Journal of Korea Robotics Society (2022) 17(1):040-047
https://doi.org/10.7746/jkros.2022.17.1.040 ISSN: 1975-6291 / eISSN: 2287-3961

시연에 의해 유도된 탐험을 통한 시각 기반의 물체 조작

Visual Object Manipulation Based on Exploration Guided by Demonstration

김 두 준[1] · 조 현 준[2] · 송 재 복†
Doo-Jun Kim[1], HyunJun Jo[2], Jae-Bok Song†

Abstract: A reward function suitable for a task is required to manipulate objects through reinforcement learning. However, it is difficult to design the reward function if the ample information of the objects cannot be obtained. In this study, a demonstration-based object manipulation algorithm called stochastic exploration guided by demonstration (SEGD) is proposed to solve the design problem of the reward function. SEGD is a reinforcement learning algorithm in which a sparse reward explorer (SRE) and an interpolated policy using demonstration (IPD) are added to soft actor-critic (SAC). SRE ensures the training of the critic of SAC by collecting prior data and IPD limits the exploration space by making SEGD's action similar to the expert's action. Through these two algorithms, the SEGD can learn only with the sparse reward of the task without designing the reward function. In order to verify the SEGD, experiments were conducted for three tasks. SEGD showed its effectiveness by showing success rates of more than 96.5% in these experiments.

Keywords: Imitation Learning, Manipulation, Variational Autoencoder, Reinforcement Learning

1. 서 론

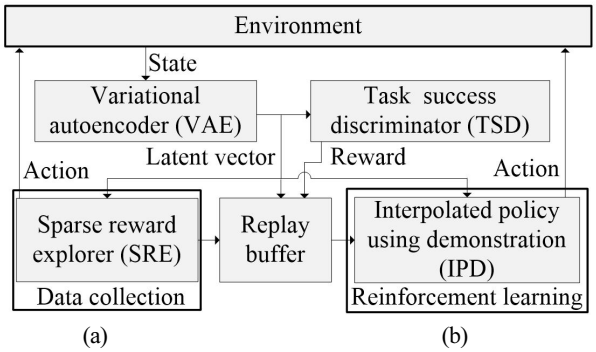
시각 기반의 강화학습 알고리즘이 발전함에 따라 로봇이 다양한 작업을 학습하여 수행할 수 있게 되었다. 다양한 작업을 학습하여 수행하기 위해 강화학습 알고리즘은 작업 대상 물체 혹은 로봇의 위치, 위치 변화량, 속도 등의 다양한 상태를 이용하여 특정 작업에 적합한 보상함수가 필요하다[1-3]. 그러나 작업 환경에 따라 로봇이 얻을 수 있는 정보가 제한될 수 있으며, 이와 같은 경우에는 적절한 보상함수를 설계하기 어렵다.

이러한 문제를 해결하기 위하여 전문가 시연(expert demonstration)을 모방하는 시각 기반의 강화학습 알고리즘[4,5]이 제안되었다. 이들 알고리즘에서는 강화학습 정책이 전문가의 행동을 추종하도록 함으로써 학습에 필요한 시간을 단축하거나[6] 정책이 전문가 시연에서 크게 벗어나는 비정상적인 행동을 취하는 것을 방지하였다[7]. 이러한 장점으로 인해 이들은 휴머노이드 로봇, 로봇 팔과 손 등 다양한 분야에서 사용되고 있지만, 이들 방법은 많은 시행착오가 필요하다. 이는 강화학습이 추정한 행동과 전문가 시연 간의 유사성을 평가하는 기준을 설정하는 명확한 가이드라인이 아직 존재하지 않아서, 시행착오를 통해 최적의 기준을 마련할 수밖에 없기 때문이다.

한편, 전문가 시연의 수집 시에, 전문가의 행동에 대한 모든 정보를 수집하기에는 큰 비용이 요구되므로, 전문가의 시연 영상만을 수집하기도 하는데, 이 경우에는 인공신경망으로 구성된 보상함수를 이용한다[8-10]. 이들 알고리즘에서는 인공신경망이 영상으로부터 보상을 추정하고, 이 보상을 이용하여 강화학습을 진행함으로써 강화학습의 정책이 전문가의 시연을 모방하도록 유도한다. 다만, 이들은 학습 중에 로봇의 탐험 범위가 전문가 시연에서 나타난 행동 범위를 벗어날 수 있는 환경에서는 학습이 제대로 이루어지지 않는다.

