Imperial College London

The Alan Turing Institute

Inroads into Autonomous Network Defence using Explained Reinforcement Learning

Myles Foley | Mia Wang | Zoe M | Chris Hicks | Vasilios Mavroudis

AGENDA

- REINFORCEMENT LEARNING!
- PROBLEM SCENARIO: NETWORK DEFENCE AS A GAME
- HIERARCHICAL RL : CONTROLLERS AND SUBAGENTS
- EVALUATION AND EXPLAINABILITY

REINFORCEMENT LEARNING

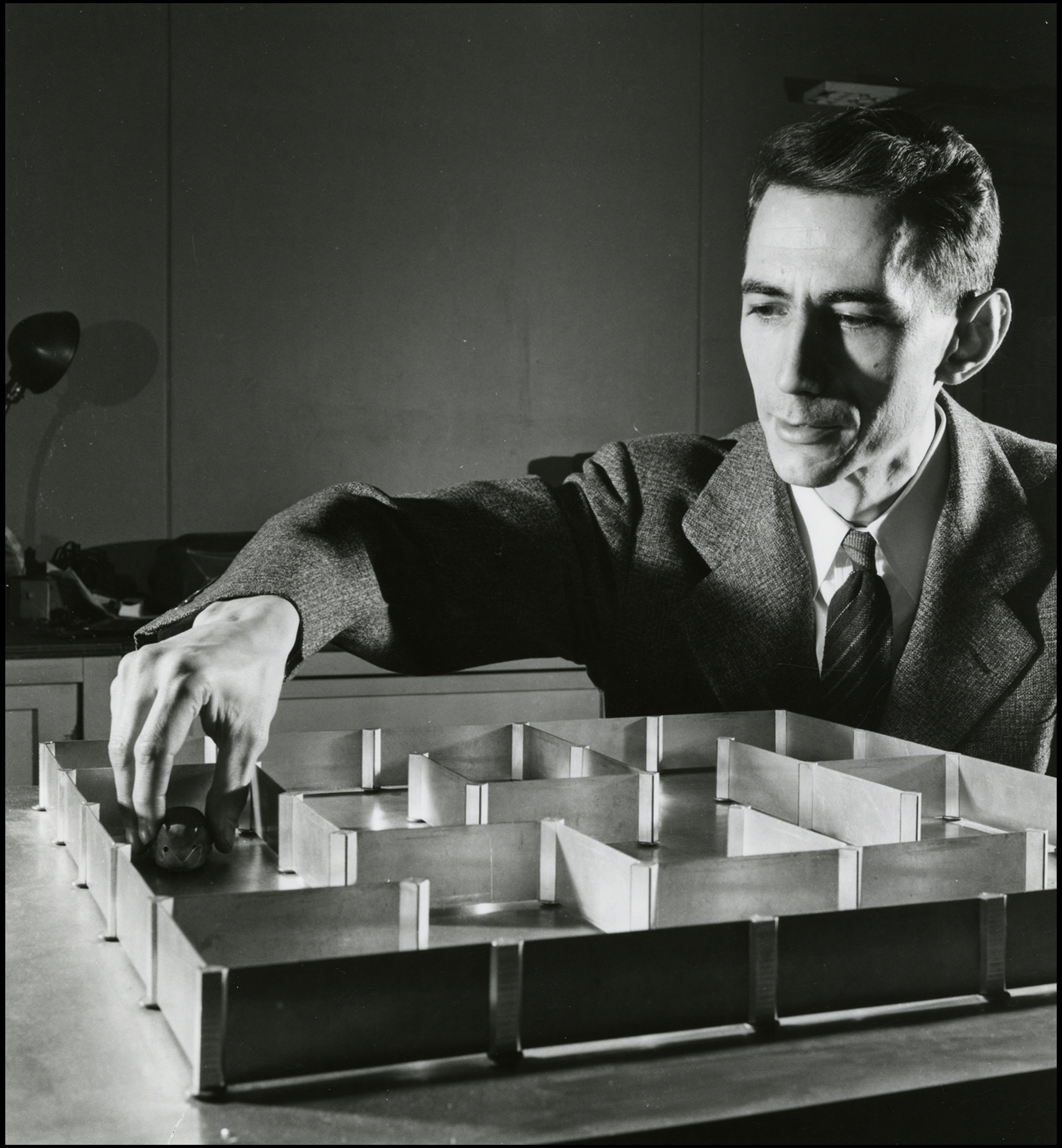
WHAT IS REINFORCEMENT LEARNING?

