Cooperative Multi-Agent Control using Deep Reinforcement Learning

2019. 02. 11

배영민

Index

- Cooperative Multi-Agent Control using Deep Reinforcement Learning, Gupta, et al.

1. Overview

- Cooperative Multi-Agent Control using Deep Reinforcement Learning, Gupta, et al.

Objectives

"Learning cooperative polices in complex(high-dimension), partially observable environment without explicit communication."

Problems

- Approximating Large and complex observation spaces.
- Solving Dec-POMDPs (Decentralized Partially Observable Markov Decision Process) efficiently.
- Scaling .
- Cooperative Multi-agent system without explicit communication between agents.

1. Overview

- Cooperative Multi-Agent Control using Deep Reinforcement Learning, Gupta, et al.

Problems

- Approximating Large and complex observation spaces.
  ➔ DQN, DDPG, TRPO

- Solving Dec-POMDPs (Decentralized Partially Observable Markov Decision Process) (Decentralized Control Problem) efficiently.
  ➔ Reword Structure, Curriculum Learning

- Scaling
  ➔ Decentralized parameter sharing training protocol

- Cooperative Scalable Multi-agent system without explicit communication between agents
  ➔ Use all of the above methods.

Solution

➔ Decentralized parameter sharing neural network policy(PS-TRPO)

2. Introduction

Dec-POMDP, POMDP

Dec-POMDP(Decentralized – Partially Observable Markov Decision Process.)

The Dec-POMDP just extends single-agent POMDP by considering joint actions and observations.
POMDP only consider a single agent, but Dec-POMDP deal with the effect of uncertainty with respect to other agents. [1]

Definition 1 (Dec-POMDP). A *decentralized partially observable Markov decision process* is defined as a tuple $\langle \mathcal{D}, S, \boldsymbol{A}, T, R, \boldsymbol{O}, O, h, I \rangle$, where

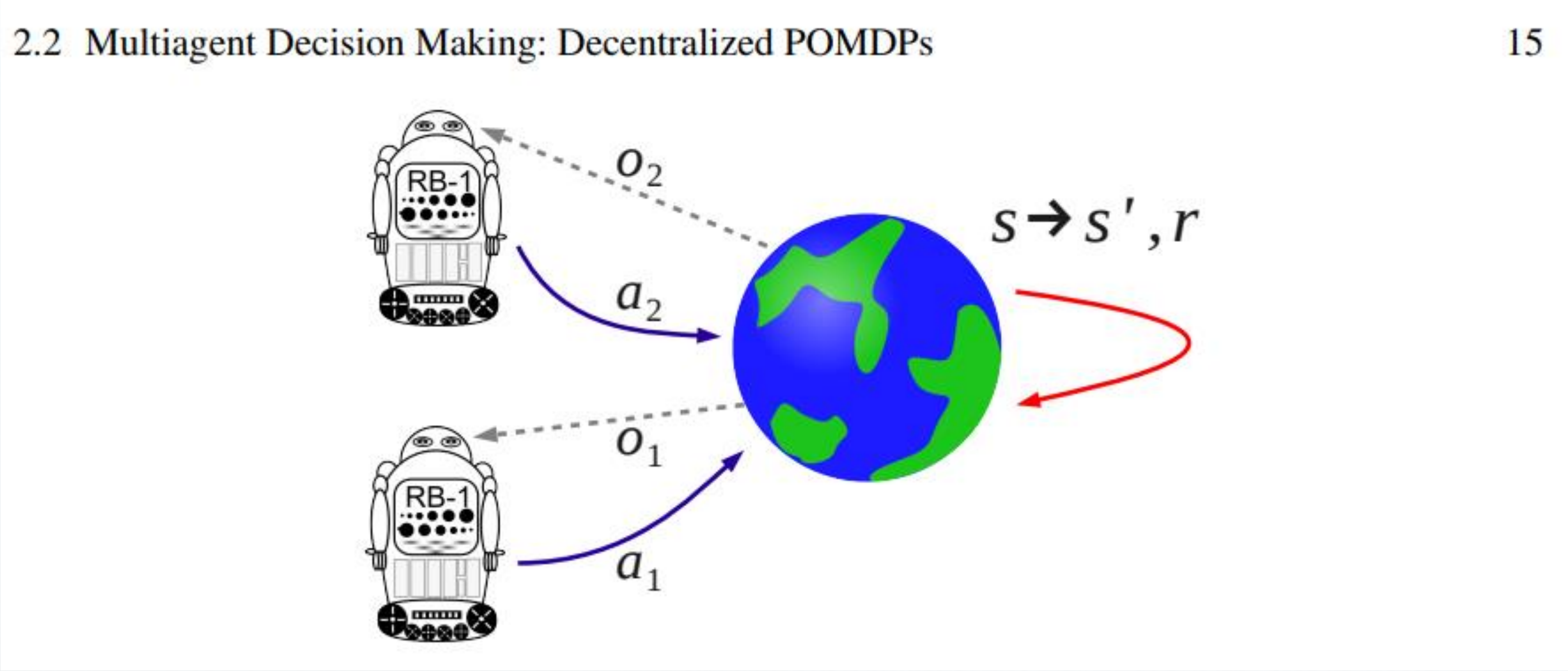
- $\boxed{\mathcal{D} = \{1, \ldots, n\} \text{ is the set of } n \text{ agents.}}$
- S is a finite set of states s in which the environment can be.
- $\boldsymbol{A}$ is the finite set of $\boxed{\text{joint actions.}}$
- T is the transition probability function.
- R is the immediate reward function.
- $\boldsymbol{O}$ is the finite set of $\boxed{\text{joint observations.}}$
- O is the observation probability function.
- h is the horizon of the problem.
- $I \in \mathcal{P}(\mathcal{S})$, is the initial state distribution at stage $t = 0$.