본 연구에서는 [Fig. 1]과 같은 구조를 갖는 전문가 시연 기반의 강화학습 알고리즘인 Stochastic Exploration Guided by Demonstration (SEGD)를 제안한다. SEGD는 기존의 전문가 시연 기반 알고리즘에서 복잡한 보상함수를 사용한 것과 다르게,

Received : Dec. 7. 2021; Revised : Jan. 24. 2022; Accepted : Jan. 24. 2022
※ This work was supported by IITP grant funded by the Korea Government MSIT. (No. 2018-0-00622)
1. Principal Researcher, Mechanical Engineering, Korea University, Seoul, Korea (rlaenwns95@korea.ac.kr)
2. Secondary Researcher, Mechanical Engineering, Korea University, Seoul, Korea (jhj0630@korea.ac.kr)
† Professor, Corresponding author: Mechanical Engineering, Korea University, Seoul, Korea (jbsong@korea.ac.kr)



[Fig. 1] Pipeline of SEGD: (a) Data collection with SRE and (b) reinforcement learning with IPD

작업 성공 시 발생하는 희소 보상만으로 학습하기 위하여 Soft Actor-Critic (SAC)에 두 가지 요소를 추가하였다[11]. 먼저, 강화학습을 시작하기 전에 로봇을 직접 제어하여 강화학습에 필요한 사전 데이터를 수집하는 Sparse Reward Explorer (SRE)를 사용한다. 로봇이 움직이면서 발생한 데이터는 리플레이 버퍼(replay buffer)에 현재 상태 $\boldsymbol{s}$, 현재 행동 $\boldsymbol{a}$, 보상 r, 그리고 다음 상태 $\boldsymbol{s'}$을 종합한 $(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s'})$ 형태의 전이(transition)로 저장된다. 다음으로, SAC에 일반적으로 정책으로 사용되는 Gaussian policy (GP) 대신에 Interpolated Policy using Demonstration (IPD)를 사용한다. IPD는 전문가와 유사한 행동 후보들을 추출하여 탐험을 수행하므로 탐험 공간이 축소되고 학습시간이 단축된다. 이러한 두 요소를 통하여 SEGD는 보상함수의 설계 없이 희소 보상만을 통한 학습이 가능하다.

본 연구의 기여는 다음과 같다. 첫째, 보상함수의 설계가 필요하지 않은 SEGD를 제안하였다. 둘째, 학습에 필요한 희소 보상을 수집하는 SRE를 제안하였다. 셋째, 전문가와 유사한 행동을 기반으로 탐험을 진행하는 IPD를 제안하였다.

본 논문의 구성은 다음과 같다. 2장에서는 SEGD의 전체적인 구조와 구성 요소들을 설명하고, 3장에서는 세 가지 다른 환경에서의 실험 결과를 분석한다. 마지막으로, 4장에서는 결론을 도출한다.

2. SEGD

본 연구에서 제안하는 시각 기반의 물체 조작 알고리즘인 SEGD는 영상을 잠재벡터(latent vector)로 압축하는 Variational Auto-Encoder (VAE), 전문가의 시연을 모방하는 Behavioral Cloning (BC), 그리고 작업 성공을 판단하는 Task Success Discriminator (TSD)의 학습을 우선적으로 수행한다[12,13]. 이후 이들을 이용하여 [Fig. 1]의 (a)와 같이 SRE를 통해 사전 데이터 수집 과정을 거친다. SRE는 입력된 RGB 영상을 VAE에 의해 압축한 잠재벡터를 기반으로 로봇의 말단 위치를 조작한다. 이 과정에서 $(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s'})$ 형태의 전이를 리플레이 버퍼에 저장한다. 마지막으로, [Fig. 1]의 (b)와 같은 강화학습 과정에서는 로봇을 직접 제어하여 데이터를 리플레이 버퍼에 수집하는 과정과 리플레이 버퍼에 수집된 데이터를 기반으로 정책인 IPD의 성능을 개선하는 과정을 반복한다. 먼저 로봇을 직접 제어하여 데이터를 리플레이 버퍼에 수집하는 과정에서 VAE는 입력된 RGB 영상을 잠재벡터로 압축하여 IPD에 잠재벡터를 전달한다. 그리고 IPD는 이 잠재벡터를 입력 받아 로봇의 말단 위치 변화량과 그리퍼의 개폐 상태가 포함된 행동을 출력한다. 이때 출력된 IPD의 행동을 통해 실제 로봇의 말단 위치를 제어하고, 그리퍼의 개폐를 조절한다. 다음으로, 수집된 데이터를 기반으로 정책 IPD의 성능을 개선하는 과정에서는 리플레이 버퍼에서 임의로 추출한 잠재벡터, 로봇의 말단 위치 변화량, 그리퍼의 개폐 상태, 그리고 보상의 정보가 포함된 전이를 기반으로 IPD의 학습을 수행한다.

