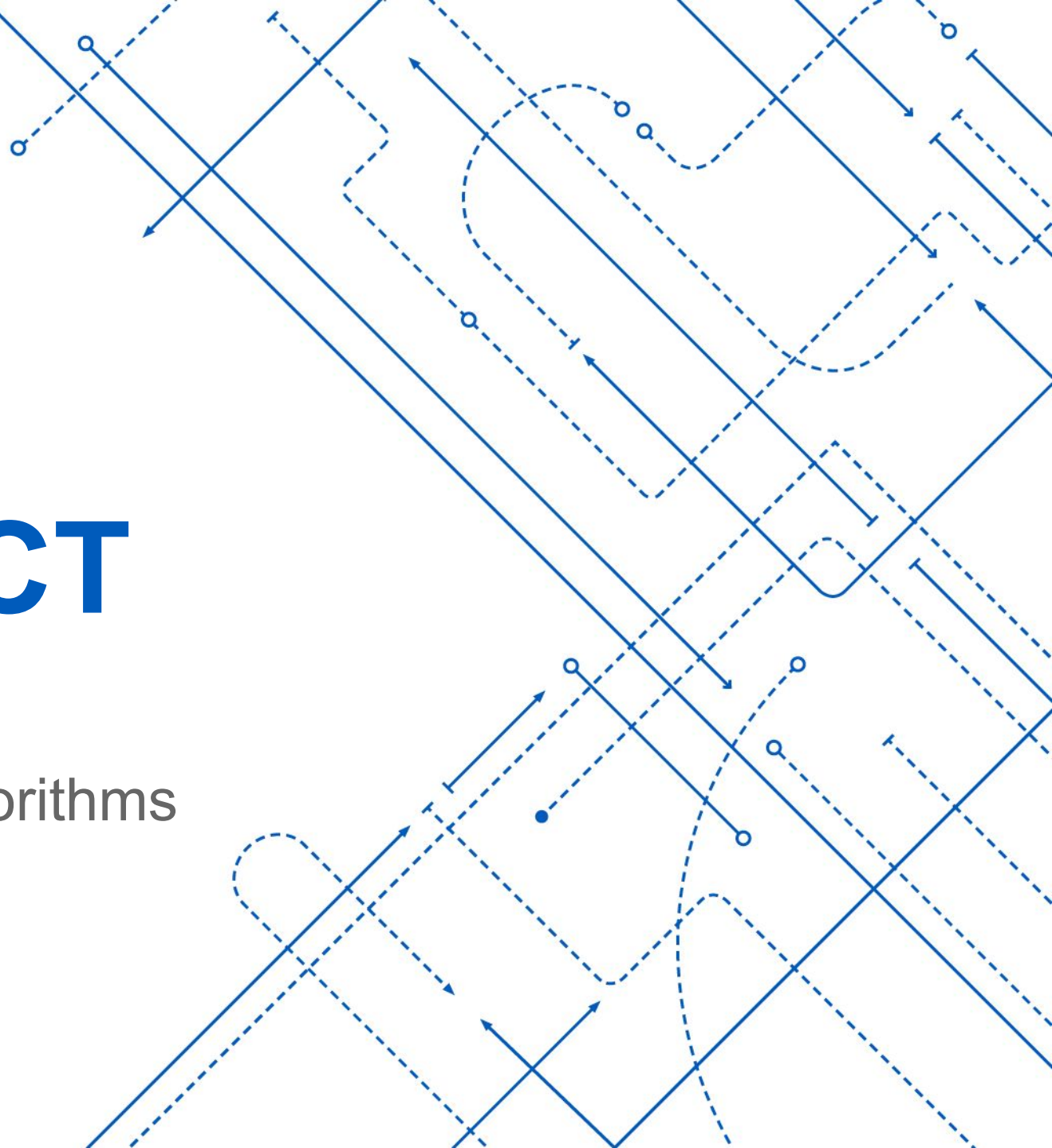


CSE 546
FINAL PROJECT

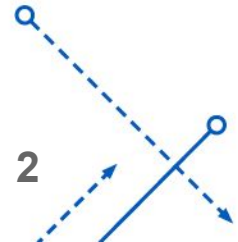
Solving Several Multi-Agent RL Environments using Different Algorithms

Sougata Saha, Zebin Li

December 8th, 2020

University at Buffalo The State University of New York

Table of Content



Background and Project Description

- Background:
  - Multi-agent RL includes more than one agent in the environment. These agents interact with the environment, obtain rewards and choose actions based on their own learning, or sharing one brain across them. The relationship of those agents can be cooperative, competitive and neither of these two.

