{G}}_{\text{DAI-A}}(o_t, a_t) := \underbrace{-D_{\text{KL}}\left[\mathcal{Q}(a_t \mid o_t, o_{t+1}) \,||\, \mathcal{Q}(a_t \mid o_t)\right]}_{\text{active information bonus}} \underbrace{-\ r_t}_{\text{pragmatic gain}} \tag{13}$$

This action-space KL is retained as an auxiliary mutual-information bonus, but it is not claimed to arise from the EFE decomposition in Sect. 3. Rather, DAI- A should be understood as an information-theoretic extension that isolates the empirical effect of action disambiguation.

The posterior policy $\mathcal{Q}(a_t|o_t, o_{t+1})$ is approximated via Bayes' rule as follows:

$$\begin{aligned}&\mathcal{Q}(a_t|o_t, o_{t+1}):=\\ &\quad:=\frac{\overbrace{\mathbf{Q}_\theta(o_{t+1}|o_t, a_t)}^{\text{transition likelihood}}\cdot\overbrace{\mathbf{Q}_\phi(a_t|o_t)}^{\text{policy prior}}}{\sum_{a'}\mathbf{Q}_\theta(o_{t+1}|o_t, a')\mathbf{Q}_\phi(a'|o_t)}\end{aligned} \tag{14}$$

3. **DAI-SA**: This implementation utilises a surrogate function to approximate the same active information bonus solely using the policy module. Specifically, the bonus is approximated by a divergence measure between the policy model and the current normative policy induced by the critic:

$$\begin{aligned}&D_{\text{KL}}\left[\ \mathcal{Q}(a_t|o_t, o_{t+1})||\ \mathcal{Q}(a_t|o_t)\right]\\ &\quad\approx \mathbb{E}_{o_t\sim\mathcal{D}}\left[\left\|\mathbf{Q}_\phi(\cdot|o_t) - P(\cdot|o_t)\right\|_2^2\right]\end{aligned} \tag{15}$$

Here, $o_t \sim \mathcal{D}$ indicates that the expectation is computed over a sample of observations drawn from a replay buffer. Again, this quantity is used as a surrogate information-theoretic regulariser rather than as a term derived from the canonical EFE.

4. **DAI-F**: Finally, this implementation estimates a hybrid objective by combining the canonical DAI- P target with the active information bonus used by DAI- A:

$$\begin{aligned}&\bar{\mathcal{G}}_{\text{DAI-F}}(o_t, a_t) :=\\ &\quad\bar{\mathcal{G}}_{\text{DAI-P}}(o_t, a_t) - D_{\text{KL}}\left[\mathcal{Q}(a_t \mid o_t, o_{t+1}) \,||\, \mathcal{Q}(a_t \mid o_t)\right]\end{aligned}$$

Notice how all the formulations of $\bar{\mathcal{G}}(o_t, a_t)$ given here support off-policy learning: target values can be computed from stored trajectories and, when needed, from auxiliary inferences performed by the policy and transition networks.

4.3 Optimising the transition network

The transition network is utilised by DAI- P, DAI- A, and DAI- F. In all three cases, its optimisation is grounded in next-observation prediction. For the canonical DAI- P agent,

this prediction error is exactly the epistemic term appearing in the one-step EFE approximation. For the hybrid agents DAI- A and DAI- F, the same predictive model is additionally required to instantiate the posterior policy in (14). Accordingly, the transition network is trained through the following supervised loss:

$$\begin{aligned}\theta^* &:= \arg\min_{\theta} D_{\mathrm{KL}}\left[\, \mathcal{Q}(o_t) || \, \mathcal{Q}(o_t|o_{t-1}, a_{t-1})\right] \\ &\approx \frac{1}{2}\left\| o_t - \mathbf{Q}_\theta(o_t|o_{t-1}, a_{t-1})\right\|_2^2\end{aligned} \tag{16}$$

In this context, it is crucial to recognise that while this term serves as the loss function to minimise for optimising θ and learning the environment's transition dynamics, it simultaneously functions as the epistemic component to maximise in the canonical DAI- P and hybrid DAI- F action-value targets. Thus, the transition model is trained to become predictive, while the critic favours actions that expose informative mismatches between prior prediction and realised observation.

4.4 Optimising the policy network

The implementations that utilise a policy network are DAI- A, DAI- F, and DAI- SA. Notice that no policy-network loss is claimed to arise directly from the canonical VFE decomposition of Sect. 3. Instead, whenever a policy model is used, it is trained to approximate the current normative policy induced by the critic. Accordingly, the parameters of the policy network (ϕ) are optimised through the supervised objective:

$$\phi^* = \arg\min_{\phi} \mathbb{E}_{o_t \sim \mathcal{D}}\left[\left\| \mathbf{Q}_\phi(\cdot|o_t) - P(\cdot|o_t)\right\|_2^2\right] \tag{17}$$

This update is off-policy and reduces policy learning to a supervised problem in which the normative distribution serves as a soft target. In DAI- A and DAI- F, the resulting policy model is needed to instantiate the active information bonus through (14). In DAI- SA, the same MSE also serves as the surrogate quantity in (15). Consequently, DAI- SA exhibits a dual use of this divergence:

- as a surrogate intrinsic bonus intended to favour action-disambiguating policies; and
- as a policy loss to ensure convergence toward the current normative distribution.

Finally, recall that the target distribution $P(a_t|o_t)$ is computed using the current critic $\mathbf{G}_\psi$ via (9).

4.5 Algorithm description

Algorithm 1 provides a unified pseudo-code of the ACID training procedure, encompassing all four DAI variants and the DDQN baseline. The algorithm is parameterised by the *agent type*, which determines (i) the one-step EFE target used to compute $\bar{\mathcal{G}}$, and (ii) which auxiliary networks—transition ($\mathbf{Q}_\theta$) and/or policy ($\mathbf{Q}_\phi$)—are instantiated alongside the shared critic $\mathbf{G}_\psi$.

Algorithm 1 ACID: unified training loop for all DAI agents and DDQN.

Require: Agent type $\in$ {DAI- P, DAI- A, DAI- SA, DAI- F, DDQN}; discount γ; precision ζ; batch size B; target update period T

1: Initialise replay buffer $\mathcal{D} \leftarrow \emptyset$, step counter $t \leftarrow 0$
2: Initialise critic $\mathbf{G}_\psi$ and target critic $\hat{\mathbf{G}}_{\hat{\psi}} \leftarrow \mathbf{G}_\psi$
3: **if** agent $\in$ {DAI- P, DAI- A, DAI- F} **then**
4: Initialise transition network $\mathbf{Q}_\theta$
5: **end if**
6: **if** agent $\in$ {DAI- A, DAI- SA, DAI- F} **then**
7: Initialise policy network $\mathbf{Q}_\phi$
8: **end if**
9: **loop**
10: Observe current state o_t from environment
11: $a_t \sim \sigma\left[-\zeta\, \mathbf{G}_\psi(o_t, \cdot)\right]$ {Boltzmann, Eq. (9)}
12: Execute a_t; receive r_t, o_{t+1}; store (o_t, a_t, r_t, o_{t+1}) in $\mathcal{D}$; t += 1
13: **if** $|\mathcal{D}| \geq B$ **then**
14: Sample minibatch $\mathcal{B} = \{(o_s, a_s, r_s, o_{s+1})\}_{s=1}^{B}$ from $\mathcal{D}$
15: Compute immediate target $\bar{\mathcal{G}}(o_s, a_s)$ for all $s \in \mathcal{B}$:
16: **if** agent = DAI- P **then**
17: $\bar{\mathcal{G}} \leftarrow -D_{\mathrm{KL}}\left[\, \mathcal{Q}(o_s) || \, \mathcal{Q}(o_s|o_{s-1}, a_{s-1})\right] - r_s$ {Eq. (11)}
18: **else if** agent = DAI- A **then**
19: $\bar{\mathcal{G}} \leftarrow -D_{\mathrm{KL}}\left[\, \mathcal{Q}(a_s|o_s, o_{s+1}) || \, \mathcal{Q}(a_s|o_s)\right] - r_s$ {Eq. (13)}
20: **else if** agent = DAI- SA **then**
21: $\bar{\mathcal{G}} \leftarrow -\left\| \mathbf{Q}_\phi(\cdot|o_s) - P(\cdot|o_s)\right\|_2^2 - r_s$ {Eq. (15)}
22: **else if** agent = DAI- F **then**
23: $\bar{\mathcal{G}} \leftarrow \bar{\mathcal{G}}_{\text{DAI- P}}(o_s, a_s) - D_{\mathrm{KL}}\left[\, \mathcal{Q}(a_s|o_s, o_{s+1}) || \, \mathcal{Q}(a_s|o_s)\right]$
24: **else** {DDQN }
25: $\bar{\mathcal{G}} \leftarrow -r_s$
26: **end if**
27: $\mathcal{G}(o_s, a_s) \leftarrow \bar{\mathcal{G}}(o_s, a_s) + \gamma\, \hat{\mathbf{G}}_{\hat{\psi}}(o_{s+1}, a_{s+1})$ {TD target, Eq. (10)}
28: $\psi \leftarrow \psi - \nabla_\psi \left\| \mathcal{G}(o_s, \cdot) - \mathbf{G}_\psi(o_s, \cdot)\right\|_2^2$ {critic update}
29: **if** agent $\in$ {DAI- P, DAI- A, DAI- F} **then**
30: $\theta \leftarrow \theta - \nabla_\theta \left\| o_{s+1} - \mathbf{Q}_\theta(o_{s+1}|o_s, a_s)\right\|_2^2$ {Eq. (16)}
31: **end if**
32: **if** agent $\in$ {DAI- A, DAI- SA, DAI- F} **then**
33: $\phi \leftarrow \phi - \nabla_\phi \left\| \mathbf{Q}_\phi(\cdot|o_s) - P(\cdot|o_s)\right\|_2^2$ {Eq. (17)}
34: **end if**
35: **if** $t \bmod T = 0$ **then**
36: $\hat{\psi} \leftarrow \psi$ {periodic target network synchronisation}
37: **end if**
38: **end if**
39: **end loop**

