THE ANDREW & ERNA VITERBI FACULTY OF ELECTRICAL ENGINEERING

CRML Control Robotics and Machine Learning Laboratory

TECHNION Israel Institute of Technology

Semantic State Representation for Reinforcement Learning

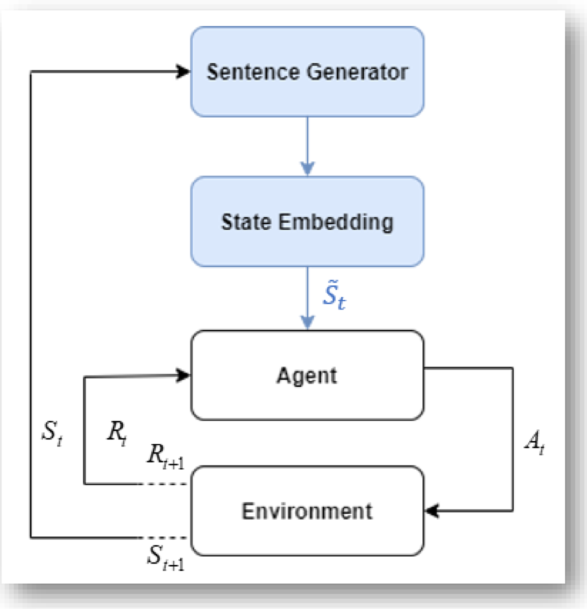
Erez Schwartz, Guy Tennenholtz, Chen Tessler, Shie Mannor

Introduction

- Recent advances in reinforcement learning have shown its potential to tackle complex real life tasks. However, as the task's dimensionality increases, reinforcement learning methods tend to struggle.
- To overcome this, we explore methods for representing the semantic information embedded in the state.
- While previous methods focused on information in its raw form (e.g., raw visual input), we propose to represent the state using natural language.
- Language can represent complex scenarios and concepts, making it a favorable candidate for representation.
- Empirical evidence, within the domain of ViZDoom, suggests that natural language based agents are more robust, converge faster and perform better than vision based agents, showing the benefit of using natural language representations for reinforcement earning.

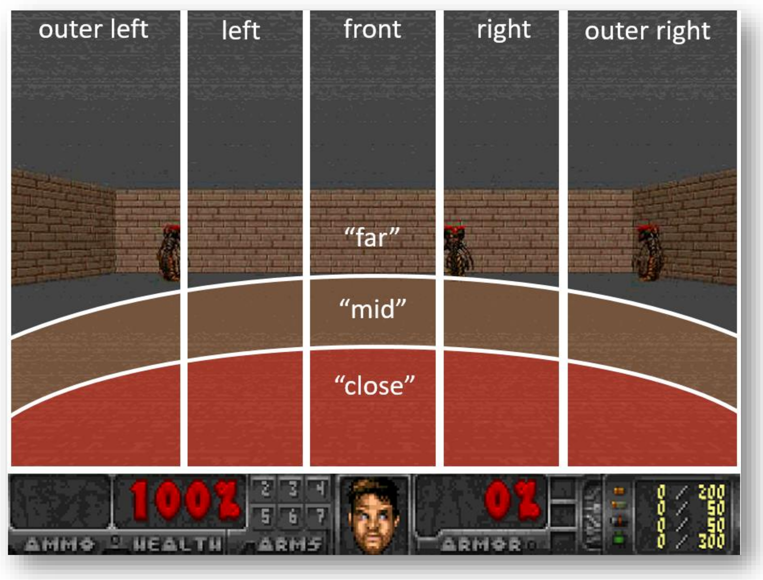
Agent Implementation

- We trained agents utilizing two common reinforcement learning methods; Deep Q-Learning and Proximal Policy Optimization
- All our agents were implemented using Convolutional neural network
- The natural language created by our simulator was processed by an agent using word embeddings



NL State Generator

- VizDoom Provides two types of state visual representation; semantic segmentation map and raw image.
- The NL state generator divides the screen vertical patches and distance regions. The generator counts how many objects of different types exist in each region and generates a matching natural language sentence
- To simulate natural language-based captioning of the state, we constructed a semantic natural language parser. To guarantee a fair comparison against visual representations, we constructed descriptions based on two criteria: limiting information and ambiguity:
- To simulate **ambiguity**, we constructed ten distinct sentence generators that described each patch of every game frame in a different manner. More specifically, at each step and for every patch, a random parser was sampled, describing the given patch, forming the final state representation. The parsers varied in word usage and sentence structure, allowing for a wide range of descriptions for every state
- To **limit information**, some of the state generators ignored certain aspects of the state (object types, color), and mentioned only objects which are visible on screen.