- Project Description:
  - Play with three Gym multi-agent RL environments which are Checkers-v0, Switch2-v0 and Switch4-v0, and solve them by Vanilla DQN, Dueling DQN, A2C and A3C, then discuss the performance of independent and central architectures of algorithms on different environments.

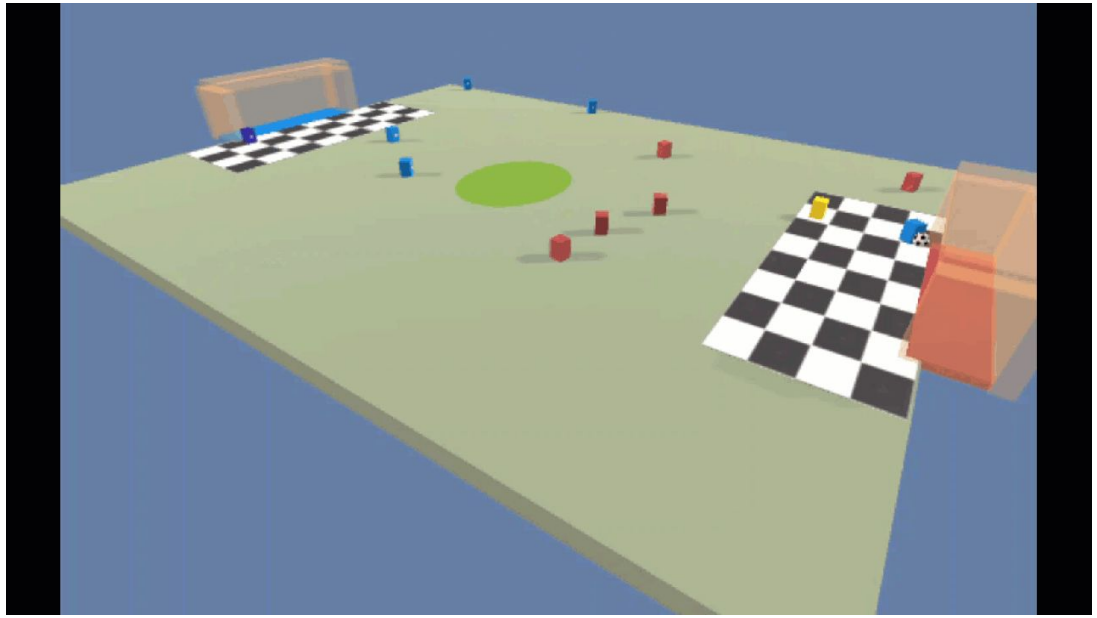


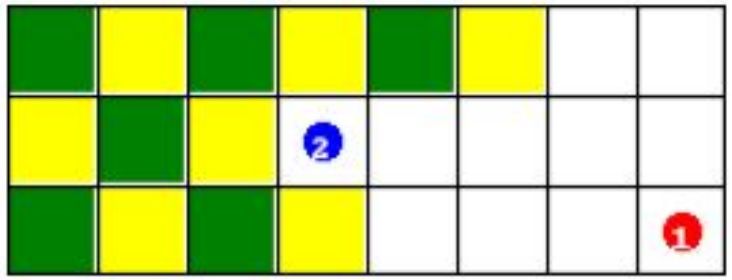
Table of Content

Introduction about the Environments

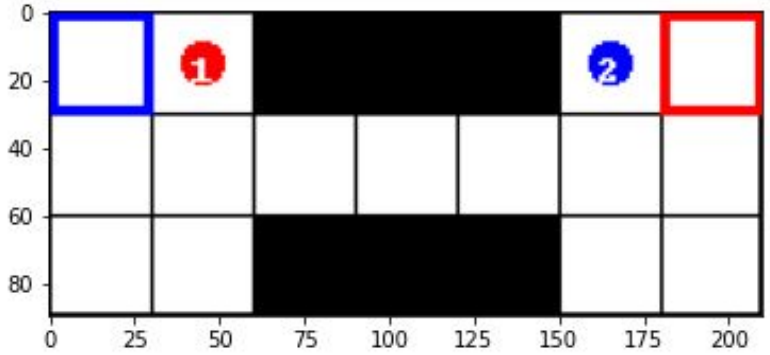
- Checkers-v0:
  - States: 3 × 8 grid world
  - Actions: up, down, left, right and wait
  - Rewards:
    - Red ball: one green square +5, one yellow square -5
    - Blue ball: one green square +1, one yellow square -1
  - Main objective: The red ball should collect as much green squares as it can.

- Switch2-v0:
  - States: 3 × 7 grid world, excluding black squares
  - Actions: up, down, left, right and wait