The algorithm proceeds as follows. **Initialisation (lines 1–6):** The replay buffer, the shared critic $\mathbf{G}_\psi$, and its target copy $\hat{\mathbf{G}}_{\hat{\psi}}$ are initialised for all agents. Depending on the agent type, a transition network $\mathbf{Q}_\theta$ (used by DAI- P, DAI- A, and DAI- F) and/or a policy network $\mathbf{Q}_\phi$ (used by DAI- A, DAI- SA, and DAI- F) are additionally instantiated. **Environment interaction (lines 7–10):** At each step, the agent observes o_t, selects an action via Boltzmann sampling over the critic's

scores (Eq. (9)), executes it, and stores the resulting transition tuple in $\mathcal{D}$. **One-step EFE target (lines 11–22):** A minibatch is sampled from the replay buffer. For each transition, the immediate intrinsic-value component $\bar{\mathcal{G}}(o_s, a_s)$ is computed according to the agent-specific formulation. DAI- P uses the canonical perceptive epistemic term together with the environment reward (Eq. (11)). DAI- A substitutes the epistemic term with the active information bonus (Eq. (13)). DAI- SA uses the surrogate policy-divergence bonus (Eq. (15)). DAI- F combines the canonical DAI- P target with the active bonus. DDQN reduces to a purely pragmatic target $-r_s$, making it a special case of the unified formulation with the epistemic component removed. **Bootstrapped TD update (line 23):** The full intrinsic-value target is completed by adding the discounted next-step critic estimate, following Eq. (10). **Network updates (lines 24–30):** The critic is updated by a gradient step on the mean-squared TD error over the minibatch. When present, the transition network $\mathbf{Q}_\theta$ is updated by minimising the next-observation prediction error (Eq. (16)), and the policy network $\mathbf{Q}_\phi$ is updated by minimising its divergence from the current normative policy (Eq. (17)). All three updates are off-policy and operate on stored transitions without requiring new environment interaction. **Target network synchronisation (lines 31–33):** The target critic $\hat{\mathbf{G}}_{\hat{\psi}}$ is periodically copied from the main critic $\mathbf{G}_\psi$ to stabilise the TD learning dynamics.

5 The LION environment and experimental setting

This paper follows the design principles introduced in the TIGER environment in [6], which is a reinforcement learning environment for benchmarking optimal CTI-acquisition policies in the context of ML-based network intrusion detection systems. In the TIGER game environment, a cyber-defensive agent—the *management module*—is in charge of blocking or accepting incoming traffic flows and eventually buying CTI insights, operationalised as labels for unknown classes of benign/malicious traffic. Acquired labels are used to train a ML-based intrusion detector, called the *inference module*, whose output serves as a (noisy) support for the decision module. The goal of the decision module is to keep the network safe, and such objective is mapped through a specific reward policy that compensates correct defensive actions while penalising for incorrect ones. CTI-acquisition actions incurr in a cost but may turn useful for training the inference module, providing a more principled decision support for the management agent. All the agent's decisions are made under a constrained budget setting.

This paper introduces LION, a synthetic proof-of-concept testbed for the DAI agents of Sect. 4 inspired by the TIGER environment. Namely LION involves a CTI-acquisition game in a self-contained environment where an agent is forced to avoid **L**east-**I**nformed Decisi**ON**s for cyber-defensive policies. While the synthetic characteristic of LION reduces the cybersecurity realism of a full network emulation with respect to TIGER, the decision-theoretic structure of the CTI-acquisition problem is preserved, and synthetic data enables controlled uncertainty and reproducible experimentation.

Network flows in LION are represented as two-dimensional feature vectors sampled from seven Gaussians: three *known* classes (K_0, K_1 benign; K_2 malicious) on which the inference module is pretrained, and four *unknown* classes whose labels the management agent must purchase. The unknowns exhibit heterogeneous strategic properties: U_A (malicious) overlaps K_0 in feature space and is therefore misclassified as benign until labelled; U_B (malicious) and U_C (benign) are clearly anomalous; U_D (benign) sits near K_2, creating a false-positive risk.

Label prices differ across unknown classes, and their sum exceeds the agent's initial budget. The agent must therefore choose *which* labels to acquire and *when*, while managing incoming flows under uncertainty—a budget-constrained curriculum learning problem in which the purchase sequence is not prescribed externally but emerges from the agent's own policy. This is a well-posed ground floor for testing Act-Inf agents: EFE epistemic-gain terms are designed precisely to identify actions that reduce model uncertainty and thus to discover the optimal label-acquisition order before delayed reward signals alone would justify it. The inference module is a prototypical classifier; when a label is bought, the corresponding cluster prototype is promoted to the known-class list, producing the sharpest one-step state change and therefore the highest perceptive epistemic gain—the mechanism that lets DAI assign credit to strategic purchases faster than pure reward-driven baselines. Following TIGER, episodes in LION reset the inference module but preserve the management agent's parameters, so policy learning is persistent across the curriculum. Further details of our experimental setup are given in Appendix 8.

5.1 Reinforcement learning baselines

Four reinforcement learning baselines are evaluated alongside the DAI variants described above. Because the synthetic proof-of-concept environment is low-dimensional and admits only a small discrete action space, random exploration was suppressed across all agents. Specifically, neither the value-based baselines nor the DAI variants rely on significant initial stochasticity for action selection: the DDQN variants start from a low ε-greedy probability ($\varepsilon_0 = 0.25$, decaying multiplicatively to $\varepsilon_\infty = 0.05$), while the Boltzmann precision used in the DAI variants is set to a large value ($\zeta = 100$), rendering sampling from (9) nearly deterministic. The one principled exception is the Soft Actor-Critic

agent, which maintains stochastic action selection by design through its maximum-entropy objective. A brief description of each baseline follows.

Double Deep Q-Network (DDQN). The DDQN baseline [53] is a purely pragmatic agent: its critic target reduces to $-r_s$ with no epistemic component whatsoever. It employs a shared two-stream neural backbone that processes exteroceptive and proprioceptive features separately before merging them into a unified Q-value estimate. Action selection follows the ε-greedy scheme described above. This agent serves as the primary ablative reference, isolating the contribution of the epistemic terms present in the DAI variants.

Dueling Double Deep Q-Network (DUELING-DQN). The DUELING- DQN agent [54] augments the DDQN backbone with a dueling architecture: the network decomposes the Q-value estimate into separate scalar value $V(o_t)$ and advantage $A(o_t, a_t)$ streams, recombined as $Q(o_t, a_t) = V(o_t) + A(o_t, a_t) - \overline{A}(o_t)$, where $\overline{A}$ denotes the mean advantage across actions. This decomposition is expected to reduce the variance of Q-estimates in states where the choice of action has little impact on value – a condition that arises frequently in early episodes when no labels have been purchased yet and the proprioceptive state is largely uninformative.

Proximal Policy Optimisation (PPO). The PPO baseline [55] is an on-policy actor-critic algorithm. It accumulates full-episode rollouts and updates both an actor and a value-function critic at the end of each episode, using Generalised Advantage Estimation (GAE, $\lambda = 0.95$) and a clipped surrogate objective ($\varepsilon_{\text{clip}} = 0.20$). A small entropy bonus ($c_{\text{ent}} = 0.01$) is retained in the objective to prevent premature policy collapse. Unlike the value-based agents, PPO performs action selection by sampling directly from the actor's softmax distribution, making it inherently stochastic even in the absence of ε-greedy noise. As an on-policy method, PPO does not maintain a replay buffer and reuses each collected trajectory for a fixed number of gradient epochs ($n_{\text{ep}} = 4$) before discarding it.