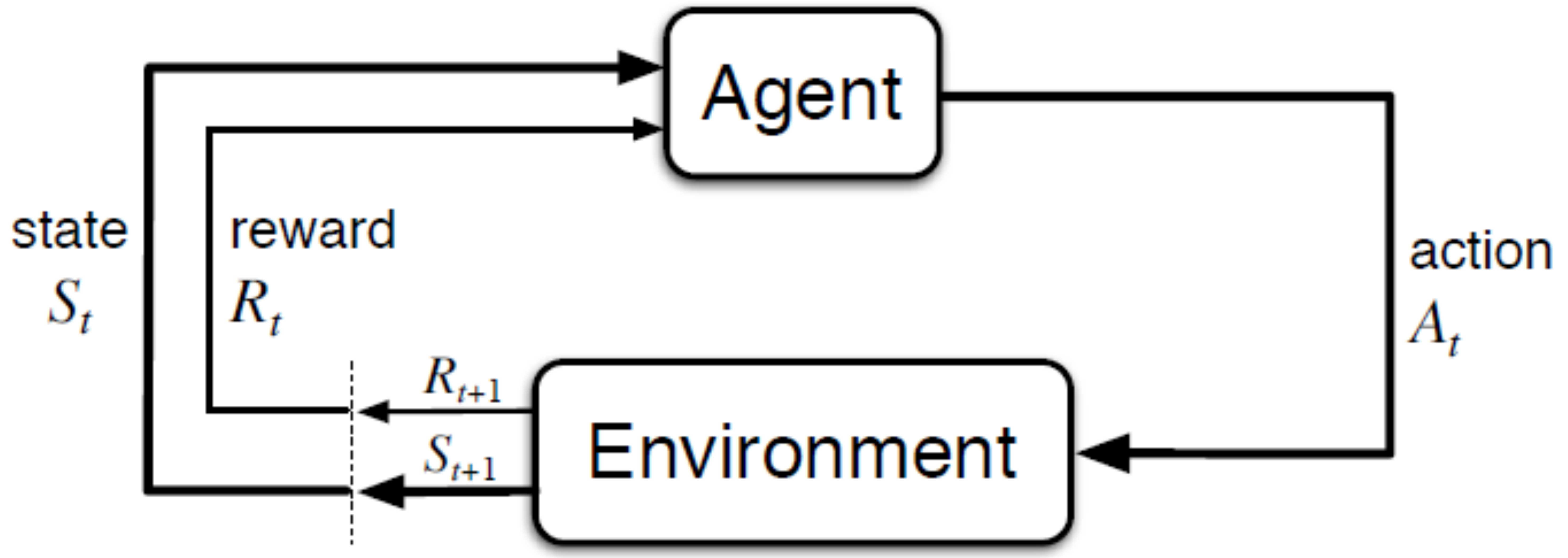
Environment

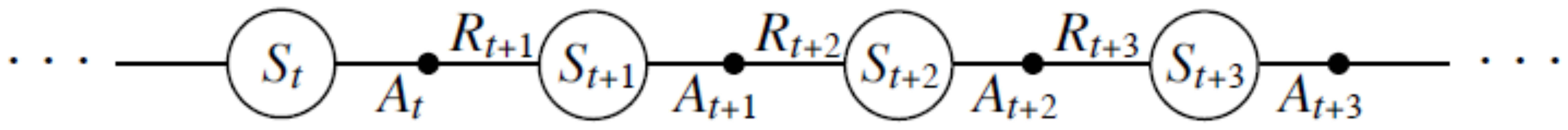
Agent

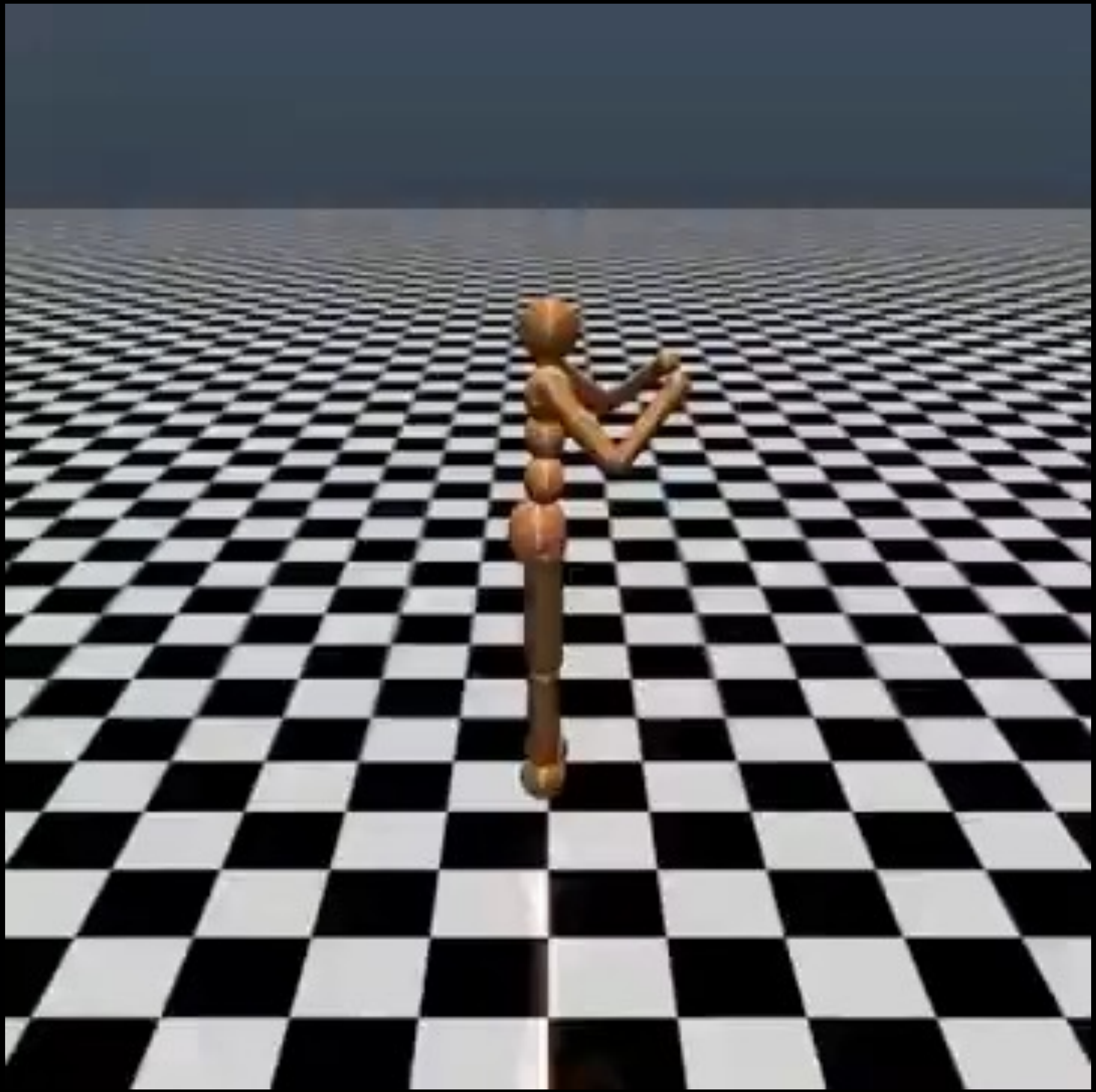
Reward

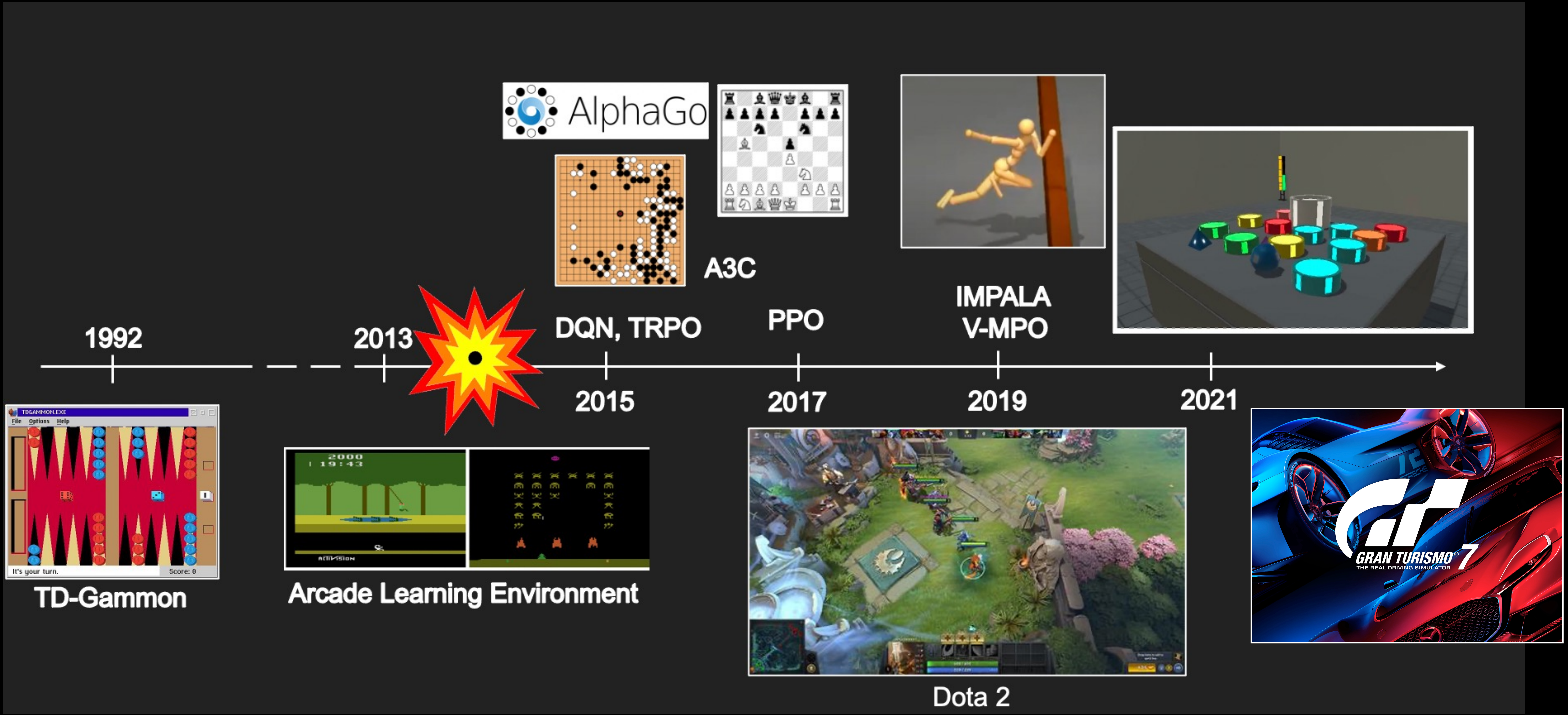
Policy




Agent
state
S_t
reward
R_t
action
A_t
R_{t+1}
S_{t+1}
Environment


S_t
A_t
R_{t+1}
S_{t+1}
A_{t+1}
R_{t+2}
S_{t+2}
A_{t+2}
R_{t+3}
S_{t+3}
A_{t+3}



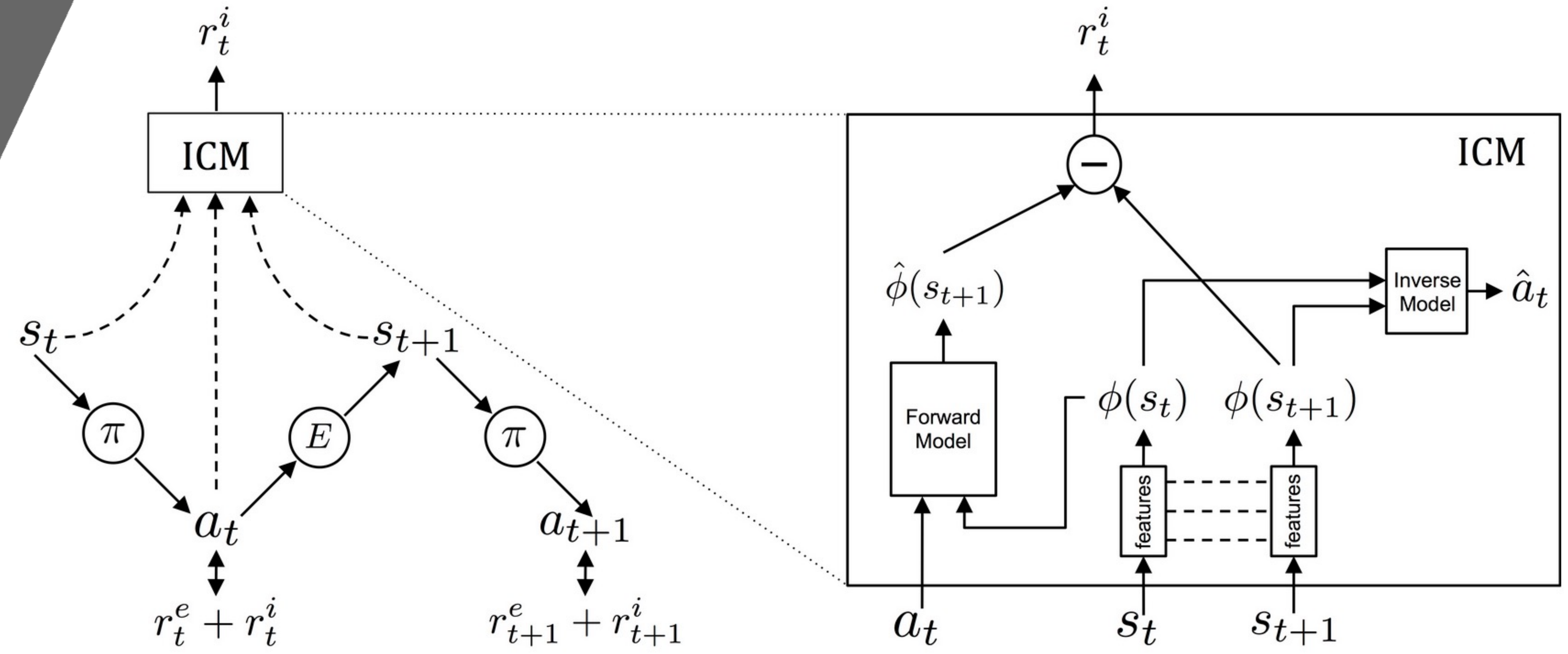


AlphaGo
A3C
1992
2013
DQN, TRPO
PPO
IMPALA
V-MPO
2015
2017
2019
2021
TDGAMMON.EXE
File Options Help
It's your turn.
Score: 0
TD-Gammon
Arcade Learning Environment
Dota 2
GRAN TURISMO 7
THE REAL DRIVING SIMULATOR

WHY REINFORCEMENT LEARNING?

- Security is hard, time consuming and continuous
- Security is often about playing ‘games’
- RL doesn’t need labeled examples (cf. supervised learning)
- RL learns by interacting with an environment
- RL goes beyond human level ability/SOTA

WHAT IS PPO? WHAT IS CURIOSITY?



Algorithm 4 PPO with Adaptive KL Penalty

Input: initial policy parameters θ_0, initial KL penalty β_0, target KL-divergence δ
for $k = 0, 1, 2, \ldots$ **do**
  Collect set of partial trajectories $\mathcal{D}_k$ on policy $\pi_k = \pi(\theta_k)$
  Estimate advantages $\hat{A}_t^{\pi_k}$ using any advantage estimation algorithm
  Compute policy update

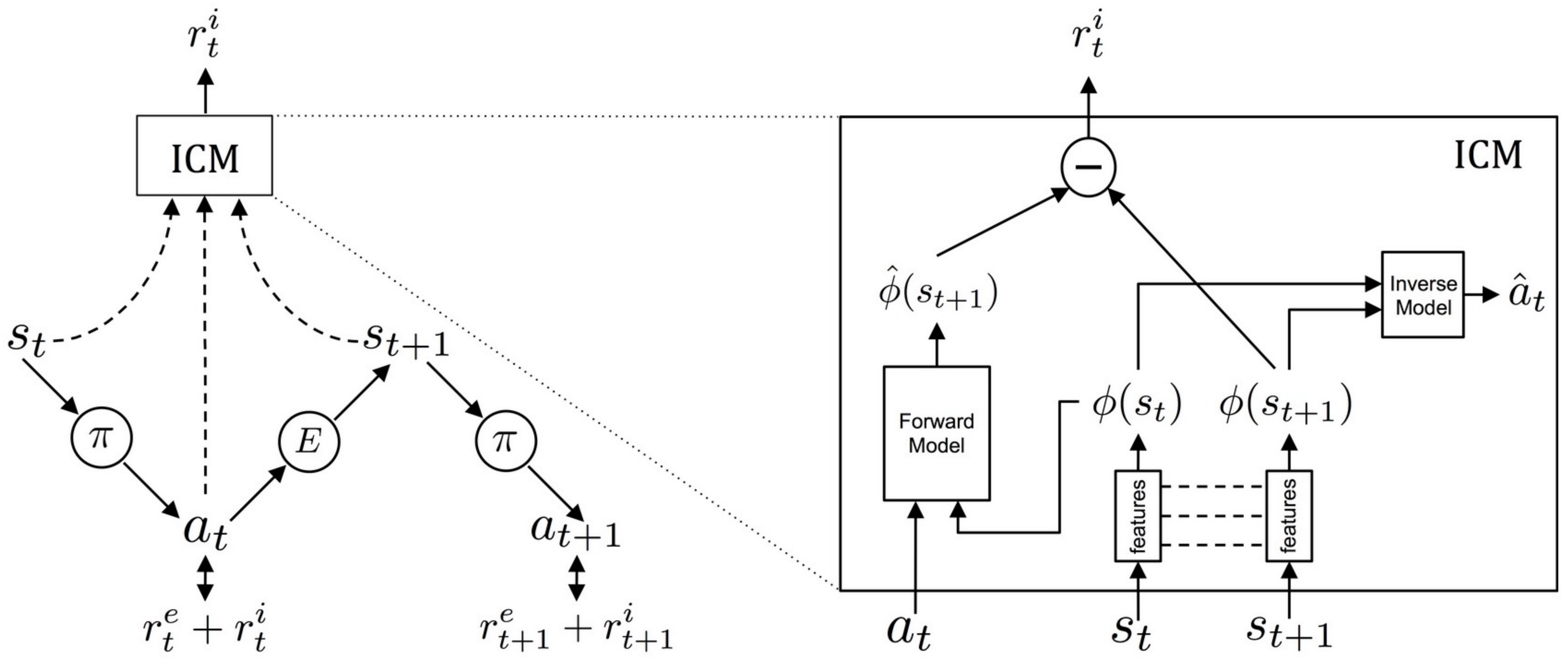
$$\theta_{k+1} = \arg\max_{\theta} \mathcal{L}_{\theta_k}(\theta) - \beta_k \bar{D}_{KL}(\theta || \theta_k)$$

  by taking K steps of minibatch SGD (via Adam)
  if $\bar{D}_{KL}(\theta_{k+1} || \theta_k) \geq 1.5\delta$ **then**
    $\beta_{k+1} = 2\beta_k$
  else if $\bar{D}_{KL}(\theta_{k+1} || \theta_k) \leq \delta/1.5$ **then**
    $\beta_{k+1} = \beta_k/2$
  end if
end for