2.1 Variational autoencoder (VAE)

SEGD에서는 RGB 영상을 이용하여 강화학습을 수행하므로, 영상으로부터 로봇과 대상 물체의 위치나 형태에 대한 정보를 분석하는 것이 중요하다. 이를 위해 입력된 영상을 압축 후 재구성하는 방법을 학습함으로써 입력된 영상의 가장 중요한 정보를 압축한 잠재벡터를 추출하는 VAE를 사용한다. 이를 위해 VAE는 엔코더(encoder)와 디코더(decoder)로 구성되며, 이들의 구체적인 구조는 [Table 1]과 같다. 이때 VAE의 엔코더는 (128, 128, 3) 크기의 영상을 압축하여 (64, 1) 크기의 잠재벡터를 출력하고, VAE의 디코더는 이 (64, 1) 크기의 잠재벡터가 입력되면 (128, 128, 3) 크기의 원본 영상을 출력한다.

[Table 1] Structure of encoder and decoder of VAE

	Input channels	Output channels	Kernel size	Stride	Padding
Encoder	3	128	5	2	2
	128	256	3	2	1
	256	512	3	2	1
	512	512	3	2	1
	512	512	3	2	1
	512	64	4	1	0
Decoder	64	512	4	1	0
	512	512	3	2	1
	512	512	3	2	1
	512	256	3	2	1
	256	128	3	2	1
	128	128	5	2	2

VAE는 전문가 시연에서의 RGB 영상을 이용하여 학습되며 다음과 같은 손실함수 L_{VAE}를 통해 학습한다.

$$L_{\mathrm{VAE}} = -E_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{o})}(\log p(\boldsymbol{o}|\boldsymbol{z})) + \beta D_{\mathrm{KL}}(q(\boldsymbol{z}|\boldsymbol{o}) \parallel p(\boldsymbol{z})) \quad (1)$$

여기서 $\boldsymbol{o}$는 입력 영상, $\boldsymbol{z}$는 잠재벡터, q와 p는 인코더와 디코더, β는 두 항의 균형을 잡기 위한 계수이다. L_{VAE}의 첫째 항은 잠재벡터로부터 디코더가 복원한 영상이 입력 영상과 유사할 가능성을 높이는 역할을 한다. 둘째 항인 p과 q의 역쿨벡-라이블러 발산(reversed Kullback-Leibler divergence)은 인코더가 잠재벡터를 압축할 때 잠재벡터가 입력 영상의 특징을 잘 표현하도록 한다.

2.2 Behavioral cloning

SEGD에서는 전문가의 시연을 직접 이용하지 않고 전문가의 시연을 학습한 BC를 이용하여 SRE와 IPD에 활용한다. 여기서 전문가 시연은 RGB 영상, 로봇의 행동, 2지 그리퍼 개폐 여부, 작업 종료 여부 등에 대한 정보를 포함하고 있는데, BC는 시연의 RGB 영상을 압축한 잠재벡터로부터 행동을 출력하는 방법을 지도학습을 통해 학습한다. 이때 BC는 3층의 완전 연결층(fully-connected layer)으로 구성된 MLP (multilayer perceptron)을 사용하며, 각 층은 512개의 노드로 구성된다. 또한, BC의 손실함수 L_{BC}는 다음과 같이 MSE 함수를 이용하여 정의한다.

$$L_{\mathrm{BC}} = \mathrm{E}_{\boldsymbol{z},\boldsymbol{a}\sim \boldsymbol{D}_d}\left[(\boldsymbol{a} - \pi_{\mathrm{BC}}(\boldsymbol{z}))^2\right] \quad (2)$$

여기서 $\boldsymbol{z}$는 잠재벡터, $\boldsymbol{a}$는 정규화된 전문가의 행동, $\boldsymbol{D}_d$는 전문가의 시연 데이터셋이다. 이 L_{BC}를 최소화함으로써, BC의 행동은 전문가 시연의 행동과 유사해진다.

2.3 Task success discriminator (TSD)

SEGD에서는 작업의 성공 여부를 판단하기 위해 TSD를 사용한다. TSD는 RGB 영상을 압축한 잠재벡터를 기반으로 작업 성공 여부를 판단하며, 성공 시에는 1을 출력하고 실패 시에는 0을 출력한다. TSD는 BC와 동일한 구조의 MLP를 사용하며, BC와 마찬가지로 전문가 시연의 RGB 영상을 압축한 잠재벡터와 시연의 작업 완료 여부를 통해 학습된다. 이때 TSD는 다음과 같이 Binary Cross-Entropy (BCE)를 손실함수 L_{TSD}로 사용한다.

$$L_{\mathrm{TSD}} = -\mathrm{E}_{\boldsymbol{z},y\sim \boldsymbol{D}}\left[y\log(\mathrm{p}(\boldsymbol{z})) + (1-y)\log(1-\mathrm{p}(\boldsymbol{z}))\right] \quad (3)$$

여기서 $\boldsymbol{z}$는 잠재벡터, y는 작업의 완료 여부, $\boldsymbol{D}$는 리플레이 버퍼이다. TSD의 손실함수를 최소화함으로써, TSD가 판단하는 작업 완료 여부가 정답에 근접하게 된다.