Soft Actor-Critic (SAC). The SAC baseline follows the discrete-action formulation of [56]. It maximises the entropy-augmented return $\mathbb{E}\big[\sum_t \gamma^t (r_t + \alpha\, \mathcal{H}(\pi(\cdot|o_t)))\big]$, where α is a temperature parameter updated automatically to track a target entropy $\mathcal{H}_{\text{target}} = 0.6 \times \log|\mathcal{A}|$. Twin critics and Polyak-averaged target networks are used to stabilise off-policy training. SAC is conceptually related to the Act-Inf framework in that both encourage the agent to maintain broad action distributions and avoid premature commitment; this connection motivates its inclusion as a theoretically informed baseline. Crucially, SAC is the only baseline for which action stochasticity is not suppressed: the maximum-entropy objective directly shapes the action distribution during training and inference, providing a principled alternative to the epistemic exploration induced by the DAI variants.

Break-Even (rule-based heuristic). The Break-Even agent is a deterministic, non-learning rule-based policy included as a lower-bound reference. It accepts flows classified as known-benign or labelled-benign, blocks flows classified as known-malicious or labelled-malicious, and handles unlabelled unknown clusters by purchasing the label of whichever cluster currently exhibits the highest anomaly score (defined as the distance of its prototype to the nearest known-class prototype in the embedding space) provided the budget allows it; otherwise, it blocks high-anomaly flows and accepts low-anomaly ones. Because this strategy prioritises visually conspicuous clusters over strategically critical ones, it should systematically misallocate the label budget in the presence of deceptive unknown classes.

6 Results and discussion

6.1 Metric computation and statistical reporting

Our main experimental results are summarised in Figs. 2 and 3. In these, all learning curves display mean ± 95% confidence interval across 10 independent random seeds. Each seed has its own generation pipeline and an independently pretrained inference modules; random streams are offset by `seed` + 1000 × `agent_index` to guarantee pairwise statistical independence across all 90 runs. Raw episode-level metrics are logged at the end of each episode. Prior to aggregation, each per-seed time series is smoothed with a causal rolling mean of window size 20 (`min_periods=1`). The confidence margin at each episode index is then $t_{9,\,0.975} \times \hat{\sigma}/\sqrt{10} \approx 2.262 \times \hat{\sigma}/\sqrt{10}$, where $\hat{\sigma}$ is the sample standard deviation (ddof = 1) across the 10 smoothed seed values.

The metrics shown in Figs. 2 and 3 are defined as follows. **Win rate: 1**[$\text{budget}_{\text{final}} \geq 40$] at episode termination. **Mean step reward:** total undiscounted episode return divided by number of steps. **Labels bought:** count of CTI labels purchased per episode. **Final budget:** remaining budget at episode end, clipped to [0, 45]. **Training loss:** mean per-step gradient loss within the episode; its definition is agent-specific (MSE on EFE targets for DAI agents; Bellman MSE for DDQN and DUELING- DQN; clipped-surrogate composite for PPO; twin-critic MSE for SAC). **Classification reward:** mean per-step reward in the *informed* regime (known or already-labelled flows). **Clustering reward:** mean per-step reward in the *uninformed anomaly* regime (unknown, unlabelled flows). **CTI acquisition cost:** sum of label prices paid during the episode. **Inference confidence:** max-softmax probability over known prototypes, averaged over all steps in the episode. **Epistemic gain:** mean per-action epistemic

bonus from the replay batch; its precise form differs by agent (transition-prediction MSE for DAI- P; KL divergence for DAI- A; policy-target MSE for DAI- SA; entropy bonus $\alpha \cdot \mathcal{H}(\pi)$ for SAC). Notice the epistemic values of DAI variants are not directly commensurate with the entropy bonuses of SAC, but these are plotted in the same axes to compare the contribution in the action's values.

6.2 DAI-P versus deep reinforcement learning baselines

Figure 2 compares DAI- P with DDQN, DUELING- DQN, PPO, SAC, and the Break-Even heuristic across six metrics.

Win rate and final budget. DAI- P achieves the highest win rate across the full training horizon, approaching 75% by episode 500 with narrow confidence bands, indicating consistent learning across seeds. DDQNand DUELING- DQN converge to a similar asymptotic level (roughly 50%) but require approximately twice as many episodes to reach DAI- P's episode-100 performance. SAC exhibits substantially weaker performance despite its entropy-regularised objective, suggesting that entropy-maximisation alone does not substitute for the targeted proprioceptive epistemic signal exploited by DAI- P. PPO shows the slowest improvement among the learned agents, consistent with the known sample-inefficiency of on-policy methods in sparse-reward environments. Interestingly, DDQN and DUELING- DQN accumulate budget at a rate comparable to DAI- P, yet their lower win rates reveal that they tend to reach the winning threshold later within each episode. This suggests that the epistemic signal of DAI- P drives earlier label purchases, allowing more time to exploit resolved uncertainty before the episode clock expires. Break-Even instead plateaus below 20% throughout training, confirming that anomaly score is a poor proxy for label priority.

Label acquisition behaviour. The labels-bought panel reveals a qualitative difference in acquisition strategy. DDQN and DUELING- DQN purchase an increasing number of labels as training progresses, eventually approaching 3–4 per episode; this pattern is consistent with broad exploration of the label space without strong prioritisation. DAI- P reaches a stable acquisition rate of roughly 2.5–3 labels per episode earlier and with less variance, suggesting that the perceptive epistemic signal guides the agent to a focused subset of labels rather than purchasing indiscriminately. PPO's low label count (≈ 1) reflects its tendency to default to conservative block-or-accept policies with minimal CTI investment. Break-Even's count (≈ 2) is consistent with its fixed heuristic: it consistently buys the two highest-anomaly-score labels (UB and one of UC/UD) but rarely reaches UA within the budget.

Loss and inference confidence. The value/policy loss panel shows that PPO exhibits the highest and most volatile training loss, reflecting the instability inherent in its clipped surrogate objective when reward signals are sparse. All other agents converge to low, stable losses within the first 100 episodes. Inference confidence (mean max-softmax probability over known prototypes) is broadly stable across agents in the range 0.83–0.90, which indicates both the fact that a pretrained classifier tends to equally help every agent and that DAI and DDQN variants tend to be more aware of their own uncertainty, which provokes blocking or buying actions.

6.3 DAI variants comparison

Figure 3 compares the four DAI variants against each other and against SAC as a reference, focusing on the epistemic and reward-composition metrics that are not visible in the baseline benchmark. Note that epistemic gain quantities are structurally heterogeneous across agents: for DAI- P it is a mean transition-prediction MSE; for DAI-A a mean KL divergence; for DAI-SA a mean policy-target MSE; for SAC the entropy bonus; they are placed on the same axis for visual comparison but should not be interpreted as commensurate quantities.

Win rate: canonical versus hybrid variants. DAI- P and DAI- A achieve the highest and nearly identical win rates ($\approx$ 70%–75% at convergence), with overlapping confidence bands. DAI- F, which combines both the perceptive and active epistemic terms, matches this level, suggesting that the additional active-information term in DAI- F does not substantially improve over the canonical perceptive term alone. DAI- SA lags by a notable margin throughout training, indicating that the surrogate divergence ($\| Q_\phi - P \|^2$) is a weaker epistemic signal than the transition-network residual used by DAI- P, despite requiring similar computational resources. SAC's win rate remains significantly lower than any DAI variant, confirming that the entropy bonus cannot replace the structured, proprioception-targeted epistemic signal.