PROXIMAL POLICY OPTIMISATION

- POLICY GRADIENT METHOD
- SCALES WITH REAL WORLD PROBLEM
- KL PENALTY / CLIPPING

“Curiouser and curiouser,” cried Alice.



INTRINSIC REWARD

INTRINSIC CURIOSITY MODULE

NOISE REDUCTION

NETWORK DEFENCE (CYBORG)

RED AGENTS

B-line Highly specialised goes straight for the production server

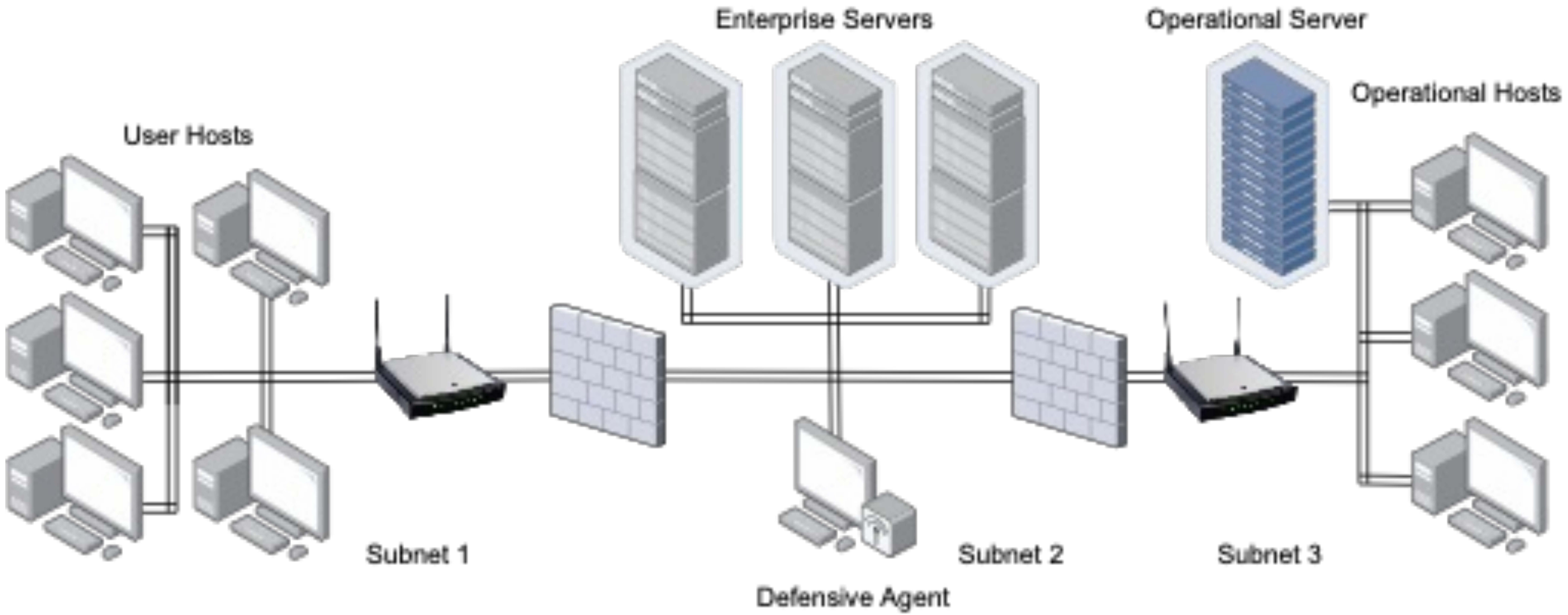
Meander More generalised but less efficient / stealthy

STOCHASTICITY

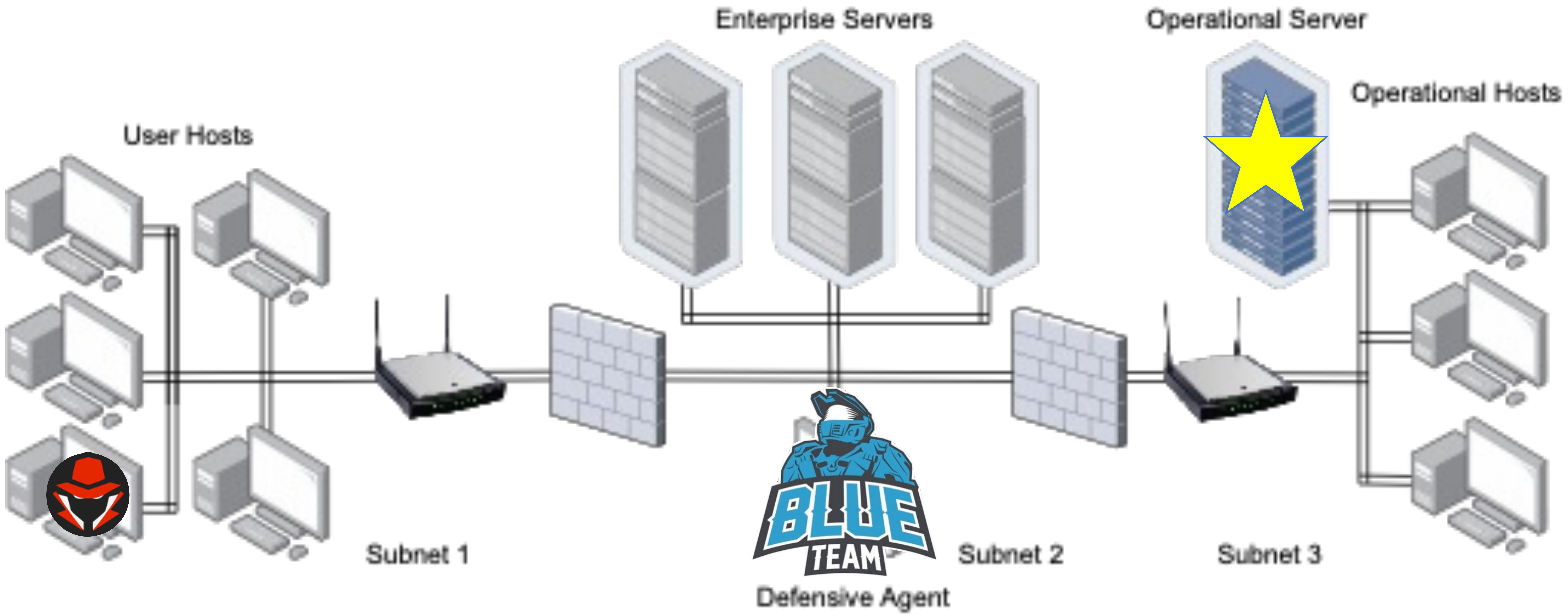
Sometimes things don't work as intended…

https://arxiv.org/abs/2108.09118

OPERATIONAL NETWORK



FLORIN'S OPERATIONAL NETWORK



RED ACTIONS

Action	Purpose	Output
Discover Remote Systems (per subnet)	ATT&CK[4] Technique T1018 Remote System Discovery. Discovers new hosts/IP addresses in the network through active scanning using tools such as ping.	IP addresses in the chosen subnet from hosts that respond to ping
Discover Network Services (per host)	ATT&CK Technique T1046 Network Service Scanning. Discovers responsive services on a selected host by initiating a connection with that host.	Ports and service information
Exploit Network Services (per host)	ATT&CK Technique T1210 Exploitation of Remote Services. This action attempts to exploit services on a remote system.	Success/Failure Initial recon of host if successful Information on why failed (unavailable port, etc.)
Escalate (per host)	ATT&CK Tactic TA0004 Privilege Escalation. This action escalates the agent's privilege on the host.	Success/FailureInternal information now available due to increased access to the host
Impact (per host)	ATT&CK Technique T1489 Service Stop. This action disrupts the performance of the network and fulfils red's objective of denying the operational service.	Success/Failure

BLUE ACTIONS

Action	Purpose	Output
Monitor (network level)	Collection of information about flagged malicious activity on the system. Corresponds to action ID 1: Scan in the OpenC2 specification[3].	Network connections and associated processes that are identified as malicious.
Analyse (per host)	Collection of further information on a specific host to enable blue to better identify if red is present on the system. Corresponds to action ID 30: Investigate in the OpenC2 specification.	Information on files associated with recent alerts including signature and entropy.
Honeypot (per host)	Setup of decoy services on a specific host. Green agents do not access these services, so any access is a clear example of red activity.	An alert if the red agent accesses the new service.
Remove user (per host)	Attempting to remove red from a host by destroying malicious processes, files and services. This action attempts to stop all processes identified as malicious by the monitor action. Corresponds to action ID 10: Stop in the OpenC2 specification.	Success/Failure
Restore (per host)	Restoring a system to a known good state. This has significant consequences for system availability. This action punishes Blue by -1. Corresponds to action ID 23: Restore in the OpenC2 specification.	Success/Failure