2. Introduction

Dec-POMDP, MOMDP

Dec-POMDP(Decentralized – Partially Observable Markov Decision Process.)

The Dec-POMDP just extends single-agent POMDP by considering joint actions and observations.
POMDP only consider a single agent, but Dec-POMDP deal with the effect of uncertainty with respect to other agents. [1]



3. Methods

Three Deep Reinforcement Learning Algorithm and etc.

논문에서 사용한 강화학습 알고리즘 : DQN, DDPG, TRPO

Deep Q-Network

Temporal-difference error

$$L_i(\theta_i) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\left(r + \gamma \max_{a'} Q(s', a'; \theta_i^-) - Q(s, a; \theta_i)\right)^2\right]$$

DDPG

Actor-Critic

$$\nabla_{\theta_\mu} J \approx \mathbb{E}_{s_t\sim\rho_\pi}\left[\nabla_a Q(s, a \mid \theta_Q)|_{s=s_t, a=\mu(s_t)} \nabla_{\theta_\mu} \mu(s \mid \theta_\mu)|_{s=s_t}\right]$$

TRPO

Policy Gradient

$$\begin{aligned} &\underset{\theta}{\text{Maximize}} \quad \mathbb{E}_{s\sim\rho_{\theta_k}, a\sim\pi_{\theta_k}}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A_{\theta_k}(s, a)\right] \\ &\text{subject to} \quad \mathbb{E}_{s\sim\rho_{\theta_k}}\left[D_{KL}(\pi_{\theta_k}(\cdot|s) \| \pi_\theta(\cdot|s))\right] \leq \Delta_{KL} \end{aligned}$$

3. Methods

Three Deep Reinforcement Learning Algorithm and etc.

Reward Structure

- Dec-POMDP에서 보상은 모든 에이전트들과 결합되어 공유되어졌음(shared jointly by all agents).
- 본 논문에서는 기존의 centralized representation(즉 모든 에이전트들과 공유된 보상)이 아닌 decentralized representation을 사용하였음.
- 이는 에이전트 각각에게 local reward를 줌.

- 효과: shared jointly된 보상이 아닌 각각의 agent에게 보상이 주어짐으로써, 학습에 필요한 샘플 수가 감소하여 학습 시간이 줄어듬

3. Methods

Three Deep Reinforcement Learning Algorithm and etc.

Curriculum Learning

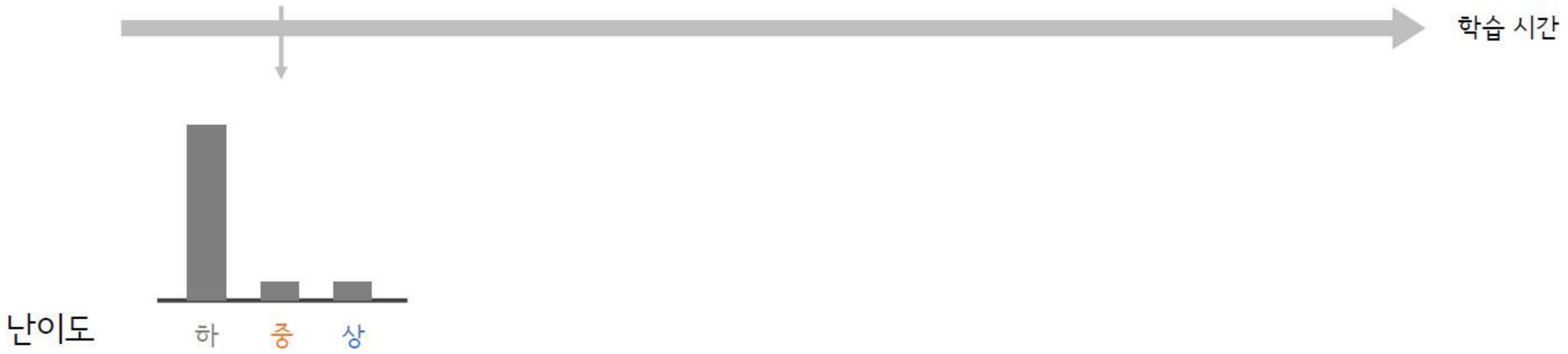
- Curriculum Learning은 쉬운 작업(simple task)부터 먼저 학습한 다음, 이후 그 기반을 바탕으로 어려운 작업(difficult task)을 난이도를 올려가며 학습함.
- 인간이 학습하는 것과 비슷하게 쉬운작업부터 차근차근 어려운작업까지 학습하는 것과 같음 [2]
- 해당 논문에서는 T라는 작업 세트(ordered set of task)를 만들고 점점 작업의 난이도를 올려가는 방법으로 커리큘럼 러닝을 구현함.

```yaml
tasks:
  l1:
    n_walkers: 2

  l2:
    n_walkers: 3

  l3:
    n_walkers: 4

  l4:
    n_walkers: 5

  l5:
    n_walkers: 6

  l6:
    n_walkers: 7

  l7:
    n_walkers: 8

  l8:
    n_walkers: 9

  l9:
    n_walkers: 10

thresholds:
  lesson: 10
  stop: 20

n_trials: 20
eval_trials: 20
metric: ret
```

3. Methods

Three Deep Reinforcement Learning Algorithm and etc.