2.4. Sparse reward explorer (SRE)

SEGD에서는 강화학습에 필요한 사전 데이터를 수집하기 위해 SRE를 이용한다. 사전 데이터를 수집할 때, 작업의 성공을 통해 보상이 발생되어야 SEGD의 평가자(critic) 학습이 수행된다. 이를 확인하기 위해 엔트로피-정규화된(entropy-regularized) 상태-행동 가치함수 Q^{π}의 재귀 벨만 방정식(recursive Bellman equation)을 살펴보자. 평가자의 현재 정책에 대한 Q^{π}는 다음과 같다.

$$Q^{\pi}(\boldsymbol{s},\boldsymbol{a}) = \mathrm{E}_{\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}'\sim \boldsymbol{D}}\left[R(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}') + \gamma V^{\pi}(\boldsymbol{s}')\right] \quad (4)$$
$$\text{where } V^{\pi}(\boldsymbol{s}') = \mathrm{E}_{\boldsymbol{a}'\sim \pi}\left[Q^{\pi}(\boldsymbol{s}',\boldsymbol{a}') + \alpha H(\pi(\cdot|\boldsymbol{s}'))\right]$$

여기서 $\boldsymbol{a}'$는 $\boldsymbol{s}'$를 기반으로 현재 정책으로부터 얻은 다음 행동이고, $\boldsymbol{D}$는 리플레이 버퍼이다. 또한, R은 보상함수, H는 엔트로피, 그리고 π는 현재 정책이다. 이때 리플레이 버퍼에 있는 보상들이 모두 0이라면 Q^{π}는 학습이 이루어지지 않는다. 이로 인해 평가자도 IPD의 행동에 대한 정확한 가치 평가를 할 수 없으므로, IPD의 학습이 수행되지 않는다. 따라서 강화학습을 진행하기 전에 SRE를 통하여 작업이 성공했을 때의 데이터를 생성함으로써 보상이 발생한 전이를 얻는 과정이 필요하다.

SRE는 작업이 성공했을 때의 데이터를 수집하기 위해 BC를 이용한다. 그러나 BC만을 이용하면 편향된 데이터가 수집되므로, 수집된 데이터 분포 이외의 행동들에 대한 잘못된 가치 판단을 하는 부트스트래핑(bootstrapping) 오류가 발생한다[14]. 이런 문제를 억제하기 위해서 SRE를 통해 수집한 데이터의 분포가 IPD가 출력하는 행동의 분포와 유사해야 한다. BC가 출력하는 행동의 분포를 IPD가 출력하는 행동의 분포와 유사하게 하기 위하여 BC를 무작위화한 randomized behavioral cloning (RBC) π_{RBC}를 다음과 같이 제안한다.

$$\pi_{\mathrm{RBC}}(\boldsymbol{z}) = 2\epsilon\pi_{\mathrm{BC}}(\boldsymbol{z}) \text{ where } \epsilon \sim \mathrm{U}(0,1) \quad (5)$$

여기서 π_{BC}는 BC, $\boldsymbol{z}$는 잠재벡터, ε은 [0, 1]의 균일 분포에서 추출한 값이다.

2.5 Soft actor-critic (SAC)

SEGD에서는 강화학습 알고리즘으로 SAC를 이용한다. SAC는 수집한 데이터를 효율적으로 사용하는 off-policy 방식으로 행동이 확률적으로 추출되는 확률적인 정책(stochastic

policy)을 최적화하는 알고리즘이다. 이때 SAC가 추구하는 최적의 정책 π^*는 다음과 같다.

$$\pi^* = \arg\max_{\pi} \mathrm{E}_{\boldsymbol{\tau}\sim\pi} \left[\sum_{t=0}^{\infty} \gamma^t \left(R(\boldsymbol{z}_t, \boldsymbol{a}_t, \boldsymbol{z}_{t+1}) + \alpha H(\pi(\cdot|\boldsymbol{z}_t))\right)\right] \quad (6)$$

여기서 $\boldsymbol{\tau}$는 경로, γ는 감쇠비, $\boldsymbol{z}$은 잠재벡터, $\boldsymbol{a}$는 SAC의 행동, α는 엔트로피 계수, H는 엔트로피, π는 정책이다. 보상함수 R은 정책이 높은 보상을 얻도록 유도하고, 엔트로피 H는 정책의 엔트로피를 최대화하여 정책이 탐험을 지향하도록 한다. 다만, 더 이상 탐험이 필요하지 않다고 판단되면, 엔트로피 계수 α를 조절함으로써 정책의 탐험을 지양할 수 있다.