  - Rewards: +5 for reaching the ending position
  - Main objective: To move to the ending position

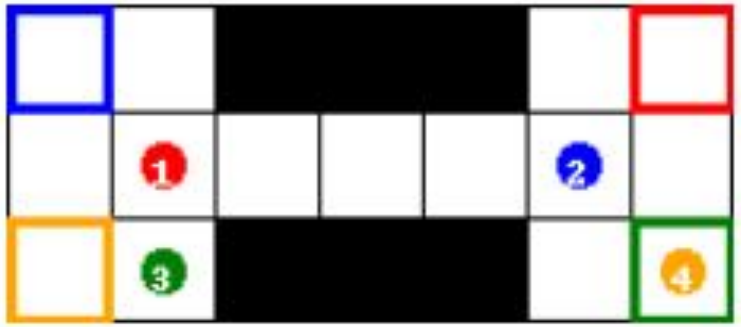
- Switch4-v0:
  - States: 3 × 7 grid world, excluding black squares
  - Actions: up, down, left, right and wait
  - Rewards: +5 for reaching the ending position
  - Main objective: To move to the ending position



Checkers-v0



Switch2-v0



Switch4-v0

Table of Content

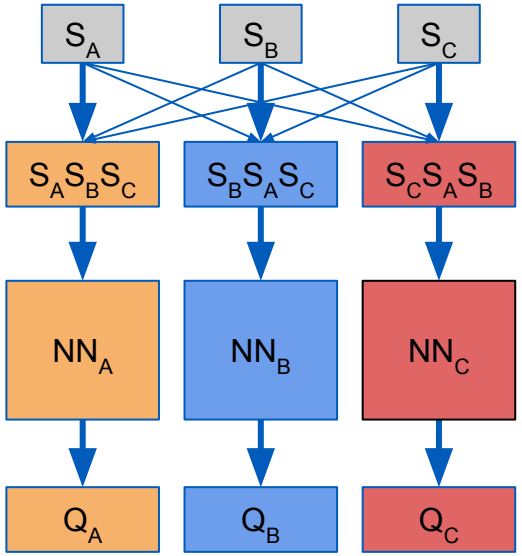
Implementation

- Vanilla DQN:
  - Use neural network to approximate the Q-values.
- Dueling DQN:
  - Separate the estimators by two new streams, one for the state value V(s) and the other for the advantage of each action A(s, a).
- A2C:
  - Use the advantage function to reduce the variance of policy gradient.
- A3C:
  - Execute a set of environments in parallel so that increasing the diversity of training data.