Curriculum Learning

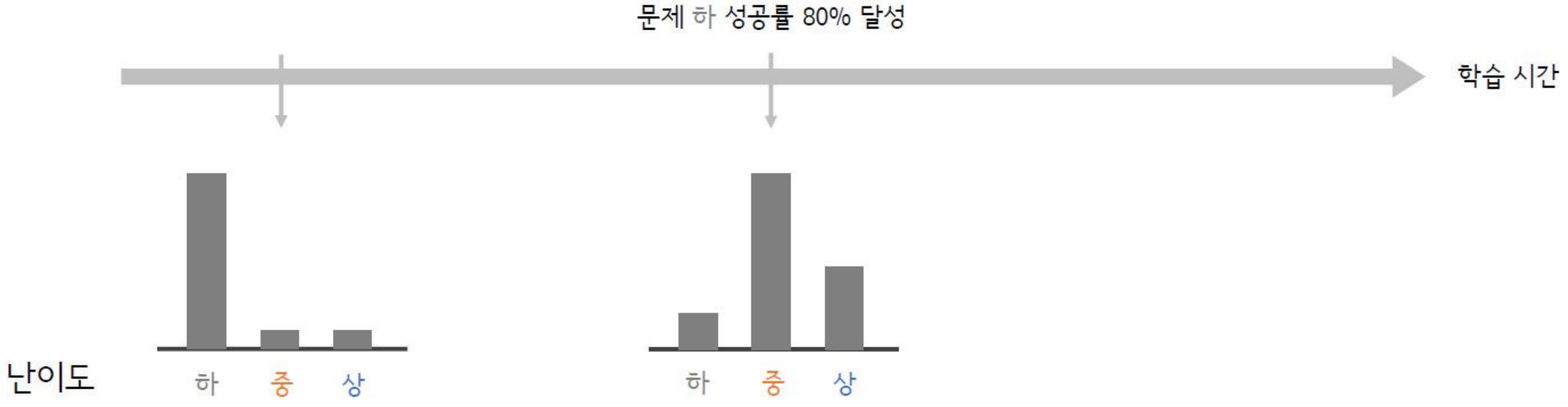


처음에는 가장 쉬운 문제를 많이 학습

3. Methods

Three Deep Reinforcement Learning Algorithm and etc.

Curriculum Learning

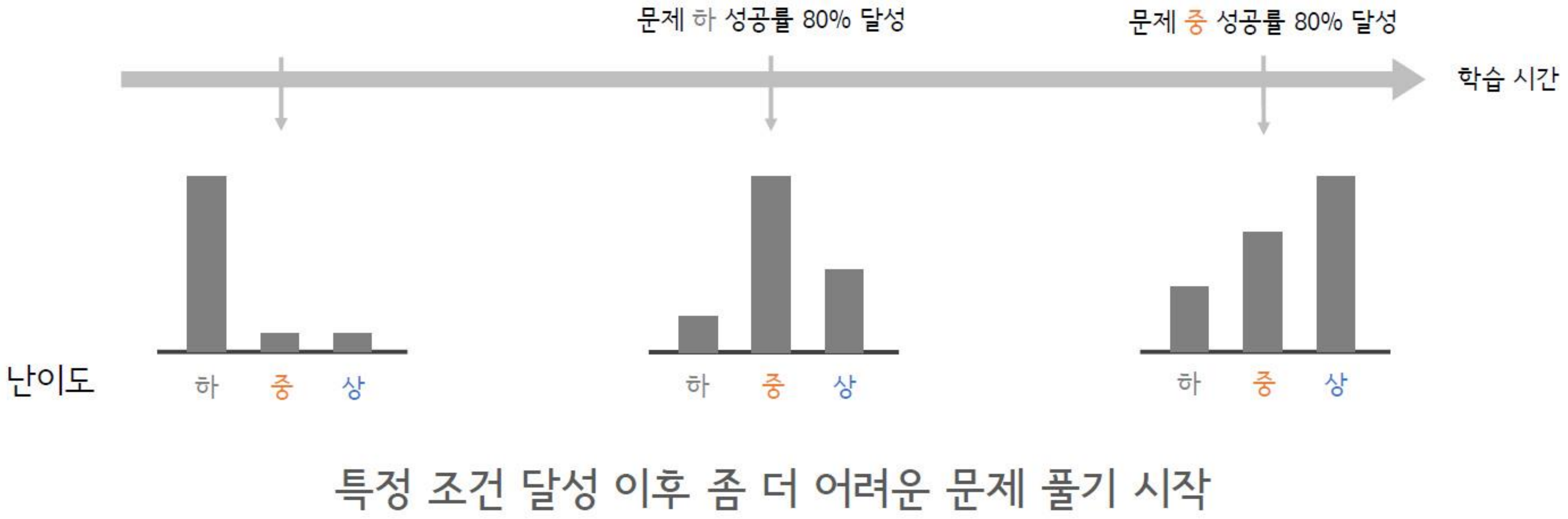


특정 조건 달성 이후 좀 더 어려운 문제 풀기 시작

3. Methods

Three Deep Reinforcement Learning Algorithm and etc.

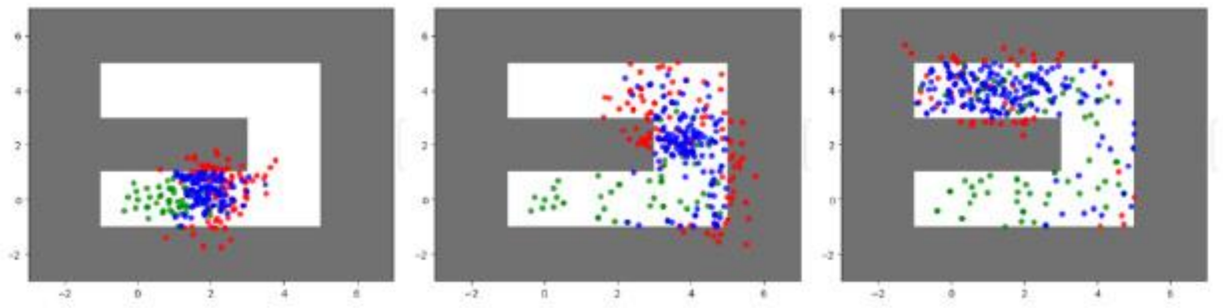
Curriculum Learning



3. Methods

Three Deep Reinforcement Learning Algorithm and etc.

