Adversarial Policy Training against Deep Reinforcement Learning

Xian Wu*, Wenbo Guo*, Hua Wei*, **Xinyu Xing**

The Pennsylvania State University

*Equal contribution

Deep Reinforcement Learning

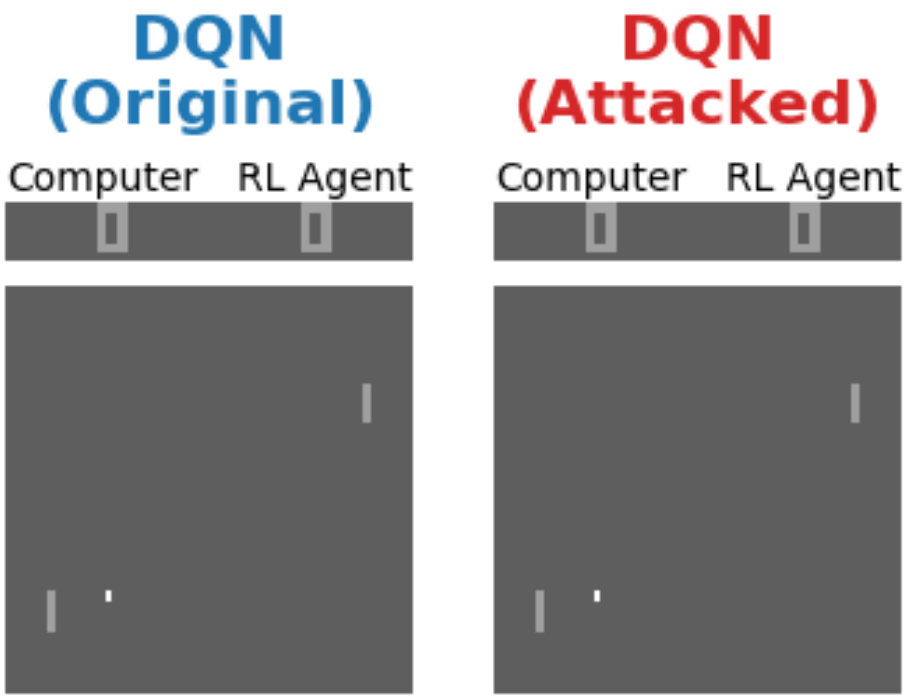
- Deep learning (DL) has dominated many fields
  - Computer vision [Krizhevsky, et al, 2012], natural language process [Socher, et al, 2011].
  - Malware detection [Wang, et al, 2017], intrusion detection [Javaid, et al, 2016].
- Integrating DL into reinforcement learning – DRL
  - Extraordinary performance on many decision-making tasks.
    - Robotic control [Kober, et al, 2009].
    - Autonomous vehicles [O'Kelly, et al, 2019].
    - Finance and business management [Cai, et al, 2018].

DRL in Games

- Board games
  - Go: DeepMind – AlphaGo, Facebook – OpenGo
  - Poker games: Texas hold'em
  - Real-time strategy games (Dota2, StarCraft)
  - Soccer game

Existing Attack on DRL

- Similar to adversarial attacks against deep neural networks, DRL is also vulnerable to adversarial attacks
- Adding invisible noise to the background canvas and thus fool the reinforcement learning agent
- Canvas perturbation is physically impractical