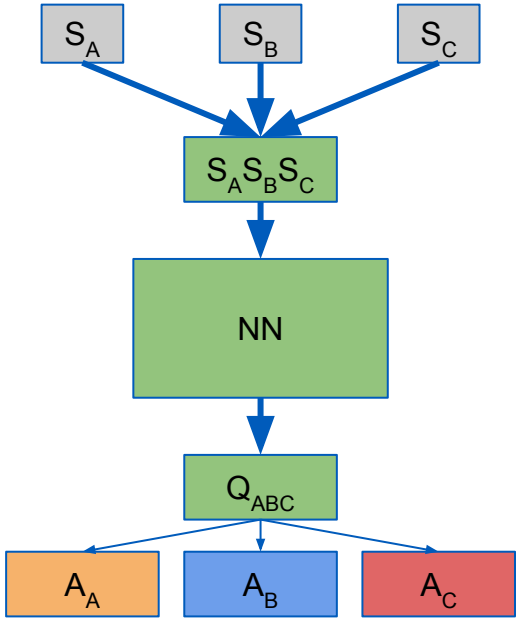
Table of Content

Experiment Setup

- We experiment with 2 architectural variants of each algorithm:
  - **Independent**: Each Agent has a separate set of parameters for estimating the value/policy function.
  - **Central**: There is a central network that takes as input the concatenated states of each agent, and estimates the overall policy/value function, which enables each agent to take action.