2.5.1 Gaussian policy (GP)

SAC는 정책으로 다음과 같은 GP를 사용한다.

$$\pi_{\mathrm{GP}}(\boldsymbol{z}) = \tanh\left(\mu(\boldsymbol{z}) + \sigma(\boldsymbol{z}) \odot \epsilon\right) \text{ where } \epsilon \sim \mathrm{N}(0, \boldsymbol{I}) \quad (7)$$

여기서 $\boldsymbol{z}$은 상태, μ와 σ는 잠재 벡터 $\boldsymbol{z}$를 기반으로 구한 평균과 분산, ϵ는 가우시안 잡음이고, $\odot$는 두 벡터를 곱하는 하다마드 곱셈(Hadamard product)이다. GP의 행동은 분산에 잡음을 곱하고 평균을 더하는 재매개화 기법(reparameterization trick)을 통해 특정되고, 하이퍼볼릭 탄젠트 함수를 거쳐 [-1, 1]의 범위로 변환된다. 이때 사용되는 잡음으로 인하여 GP는 μ와 σ를 평균과 분산으로 가지는 가우시안 분포 내에서 임의의 행동을 추출하므로, 이 행동을 통해 탐험을 할 수 있다. 한편, SAC 학습이 진행될수록 행동의 분산은 점점 작아지므로, 학습이 진행될수록 탐험이 억제된다.

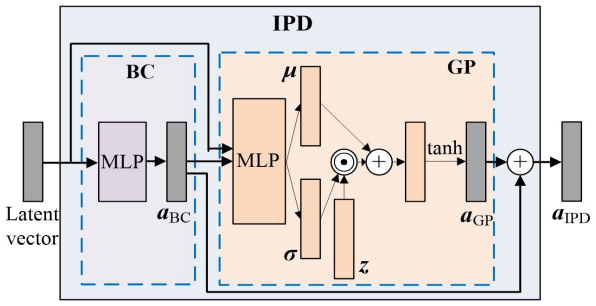
2.6 Interpolated policy using demonstration (IPD)

SEGD에서는 SAC의 정책으로 GP 대신 IPD를 사용하여 전문가와 유사한 행동을 하는 탐험을 진행한다. IPD는 VAE를 통해 구한 잠재벡터를 기반으로 로봇의 말단 위치 변화량이나 그리퍼 상태를 추정하고, 추정된 변화량을 로봇의 제어에 반영한다. 이때 π_{IPD}는 다음과 같다.

$$\pi_{\mathrm{IPD}}(\boldsymbol{z}) = \pi_{\mathrm{GP}}(\boldsymbol{\kappa}) + \alpha\pi_{\mathrm{BC}}(\boldsymbol{z}) \quad \text{where } \boldsymbol{\kappa} = \boldsymbol{z} \oplus \pi_{\mathrm{BC}}(\boldsymbol{z}) \quad (8)$$

여기서 $\boldsymbol{z}$는 잠재벡터, $\boldsymbol{\kappa}$는 $\boldsymbol{z}$와 π_{BC}를 합한(concatenate) 벡터, α는 두 항의 균형을 위한 계수이다.

IPD의 구조는 [Fig. 2]와 같다. IPD의 입력은 영상의 정보가 압축된 잠재벡터이다. BC는 이 잠재벡터로부터 BC의 행동 $\boldsymbol{a}_{\mathrm{BC}}$를 예측한다. GP는 이 잠재벡터와 $\boldsymbol{a}_{\mathrm{BC}}$를 기반으로 GP의 행동 $\boldsymbol{a}_{\mathrm{GP}}$을 추출한다. 마지막으로, IPD는 BC의 행동과 GP의 행동을

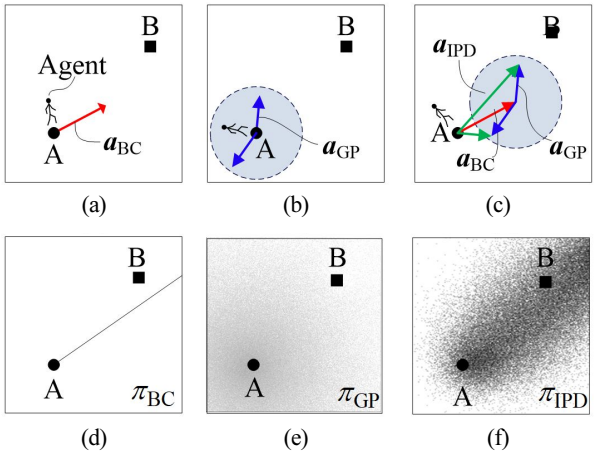


[Fig. 2] Structure of IPD

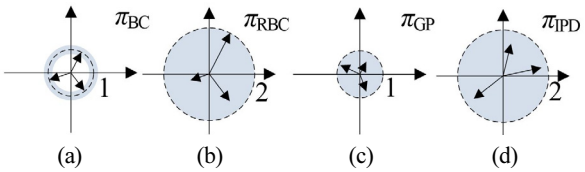
더하여 IPD의 행동인 $\boldsymbol{a}_{\mathrm{IPD}}$를 추출한다. 이 $\boldsymbol{a}_{\mathrm{IPD}}$는 로봇에 전달되어 로봇의 말단 위치나 그리퍼를 제어하는 데 이용된다.