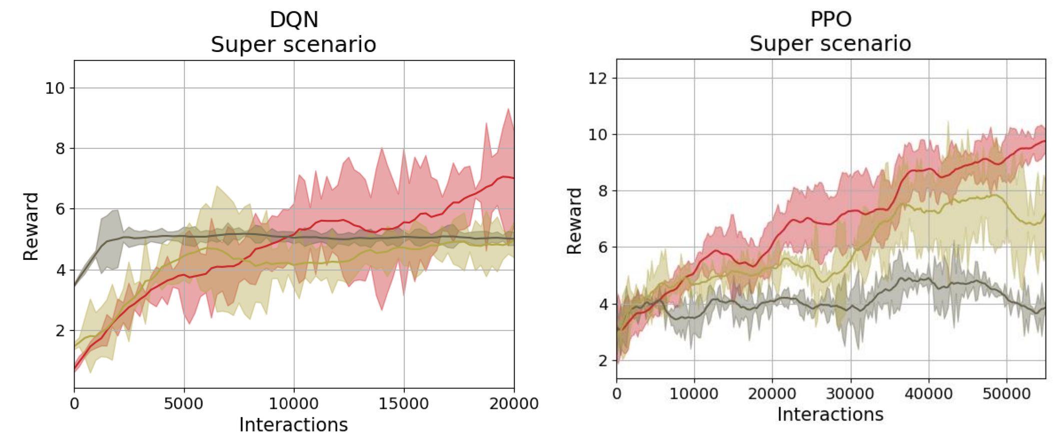


Example of state division to vertical and distance regions

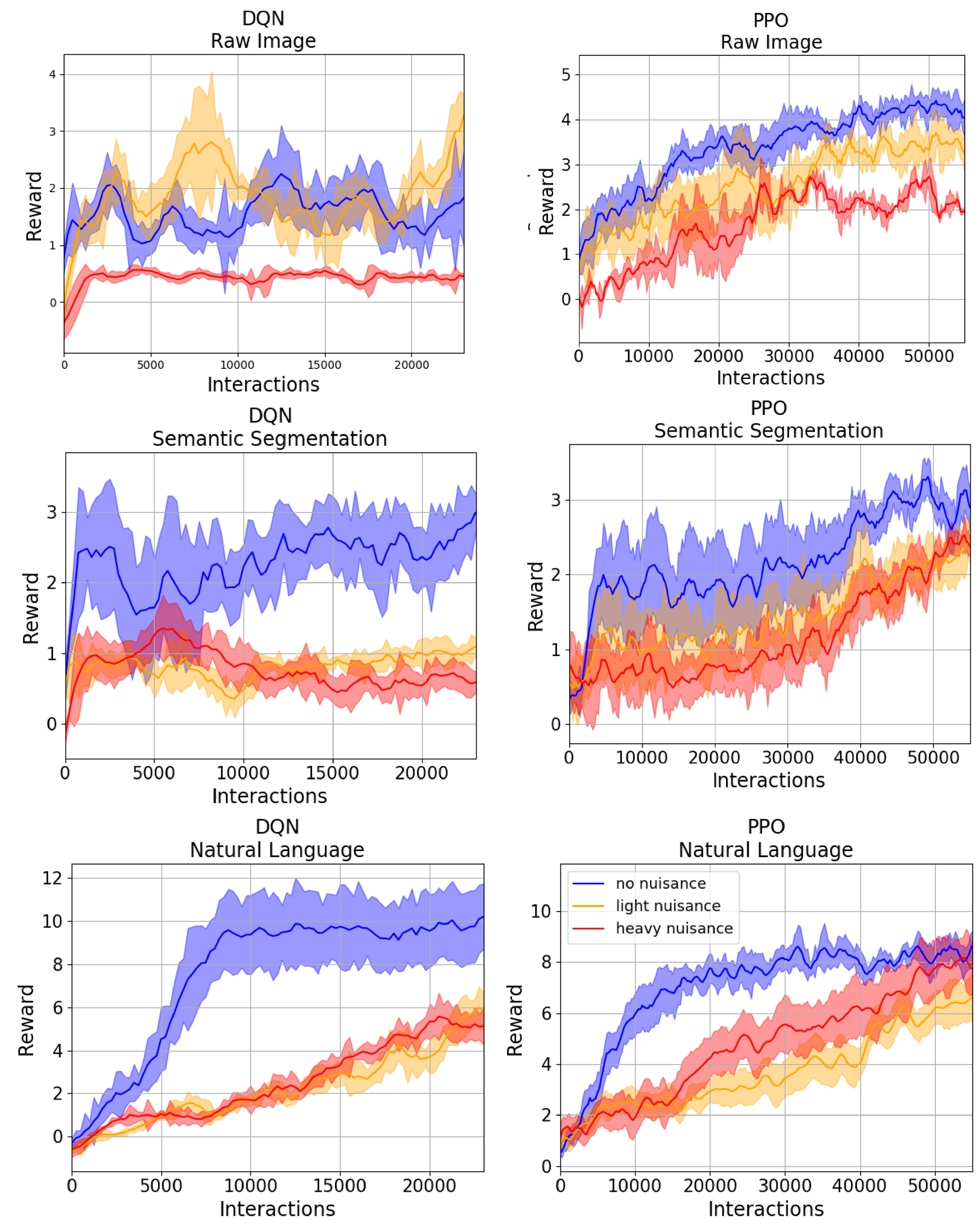
- Examples of generated natural language of the above image:
  - "You have full health. you have 50 bullets left. There is a ranged enemy on your far right"
  - "This is a first-person view, player has maximum hit points. the character has 50 rounds in the magazine. there is a fireball shooting bad guy to the player's far right"
  - "I feel great. there are 50 rounds in my gun. there is a fireball shooting creature to the player distant right"