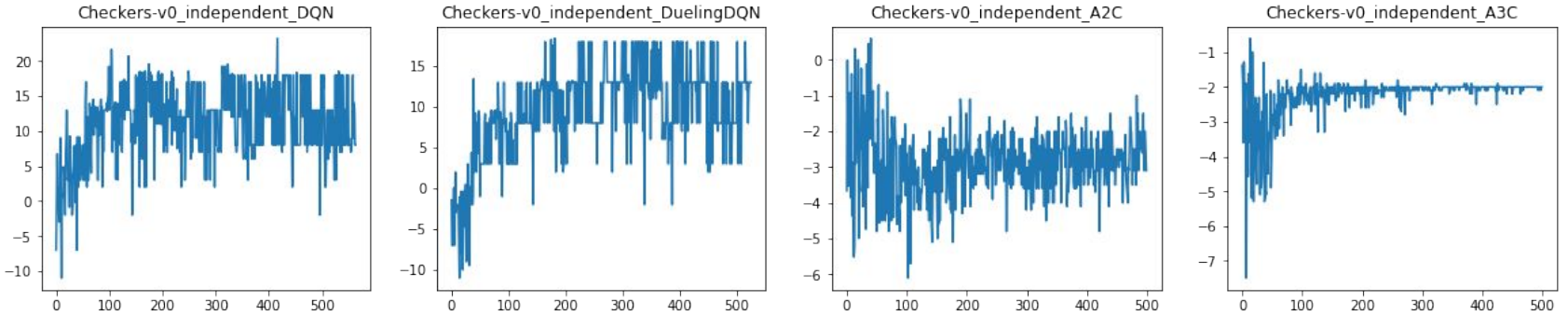
Independent Architecture

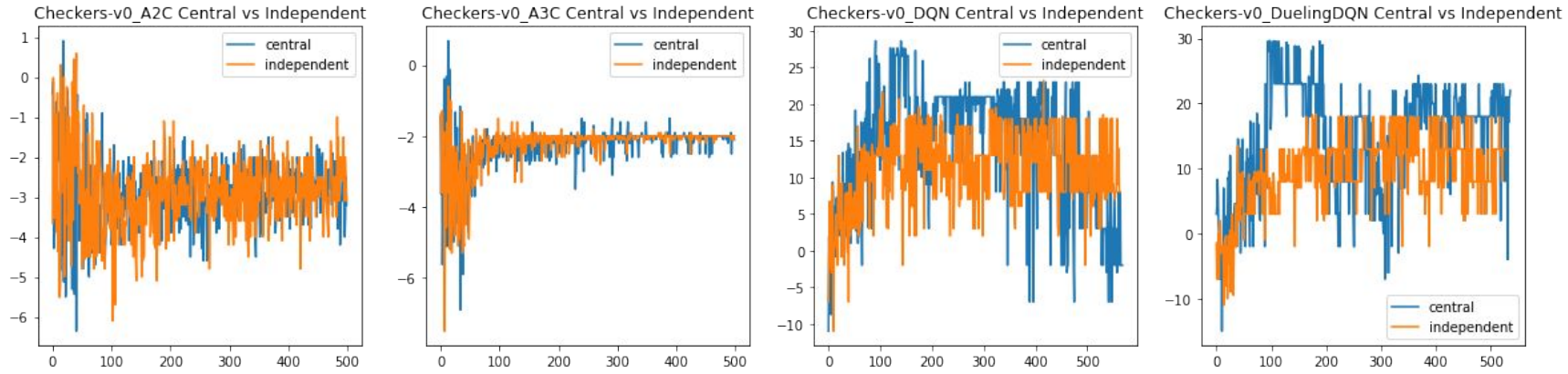


Central Architecture

Results and Analysis

- Checkers-v0

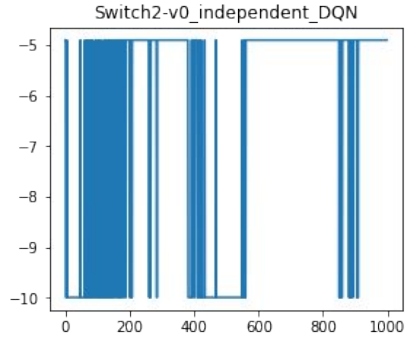


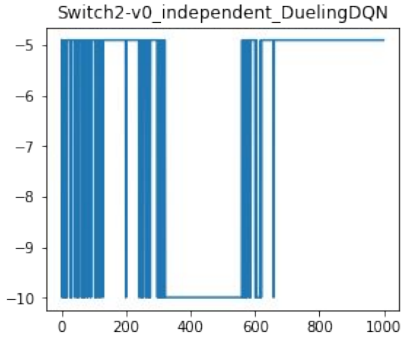


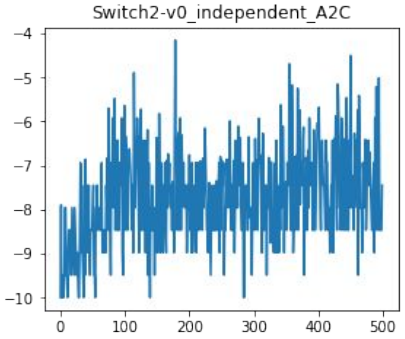
- DQN family algorithms have higher max rewards value, compared with AC family algorithms.
- A3C has smaller variation than A2C.
- Less difference of central and independent architecture for AC algorithms, while central DQN and Dueling DQN have larger max rewards value.