This Work

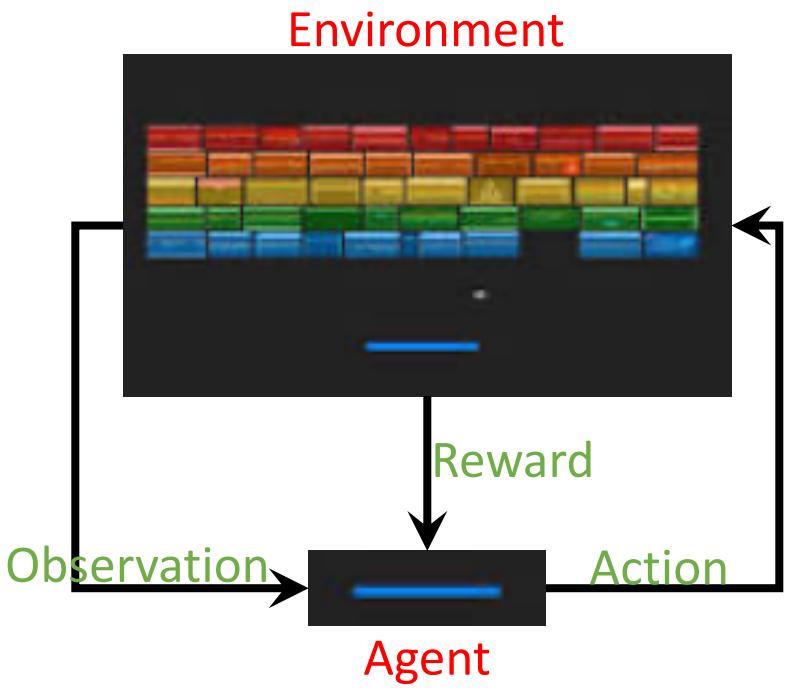
- Introducing a new physically realistic attack against reinforcement learning
- In a multi-agent setting, agents interact with each other; the physically realistic attack means
  - No change to the game's canvas
  - Constructing an adversarial agent in the multi-agent environment
  - Using that adversarial agent to trigger the weakness of the victim agent
  - Tricking the victim agent to act weirdly

Agenda

- Background of Reinforcement Learning
- Existing physically realistic attack
- Our proposed method
- Evaluation
- Conclusion

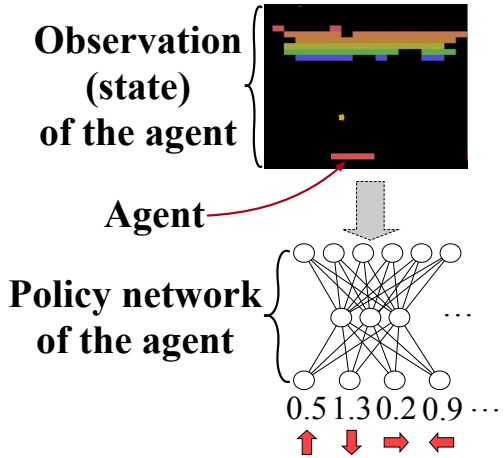
RL System Background

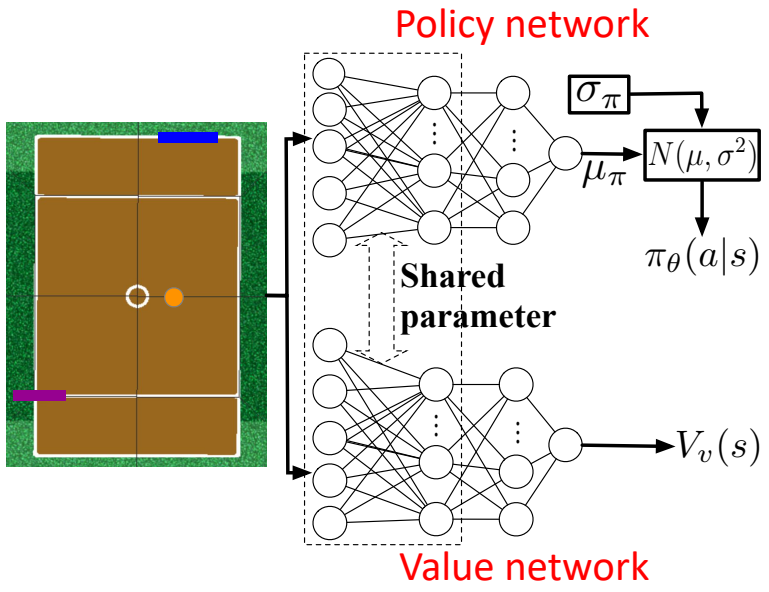
- Agent
  - Observes environment.
  - Executes an action.
- Environment
  - Receive this action.
  - Transit to the next state based on the transition dynamics.
  - Emit an reward.
  - Emit the next observation.
- Goal of RL algorithm
  - Learn an optimal policy, following which the agent could receive a maximum amount of rewards over time



Learning an Agent

- In DRL, an agent's policy is usually modeled as an DNN.
  - Policy network.
  - Taking as input the observation, and output the corresponding action.
  - Learning a policy is to solving the parameters of this neural network.
- Policy gradient methods - solving the network parameters.
  - Using another network to approximate the value-function.
  - In each iteration:
    - Updating the value network by minimizing the approximation errors.
    - Updating the policy network by maximizing the value function.



