Online Robust Policy Learning in the Presence of Unknown Adversaries

Aaron Havens†, Zhanhong Jiang* and Soumik Sarkar
Iowa State University, (Now at † UIUC, * Johnson Controls)

Neural Information Processing Systems Foundation

IOWA STATE UNIVERSITY

Motivation

- The use of Deep reinforcement learning (DRL) for cyber-physical systems has raised concerns around safety and robustness of autonomous agents.
- It is computationally feasible for a bad actor to fool a DRL policy into behaving sub optimally. Even very small perturbations can result in significant performance loss.

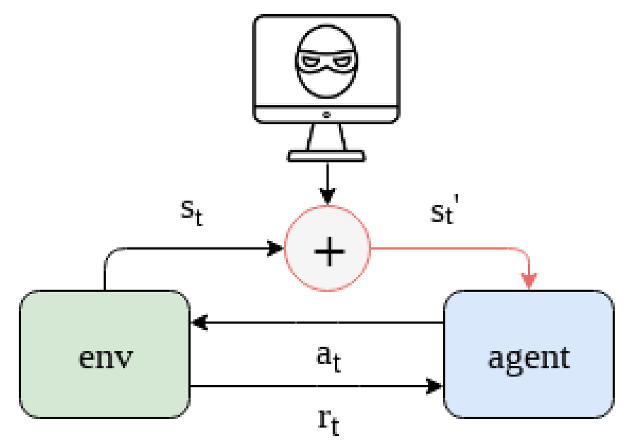

Figure 1: Model for adversary interaction with reinforcement learning system. Adversary can inject arbitrary bias into state observations.

Related Work

A table of related works

Table 1: Comparisons with different robust adversarial RL methods

Method	Online	Adaptive	Attack-model agnostic	Mitigation
VFAS (Lin et al 2017)	✓	✗	✓	✓
ARDL (Madry et al 2017)	✗	✗	✗	✓
MLAH [This paper]	✓	✓	✓	✓

Online: no offline training/retraining required, Adaptive: can adapt to a change in attack strategy, Attack-model agnostic: assumes no specific attack model, Mitigation : is the impact of the attack actively mitigated?

- Previous attempts in mitigating adversarial attacks have been successful against only assumed specific attacker models
- Robust training strategies are typically off-line (e.g., using augmented datasets) and may fail to adapt to different attacker strategies in an online fashion.

Formulation

- We want to maximize this discounted reward sum by optimizing a policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$, $\mathcal{R}(\pi) = \mathbb{E}_{s_0,a_0,\ldots}\left[\sum_{t=0}^{T} \gamma^t r(s_t)\right]$
- A finite set of MDPs $\mathcal{M}$:$\{m_0, m_{,1}, \cdots, m_n\}$, where an MDP m_i, $i \in \{0,1,\cdots,n\}$ is sampled for learning at time t and corresponding sub-policies $\Pi : \{\pi_0, \pi_1, \cdots, \pi_n\}$ which may individually be used at any instant.
- The joint hierarchical objective for $\mathcal{M}$ composed of sub-policies, which can represent adversarial and nominal conditions.

$$\mathcal{R}(\Pi) = \mathbb{E}_{s_0,\pi_0,m_0\ldots}\left[\sum_{t=0}^{T} \gamma^t r(s_t) |\, m_i, \pi_i\right]$$

Attack models

Stochastic l_∞-bounded Attacks:

$s_{i,\,adversary} = s_i + \mathcal{U}(a,b)$

$\forall s_i \in \mathbf{s} \quad \max_i |s_i - s_{i,\,adversary}| \leq \epsilon_{attack}$

Value-network Gradient Attack

$\mathbf{s}_{k+1} = \mathbf{s}_k - \alpha \nabla_s V(\mathbf{s}_k)$

Algorithm

Algorithm 1: MLAH

Input: π_{nom}, π_{adv} sub-policies parameterized by θ_{nom}, θ_{adv}; Master policy π_{master} with parameter vector ϕ.
Initialize θ_{nom}, θ_{adv}, ϕ
for *pre-training iterations [optional]* **do**
  Train π_{nom} and θ_{nom} on only nominal experiences.
end
for *learning life-time* **do**
  for *Time steps t to $t+T$* **do**
    Compute $\mathbf{A}_t$ over sub-policies (see eq. 1)
    π_{master} selects to switch or stay with sub-policy based on $\mathbf{A}_t$ observations to take action
  end
  Estimate all A_{GAE} for π_{nom}, π_{adv} over T
  Estimate all A_{GAE} for π_{master} over T with respect to $\mathbf{A}_t$ observations
  Optimize θ_{nom} based on experiences collected from π_{nom}
  Optimize θ_{adv} based on experiences collected from π_{adv}
  Optimize ϕ based on all experiences with respect to $\mathbf{A}_t$ observations
end