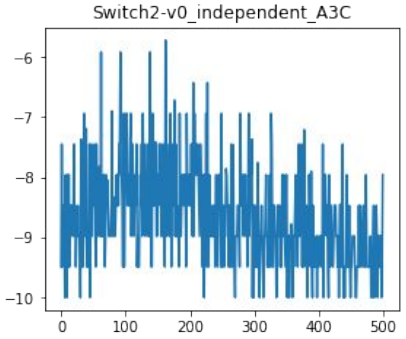
Results and Analysis

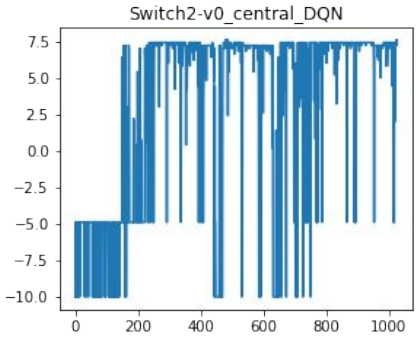
- Switch2-v0

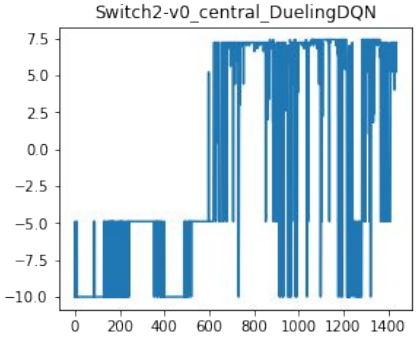


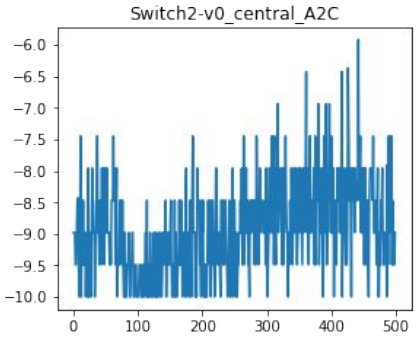


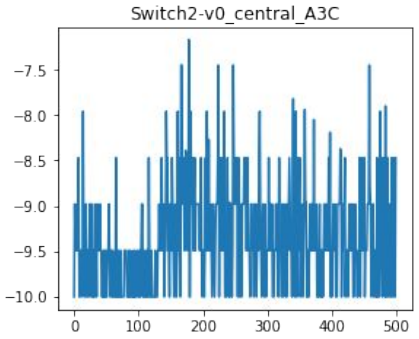


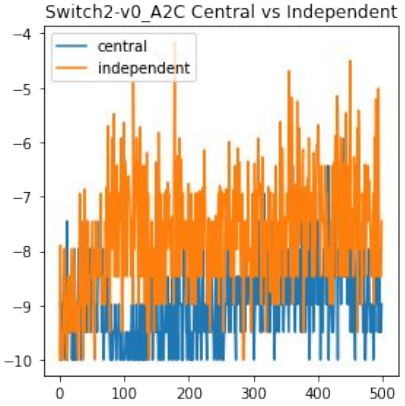


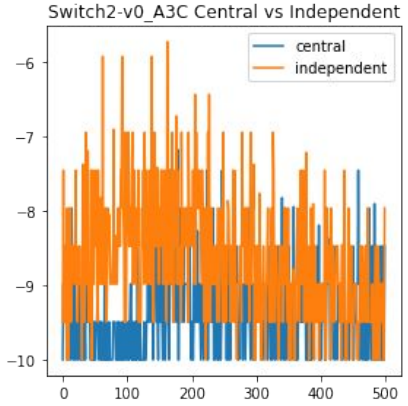


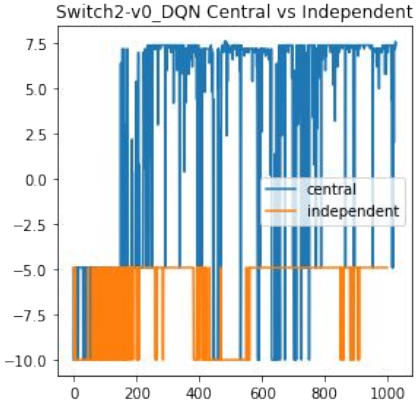


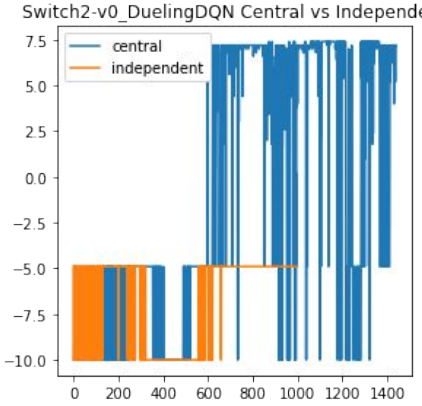








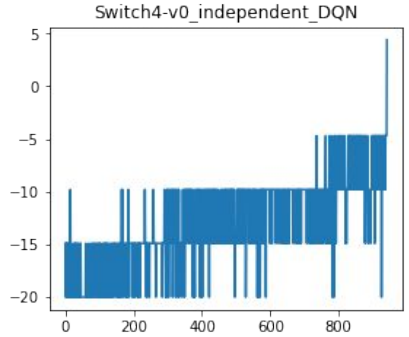


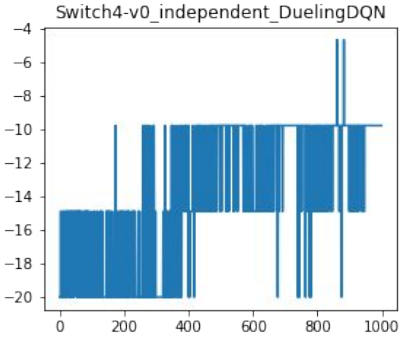


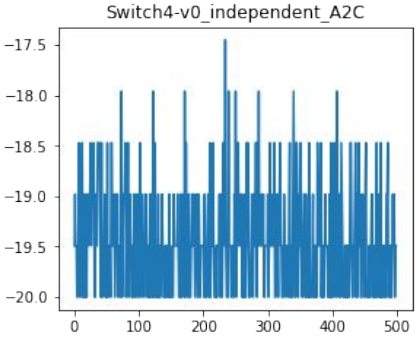
- DQN family algorithms have higher max rewards value, compared with AC family algorithms.
- The performance of A2C and A3C are similar.
- It seems that independent architecture benefits AC family algorithms, while central architecture benefits DQN family algorithms.

