CS 8803 Deep RL

Shared Visual Representations in Multi-Agent Reinforcement Learning

Fall 2024

Presenters: Biswajit Banerjee, Chinar Dankhara, Rishabh Goswami

Instructor: Animesh Garg

Georgia Tech

Motivation

- Reinforcement Learning (RL) agents struggle with redundant visual processing
- Traditional methods require each agent to learn input visual representations from scratch
- Biological systems reuse vision mechanisms across species and tasks. Our work is inspired by this
- We decided to apply shared visual representations in multi-agent RL scenarios feeding environment images directly

Formal Problem Statement

- Redundant computation in multi-agent environments by training visual processing from scratch
- Inconsistent visual interpretations among agents. Bespoke implementations increase computational costs and impede scalability
- There is need for a unified approach to visual processing in RL that focuses on using learned representations on visual inputs

Related Work

- Multi-Agent RL scenarios require effective communication and coordination
- Spatial intention maps enhance decentralized agent coordination (Wu et al, 2021). These allows agents to represent their goals in shared intention representation space
- Visual communication maps improve convergence and robustness (Nguyen et al, 2020)
- Our work is inspired by these approaches. Shared visual encoders promise reduced redundancy and better performance

Approach

- Implement a shared vision encoder ("Eyes") for all agents
- Convert raw RGB images into a 16-dimensional latent space using convolutional filters
- Enable different RL algorithms to utilize the same visual inputs produced from this module
- Aim is to enhance learning efficiency and coordination by ensuring consistent inputs

Algorithms (Part 1)

REINFORCE:

- Policy-based baseline method
- Directly learns action probabilities from states

Deep Q-Network (DQN):

- Value-based approach estimating Q-values for state-action pairs
- Utilizes experience replay and target networks for stability

Algorithms (Part 2)

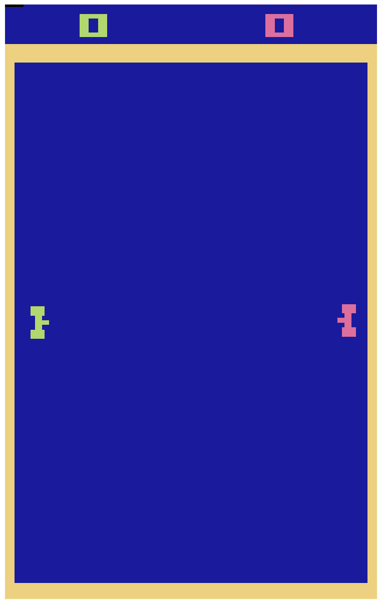
Soft Actor-Critic (SAC):

- Actor-critic method with entropy regularization
- Balances exploration and exploitation through maximum entropy framework

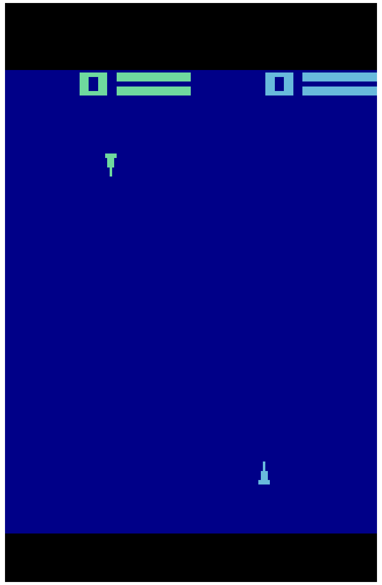
We evaluate performance across different RL paradigms using the shared visual encoder

Environments

- **Combat Tank:**
- **Description:** Two tanks compete in a 2D arena.
- **Observation Space:** RGB images (210x160x3).
- **Action Space:** 18 discrete actions (movement, shooting, etc.).
- **Reward Structure:** Winner (+1), Loser (0).



- **Space War:**
- **Description:** Two spaceships engage in combat within a 2D space.
- **Observation Space:** RGB images (210x160x3).
- **Action Space:** 18 discrete actions (movement, firing, etc.).
- **Reward Structure:** Winner (+1), Loser (-1).



Benchmark Challenges

- Since it’s adversarial training it’s hard to benchmark.
- Both of those agents are getting efficient, and reward depends on both of those agents' actions.
- We developed several techniques to make sure our agent is improving
- Also, some benchmarks to see what is going on

Training Methods

- Policy Zero (warm up) –
  - Opponent never moves, always action is zero
  - Target kill off static opponent.
- Balanced training –
  - Same policy and vision copied to both agents
  - They fight each other and back propagate.
  - Only optimize the looser.
  - Helped stabilize process and reduce mode collapse issue.
- Vision / policy transfer –
  - We eventually had to copy the vision encoder or the policy to the opponent

Offender and Victim (Behavior Engineering)

Offender:

- Rewarded highly for killing opponent (victim).
- Penalized for not killing or for taking non lethal actions.