Epistemic gain trajectories. The epistemic gain panel highlights an important self-regularising property of the DAI- P and DAI- A families. Early in training, the transition network is poorly calibrated and incurs large prediction errors on label-purchase transitions, producing high epistemic gain values (up to $\approx$ 0.35 for DAI- SA in the first 50 episodes). As the transition network learns the proprioceptive update rule associated with each label purchase, its residual error decreases and the intrinsic bonus diminishes accordingly, yielding near-zero epistemic gain after episode 100. This is the expected and desirable behaviour: the epistemic signal is strong when it is most needed (early exploration) and gracefully vanishes once the agent has internalised the consequence of each label purchase. SAC's epistemic gain

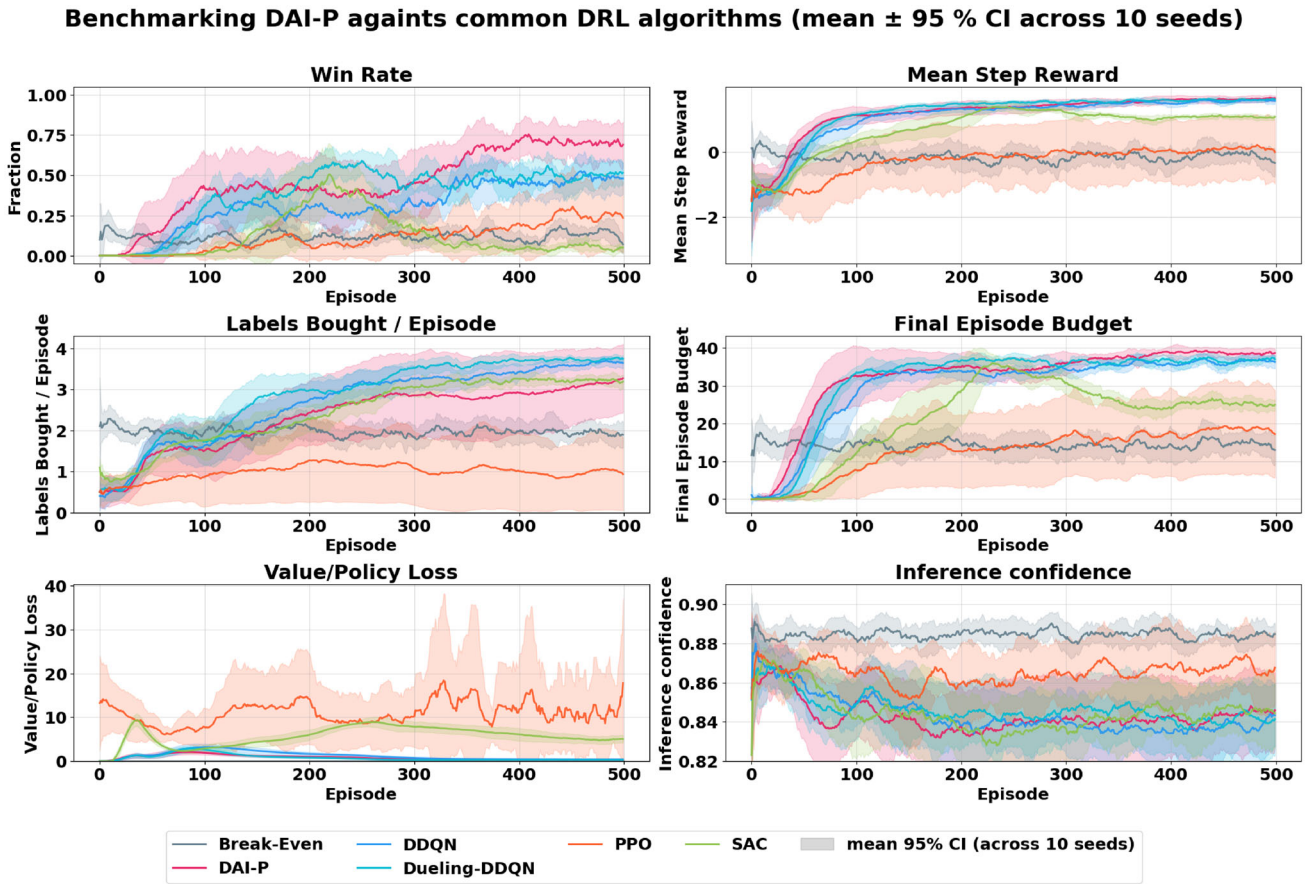


Fig. 2 Mean ± 95% confidence interval across 10 independent seeds for DAI- P and the four DRL baselines (DDQN, DUELING- DQN, PPO, SAC and the deterministic Break-Even heuristic) over 500 training episodes. Each curve is obtained by first applying a rolling smoothing window of 20 episodes to each seed's time series independently, then aggregating across seeds using a t-distribution-based interval. Panels show (top) win rate and mean per-step reward, (middle) CTI labels purchased per episode and final budget, (bottom) training loss and mean inference confidence

(here reported as the entropy bonus $\alpha H(\pi)$) shows the opposite trajectory: it *increases* monotonically throughout training as the auto-tuned temperature α adapts to higher policy entropy, reaching values above 1.0 by episode 500. This divergence in trajectories underscores the conceptual difference between entropy-driven exploration (SAC) and information-gain-driven exploration (DAI): the former is a global, undirected regulariser, while the latter targets the specific transitions that reduce proprioceptive uncertainty.

Reward decomposition. The classification reward (mean per-step reward on known and labelled flows) converges to ≈ 2.0 for all DAI variants within 100 episodes, close to the theoretical maximum of $+2.0$ for correctly accepting benign flows. SAC lags behind, particularly in the first 200 episodes, consistent with its slower win-rate improvement. The clustering reward (mean per-step reward on unknown unlabelled flows) also converges to $\approx$ 1.0–1.1 for DAI- P, DAI- A, and DAI- F, suggesting that even before a label is purchased, these agents have learned to block dangerous flows (yielding $+0.5$) rather than accept them (yielding -7.0). DAI- SA and SAC show lower clustering rewards throughout, suggesting incomplete disambiguation of unknown traffic. For DAI- SA, this is consistent with the surrogate epistemic signal being less sensitive to the sharp proprioceptive state changes that accompany label purchases, as discussed below.

A further explanation for DAI- SA's underperformance relates to the high Boltzmann precision used ($\zeta = 100$). Under this setting, the normative target $P(a_t|o_t)$ is nearly a one-hot vector on the greedy action, so the surrogate gain $\|Q_\phi - P\|^2$ primarily reflects how far the policy network is from the current critic's greedy recommendation – a measure of policy-network convergence only indirectly reflecting environmental uncertainty. Unlike the transition-prediction residual of DAI- P, which is directly tied to how much a label purchase surprises the world model, the surrogate signal does not directly quantify the different effects of actions that reduce proprioceptive uncertainty and those that do not. This