2.6.1 IPD의 특징

IPD는 BC로 인해 전문가의 행동과 유사한 행동들을 추출하므로, 순수한 GP에 비해 효율적인 탐험을 수행한다. 이를 확인하기 위해 [Fig. 3]과 같은 2차원 공간에서 점 A에서 시작하여 점 B로 가는 상황을 살펴보자. [Fig. 3]의 (a), (b), (c)는 각각 BC, GP, IPD의 행동을 시각화한 그림이다. BC는 [Fig. 3]의 (a)와 같이 정확하진 않지만, 목적지인 점 B를 향하는 행동을 추출한다. GP는 [Fig. 3]의 (b)와 같이 목적지인 점 B를 향하는 방향과 상관없는 무작위 방향의 행동을 추출한다. 여기서 BC는 동일한 입력에 대해 같은 행동을 출력하지만, GP는 동일한 입력에 대해 다른 행동을 출력한다. 이는 GP는 주어진 가우시안 분포에서 행동을 추출하기 때문이다. IPD의 행동은 [Fig. 3]의 (c)와 같이 BC의 행동과 GP의 행동을 더한 것이므로, 항상 다른 행동을 출력한다. 이때, BC는 점 B를 향해 이동하려고 하는 성질을 가지고 있으므로, IPD는 점 B를 향해 이동하면서 탐험을 수행할 수 있다.



[Fig. 3] Visualizing the problem of reaching B starting from A in a 2D space with the agent: (a) $\boldsymbol{a}_{\mathrm{BC}}$ at A, (b) a range of $\boldsymbol{a}_{\mathrm{GP}}$ at A, and (c) $\boldsymbol{a}_{\mathrm{IPD}}$ at A. A visitation map of (d) π_{BC}, (e) π_{GP}, and (f) π_{IPD}



[Fig. 4] Distributions of (a) π_{BC}, (b) π_{RCB}, (c) π_{GP} and (d) π_{IPD}

[Fig. 3]의 (d) ~ (f)는 BC, GP, 그리고 IPD를 통하여 에이전트가 5,000 에피소드 동안 지나간 곳을 표시한 방문 지도(visitation map)이다. (d)는 BC를 사용한 경우의 방문 지도이며, 항상 동일한 방향으로 움직이는 것을 볼 수 있다. (e)는 GP를 사용한 경우의 방문 지도이며, GP가 시작 지점으로부터 사방으로 넓은 범위를 탐색하는 것을 볼 수 있다. 반면에, (f)는 IPD를 사용하는 경우이며, IPD가 점 B를 향하는 방향을 집중적으로 탐색하는 것을 볼 수 있다. 이로 인해 IPD는 전문가의 시연과 유사하지만 탐험을 시도하는 성질을 가지므로, 일반적인 강화학습에 비해 탐험하는 공간이 작다. 이로 인해 강화학습에 필요한 학습 시간이 단축된다.

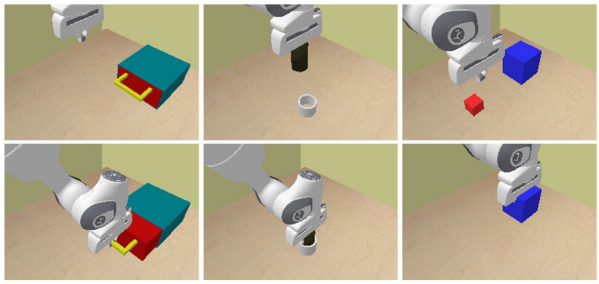
2.6.2 IPD와 RBC의 관계

한편, SRE에서는 부트스트래핑 오류를 억제하기 위해 IDP와 출력하는 행동의 분포와 유사한 RBC를 사용하여 사전 데이터를 수집한다. RBC가 출력하는 행동의 분포가 IPD가 출력하는 행동의 분포와 유사함을 확인하기 위해 BC, RBC, GP와 IPD가 출력하는 행동의 분포를 [Fig. 4]와 같이 시각화하여 비교하였다. 행동 분포는 파란색 영역으로 표현하였으며, 그 영역에서 추출된 하나의 행동을 검은 화살표로 나타냈다. BC는 길이가 1인 행동을 출력하므로 (a)와 같이 반지름이 1인 원의 둘레의 행동 분포를 가진다. RBC의 행동은 BC와 동일한 방향을 갖지만, 그 크기가 0에서 2이다. 따라서 (b)와 같은 반지름이 2인 원 형태의 행동 분포를 가진다. GP의 행동은 원점으로부터 길이가 1이내인 벡터를 추출하므로, (c)와 같이 반지름이 1인 원의 행동 영역을 가진다. IPD의 행동은 BC의 행동 (a)와 GP의 행동 (c)를 합하므로, (d)와 같은 반지름이 2인 원의 행동 영역을 가진다. 이를 통해 (b)와 같은 RBC가 출력하는 행동의 분포가 (d)와 같은 IPD가 출력하는 행동의 분포와 유사함을 확인할 수 있다.