Curriculum Learning



(a) Iteration 5 (b) Iteration 90 (c) Iterartion 350

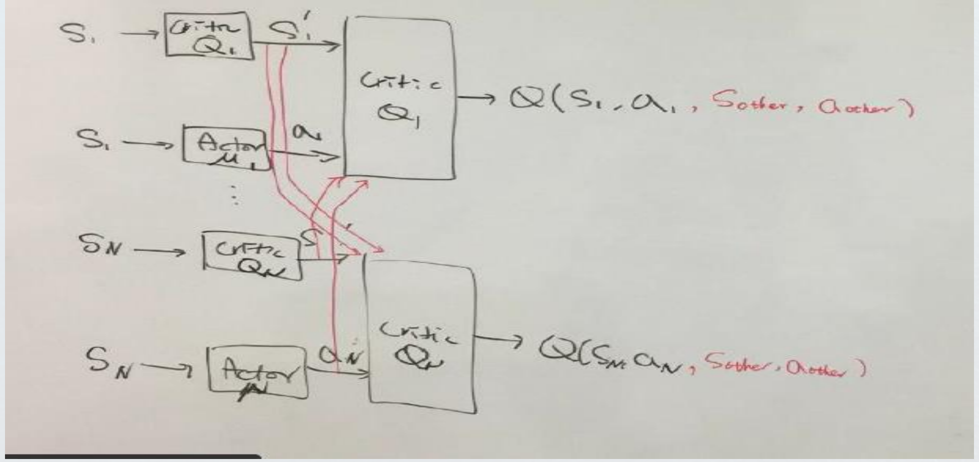
3. Methods

Three Training Schemes

Centralized

- 모든 에이전트들의 행동이 결합되어있는 상태에서 학습을 함(모든 에이전트가 policy를 알고 있음). 아래 사진 참조
- 단점: 모든 에이전트가 결합되어 있음으로, 관측 공간 (Observation space), 행동 공간(Action space)이 에이전트의 수에 따라 기하 급수적으로 많아짐.

$$P(\vec{a}) = \prod_i P(a_i)$$



Decentralized

- 모든 에이전트는 독립된 정책망을 가지고 있음(Sub-policies that map the joint observation to an action for a single agent)
- 따라서 하나의 에이전트는 Joint action distribution이 아닌 독립된 정책망에서 행동을 결정함. Action space 가 상당히 줄어듬

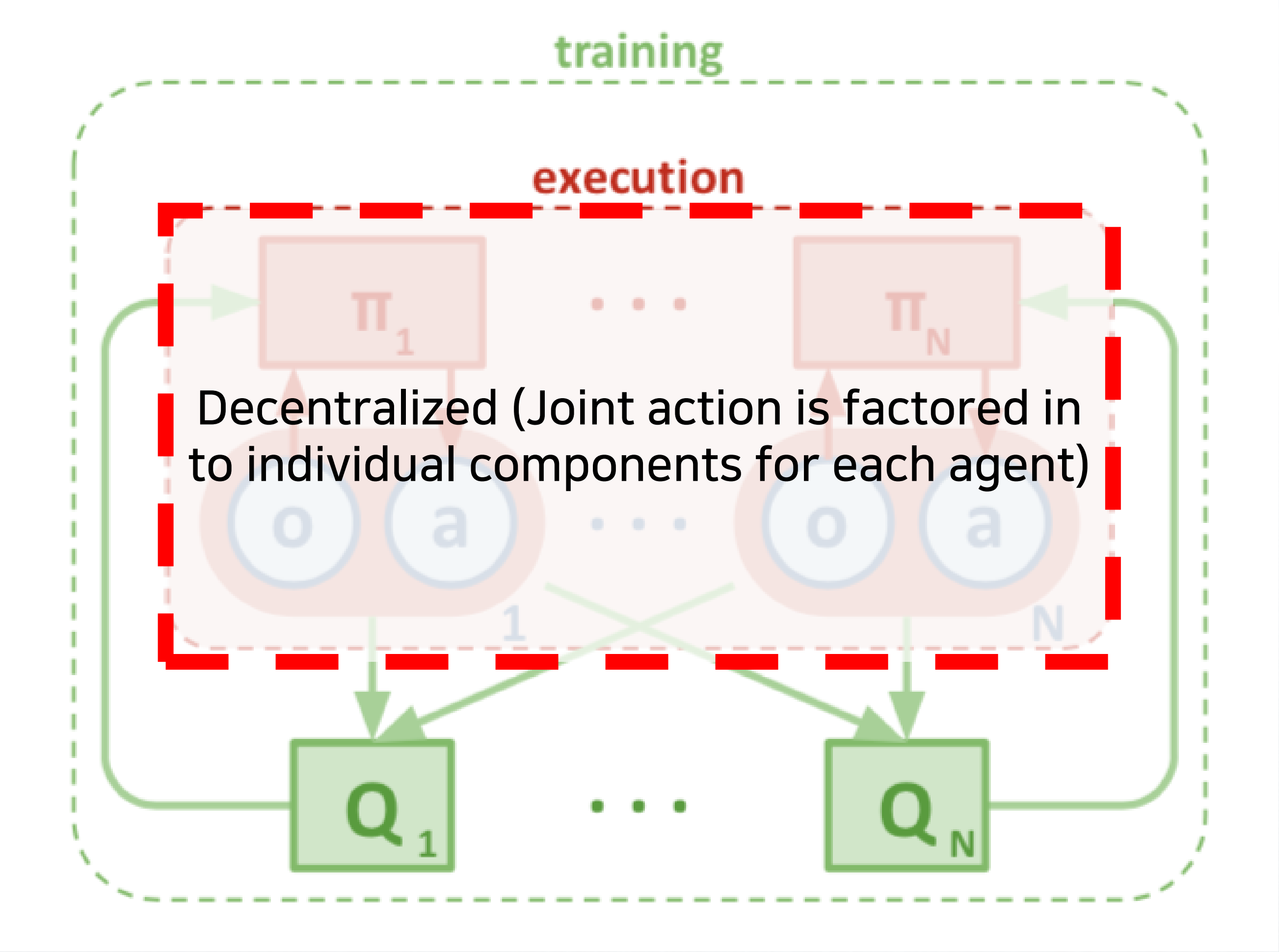
space from $|\mathcal{A}|^n$ to $n|\mathcal{A}|$