Experiments & Results

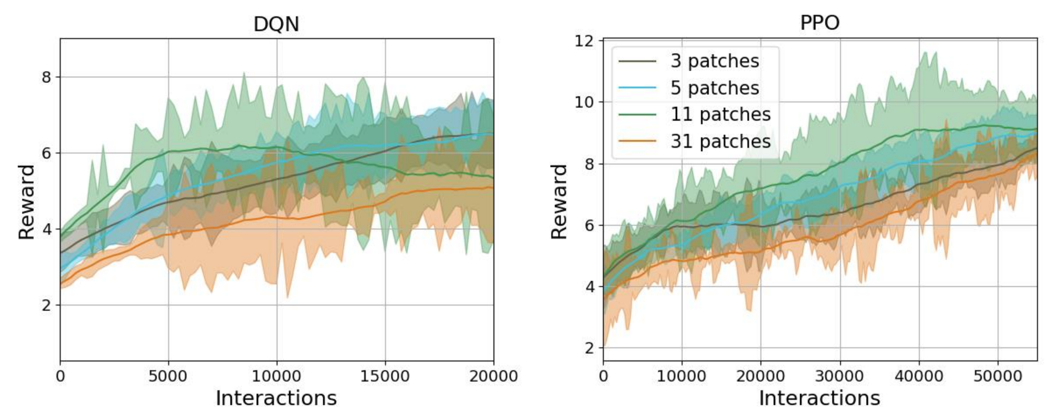
- We tested our NL and vision based agent in various aspects:
  - Performance and task completion
  - Noise/nuisance robustness
  - Multiple VizDoom game scenarios
- Our results indicate the following:
  - In this setting, NLP based agents converge faster, and outperform vision based ones.
  - PPO and NL agents are more noise robust than DQN and vision based agents.
  - NL agents are mostly invariant to state discretization granularity



Average performance per state representation



Robustness of each representation type with respect to amount of patches



Robustness of each representation type with respect to amount of nuisance

Conclusions

- NLP based agents can complete complicated tasks in environments that are naturally visual
- We were able to experiment solely with the VizDoom environment, and we encourage the development of natural language representations for other platforms
- NLP based agents are more robust converge faster and occasionally perform better than vision-based ones
- A problem might be solved more efficiently using a different representation
- **Thus, there is a noticeable benefit of using natural language representations for Reinforcement Learning.**

December 2020