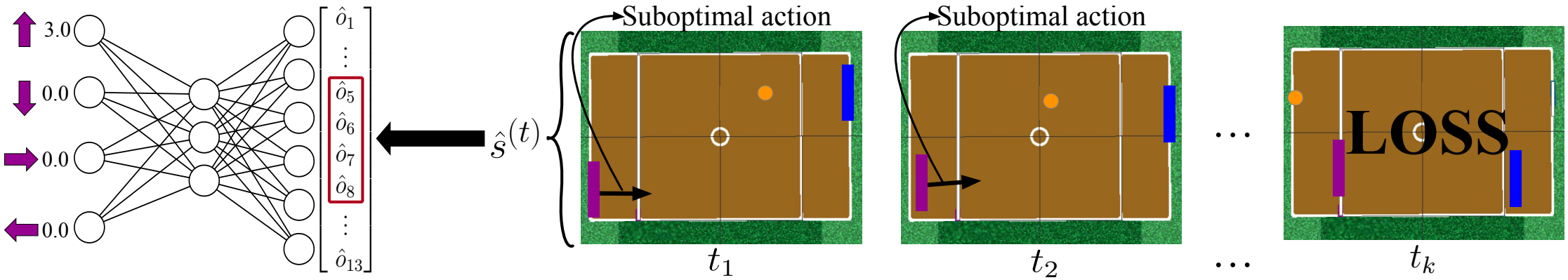


Existing Physically Realistic Attack

- Existing technique [Gleave, et al, 2020].
  - Treating the victim agent as a part of the environment.
  - Training an adversarial agent to collect maximum rewards (i.e., maximizing the adversarial agent's value function) from the environment which the victim is part of.
  - Cannot always demonstrate the efficacy against multi-party games because
    - The method still uses the loss function of an existing RL algorithm (PPO)
    - The loss provides less guidance when learning an adversarial agent to discover the weakness of the victim

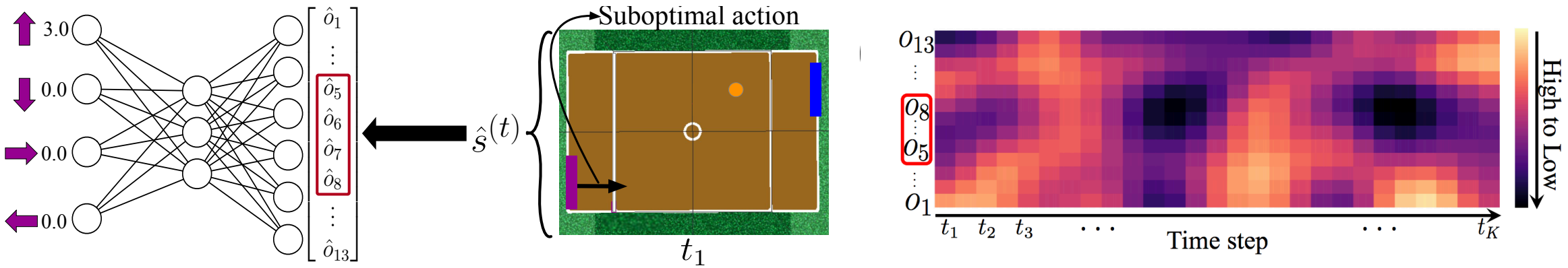
Our method

- When training adversarial agent, still maximizing the reward the agent could gather
- Letting the adversarial agent take an action that deviates the victim's next action.



Our Method (cont.)