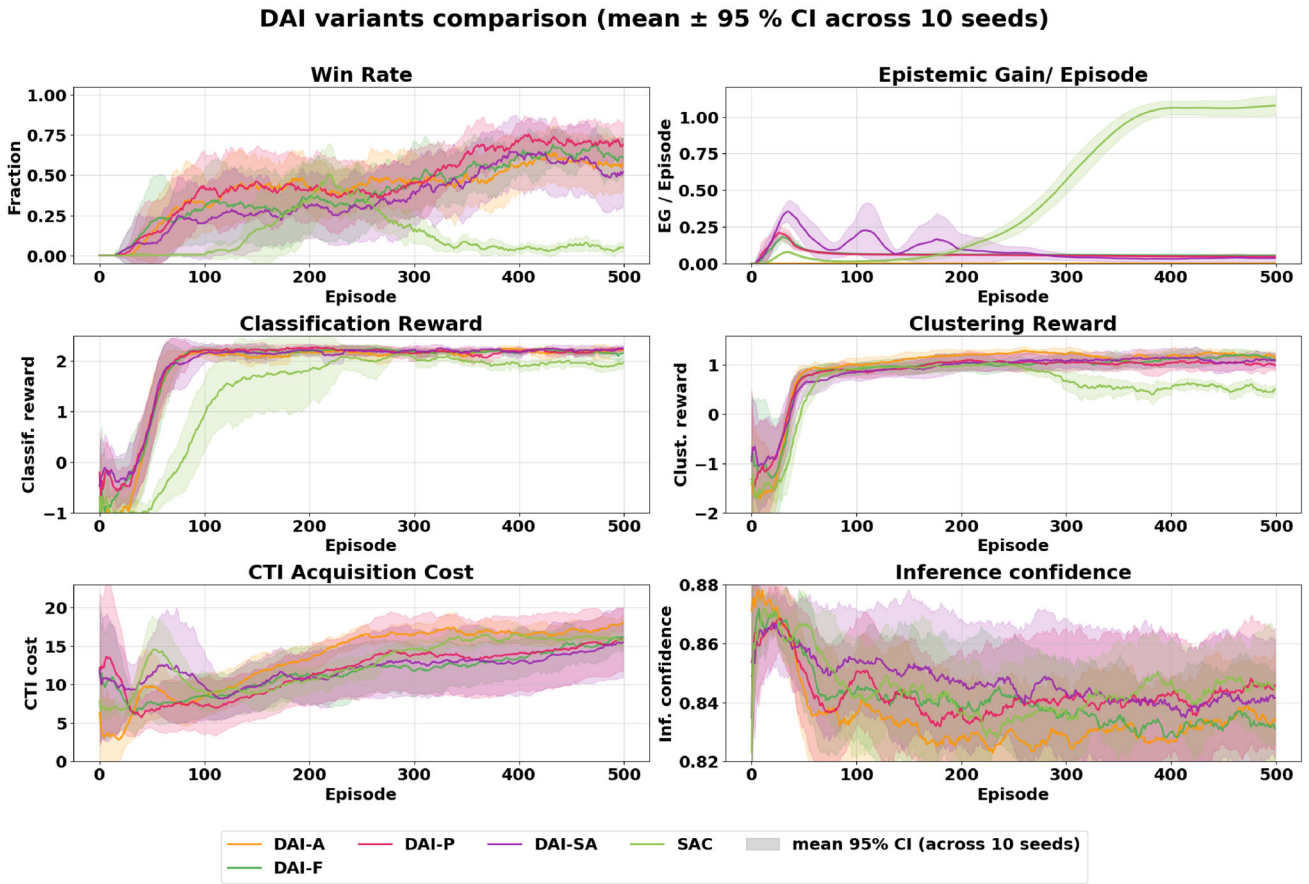


Fig. 3 Mean ± 95% confidence interval across 10 independent seeds for the four DAI variants (DAI- P, DAI- A, DAI- SA, DAI- F) and SAC over 500 training episodes. Smoothing and CI computation follow the same protocol as Figure 2. Panels show (top) win rate and mean epistemic gain per episode, (middle) classification reward in the informed regime and clustering reward in the uninformed anomaly regime, (bottom) CTI acquisition cost and mean inference confidence

may explain both the weaker epistemic gain trajectory and the incomplete clustering disambiguation observed for DAI- SA. Such a design may reveal an intrinsic limitation of the surrogate epistemic gain term in (15). However, future work could explore contexts in which DAI- SA with lower precision for the surrogate target may result in a useful epistemic gain.

CTI acquisition cost and inference confidence. The CTI acquisition cost panel shows that all DAI variants increase their label expenditure as training progresses, converging to $\approx$ 15–17 budget units per episode – corresponding to approximately three labels purchased per episode on average. The early episodes exhibit higher variance (wide confidence bands), reflecting the stochastic exploration of different label orderings before the optimal strategy is discovered. Inference confidence remains stable across all DAI variants at $\approx$ 0.83–0.88, confirming that the prototypical classifier maintains consistent performance regardless of the agent's label-acquisition decisions.

Computational complexity and capacity analysis. Table 1 reports the neural complexity of each agent. The *Module params* row reports the standalone parameter count of each network module. Agent rows indicate which modules are instantiated (✓) or absent (—), and report the resulting trainable parameter total $|\theta|$ and the approximate floating-point operations (FLOPs) per gradient step for mini-batch size B=64. FLOPs are estimated as $3 \times \sum_{\ell} 2B\, d_{\ell-1} d_{\ell}$ over all linear layers ℓ (the factor of 3 approximates the combined forward and backward pass cost). For DAI- A and DAI- F, the KL-based epistemic gain requires $|\mathcal{A}|$=3 parallel TransitionNet forward passes to evaluate $\log P(o'|o, a)$ for each action, which accounts for their substantially higher per-step cost relative to DAI- P. All agents share the same *EFE-net* backbone ($\mathbf{G}_{\psi}$, a two-stream network with 5,763 parameters), which is the sole module performing action selection via (9). The TransitionNet ($\mathbf{Q}_{\theta}$, 3,657 parameters) is used by DAI- P, DAI- A, and DAI- F to compute the perceptive epistemic gain and, in the hybrid variants, to instantiate the posterior policy (14). The PolicyNet ($\mathbf{Q}_{\phi}$, 5,763 param-

Table 1 Neural complexity per agent

	Modules				
	$\mathbf{G}_\psi$ (EFE-net)	$\mathbf{Q}_\theta$ (TransNet)	$\mathbf{Q}_\phi$ (PolicyNet)	$\lvert\theta\rvert$	FLOPs ($\times 10^6$)
Module params	*5 763*	*3 657*	*5 763*	–	–
DDQN	✓	–	–	5 763	2.14
DAI- P	✓	✓	–	9 420	3.50
DAI- SA	✓	–	✓	11 526	4.28
DAI- A	✓	✓	✓	15 183	8.37
DAI- F	✓	✓	✓	15 183	8.37

eters) is used by DAI- A, DAI- SA, and DAI- F exclusively to compute auxiliary epistemic bonuses; it is never used directly for action selection.

These architectural differences translate directly into per-step training cost: DAI- P requires approximately 3.50×10^6 FLOPs per gradient step, whereas DAI- A and DAI- F require 8.37×10^6—a 2.4× overhead attributable to the $|\mathcal{A}|$=3 parallel TransitionNet passes needed to evaluate the action-posterior KL (14). Crucially, DAI- P achieves its leading win rate with the smallest parameter budget among all DAI variants, having 38% fewer trainable parameters than DAI- A and DAI- F (9,420 vs. 15,183). Notice that the agent with strictly fewer parameters and a lower per-step cost outperforms its more complex counterparts, confirming that the performance gap is attributable to the EFE formulation rather than to model size.

Summary. Taken together, the results support three conclusions. First, the canonical perceptive epistemic term in DAI- P provides a clear and consistent advantage over all deep RL baselines in terms of both win rate and speed of convergence, demonstrating that targeted intrinsic motivation derived from proprioceptive prediction error constitutes an effective inductive bias for strategic CTI label acquisition. Second, the active mutual-information extension DAI- A matches DAI- P's performance without degrading it, while DAI- SA's surrogate formulation is consistently inferior, suggesting that a faithful model of state transition dynamics is more important than the specific information-theoretic quantity being minimised. Third, SAC's entropy regularisation does not substitute for structured epistemic gain: despite higher raw exploration, SAC achieves lower win rates and lower reward across all decomposition metrics, highlighting that the *direction* of exploration – towards proprioceptively informative transitions – matters as much as its magnitude. Table 2 summarises these conclusions.

7 Conclusions and future works

This paper has explored the use of Active Inference for automatically managing the Cyber-Threat Intelligence curricula of online-learning network intrusion detection systems. Through an original decomposition of the Free Energy Minimisation principle in fully observable Markov Decision Processes, four distinct Deep Active Inference agents were developed and tested in a modified version of the TIGER environment against common Deep Reinforcement Learning-based agents.

The conducted experiments lead to the main conclusion that incorporating the epistemic gain terms derived from the expected free energy functional into the policy value function is beneficial for shaping an agent's effective exploratory behaviour. Specifically, this work demonstrates that EFE-driven exploration can optimise effective cumulative pragmatic returns beyond what simpler exploration techniques, such as action-sampling and entropy maximisation, are capable of achieving. These conclusions not only align with current literature in the field [31, 57, 58], but are also supported by recent insights from the reinforcement learning community [59, 60]. This is consistent with the clarification in Sect. 3: even when the pragmatic component is implemented directly from the environment reward, the epistemic terms still provide a practically useful inductive bias for exploration, which is precisely where the LION results show the strongest benefit. It should be noted, however, that our claims are specifically validated within the unique parameter subspace of the LION setting. Therefore, further research is needed to enlarge the applicability of these techniques across a broader conceptualisation of CTI management. Additional conclusions of this paper provide preliminary insights for these future research directions and are outlined below:

- **Transition dynamics:** In this work, perceptive epistemic gains are modelled as a function of the transition model within the fully-observable MDP framework. This design choice implies that the successful approximation of these epistemic gains is contingent upon accurately learning the transition dynamics. Consequently, if predicting state transition dynamics proves challenging in any CTI management model, then the perceptive epistemic term may be inaccurately estimated, thereby compromising the correctness of the EFE estimation. Given that the state-space in automated CTI management is intrinsically linked to the learning dynamics of cyber-defensive systems, further research should investigate particular modelisation techniques that favour the learnability of the transition dynamics in this context.
- **Scale-free Active Inference.** Active-inference is well known as a scale-free paradigm [35, 61], where com-

Table 2 Ablation summary: component comparisons and their key empirical findings

Comparison	Component isolated	Key finding
DDQN vs. DAI- P	Perceptive epistemic term	DAI- P wins more often; DDQN accumulates reward at similar rate but too late per episode
DAI- P vs. DAI- A	Active MI bonus (exact KL)	No incremental benefit; win rates overlap
DAI- P vs. DAI- SA	Active MI bonus (surrogate MSE)	DAI- SA lags; surrogate insensitive to proprioceptive state changes at high precision
DAI- A vs. DAI- SA	Exact KL vs. surrogate MSE	Exact formulation substantially outperforms
DAI- P vs. DAI- F	Perceptive alone vs. combined	No additional benefit; perceptive term sufficient
DAI vs. SAC	Structured vs. undirected bonus	Entropy bonus cannot replace targeted proprioceptive epistemic signal

plex intelligent behaviours can be explained as the composition of multiple lower-level agents, each with a simple, goal-directed behaviour [62]. In this vein, cyber-defensive research for the IoT could significantly benefit from an active inference-based modelisation of low-level Markov blankets associated with intelligent sensors and actuators. It is worthwhile to study which simple modelisation schemes effectively compose macroscopic defensive behaviours, such as orchestrated cyber-physical proactive defence.