REWARDS

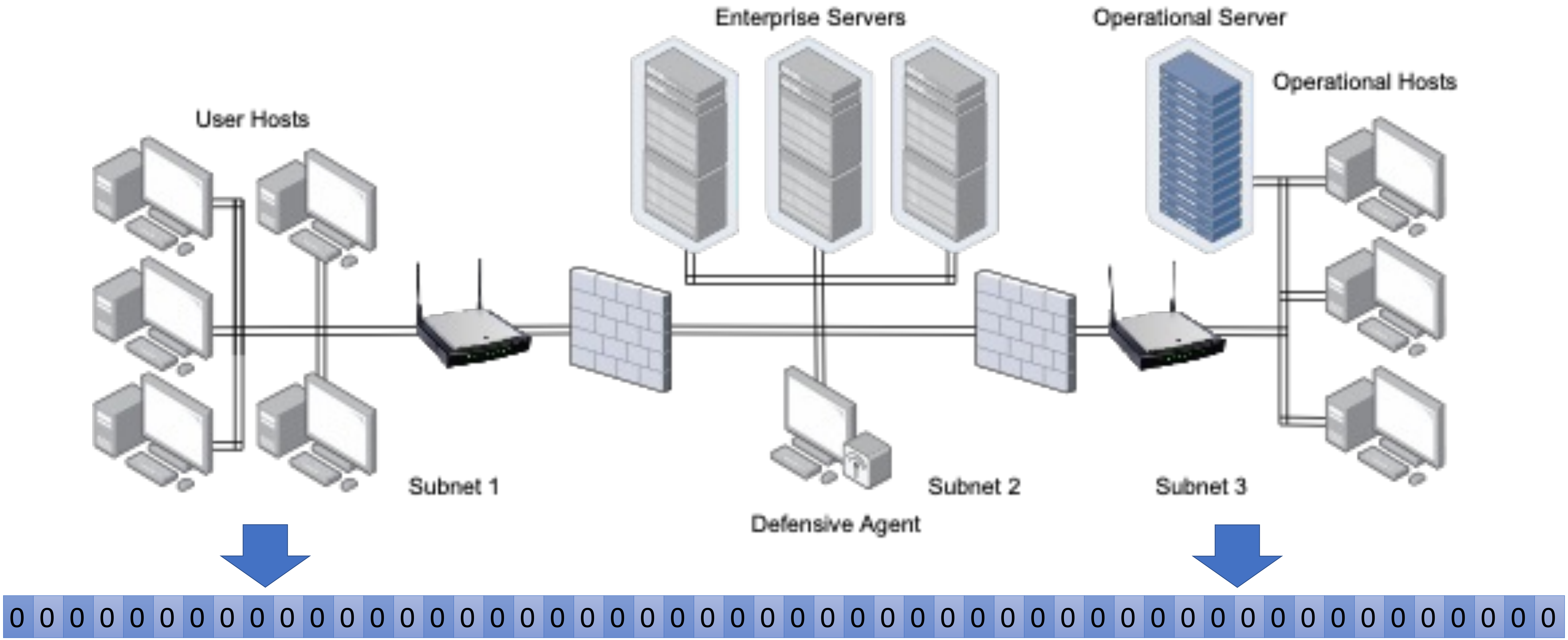
Subnet	Hosts	Blue Reward for Red Access (per turn)
Subnet 1	User Hosts	-0.1
Subnet 2	Enterprise Servers	-1
Subnet 3	Operational Server	-1
Subnet 3	Operational Hosts	-0.1

Table 1: Blue rewards for red administrator access (per turn)

Agent	Hosts	Action	Blue Reward (per turn)
Red	Operational Server	Impact	-10
Blue	Any	Restore	-1

Table 2: Blue rewards for successful red actions (per turn)

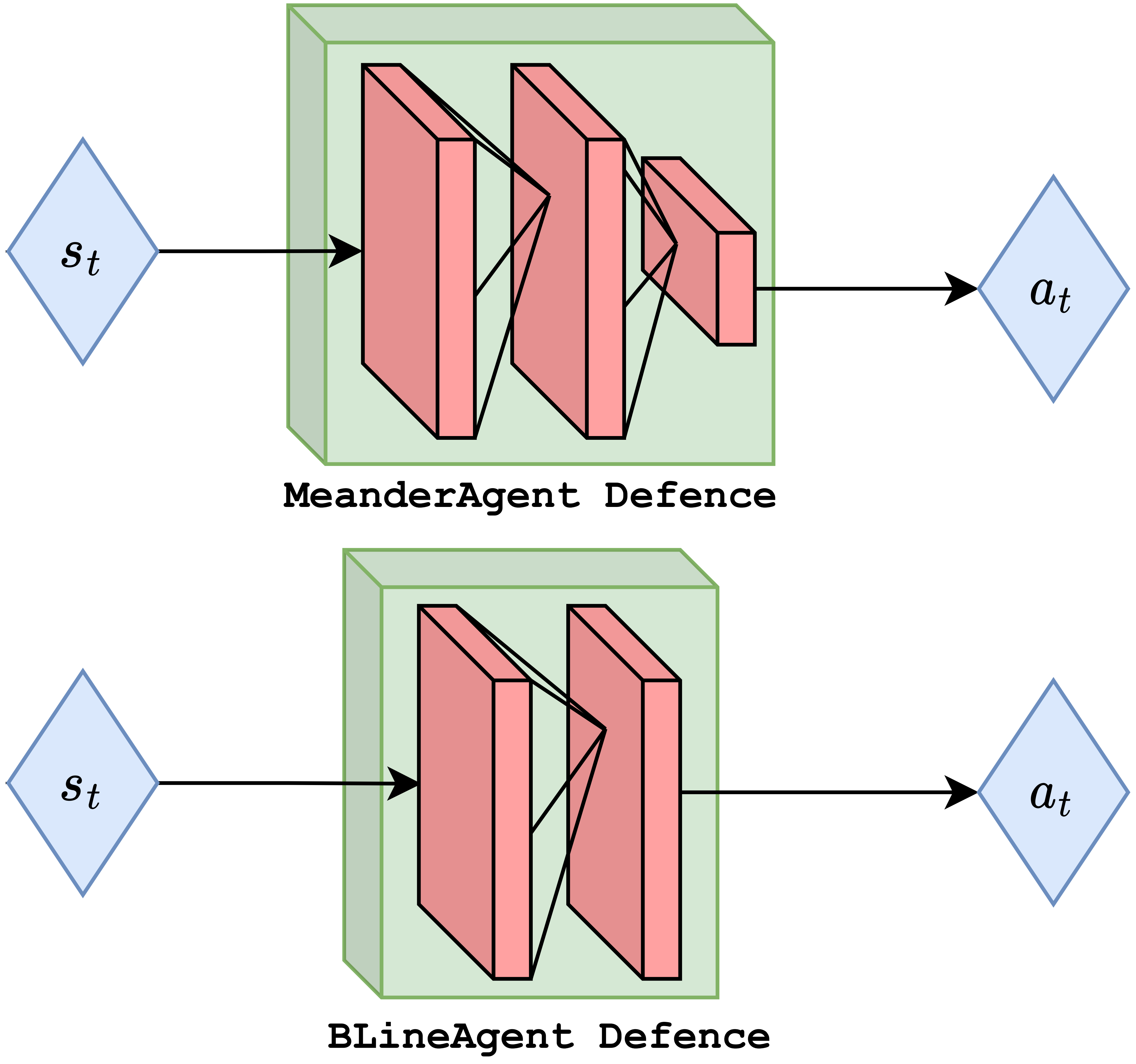
WHAT DO WE SEE?



RL TO THE RESCUE

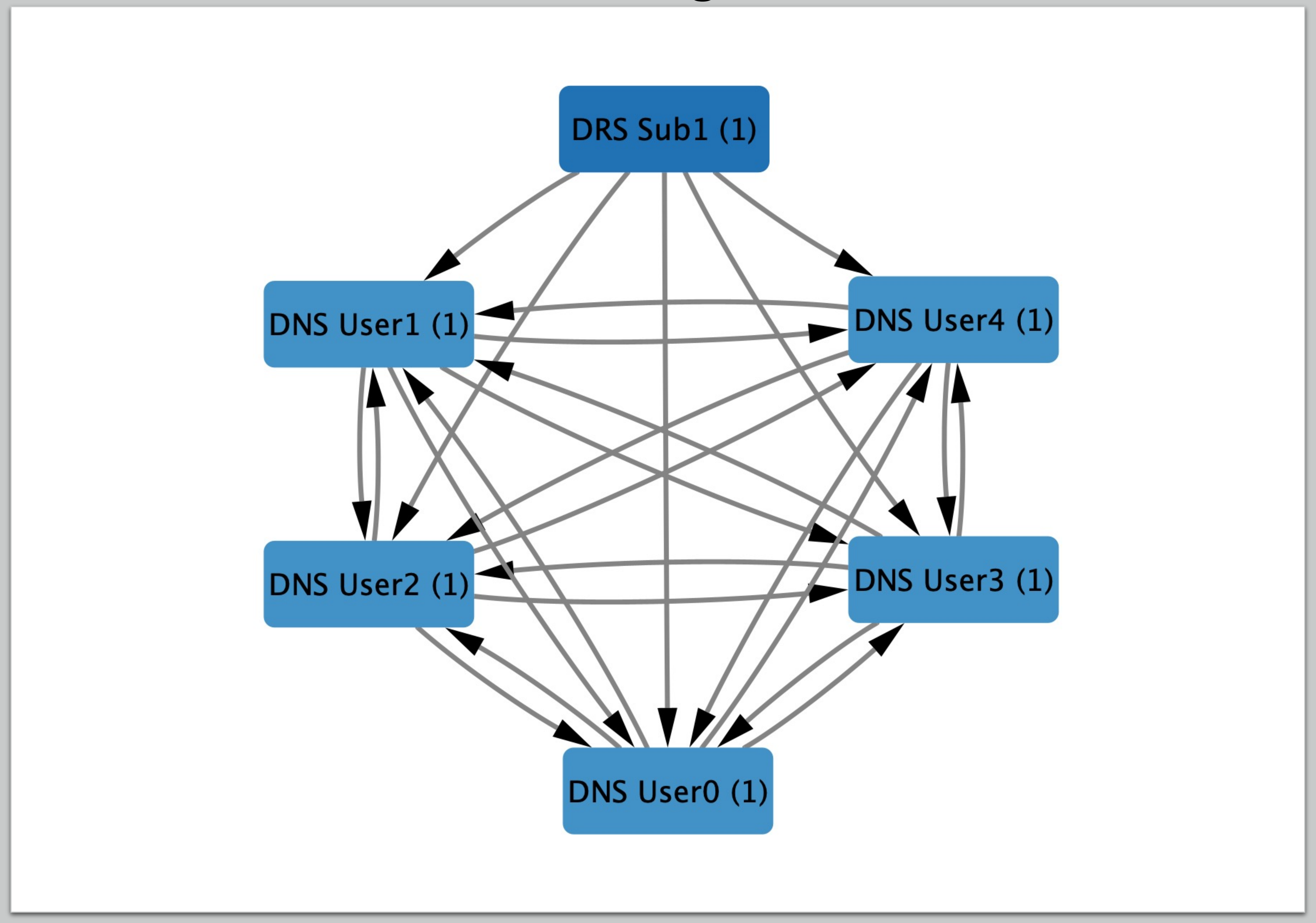
A TAILORED DEFENCE

- Defensive sub agents
  - MeanderAgent Defence:
    - Three layer network
    - No curiosity
  - BLineAgent Defence:
    - Two layer network
    - Curiosity
    - Prior Action Knowledge
    - State Representations