3. 실험 및 결과

3.1 SEGD 실험 환경

SEGD의 성능을 검증하기 위하여 [Fig. 5]와 같이 물체 조작이 가능한 가상환경을 구성하였는데, 시뮬레이터는 PyBullet



[Fig. 5] Start (top) and end (bottom) of the experimental environments of (a) opening drawer, (b) peg-in-hole, and (c) pick-and-place

를 사용하였으며, 로봇은 2지 그리퍼가 부착된 Franka Emika의 Panda 로봇을 사용하였다. 이 가상환경을 바탕으로 서랍 열기(opening a drawer), 펙인홀(peg-in-hole), 그리고 집기-놓기(pick-and-place) 작업을 수행하여 실험을 수행하였다. 각 실험에서 로봇의 말단과 모든 물체는 에피소드 시작 시, 20cm × 20cm × 20cm 공간 내의 임의의 위치에 배치된다. 또한, 로봇 말단의 이동 거리는 각 스텝마다 최대 1cm로 제한하였다.

서랍 열기 작업은 [Fig. 5(a)]와 같은 환경에서 로봇과 그리퍼를 조작하여 서랍을 10cm 이상 여는 것을 목표로 한다. 이러한 환경에서 50개의 전문가의 시연을 통해 BC를 학습하였고, 학습된 BC는 IDP에 결합된다. 이와 같은 정책함수로 매 스텝마다 로봇의 말단 위치와 그리퍼를 조작한다. 이때, 에피소드는 작업을 완료하면 종료되며, 최대 스탭은 100으로 설정하였다

펙인홀 작업은 [Fig. 5(b)]와 같은 환경에서 로봇을 조작하여 그리퍼 대신 로봇 말단에 부착된 펙을 공차가 5mm인 홀에 5cm 이상 삽입하는 것을 목표로 한다. 이러한 환경에서 50개의 전문가의 시연을 통해 BC를 학습하였고, 학습된 BC는 IDP에 결합된다. 이와 같은 정책 함수로 매 스텝마다 로봇의 말단 위치를 조작한다. 실험의 에피소드 설정 조건은 서랍 열기 작업과 동일하다.

집기-놓기 작업은 [Fig. 5(c)]와 같은 환경에서 로봇과 그리퍼를 조작하여 임의의 위치에 존재하는 빨간 정육면체를 집어 파란 공간 안에 넣는 것을 목표로 한다. 이러한 환경에서 200개의 전문가의 시연을 통해 BC를 학습하였고, 학습된 BC는 IDP에 결합된다. 이와 같은 정책함수로 매 스텝마다 로봇의 말단 위치와 그리퍼를 조작한다. 이때, 에피소드의 최대 스텝은 200으로 설정하였다.

각 실험 환경에서 학습은 다음과 같은 순서로 진행되었다. 먼저 가상환경에 Panda 로봇과 대상 물체를 배치하여 실험환경을 구축하였다. 이후 가상환경 상의 모든 물체의 위치를 알 수 있으므로, 이에 기반하여 사람이 특정 작업을 수행하는 전략을 수립하고, 이를 프로그램으로 구현하였다. 집기-놓기 작업을 예로 들어보자. 이 작업을 위해서는 로봇이 빨간 물체의

중심점과 그리퍼의 중심점이 일치하는 위치로 움직인 후, 빨간 물체를 파지하여 파란 공간 안으로 이동하여 빨간 물체를 놓아야 한다. 이와 같은 일련의 동작을 사람이 직접 전략을 수립하여 프로그래밍하는 것이다. 이와 같이 구성된 알고리즘을 통해 작업을 수행하는 도중에 시연 데이터를 100ms 단위로 수집하였다. 이때 수집된 시연 데이터는 영상, 로봇의 말단 위치 변화량, 그리퍼의 개폐 여부, 그리고 작업 완료 여부로 구성되어 있다. 다음으로, 이와 같이 수집된 시연 데이터의 영상을 기반으로 VAE를 학습하였으며, 학습된 VAE의 엔코더를 통해 추정한 시연 영상의 잠재벡터, 로봇의 말단 위치 변화량, 그리고 그리퍼의 개폐 여부를 기반으로 BC를 학습하였다. 또한, 학습된 VAE의 엔코더를 통해 구한 시연 영상의 잠재벡터와 작업 완료 여부를 기반으로 TSD를 학습하였다. 수집된 시연 데이터를 통해 VAE, TSD, 그리고 BC의 학습을 충분히 진행한 후, SRE를 이용하여 IPD의 학습에 필요한 사전 데이터를 수집하였다. 이때 SRE는 200 에피소드의 RGB 영상, 로봇의 말단 위치 변화량, 그리퍼의 상태, 그리고 희소 보상 등을 리플레이 버퍼에 저장하였다. 마지막으로, SEGD는 해당 작업을 반복하여 강화학습을 진행한다.