Victim:

- Rewarded slightly for surviving.
- Penalized for not killing or for taking non lethal actions.

Offender Rewards:

Action	Reward	Reason
Successfully kills victim	+4.0	Major objective achieved.
Is unable to kill	-0.2	Penalize wasteful actions.
Takes non-lethal action	-0.1	Slight penalty to incentivize aggression.

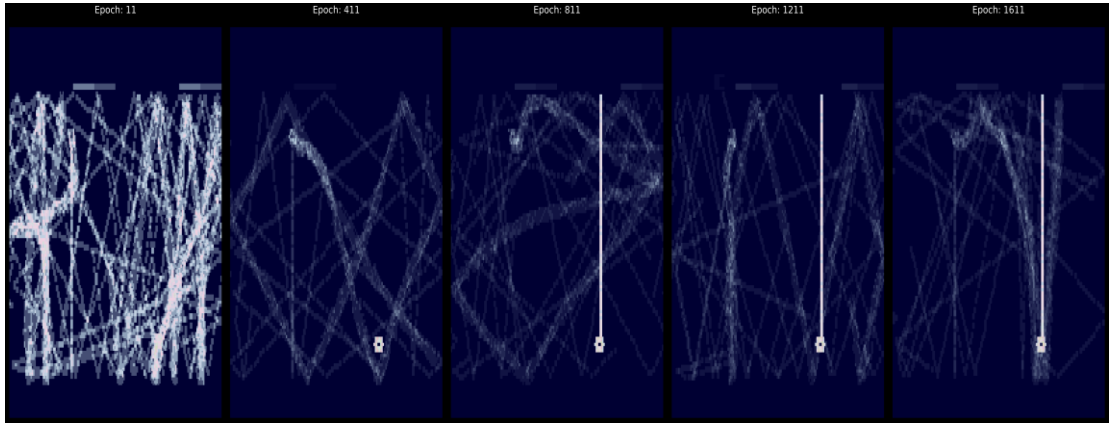
Victim Rewards:

Action	Reward	Reason
Successfully kills Offender	+4.0	Major objective achieved.
Can survive	+0.2	Reward for staying alive.
Takes lethal action	-0.3	penalty to reduce aggression.

Offender and Victim Behaviors

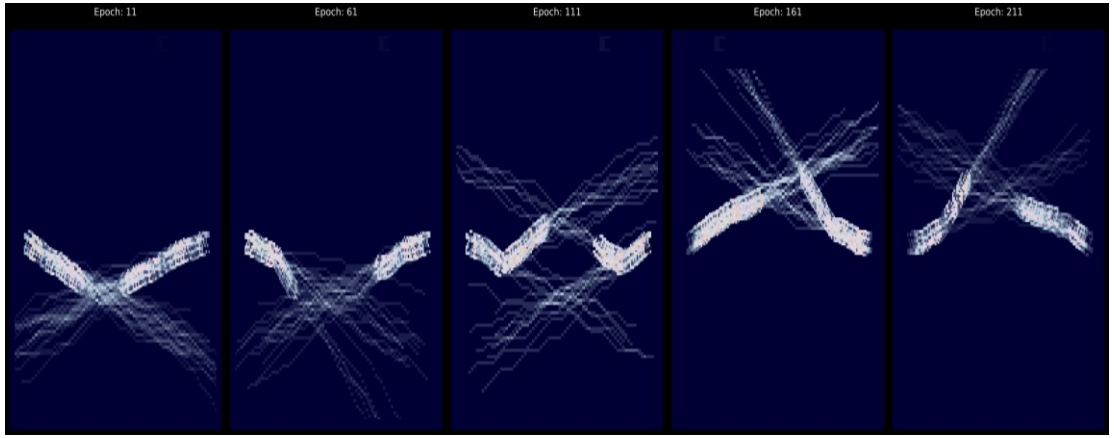
Space War:

- Offender (top left): More precise following the victim and controlled shooting
- Victim (bottom right): adapted to vertical movement as the most efficient.



Battle Tanks:

- Offender (left): Tries to intercept victim in its path.
- Victim (right): Confuses offender by shooting in one direction and running to other.



Conclusion

- Shared Vision Encoder reduces computational redundancy and ensures consistent visual inputs across agents
- SAC outperforms REINFORCE and DQN in both Combat Tank and Space War
- DQN struggles with sparse and negative rewards

Future Improvements:

- Test in environments with denser rewards
- Enhance encoder for more complex representations