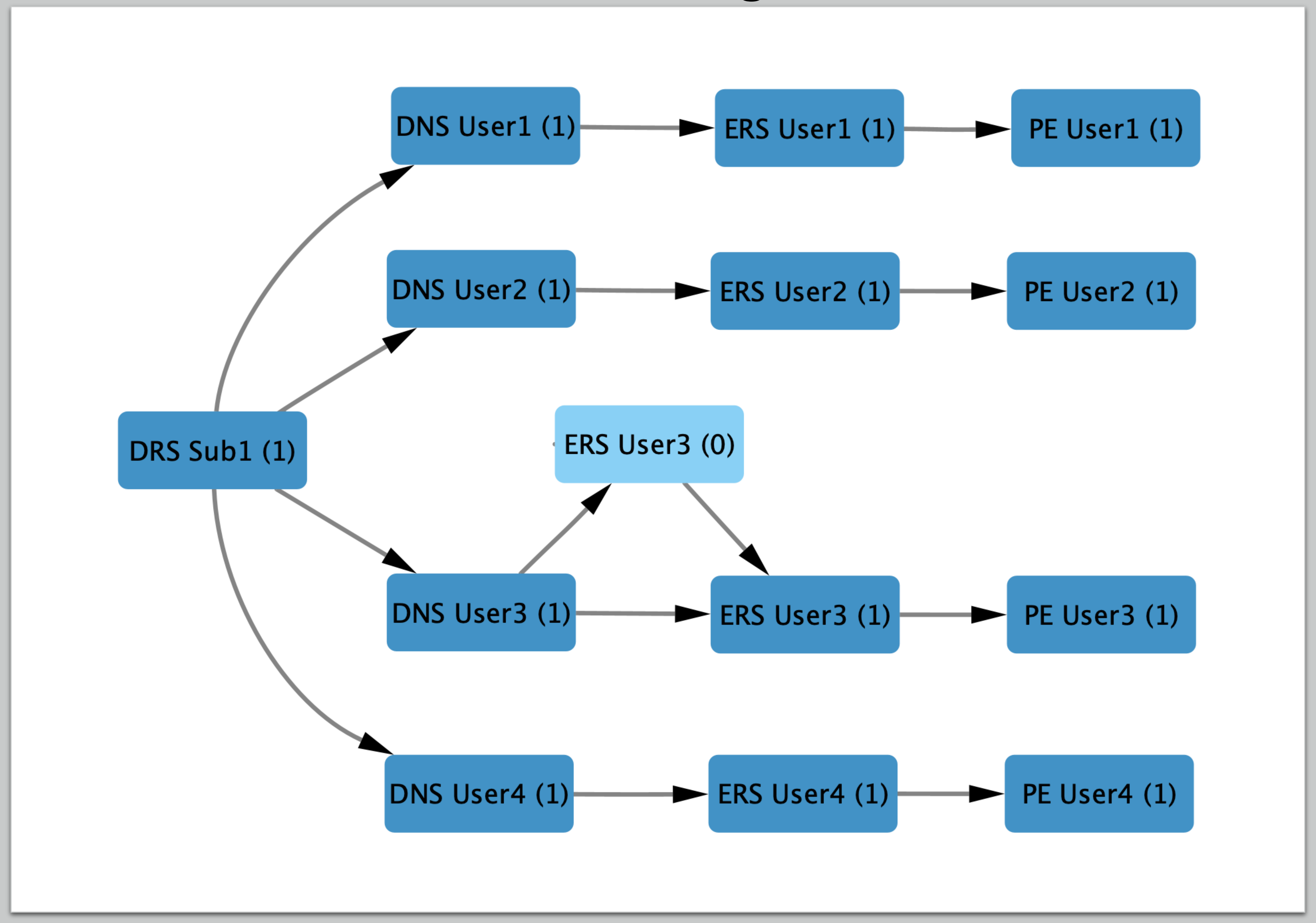


RED BEHAVIOURS

BLineAgent

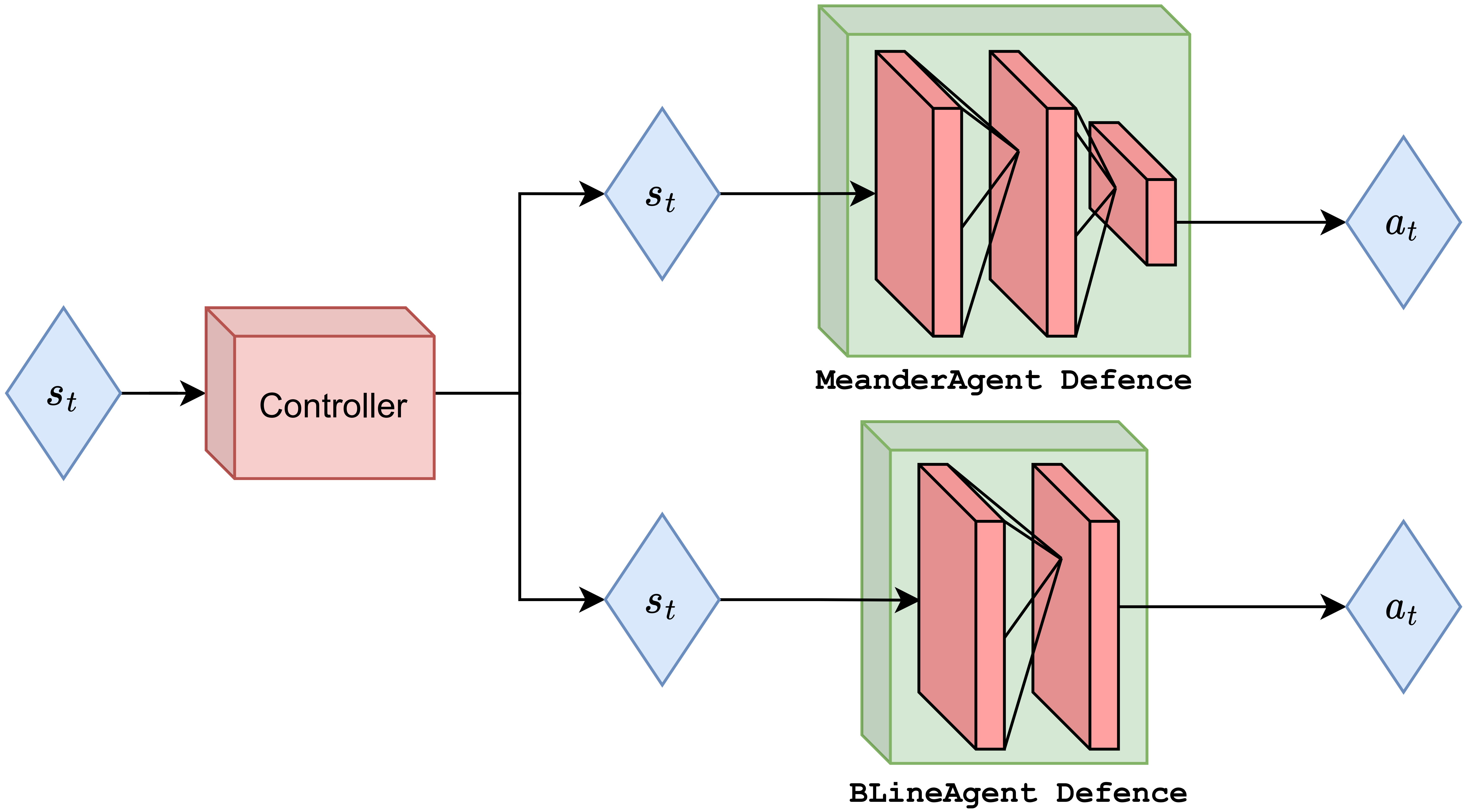


MeanderAgent



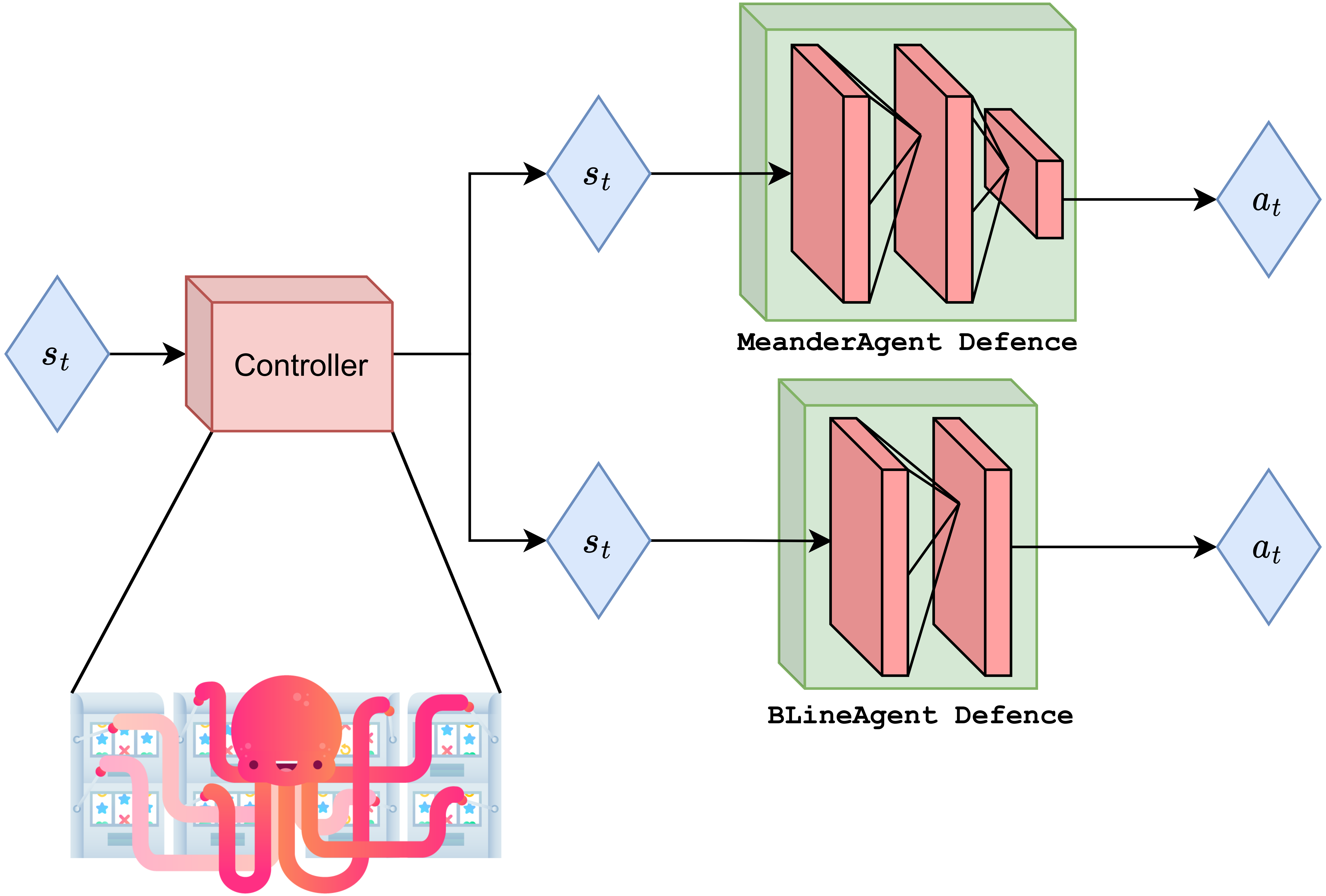
ONE AGENT TO RULE THEM ALL

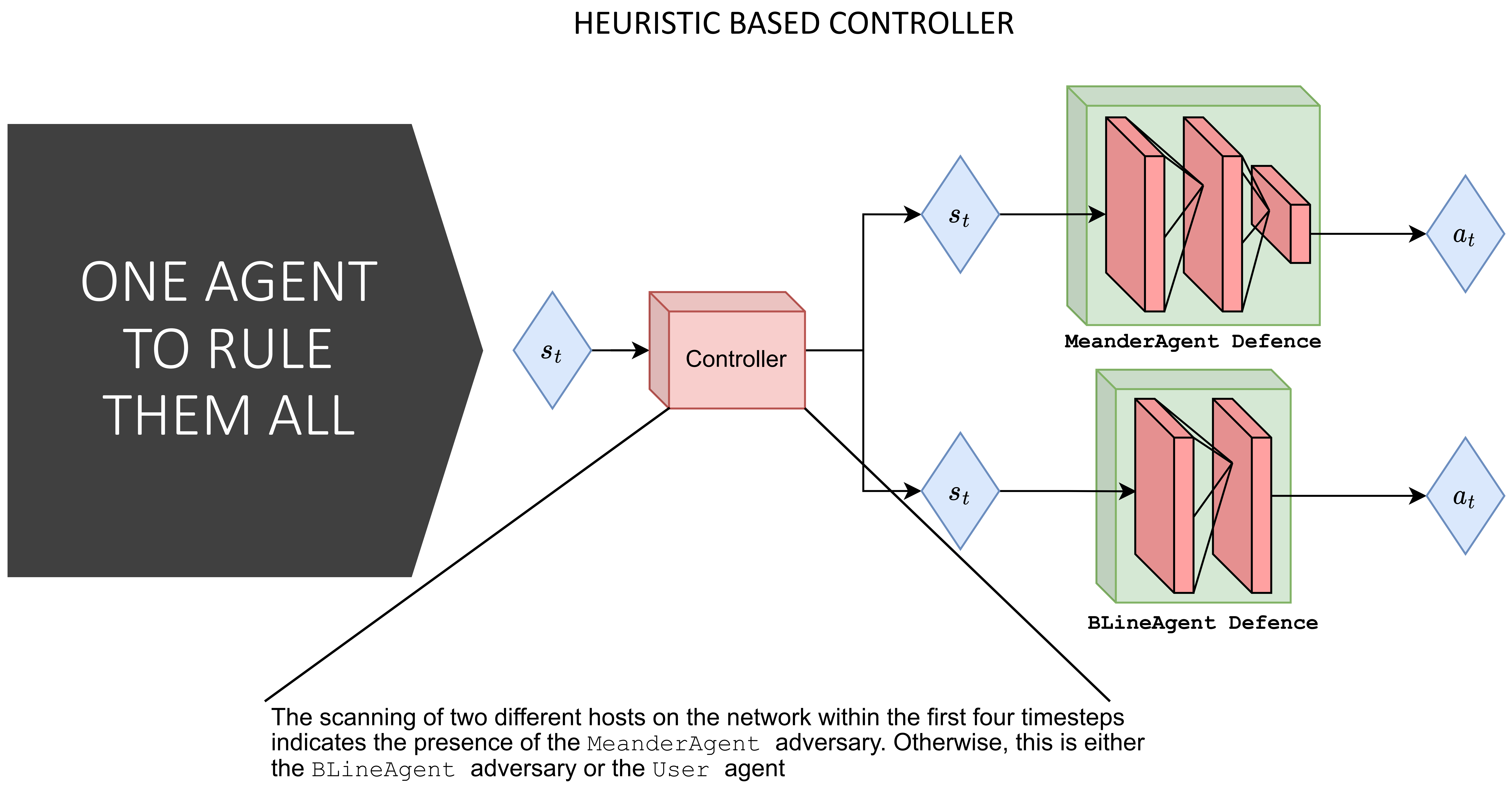
- TWO CONTROLLER AGENTS:
  - BANDIT BASED
  - HEURISTIC BASED



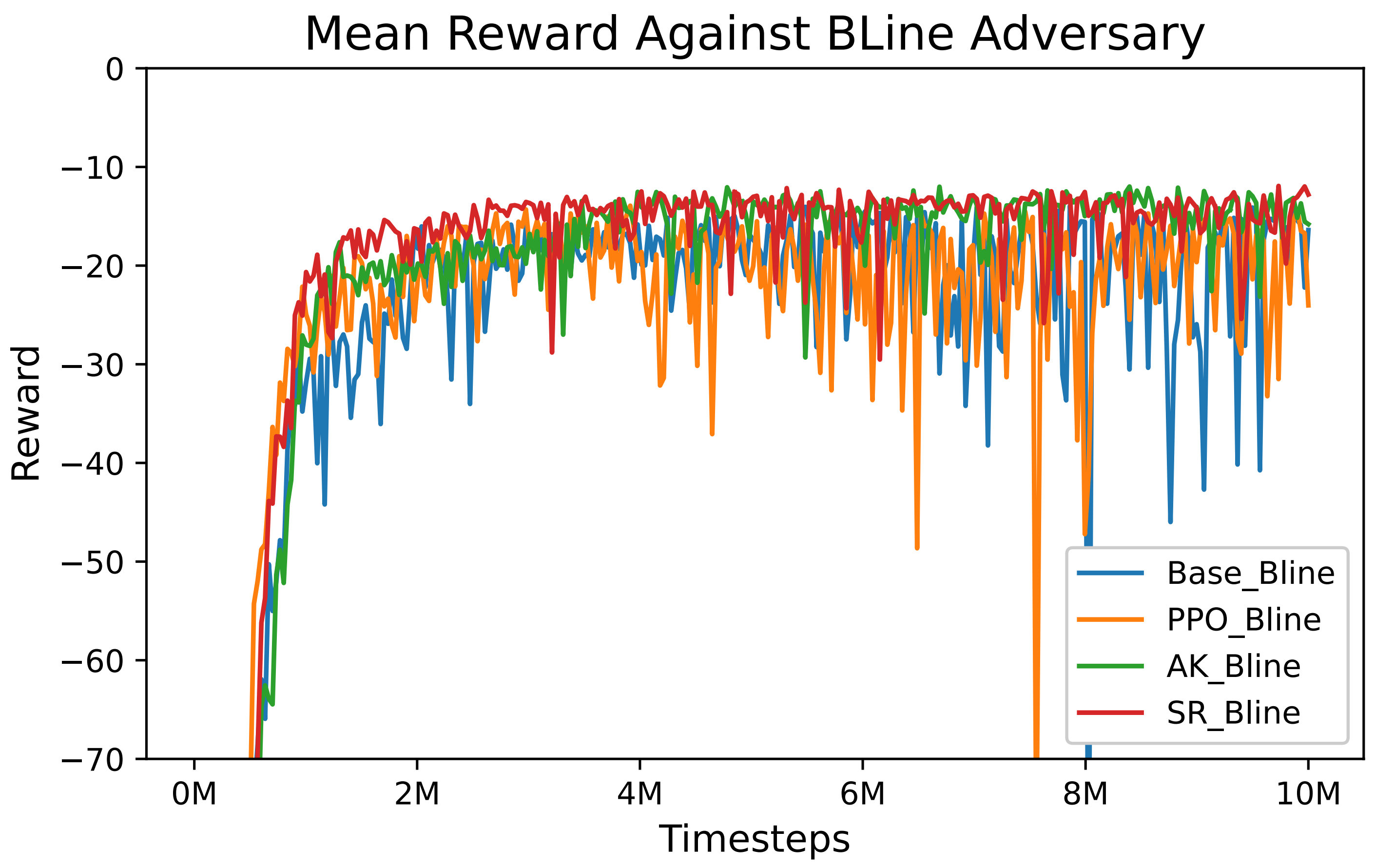
ONE AGENT TO RULE THEM ALL

- BANDIT BASED CONTROLLER