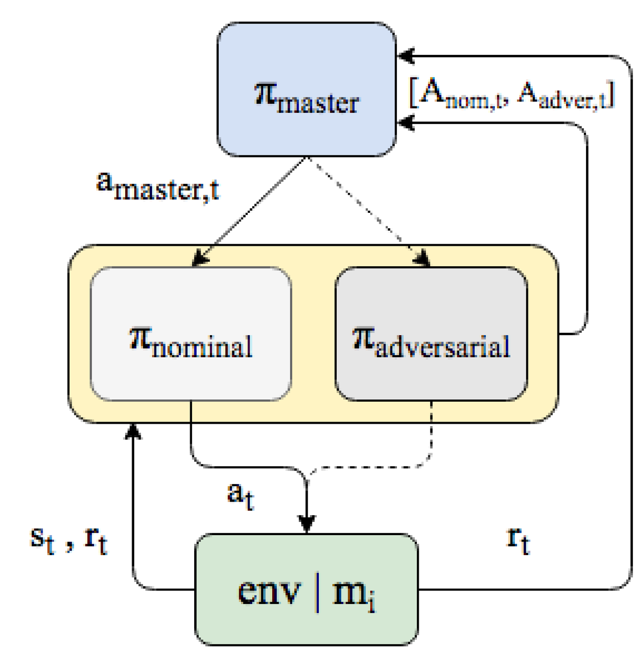

Figure 2: The MLAH framework. A hierarchical meta reinforcement learning task takes place at the same time the adversary mitigation policy is learned.

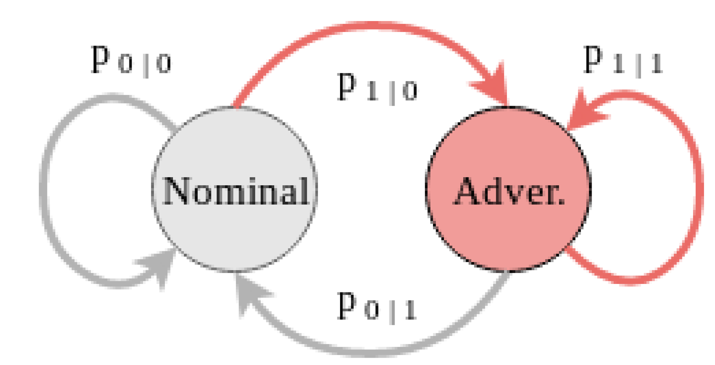

Figure 3: The assumed attack transition mechanism (only for analysis purpose) for nominal to adversary state transitions: $p_{1|0} = m$ is the probability that a nominal state transits to an adversarial state and $p_{0|1} = n$ is the probability that an adversarial state transits to a nominal state.

$$\mathbf{A}_t = \left[A_{GAE,t-h}|\pi_{nom}, A_{GAE,t-h}|\pi_{adv}\right] \in \mathbb{R}^2, \quad A_{GAE,t} = \sum_{i=0}^{T} (\gamma\lambda)^i \zeta_{t+i} \quad (1)$$

$$a_{master,t} = \pi_{*,t} = \underset{a}{\operatorname{argmax}}\, \mathbb{E}_{s_t,\pi_i,m_i\ldots}\left[\sum_{t=0}^{T} \gamma^t r(s_t, a) |\, m_i\right] \in \{0: \text{stay}, 1: \text{switch}\} \quad (2)$$

Expected Return of Conditioned Policies

The unconditioned scheme refers that of a single DRL agent with one policy. The expected discounted reward under adversarial attacks can be expressed as (m and n are Markov switching probabilities):

$$\mathbb{E}_{unc,s\sim\mathcal{S}} V(s) = V_0 p_0 + V_1 p_1 = V_0 \frac{n}{1-m+n} + V_1 \frac{1-m}{1-m+n} \quad (3)$$

The conditioned expected discounted reward of two sub-policies (one given the nominal state and other given the adversarial state) based on the proposed MLAH framework:

$$\mathbb{E}_{con,s\sim\mathcal{S}|0} V(s) = V_0 p_{0|0} + V_1 p_{1|0} = V_0 m + V_1 (1-m) \quad (4)$$

Analysis of return Lower-bound:

If $\Delta\hat{\delta} < C\Delta V$, where $C \geq \frac{(m-n)(1-m)(4\gamma\alpha^2+1-\gamma)}{(1-m+n)(1-\gamma)}$ and $\Delta V = V_0 - V_1$, then the conditioned policy has a higher lower bound of expected discounted reward compared to that of the unconditioned policy.

Experimental Results

- We sample returns of perfect switching under the MLAH framework during several switching frequencies (m and n) to evaluate our previous analysis.