- **Multi-domain evaluation.** The experimental validation presented in this paper is intentionally scoped as a first principled investigation of Deep Active Inference in a cyber-defence-inspired setting. Extending this evaluation to more diverse environments – encompassing varying network topologies, attack types, and scales – is an important and natural next step, which would further substantiate the generalisability of the epistemic gain mechanisms identified here.

8 Experimental setting details

8.1 Environment

Experiments are conducted on a controlled budget-episode environment designed to isolate the strategic value of Cyber Threat Intelligence (CTI) label acquisition under a hard budget constraint. At each step the agent receives one network flow drawn uniformly at random from seven two-dimensional Gaussian clusters and must choose one of three actions: *block* (reject the flow), *accept* (forward it), or *buy-label* (spend budget to reveal the true type of the flow's cluster). An episode terminates when the running budget reaches or exceeds the win threshold of 40, falls to 0, or 30 steps have elapsed (Fig. 4).

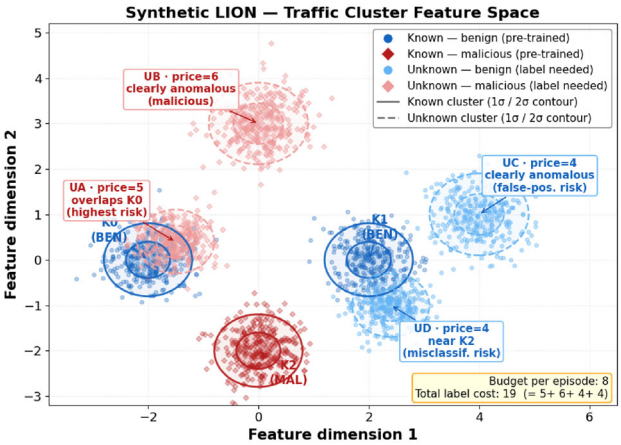


Fig. 4 Sampling data-points from the synthetic 2D feature space in the LION environment. In this environment Clusters' geometrical distributions, reward policies and other parameters were designed on purpose to force the agents to adopt a specific CTI-acquisition strategy

Three clusters are *known*: K0 (benign, $\mu=(-2, 0)$), K1 (benign, $\mu=(2, 0)$), and K2 (malicious, $\mu=(0, -2)$), all with $\sigma=0.4$. Four clusters are *unknown*: UA (malicious, $\mu=(-1.5, 0.4)$, $\sigma=0.35$) – deliberately placed to overlap the benign cluster K0; UB (malicious, $\mu=(0, 3)$, $\sigma=0.45$) – clearly anomalous; UC (benign, $\mu=(4, 1)$, $\sigma=0.45$) – visually conspicuous but harmless; and UD (benign, $\mu=(2.4, -1)$, $\sigma=0.35$) – near the malicious K2. The inference module classifies known flows correctly with high confidence after pretraining; unknown flows are either misclassified as the nearest known class or flagged as anomalous, depending on whether their distance to the nearest known prototype exceeds the anomaly threshold $\delta=1.8$ in the encoder's hidden space.

8.2 Reward structure and strategic difficulty

The reward structure is designed so that no single decision dominates the episode outcome; winning requires coordi-

Table 3 Synthetic LION reward structure

Cluster	Type	Accept	Block	Price
K0	Benign (known)	+2.0	−2.5	–
K1	Benign (known)	+2.0	−2.5	–
K2	Malicious (known)	−5.0	+2.5	–
UA	Malicious (unknown)	−7.0	+0.5	5
UB	Malicious (unknown)	−3.5	+1.5	6
UC	Benign (unknown)	+0.5	−1.5	4
UD	Benign (unknown)	+0.5	−1.0	4

nating flow-level classification decisions with a strategic label-acquisition plan. For known or already-labelled flows the rewards are symmetric and informed: accepting a benign flow yields +2.0, blocking it costs −2.5; accepting a malicious flow costs −5.0, blocking it yields +2.5. For unlabelled unknown flows, rewards reflect the a-priori ambiguity of each cluster: accepting UA yields −7.0 (blocking +0.5); accepting UB yields −3.5 (blocking +1.5); accepting UC yields +0.5 (blocking −1.5); and accepting UD yields +0.5 (blocking −1.0). A successful label purchase incurs only a tiny immediate signal (−price × 0.05), while the full label price is debited from the budget; an invalid purchase yields −0.3.

The core difficulty of the task stems from the combination of two constraints. First, the total cost of all four labels ($5 + 6 + 4 + 4 = 19$) exceeds the initial budget of 8, so the agent cannot afford to label every cluster; it must choose a strict subset and the *order* in which to acquire them. Second, the highest-value label – UA, whose uninformed acceptance penalty of −7.0 is the largest in the environment – belongs to the cluster with the *lowest* anomaly score, because UA overlaps the benign K0 region and is never flagged by the anomaly detector. An agent that naively prioritises suspicious-looking clusters (high anomaly score) will spend its budget on UB or UD first, exhaust the budget before reaching UA, and continue to incur heavy penalties from UA flows for the remainder of the episode. Winning consistently therefore requires discovering that UA is the highest priority label despite being the least visually anomalous cluster – precisely the inference that the canonical perceptive-epistemic term of DAI- P is designed to make. Table 3 summarises the reward policy in the LION environment. In such a table, columns *Accept* and *Block* show per-step rewards in the uninformed regime (before a label is purchased). Known-class flows always receive the informed rewards shown.

8.3 State representation

The 14-dimensional state vector is partitioned into two streams: a 5-dimensional *exteroceptive* vector (predicted known class, classifier confidence, anomaly score, and the raw two-dimensional normalised input coordinates) and a 9-dimensional *proprioceptive* vector (four per-cluster EMA confidence values `unk_conf`[0:3], fraction of labels bought, budget fraction, price fraction of the current cluster, time fraction, and a repeated confidence scalar).

The four proprioceptive confidence slots serve as the primary signal for the transition network of the DAI- P agent: when a label is purchased for cluster u, `unk_conf`[u] is immediately zeroed, producing a sharp, localised change in the proprioceptive state that the transition network cannot anticipate from the previous state alone, and therefore registers as maximum prediction error – equivalently, maximum perceptive epistemic gain.

8.4 Agents and hyperparameters

Nine agents are evaluated: the four proposed DAI variants (DAI- P, DAI- A,DAI- SA, DAI- F) and five baselines (DDQN, DUELING- DQN, PPO, SAC, and the deterministic Break-Even heuristic). All learned agents share the same two-stream network backbone and are trained with Adam ($\text{lr} = 4 \times 10^{-4}$). The shared replay buffer has capacity 6,000; training begins after 128 transitions have been collected; mini-batch size is 64; and target networks are hard-copied every 20 gradient steps. The discount factor is $\gamma = 0.95$. DAI- P, DAI- A, and DAI- F use an epistemic weight of $\varepsilon = 0.5$ and Boltzmann action selection with precision $\beta = 100$. DDQN and DUELING- DQN use ε-greedy exploration with $\varepsilon_{\text{start}} = 0.25$, $\varepsilon_{\text{end}} = 0.05$, and multiplicative decay 0.9995 per gradient step. PPO collects on-policy rollouts over full episodes, applies GAE with $\lambda = 0.95$, four gradient epochs per episode, clipped surrogate loss ($\epsilon = 0.20$), value coefficient 0.5, and entropy bonus 0.01. SAC (discrete-action formulation [56]) uses twin critics with polyak averaging ($\tau = 0.005$) and auto-tuned temperature with target entropy $\log(3) \times 0.6$.