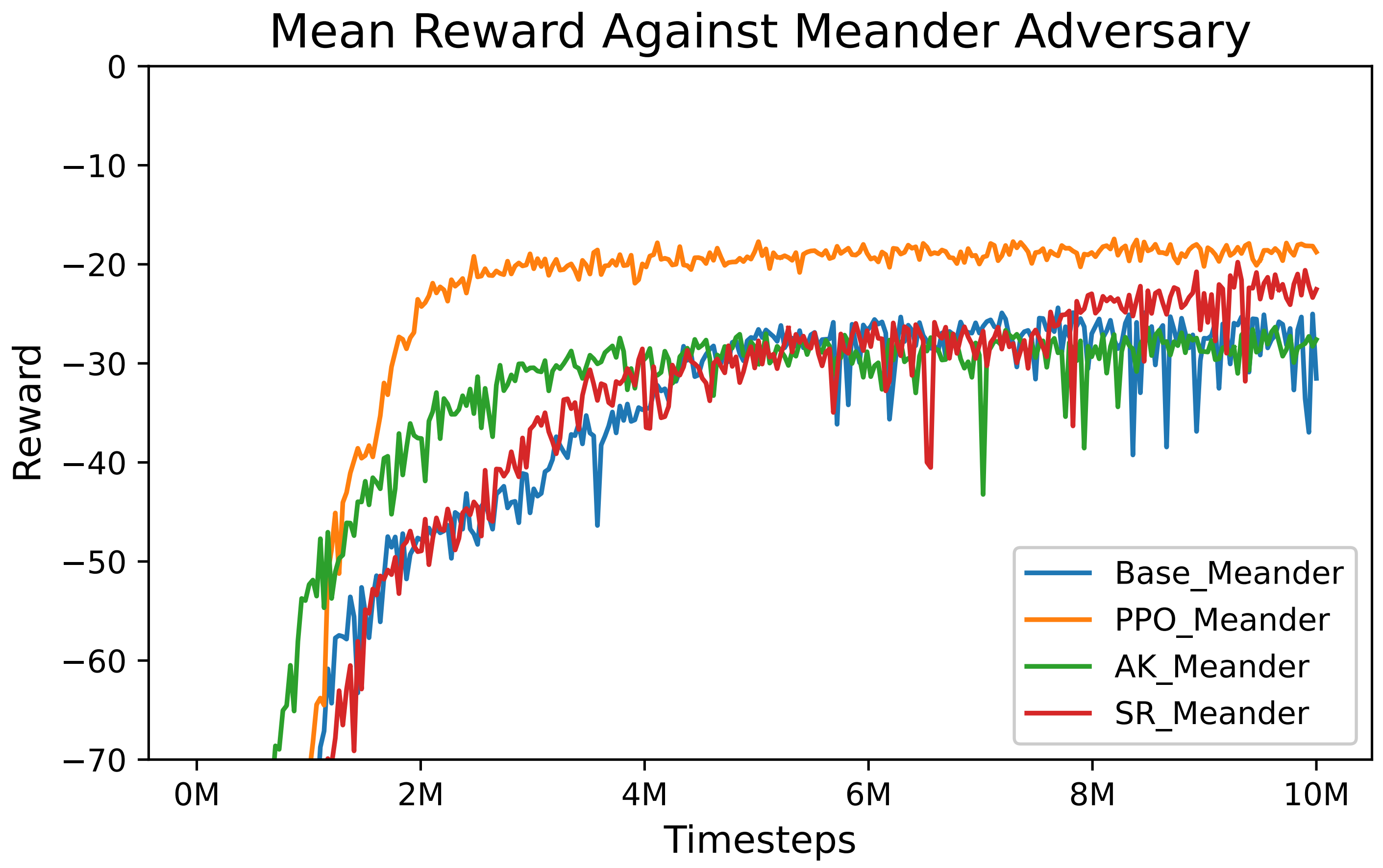



HEURISTIC BASED CONTROLLER
ONE AGENT TO RULE THEM ALL
s_t
Controller
s_t
s_t
MeanderAgent Defence
BLineAgent Defence
a_t
a_t
The scanning of two different hosts on the network within the first four timesteps indicates the presence of the MeanderAgent adversary. Otherwise, this is either the BLineAgent adversary or the User agent