- Training an adversarial agent to deviate victim's action could potentially make itself perform badly
- Do not vary adversary's motions too aggressively
- Varying the adversary motion only at the time when adversary's motion could influence the victim agent the most
- Using explainable AI to identify the time most critical for influencing the victim



Our Method (in Math)

- Introducing the following loss term into the PPO loss function (i.e., the loss function used in existing RL algorithm)

Indicate the importance of each time step as to the impact of the adversary's motion

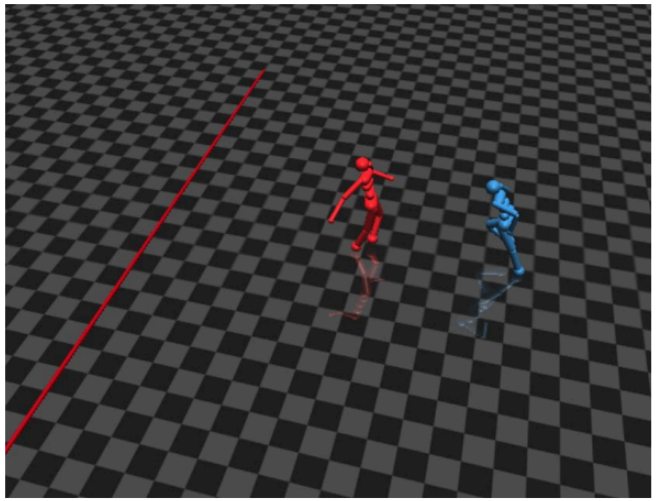
$$L_{ad} = \text{maximize}\, \lambda^{(t)} \left(\|\hat{a}_v^{(t)} - a_v^{(t)}\|_p + \|\hat{o}_v^{(t)} - o_v^{(t)}\|_p \right).$$

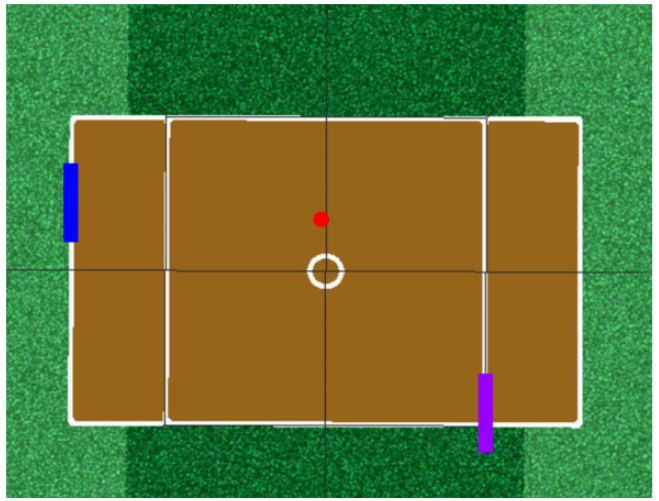
Deviate the action of victim agent as much as possible

Reduce the motion of the adversary as much as possible

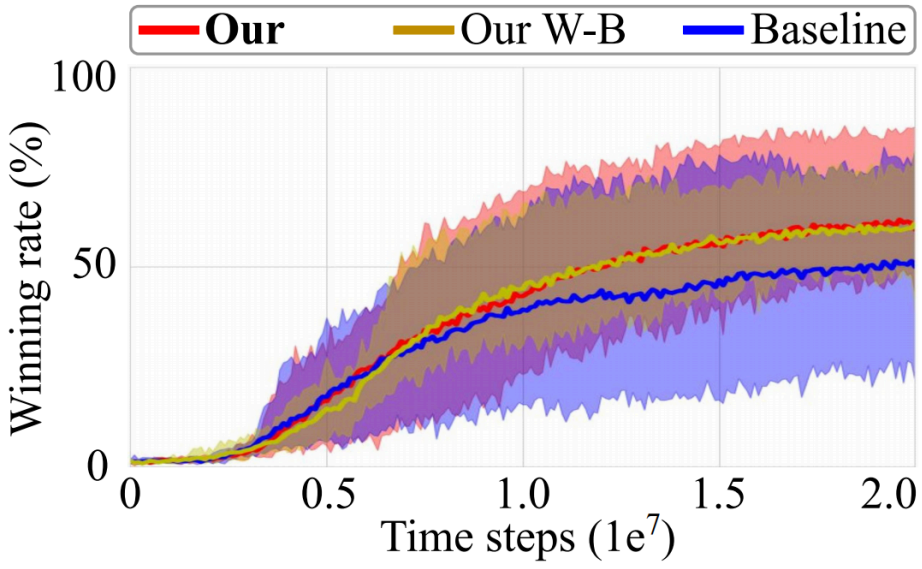
Evaluation

- Selecting two commonly used games
  - MuJoCo -- You Should not pass
  - Robotschool – Pong
- Measuring the winning rate of the adversarial agent each time its policy is updated during the training process