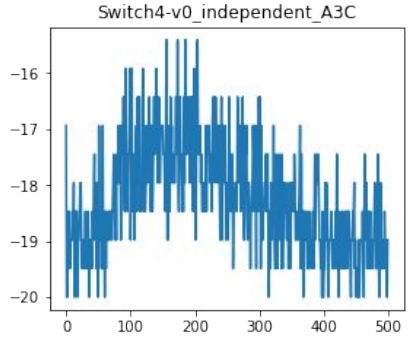
Results and Analysis

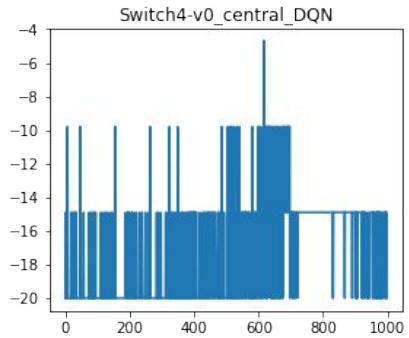
- Switch4-v0

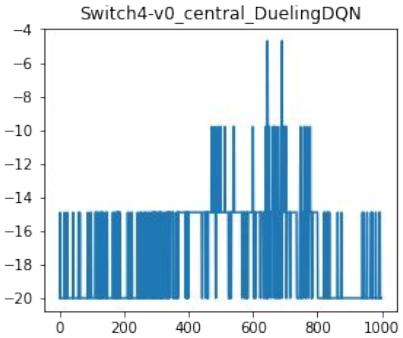


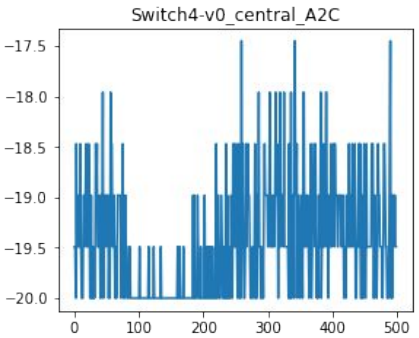


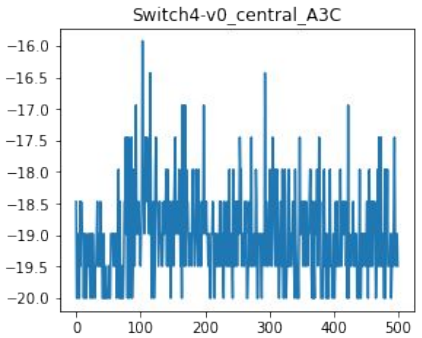


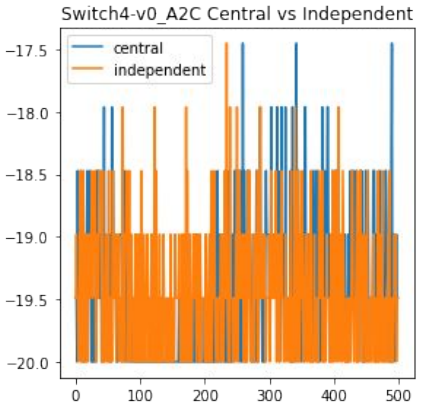


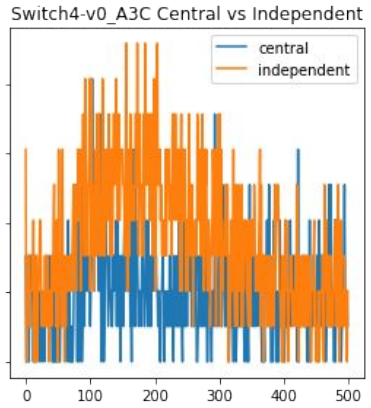


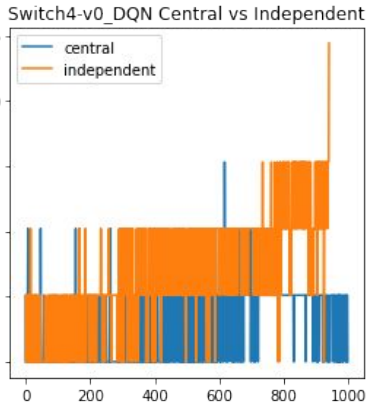


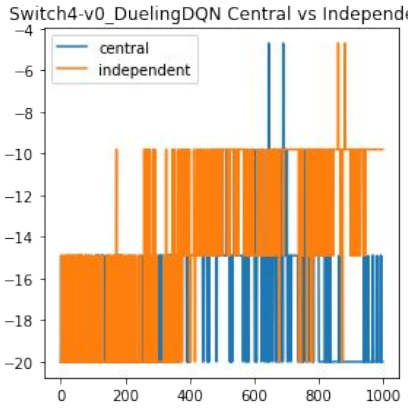












- DQN family algorithms have higher max rewards value, compared with AC family algorithms.
- It seems that A3C has smaller variation.
- Less difference of central and independent architecture for A2C, while for A3C, independent architecture seems unstable. For DQN family, it seems that independent architecture overperforms central architecture.

Results and Analysis

- Summary table:

Env	Type	Algo	Reward
Checkers-v0	central	DuelingDQN	22.00
Checkers-v0	independent	DQN	14.62
Checkers-v0	independent	DuelingDQN	13.00
Checkers-v0	independent	A3C	-2.00
Checkers-v0	central	DQN	-2.00
Checkers-v0	central	A2C	-2.00
Checkers-v0	central	A3C	-2.00
Checkers-v0	independent	A2C	-7.00

Env	Type	Algo	Reward
Switch2-v0	central	DQN	7.40
Switch2-v0	independent	DQN	-4.90
Switch2-v0	independent	DuelingDQN	-4.90
Switch2-v0	independent	A2C	-4.90