EVALUATION AND EXPLAINABILITY



Controller Agent	Prediction Accuracy	
	`BLineAgent`	`RedMeander`
PPO with curiosity (4 steps)	76.8%	0.0%
PPO with curiosity (100 steps)	30.3%	42.9%
Heuristic	100.0%	100.0%
Bandit	100.0%	100.0%

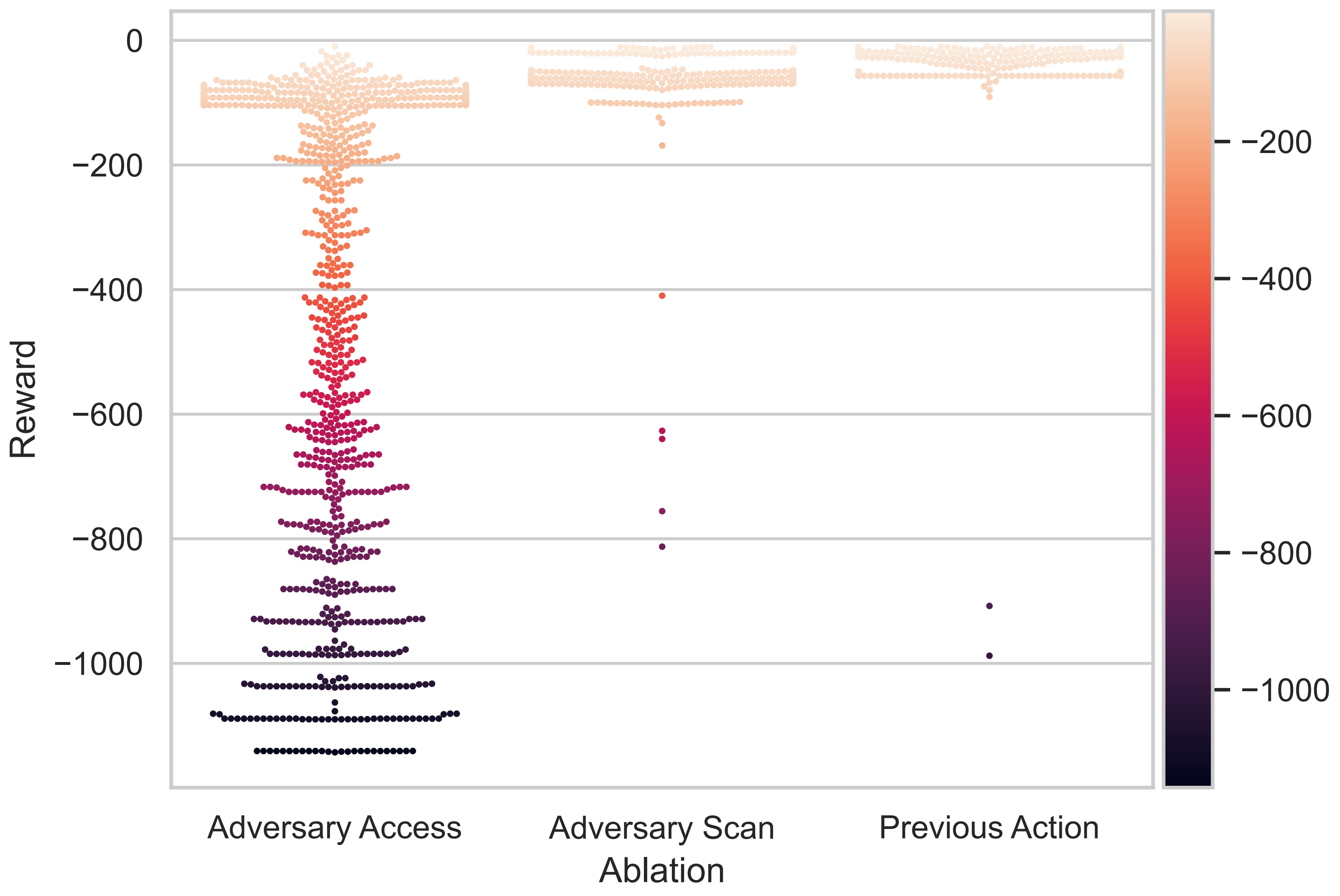


TRAIN ALL THE AGENTS

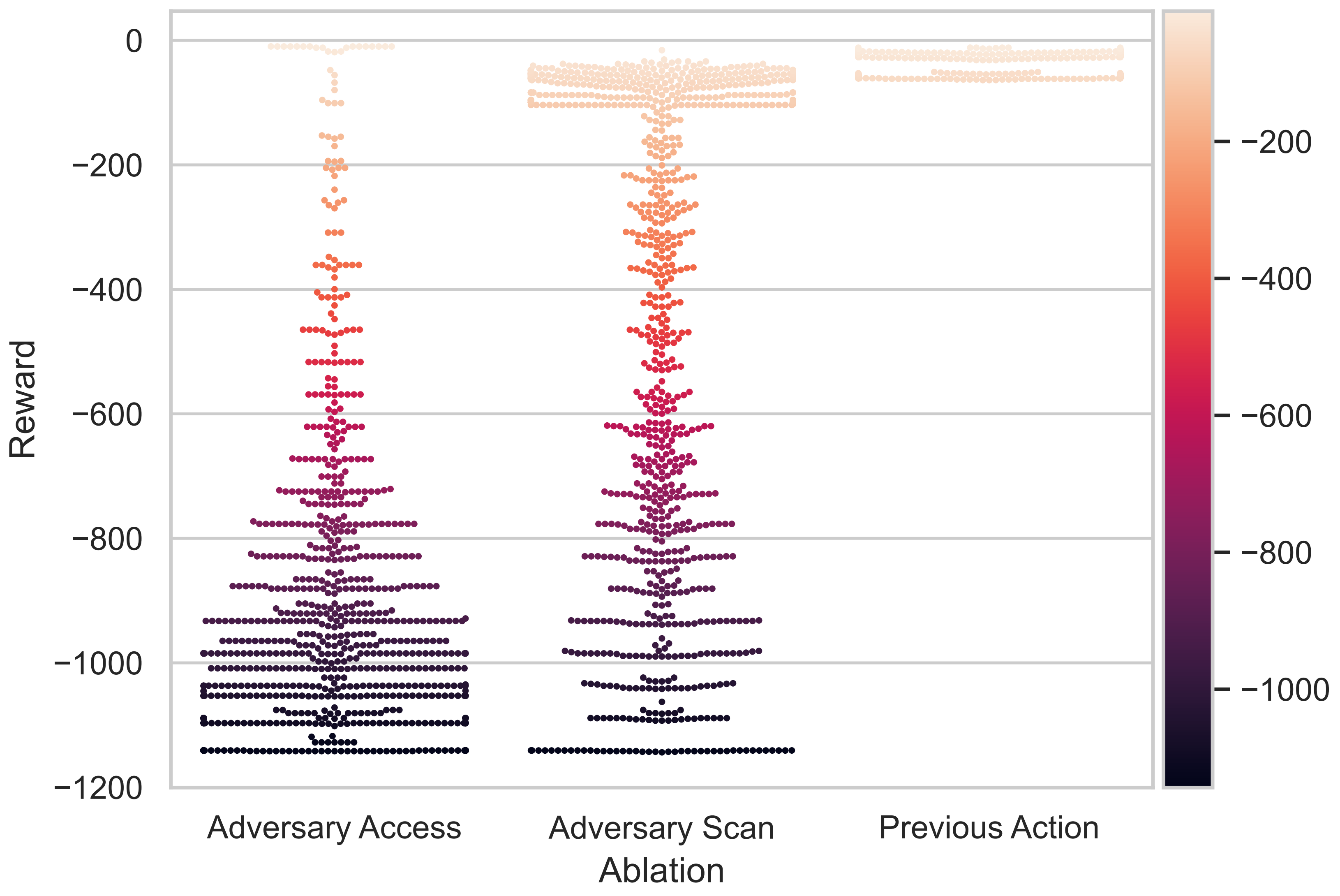
END-TO-END PERFORMANCE

Controller	Subagents	30 steps		50 steps		100 steps	
		BLineAgent	MeanderAgent	BLineAgent	MeanderAgent	BLineAgent	MeanderAgent
Bandit	PPO + AK	**-3.56±2.03**	**-6.80±1.40**	-6.79±13.00	-10.10±2.30	-13.54±15.95	**-17.30±4.27**
	PPO + SR	-3.62±2.04	-6.88±1.42	-6.26±3.18	-10.06±2.15	**-13.00±6.28**	-17.56±4.51
Heuristic	PPO + AK	-3.56±2.04	**-6.80±1.40**	-6.79±13.00	**-9.96±2.33**	-14.07±27.73	-17.57±4.82
	PPO + SR	-3.71±2.09	-6.86±1.48	**-6.17±3.40**	-10.04±2.32	-13.06±6.14	-17.32±4.35
Baseline (PPO Controller)	PPO + AK	-4.35±2.42	-7.19±1.69	-7.45±4.27	-10.84±2.62	-14.97±8.09	-19.33±5.38
	PPO + SR	-3.95±2.18	-7.36±1.74	-6.38±3.20	-11.33±3.00	-13.14±6.45	-21.21±6.10
Baseline (PPO Controller)	Baseline (PPO subagents)	4.82±4.22	-8.78±3.21	-9.20±16.01	-19.00±20.86	-18.49±34.40	-47.60±88.16