8.5 Evaluation protocol

Each agent is trained for 500 episodes across 10 independent random seeds (seeds 0–9), yielding 90 runs in total (9 agents × 10 seeds). Each seed is associated with its own `DataGenerator` and an independently pretrained `InferenceModule`; pretraining runs for 120 Adam epochs over 200 samples per known class using a prototypical NLL loss. The pretrained encoder is serialised once per seed and shared identically across all agents for that seed, so differences in performance cannot be attributed to differences in the initial classifier quality. Agent-specific random streams are offset by `seed`$+1000\times$`agent_index` to guarantee statistical independence across all (seed, agent) pairs. Results are reported as mean ± 95% confidence interval, computed via the t-distribution with 9 degrees of freedom ($t_{9,\,0.975} \approx 2.262$) applied to the per-seed time series after

Table 4 Hyperparameter summary

Group	Parameter	Value
Environment	Max steps per episode	30
	Initial budget	8
	Win threshold	40
	Label prices (A, B, C, D)	5, 6, 4, 4
	Invalid-action penalty	-0.3
Inference module	Encoder input dimension	2
	Encoder hidden dimension	24
	Prototypical temperature τ_p	3.0
	Anomaly threshold δ	1.8
	Unknown-prototype EMA coeff. α_{ema}	0.08
	Pretraining learning rate	10^{-3}
	Pretraining epochs	120
	Pretraining samples per class	200
Network backbone	State dimension $\lvert\mathbf{s}\rvert$	14
	Exteroceptive stream dim	5
	Proprioceptive stream dim	9
	Hidden units per stream (TwoStreamNet)	48
	Hidden units (TransitionNet)	48
	Action space $\lvert\mathcal{A}\rvert$	3
Shared RL (all learned agents)	Optimiser	Adam
	Learning rate	4×10^{-4}
	Replay buffer capacity	6 000
	Min. transitions before training	128
	Mini-batch size	64
	Target-network update (steps)	20
	Discount factor γ	0.95
DAI variants	Epistemic weight ε	0.5
	Boltzmann precision ζ	100
DDQN/ DUELING- DQN	$\varepsilon_{\text{start}}$	0.25
	ε_{end}	0.05
	ε decay (per step)	0.9995
PPO	GAE λ	0.95
	Gradient epochs per episode	4
	Clip ε	0.20
	Value coefficient	0.5
	Entropy bonus coefficient	0.01
SAC	Polyak averaging τ	0.005
	Initial temperature α_0	0.1
	Target entropy (auto-tunes α)	$\log(3) \times 0.6$
Evaluation protocol	Training episodes	500
	Random seeds	10
	Smoothing window (episodes)	20

Parameters already discussed in the text are included for completeness

a rolling smoothing window of 20 episodes. Table 4 contains a more complete list of the hyper-parameters used in all our experiments. For reproducing our results, the interested reader can use refer to our publicly available repository at github.com/DISTA-IoT/lion. All experiments were conducted on a shared computing infrastructure equipped with eight NVIDIA A100-SXM4-80GB GPUs (80 GB HBM2e VRAM each, TDP 400 W, CUDA 12.3, driver 535.247.01) and dual AMD EPYC 7742 processors (2 sockets × 64 cores × 2 threads, 256 logical cores in total), with 1 TB of system RAM. The 90 independent runs (9 agents × 10 seeds) were dispatched in parallel across the available GPUs using a round-robin assignment scheme, with each (seed, agent) pair assigned to a dedicated GPU worker to guarantee full statistical independence between runs.

Author Contributions The authors equally contributed to the design and development of the proposed system and approach.

Funding Open access funding provided by Università degli Studi dell'Insubria within the CRUI-CARE Agreement.

Data Availability Our code implementation, datasets, and pre-trained models are publicly available in the third and latest release of the SmartVille repository, accessible at https://github.com/ DISTA-IoT/smartville/releases/tag/3.0_ACID. Leveraging the Hydra [4] library, all of our experiments are easily reproducible using the corresponding override configuration, which contains the full list of hyperparameters used.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Wang M, Yang N, Gunasinghe D, Weng N (2023) On the robustness of ml-based network intrusion detection systems: an adversarial and distribution shift perspective. De Computis 12(10):209–209 (**MAG ID: 4387705798 S2ID: e569ba3199e0c4bc62b2ef5128a02d3bf8580888**)
2. Cong Z, Luo X, Jian P, Zhu F, Zhang Y (2021) Data pricing in machine learning pipelines
3. Yang J, Chen X, Chen S, Jiang X, Tan X (2021) Conditional variational auto-encoder and extreme value theory aided two-stage learning approach for intelligent fine-grained known/unknown intrusion detection. IEEE Trans Inf Forensics Secur 16:3538–3553
4. Cevallos J, Rizzardi A, Sicari S, Coen-Porisini A (2024) Nero: neural algorithmic reasoning for zero-day attack detection in the iot: a hybrid approach. Comput Secur 142:103898
5. Amalapuram SK, Channappayya S, Tamma BR (2023) Augmented memory replay-based continual learning approaches for network intrusion detection. Adv Neural Inf Process Syst 36:17156–17169
6. Cevallos J, Rizzardi A, Sicari S, Coen-Porisini A (2025) Tiger: an open-source cyber-threat intelligence game environment for reinforcement learning. Submitted
7. Parr T, Pezzulo G, Friston KJ (2022) Active inference: the free energy principle in mind, brain, and behavior. MIT Press, Cambridge
8. Friston K, Da Costa L, Sajid N, Heins C, Ueltzhöffer K, Pavliotis GA, Parr T (2023) The free energy principle made simpler but not too simple. Phys Rep 1024:1–29
9. Zhang Z, Xu F (2024) An overview of the free energy principle and related research. Neural Comput 36(5):963–1021
10. Pio-Lopez L, Kuchling F, Tung A, Pezzulo G, Levin M (2022) Active inference, morphogenesis, and computational psychiatry. Front Comput Neurosci 16:988977
11. Friston K, FitzGerald T, Rigoli F, Schwartenbeck P, O'Doherty J, Pezzulo G (2016) Active inference and learning. Neurosci Biobehav Rev 68:862–879
12. Kirchner L, Eckert A-L, Berg M (2022) From broken models to treatment selection: active inference as a tool to guide clinical research and practice. Clin Psychol Europe 4(2):e9697
13. Albarracin M, Pitliya RJ, Smithe TSC, Friedman D, Friston K, Ramstead M (2024) Shared protentions in multi-agent active inference. Entropy **(S2ID: 7b2aca8cb0f297c53c539ece9c36fe9a06554f4d)**
14. Friston K, FitzGerald T, Rigoli F, Schwartenbeck P, Pezzulo G (2017) Active inference: a process theory. Neural Comput 29(1):1–49
15. Ororbia A, Mali A, Kohan A, Millidge B, Salvatori T (2024) A review of neuroscience-inspired machine learning. arXiv:2403.18929 [cs]
16. Ramstead M, Friston K, Hipólito I (2020) Is the free-energy principle a formal theory of semantics? From variational density dynamics to neural and phenotypic representations
17. Aldweesh A, Aldweesh A, Derhab A, Derhab A, Emam A, Emam A (2020) Deep learning approaches for anomaly-based intrusion detection systems: a survey, taxonomy, and open issues. Knowl Based Syst 189:105124 (**MAG ID: 2981025625 S2ID: 6a0e4209529b827cd86a52a4766968001a3f187c**)
18. Alotaibi F, Maffeis S (2024) Mateen: adaptive ensemble learning for network anomaly detection. In: International symposium on recent advances in intrusion detection **(S2ID: 0a584a51cf00d719f019ff936f5ba34ae728318b)**
19. Hendrycks D, Gimpel K (2017) A baseline for detecting misclassified and out-of-distribution examples in neural networks. In: 5th international conference on learning representations, ICLR 2017, Toulon, France, April 24–26, 2017, conference track proceedings. OpenReview.net
20. Cevallos JFM, Rizzardi A, Sicari S, Coen-Porisini A (2025) HERO: from high-dimensional network traffic to zERO-Day attack detection. Comput Netw **(in press)**
21. Jan STK, Hao Q, Hu T, Pu J, Oswal S, Wang G, Viswanath B (2020) Throwing darts in the dark? Detecting bots with limited data using neural data augmentation. In: 2020 IEEE symposium on security and privacy (SP), May 2020, pp 1190–1206 **(ISSN: 2375-1207)**
22. Amalapuram SK, Tadwai A, Vinta R, Channappayya SS, Tamma BR (2022) Continual learning for anomaly based network intrusion detection. In: 2022 14th international conference on communication systems & NETworkS (COMSNETS). IEEE, pp 497–505
23. Han D, Wang Z, Chen W, Wang K, Yu R, Wang S, Zhang H, Wang Z, Jin M, Yang J, Shi X, Yin X (2023) Anomaly detection in the open world: normality shift detection, explanation, and adaptation. In: Proceedings 2023 network and distributed system security symposium, San Diego, CA, USA, 2023. Internet Society