3. Methods

Three Training Schemes

Centralized

Decentralized

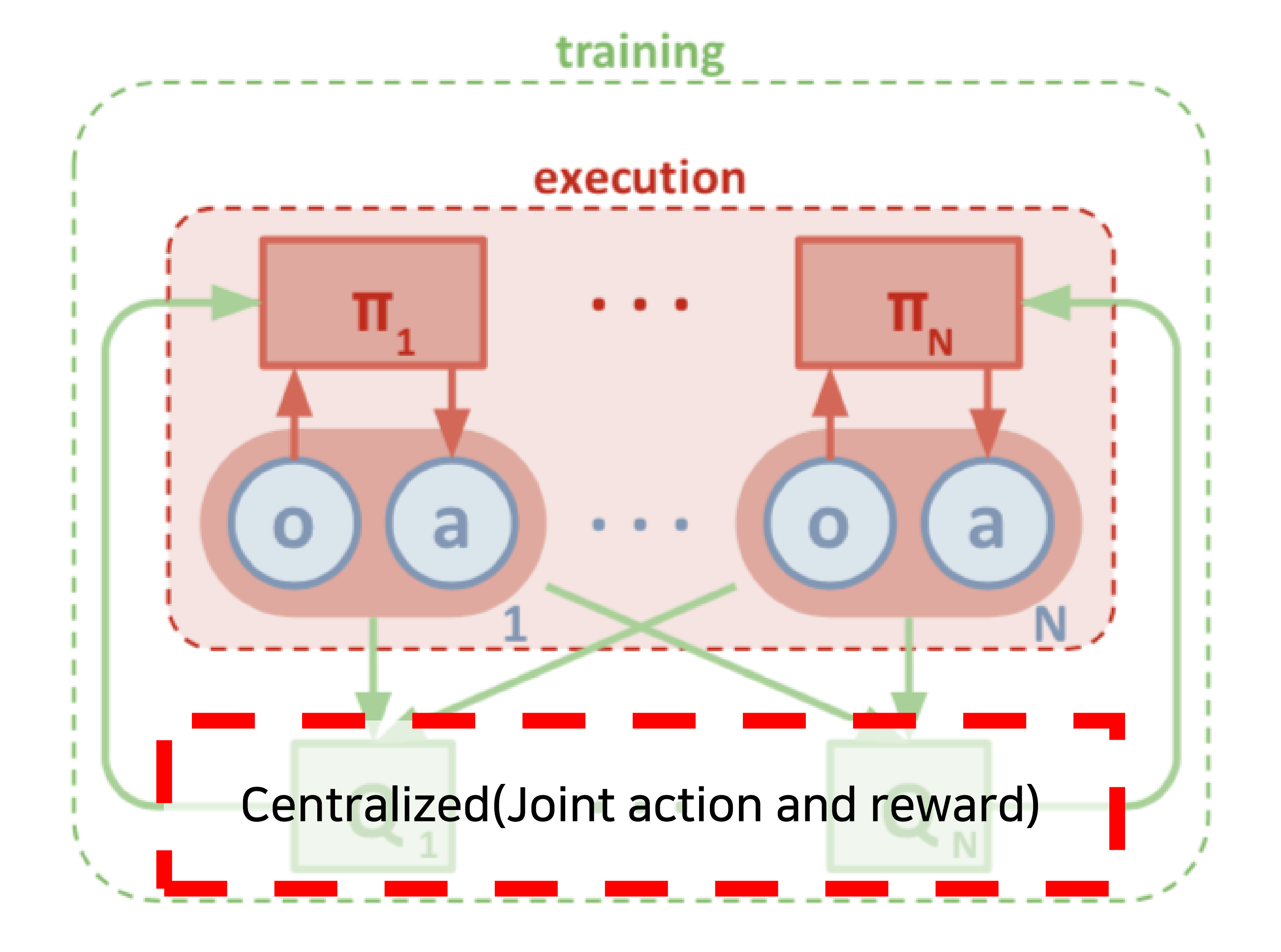


3. Methods

Three Training Schemes

Centralized

Decentralized



3. Methods

Three Training Schemes

Concurrent (discussion needed)

- In concurrent learning, each agent learns its own individual policy.
- Concurrent policies map an agent's private observation to an action for that agent. Each agent's policy is independent.
- In the policy gradient approach, this means optimizing multiple polices simultaneously from the joint reward signal.

3. Methods

Three Training Schemes

Parameter Sharing

- 파라미터 공유는 하나의 경험에 대해 모든 에이전트가 동시에 학습함.
- 파라미터 공유때문에 모든 에이전트가 같은 행동을 하진 않음. (모든 에이전트마다 각자 다른 관측치를 받기 때문)

- 본 논문은 파라미터 공유(Parameter Sharing)과 Decentralized를 사용해서 확장가능한(Scalable) Multi-agent control 이 가능해짐.