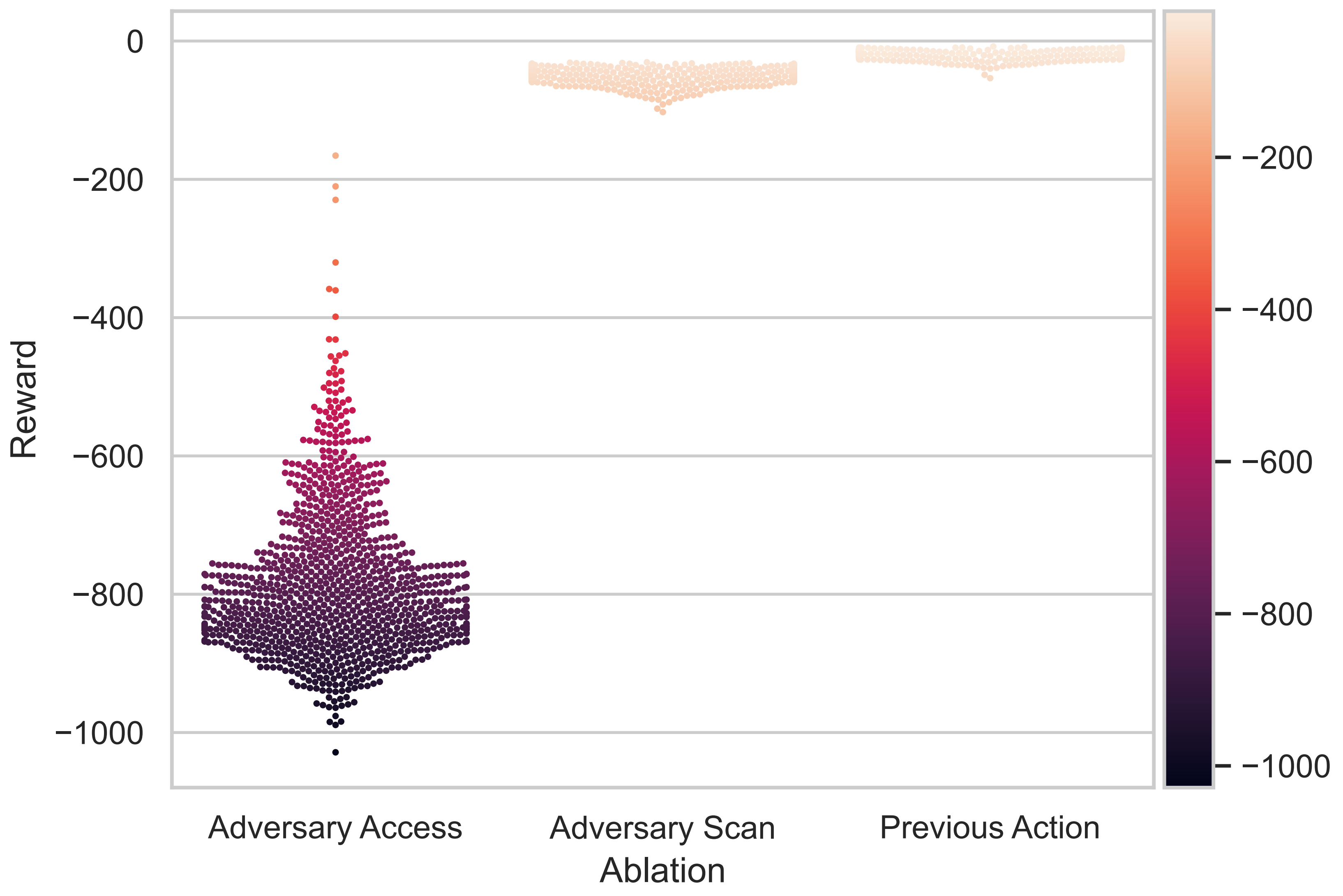
Table 3: Performance of all subagents-controller combinations, evaluated over 1,000 episodes with a length of 30, 50 and 100 steps each.



AK BLineAgent Defence



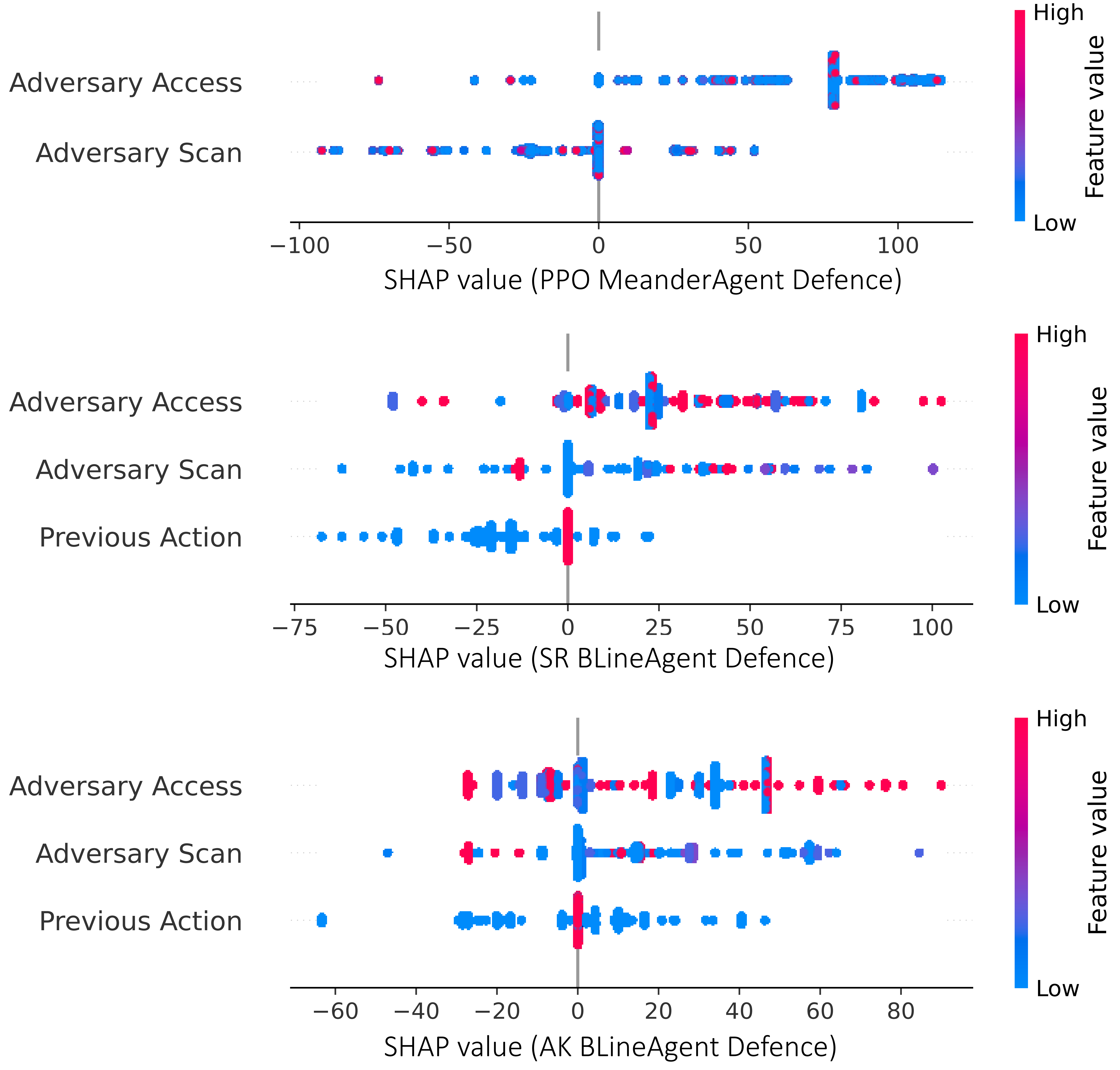
SR BLineAgent Defence



PPO MeanderAgent Defence

ABLATION STUDY

FEATURE IMPORTANCE STUDY



Thanks to our funders:

dstl

The Science Inside

GCHQ

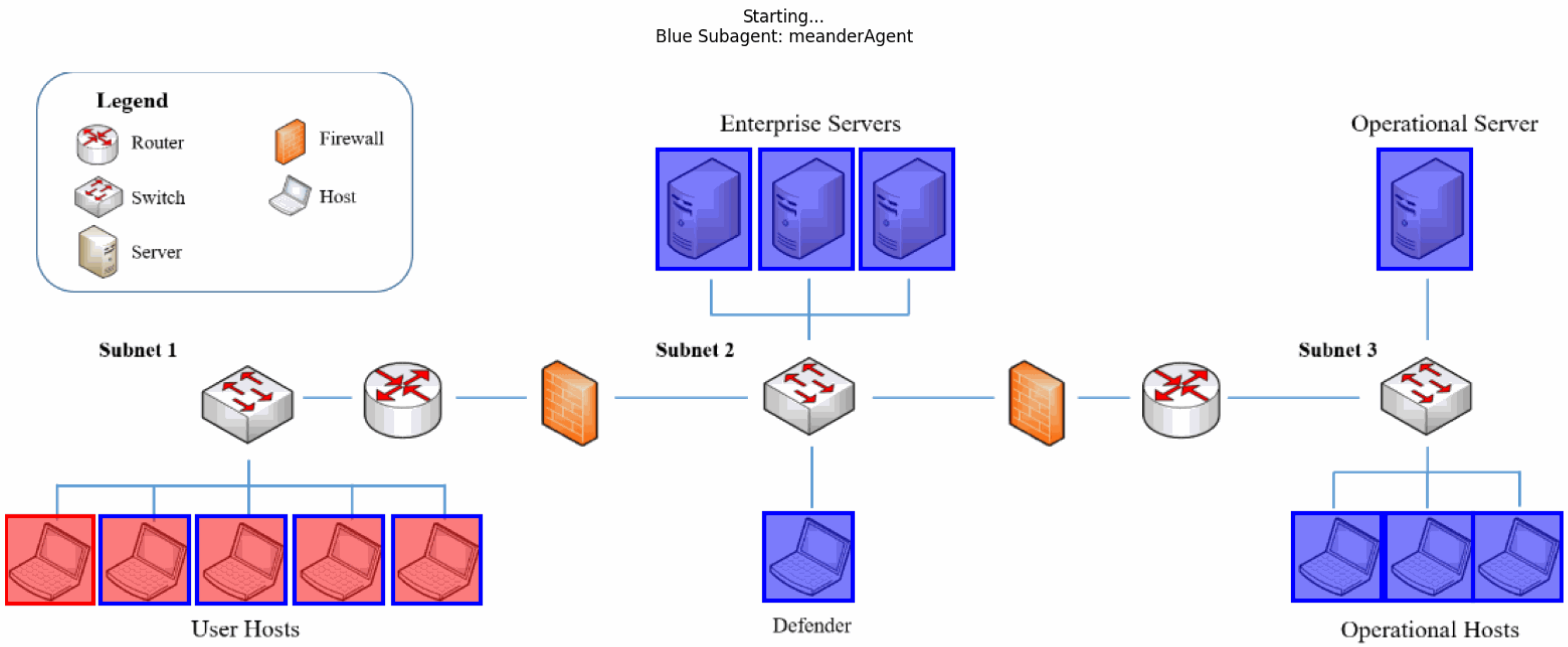
and thank you for listening!

Get in touch:

Personal email: m.foley20@imperial.ac.uk

Group Email: mindrake@turing.ac.uk

DEFEND THAT NETWORK!



Algorithm 1 Bandit Controller Learning Algorithm.

```
Initialise the known states, s_n
Initialise set of bandits, B
Initialise for a = 1 to k:
    bandit_0.Q(a) ← 0 // Initialise Q values and action counter for the first bandit
    bandit_0.N(a) ← 0
Predict(s):
    if s ∉ s_n:
        s_n ← s
        Initialise bandit_s
        B ← bandit_s
```

$$A \leftarrow \begin{cases} argmax_a(bandit_s.Q(a)) & \text{with probability } 1-\epsilon \\ random\ action & \text{with probability } \epsilon \end{cases}$$

$R \leftarrow prediction\_result(A)$

$bandit_s.N(A) \leftarrow bandit_s.N(A) + 1$

$bandit_s.Q(A) \leftarrow bandit_s.Q(A) + \frac{1}{N(A)}\left[R - bandit_s.Q(A)\right]$

Full Bline Behaviour