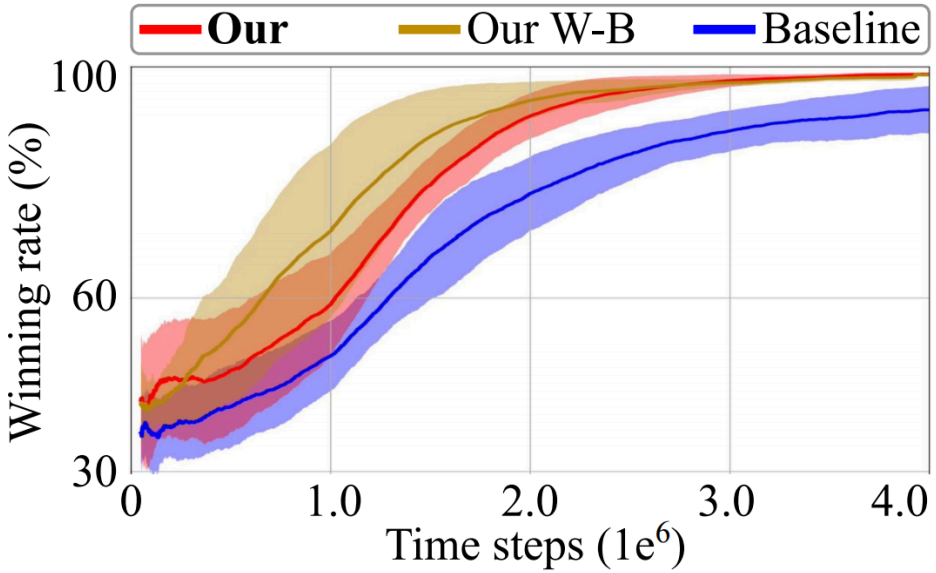




Results



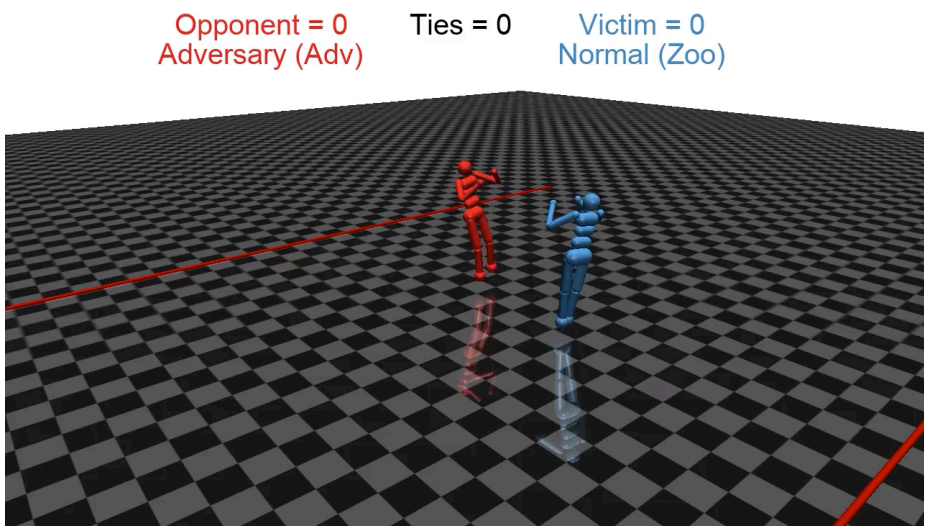
MuJoCo



Robotschool Pong

Demo & Conclusion

- Reinforcement learning agent suffers from physically realistic adversarial attack
- Explainable AI technique could escalate the exploitability of the adversarial agent
- New defense against physically realistic attack needs to be developed



Thank you
Q & A

Code repository: https://github.com/psuwuxian/rl_attack