3. Methods

Three Training Schemes

(Decentralized) PS-TRPO

- (Decentralized) PS-TRPO는 Policy gradient 계열의 TRPO에 decentralized policy(Algorithm 1)와 parameter sharing (Equation 1)을 결합한 논문에서 새로 주장하는 알고리즘임.

- 기존의 TRPO알고리즘과 비슷하나, multi-agent를 위해 m이라는 agent index를 도입함.

Algorithm 1 PS-TRPO

Input: Initial policy parameters Θ_0, trust region size Δ
for $i \leftarrow 0, 1, \ldots$ **do**
  Rollout trajectories for all agents $\vec{\tau} \sim \pi_{\theta_i}$
  Compute advantage values $A_{\pi_{\theta_i}}(o^m, m, a^m)$ for each agent m's trajectory element.
  Find $\pi_{\theta_{i+1}}$ maximizing Eq. ($\boxed{1}$)
    subject to $\overline{D}_{KL}(\pi_{\theta_i} \| \pi_{\theta_{i+1}}) \leq \Delta$

$$L(\theta) = \mathbb{E}_{o \sim \rho_{\theta_k}, a \sim \pi_{\theta_k}} \left[\frac{\pi_\theta(a \mid o, m)}{\pi_{\theta_k}(a \mid o, m)} A_{\theta_k}(o, m, a) \right] \qquad (1)$$

4. Experiment & Result

Experiment Overview

Overview

- 논문에서는 앞에서 언급한(decentralized, parameter sharing) 방법들을 TRPO, DDPG, DQN에 적용하여 Discrete environment 와 Continuous environment에서 agent들의 cooperative behavior을 테스트함.

- PS-TRPO 알고리즘에서 decentralized, concurrent, centralized training scheme을 비교 분석하였음.
- Discrete Control Task에선 TRPO 와 DQN을 비교하였음.
- Continuous Control Task에선 TRPO와 DDPG를 비교하였음.
- Multi-Walker domain(Observation history가 중요한 domain)에서 4개 training scheme(Parameter sharing, centralized, concurrent, curriculum)에 대해 비교 분석하였음.

4. Experiment & Result

Experiment Overview

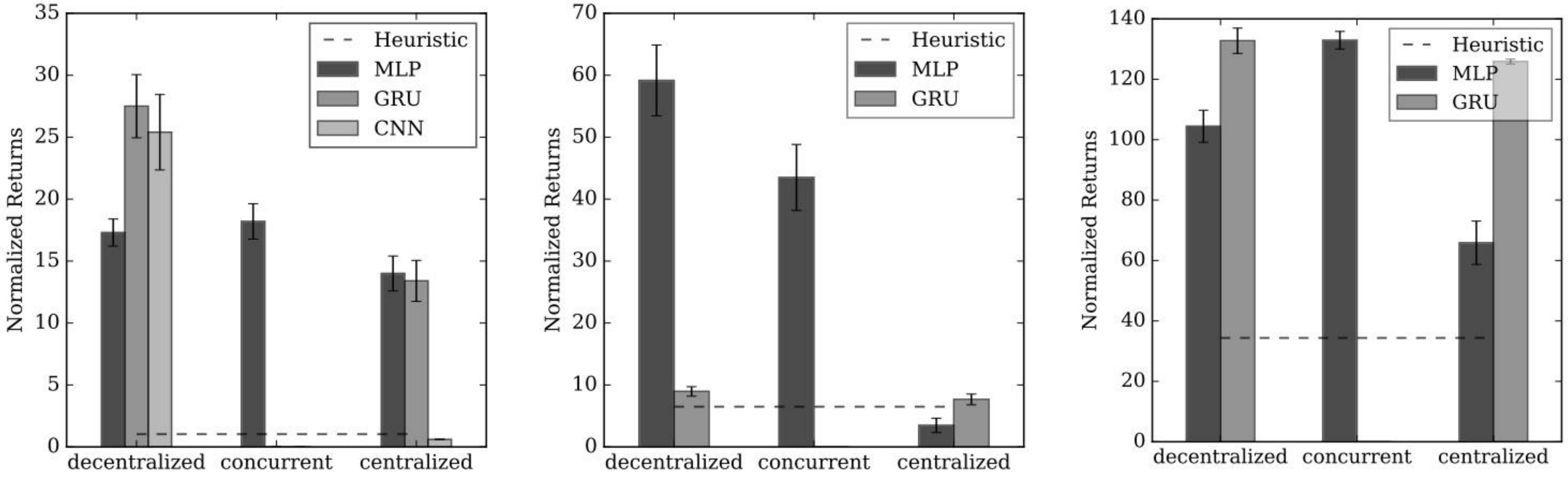
Overview

- 논문에서는 앞에서 언급한(decentralized, parameter sharing) 방법들을 TRPO, DDPG, DQN에 적용하여 Discrete environment 와 Continuous environment에서 agent들의 cooperative behavior을 테스트함.

- PS-TRPO 알고리즘에서 decentralized, concurrent, centralized training scheme을 비교 분석하였음.
- Discrete Control Task에선 TRPO 와 DQN을 비교하였음.
- Continuous Control Task에선 TRPO와 DDPG를 비교하였음.
- Multi-Walker domain(Observation history가 중요한 domain)에서 4개 training scheme(Parameter sharing, centralized, concurrent, curriculum)에 대해 비교 분석하였음.

4. Experiment & Result

Normalized average returns for multi-agent policies.



(a) Pursuit (b) Waterworld (c) Multi-Walker

4. Experiment & Result

Discrete/Continuous Task, Reward Structure 비교 분석.