Table 2: Performance evaluation of Oracle-MLAH

	Normalized avg. training return		Normalized avg. evaluation return	
m/n	Vanilla	Oracle-MLAH	Vanilla	Oracle-MLAH
1.0/−	0.96 ± 0.03	0.96 ± 0.03	1.0	1.0
0.995/0.005	$0.238 \pm .082$	0.553 ± 0.242	0.471 ± 0.051	0.99 ± 0.001
0.95/0.05	$0.612 \pm .08$	0.677 ± 0.149	0.644 ± 0.078	0.99 ± 0.001
0.8/0.2	0.613 ± 0.043	0.728 ± 0.063	0.539 ± 0.023	0.994 ± 0.165
0.5/0.5	0.749 ± 0.093	0.764 ± 0.078	0.787 ± 0.010	0.948 ± 0.086

Comparison of the returns of Vanilla PPO and Oracle-MLAH under attacks over 40 policy optimization iterations in *MountainCarContinuous-v0* with 1σ uncertainty bounds. The training return uses a stochastic policy for exploration and evaluation acts deterministically

Meta-Learned Switching and Adversary Mitigation

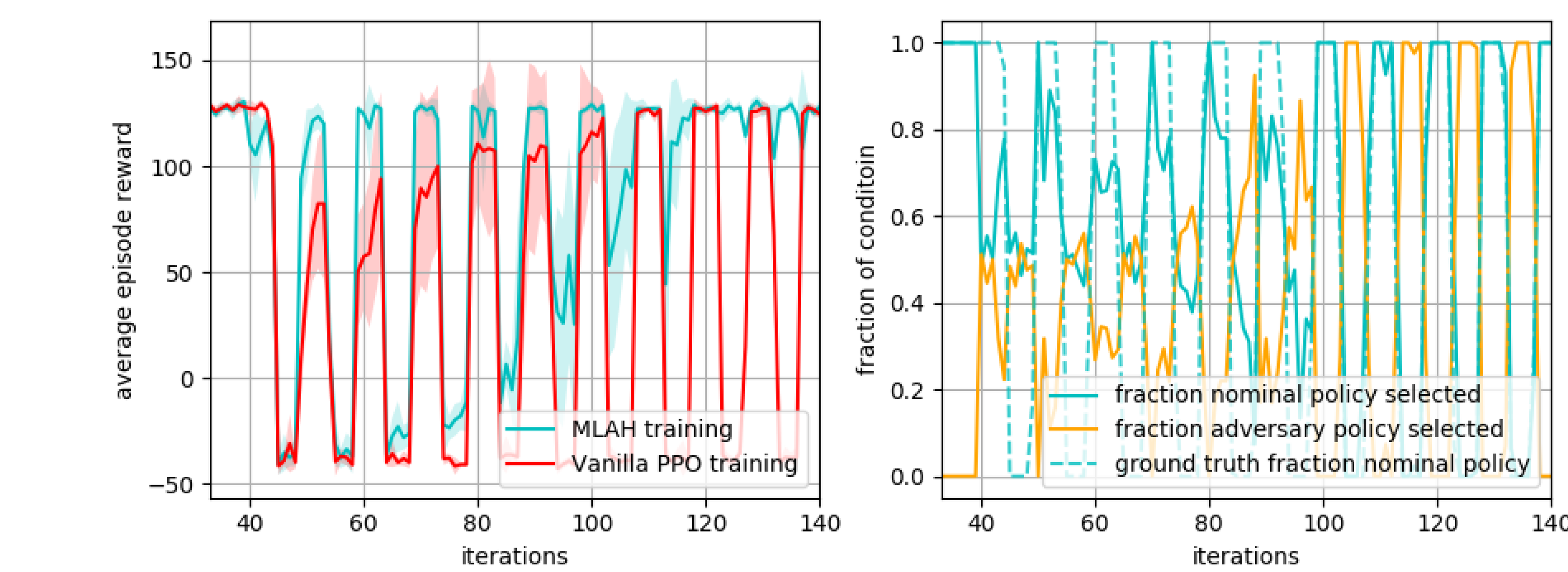

Figure 4: Shown is MLAH simultaneously learning to switch policies and a mitigation strategy on a small 11×11 grid world. The adversary simply gives the agent a deterministic mirrored column observation about the center of the grid, making it so that the optimal policy is different for every state given there is an attack. The attacks are applied intermittently on intervals of 5000 actions, showing $1-\sigma$ variance.

- The master agent optimizes to use one policy for the nominal and the other for the adversarial conditions to optimize its reward.
- This attack is interesting because it is completely deterministic and impossible for a single policy to be optimal in both nominal and adversarial conditions.
- Once attacks are introduced, it takes MLAH $\approx$ 50000 actions to both solve the meta task of switching policies and learn the adversary mitigation policy from a random initial policy.

Conclusion and Future Work

Conclusions:

- MLAH framework proven to be useful for handling adversarial attacks in an online manner specifically in the context of RL
- The return lower bound is improved when compared to a single policy agent
- MLAH presenting a way of examining adversarial attacks in the temporal domain

Future Work:

- MLAH may generalize to multi-task environments.
- There may be a need to extend framework to more than two policies and improve meta-learning stability via entropy penalties.

Find all code and future developments here: `https://github.com/AaronHavens/safe_rl`.

Self-aware Complex Systems Laboratory

References: [1] Havens, A., Jiang, Z.,& Sarkar, S.. Online Robust Policy Learning in the Presence of Unknown Adversaries. arXiv preprint arXiv:1807.06064 (2018).

Acknowledgements: This work has been supported in part by the U.S. AFOSR under the YIP grant FA9550-17-1- 0220.