Switch2-v0	central	DuelingDQN	-4.90
Switch2-v0	independent	A3C	-4.90
Switch2-v0	central	A2C	-10.00
Switch2-v0	central	A3C	-10.00

Env	Type	Algo	Reward
Switch4-v0	independent	DQN	-4.70
Switch4-v0	independent	DuelingDQN	-9.80
Switch4-v0	central	DQN	-14.90
Switch4-v0	independent	A2C	-20.00
Switch4-v0	independent	A3C	-20.00
Switch4-v0	central	DuelingDQN	-20.00
Switch4-v0	central	A2C	-20.00
Switch4-v0	central	A3C	-20.00

- DQN family algorithms almost always have higher rewards than AC family algorithms, no matter independent or central.
- Central architecture algorithms almost always have higher rewards than independent architecture algorithms in environments with lesser number of agents.

Demo

Checkers-v0

Switch2-v0

Switch4-v0

Table of Content

Summary

- The performance of different algorithms on different environments may vary, therefore, we cannot say which algorithm is the best, it depends on the properties of the environment.
- It seems that central architecture is better than independent architecture, in environments with fewer number of agents.
- It seems that DQN family algorithms tend to obtain higher rewards than AC family algorithms, this may because the simpler structures of DQN, so that easy to overestimate.

Thank you !