Discrete/Continuous Task

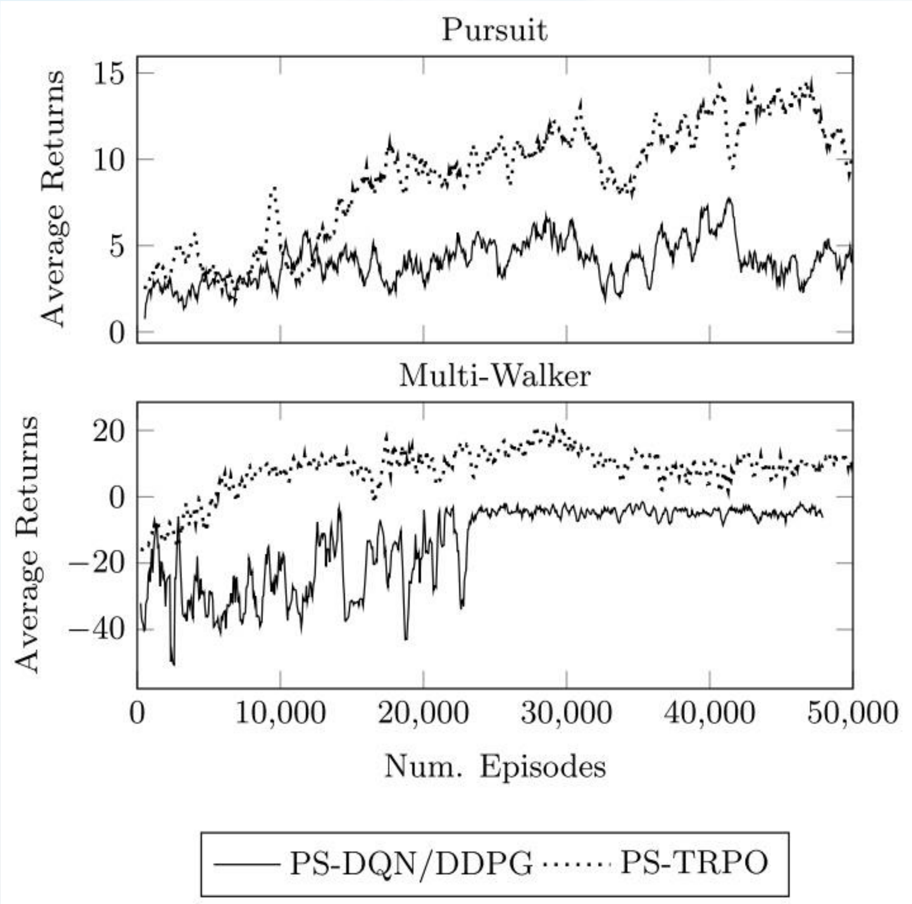
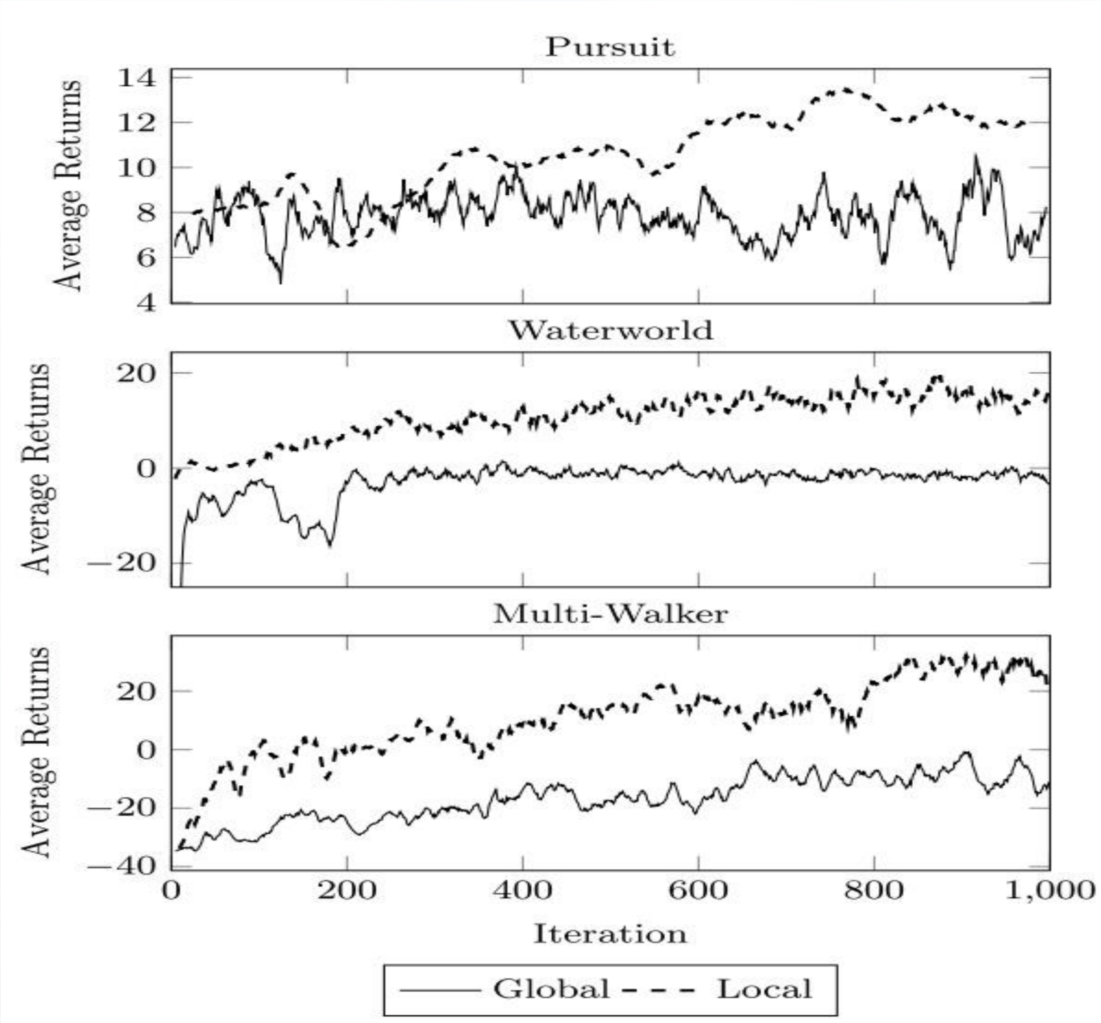


Figure 3: Average returns comparing PS-TRPO and DQN/DDPG in Pursuit and Multi-Walker Domains.