24. Amalapuram SK, Tamma BR, Channappayya SS (2024) Spider: a semi-supervised continual learning-based network intrusion detection system. In: IEEE INFOCOM 2024-IEEE conference on computer communications. IEEE, pp 571–580
25. Talpini J, Sartori F, Savi M (2024) Hierarchical multi-class continual learning for network intrusion detection. In: IEEE conference on network softwarization **(S2ID: 89f042a6d1ef13d1c72667a2d75653a0a291e561)**
26. Cevallos J, Rizzardi A, Sicari S, Coen-Porisini A (2024) Asap: automatic synthesis of attack prototypes, an online-learning, end-to-end approach. Comput Netw 110828
27. Li A, Qiu C, Kloft M, Smyth P, Rudolph M, Mandt S (2023) Zero-shot anomaly detection via batch normalization. Adv Neural Inf Process Syst 36:40963–40993
28. Du M, Du M, Chen Z, Chen Z, Liu C, Liu C, Oak R, Oak R, Oak R, Oak R, Song D, Song D (2019) Lifelong anomaly detection through unlearning. In: Conference on computer and communications security, pp 1283–1297 **(MAG ID: 2988337058 S2ID: 2a28b9e66d1b65805f90e101fcba71c1af813be8)**
29. Wang W, Yang H, Meinel C, YaizÖzkan H, Serna CB, Machuca CM (2024) Feature distribution shift mitigation with contrastive pretraining for intrusion detection. In: 2024 IEEE international conference on machine learning for communication and networking (ICMLCN), 2024. arXiv: 2404.15382 **(S2ID: 4cf6774d4963f6798628fc96bdacba7c5b362b45)**
30. Tran D-HM (2025) Park enhancing GNN-based network intrusion detection systems through memory-replay approach. In: International conference on information networking **(S2ID: 29e6b1301137f9237786bbc376bb08487a4bd0f5)**
31. Millidge B (2020) Deep active inference as variational policy gradients. J Math Psychol 96:102348
32. Friston K, Smith R (2020) The relationship between dynamic programming and active inference: the discrete, finite-horizon case. arXiv:2009.08111
33. Smith R, Friston KJ, Whyte CJ (2022) A step-by-step tutorial on active inference and its application to empirical data. J Math Psychol 107:102632
34. Fountas Z, Sajid N, Mediano P, Friston K (2020) Deep active inference agents using Monte-Carlo methods. Adv Neural Inf Process Syst 33:11662–11675
35. Friston K, Heins C, Verbelen T, Da Costa L, Salvatori T, Markovic D, Tschantz A, Koudahl M, Buckley C, Parr T (2024) From pixels to planning: scale-free active inference. arXiv:2407.20292 [cs]
36. Van Der Himst O, Lanillos P (2020) Deep active inference for partially observable MDPs. In: Verbelen T, Lanillos P, Buckley CL, De Boom C (eds) Active inference, vol 1326. Communications in Computer and Information Science. Springer International Publishing, Cham, pp 61–71
37. Champion T, Grześ M, Bonheme L, Bowman H (2023) Deconstructing deep active inference. arXiv:2303.01618 [cs]
38. Cevallos J, Rizzardi A, Sicari S, Coen-Porisini A (2023) Deep reinforcement learning for intrusion detection in internet of things: Best practices, lessons learnt, and open challenges. Comput Netw 236:110016
39. Elderman R, Pater LJJ, Thie AS, Drugan MM, Wiering MA (2017) Adversarial reinforcement learning in a cyber security simulation. In: 9th international conference on agents and artificial intelligence (ICAART 2017). SciTePress Digital Library, pp 559–566
40. Microsoft Defender Research Team (2021) Cyberbattlesim. https://github.com/microsoft/cyberbattlesim. 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
41. Sáez-de Cámara X, Flores JL, Arellano C, Urbieta A, Zurutuza U (2023) Gotham testbed: a reproducible IoT testbed for security experiments and dataset generation. IEEE Trans Depend Secure Comput 21(1):186–203
42. Nguyen HPT, Chen Z, Hasegawa K, Fukushima K, Beuran R (2024) Pengym: pentesting training framework for reinforcement learning agents. In: ICISSP, pp 498–509
43. Janisch J, Pevnỳ T, Lisỳ V (2023) Nasimemu: network attack simulator & emulator for training agents generalizing to novel scenarios. In: European symposium on research in computer security. Springer, pp 589–608
44. Hammar K, Stadler R (2022) Intrusion prevention through optimal stopping. IEEE Trans Netw Serv Manag 19(3):2333–2348
45. Millidge B, Tschantz A, Buckley CL (2021) Whence the Expected Free Energy? Neural Comput 33(2):447–482
46. Jaynes ET (1957) Information theory and statistical mechanics. Phys Rev 106(4):620–630
47. Friston K, Samothrakis S, Montague R (2012) Active inference and agency: optimal control without cost functions. Biol Cybern 106(8):523–541
48. Friston KJ, Daunizeau J, Kiebel SJ (2009) Reinforcement learning or active inference? PLoS ONE 4(7):e6421
49. Tschantz A, Millidge B, Seth AK, Buckley CL (2020) Reinforcement learning through active inference. arXiv:2002.12636. Presented at the bridging AI and cognitive science workshop (BAICS), ICLR 2020
50. Sajid N, Ball PJ, Parr T, Friston KJ (2021) Active inference: demystified and compared. Neural Comput 33(3):674–712
51. Millidge B (2020) Deep active inference as variational policy gradients. J Math Psychol 96:102348
52. Shin J, Kim C, Hwang HJ (2022) Prior preference learning from experts: designing a reward with active inference. Neurocomputing 492:67–86
53. Van Hasselt H, Guez A, Silver D (2016) Deep reinforcement learning with double q-learning. In: Proceedings of the AAAI conference on artificial intelligence, vol 30
54. Wang Z, Schaul T, Hessel M, Hasselt H, Lanctot M, Freitas N (2016) Dueling network architectures for deep reinforcement learning. In International conference on machine learning. PMLR, pp 1995–2003
55. Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O (2017) Proximal policy optimization algorithms. arXiv:1707.06347
56. Christodoulou P (2019) Soft actor-critic for discrete action settings. arXiv:1910.07207
57. Mazzaglia P, Verbelen T, Dhoedt B (2021) Contrastive active inference. In: Advances in neural information processing systems, vol 34. Curran Associates, Inc., pp 13870–13882
58. Nguyen VD, Yang Z, Buckley CL, Ororbia A (2024) R-AIF: solving sparse-reward robotic tasks from pixels with active inference and world models. arXiv:2409.14216 [cs]
59. Tschantz A, Millidge B, Seth AK, Buckley CL (2020) Reinforcement learning through active inference. arXiv:2002.12636 [cs]
60. Liu G, Tang M, Eysenbach B (2024) A single goal is all you need: skills and exploration emerge from contrastive RL without rewards, demonstrations, or subgoals. arXiv:2408.05804 cs
61. Pezzato C, Çatal O, Van de Maele T, Pitliya RJ, Verbelen T (2025) Mobile manipulation with active inference for long-horizon rearrangement tasks. arXiv:2507.17338 [cs]
62. B de Vries, Nuijten W, van de Laar T, Kouw W, Adamiat S, Nisslbeck T, Lukashchuk M, Nguyen HMH, Araya MH, Tresor R, Jenneskens T, Nikoloska I, Subramanian RG, van Erp B, Bagaev D, Podusenko A (2025) Expected free energy-based planning as variational inference. arXiv:2504.14898 [stat]

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Jesús F. Cevallos-Moreno received a Ph.D. in Computer Science Engineering from Sapienza University (Rome) in 2022. He now holds a post-doc researcher position at the University of Insubria (Varese). His main research interests are industrial/ethical applications of Deep Learning with a special focus on Optimal Control and Neural Algorithmic Reasoning.



Alessandra Rizzardi is an Associate Professor at the University of Insubria (Varese), where she received BS/MS degree in Computer Science 110/110 cum laude in 2011 and 2013, respectively. In 2016, she got Ph.D. in Computer Science and Computational Mathematics at the same university, under the guidance of Prof. Sabrina Sicari. Her research activity is on WSN and IoT security issues. She is a member of the ETT, ITL, and Sensors editorial boards. She is an IEEE member.



Sabrina Sicari is a Full Professor at the University of Insubria (Varese). She received the M.Sc. degree in Electronic Engineering, 110/110 cum laude, from the University of Catania in 2002, where, in 2006, she got Ph.D. in Computer and Telecommunications Engineering, under the supervision of Prof. Aurelio La Corte. She is a COMNET, IEEE IoT, ETT, and ITL editorial board member. Her research concerns security, privacy, and trust in WSN, WMSN, IoT, and distributed systems. She is an IEEE senior member.



Alberto Coen-Porisini received a Dr. Eng. degree and Ph.D. in Computer Engineering from Politecnico di Milano in 1987 and 1992. He has been a Full Professor of Software Engineering at UniversitÃ degli Studi dell' Insubria since 2001, Dean of the School of Science from 2006, and Dean from 2012 to 2018. His research regards specification/design of real-time systems, privacy models, and WSN.