Reward Structure



4. Experiment & Result

Continuous Control Task에서 4개 training scheme 비교 분석

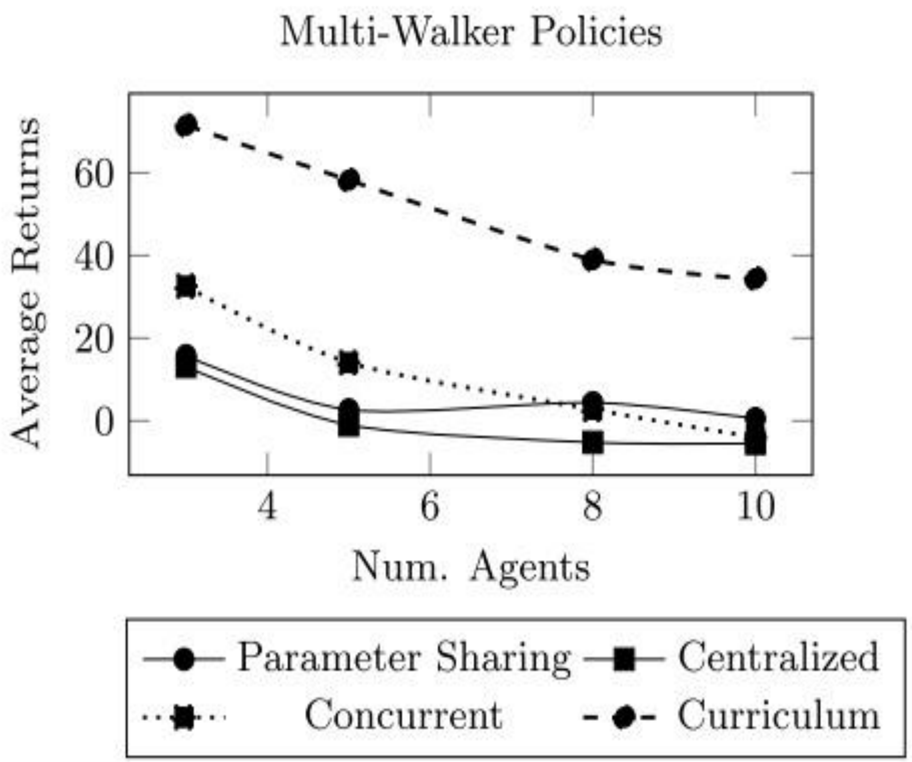


Figure 7: Performance of multi-walker policies as a function of the number of agents during training. Each data point in the decentralized, centralized, and concurrent curves was generated by training and evaluating a policy with a fixed number of agents. The curriculum curve was generated by evaluating a single policy with varying number of agents.

4. Experiment & Result

Scaling

Scaling Problem

- Decentralized method 덕분에 큰 관측공간(observation space)와 많은 multi-agent를 학습시킬 수 있게 됨.
- 다만 에이전트 수가 늘어날때마다 학습이 안되는 상황이 발생하고, 이를 막기 위해 curriculum learning 을 도입함.

- 그럼에도 불구하고 decentralized policy 학습을 다량의 에이전트에 적용하는것(generalizing)은 불가능하고있음.
- 본 논문에서는 Decentralized method + Curriculum learning 을 통해 좀 더 improvement할 가능성이 있다고 봄.

Algorithm 2 Curriculum Training

Input: Curriculum $\mathcal{T}$, Iteration n, Policy π_Θ, $r_{\text{threshold}}$
$\alpha_{\mathcal{T}} \leftarrow [\text{length}(\mathcal{T}), 1, 1, \ldots]$
while $r_{\text{min}} < r_{\text{threshold}}$ **do**
  {Sample task from the task distribution.}
  $w \sim \text{Dirichlet}(\alpha_{\mathcal{T}})$
  $i \sim \text{Categorical}(w)$
  {Apply optimization step for a few iterations.}
  PS-TRPO $(\mathcal{T}_i, \pi_\theta, n)$
  {e_{curr} is the task with the highest weight $\alpha_{\mathcal{T}}$.}
  $r_{e_{\text{curr}}} \leftarrow \text{Evaluate}(\pi_\theta, e_{\text{curr}})$
  if $r_{e_{\text{curr}}} > r_{\text{threshold}}$ **then**
    Circular shift $\alpha_{\mathcal{T}}$ weights to the next task
  {Find the minimum average reward across tasks.}
  $r_{\text{min}} \leftarrow \min_{\mathcal{T}} \mathbb{E} r_{\mathcal{T}}$

Conclusion

- Cooperative Multi-Agent Control using Deep Reinforcement Learning, Gupta, et al.

논문에서 풀고자 했던 목표: “Learning cooperative polices in complex(high-dimension), partially observable environment without explicit communication.”

본 논문에서는 Decentralized parameter sharing neural network policy**(PS-TRPO)** 을 제안함으로서 기존의 아래와 같은 복합적인 문제들을 해결하고 high-dimensional, partially observable domain 에서 안정적으로 multi-agent control task를 풀었음.

1) Difficulty of approximating high-dimensional observation spaces
2) Difficulty of control large number of agents
3) Difficulty of accommodating partial observability(POMDPs)
4) Difficulty of handling continuous action spaces.

감사합니다.