3.2 타 알고리즘과의 비교실험 및 결과

SEGD의 성능을 검증하기 위해 강화학습의 정책이 예측한 로봇의 행동과 시연 데이터에서의 행동 간의 차이에 대한 보상과 작업 완료 시 발생하는 희소 보상의 선형 결합으로 구성된 보상함수를 이용하는 deepmimic과 SEGD를 [Table 2]와 같은 3가지 작업에 비교 실험을 진행하였다[4,6]. 각 작업별로 4번씩 강화학습을 수행하여 4개의 서로 다른 정책을 얻었으며, 이 4개의 정책 각각에 대해 작업별로 50번씩 실험하여 작업 성공률을 얻었다.

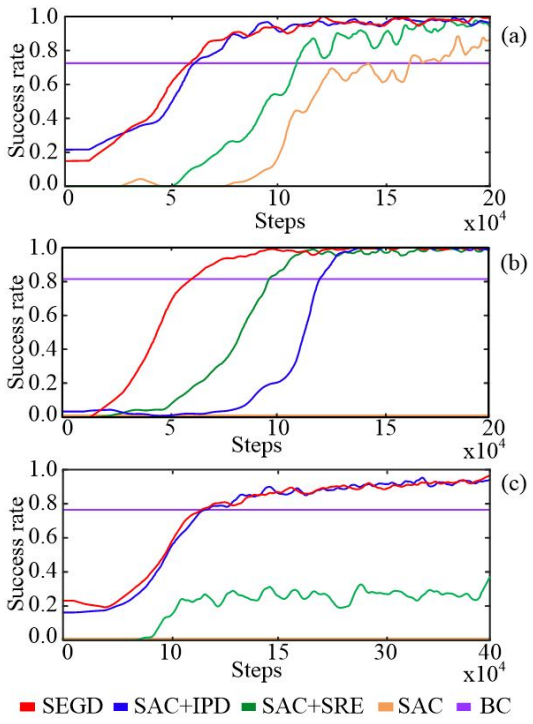
[Table 2]를 통해 모든 작업에서 deepmimic의 작업 성공률보다 SEGD이 높게 나타남을 알 수 있다. 이와 같은 결과는 시연을 모방하는 방식의 차이에 의해 발생한다. Deepmimic의 보상함수는 시연과 정책의 유사성에 대한 보상을 가지고 있는데, 이로 인해 정책의 출력은 학습이 진행될수록 시연과 유사해진다. 이는 학습 초기에는 정책의 학습에 긍정적인 영향을 주지만, 학습이 어느 정도 진행된 후에는 정책이 최적의 정책을 찾을 수 없도록 하는 효과가 있다. 따라서 최종적으로는 최적에 근접하지만 최적은 아닌 근접 최적(near-optimal) 정책을 학습하게 된다. 반면에, SEGD의 경우 희소 보상만을 사용하였으므로 전문가의 시연보다 작업 성공에 집중한다. 따라서 작업의 성공률을 최대한 끌어올릴 수 있다. 이와 같은 차이로 인해 결과적으로 deepmimic보다 SEGD가 더 높은 작업 성공률을 달성한 것으로 보인다.

[Table 2] Task success rates of deepmimic and SEGD

Algorithms	Tasks		
	Opening drawer	*Peg-in-hole*	*Pick-and-place*
Deepmimic[4]	87.0%	98.5%	57.5%
SEGD	99.0%	99.5%	96.5%

3.3 IPD와 SRE의 실험 및 결과

IPD와 SRE가 SEGD에 끼치는 영향을 확인하기 위해 3가지 작업에 대해 SAC에 IPD와 SRE를 다양하게 조합해 총 4가지 알고리즘으로 실험하였다. 4가지 조합은 1) 순수 SAC, 2) SRE만 적용한 SAC+SRE, 3) IPD만 적용한 SAC+IPD, 4) IPD와 SRE 모두 적용한 SEGD이다.



[Fig. 6] Success rates of SEGD, SAC+IPD, SAC+SRE and SAC for (a) opening drawer, (b) peg-in-hole, and (c) pick-and-place

[Table 3] Task Success rates

Algorithms	Tasks		
	Opening drawer	*Peg-in-hole*	*Pick-and-place*
SAC	85.0%	0.0%	0.0%
SAC+ SRE	98.0%	98.0%	37.0%
SAC+ IPD	98.5%	99.0%	94.0%
SEGD	99.0%	99.5%	96.5%
BC	72.5%	81.0%	76.5%

실험에 대한 학습 결과는 [Fig. 6], 실험에 대한 결과는 [Table 3]에 나타내었다. [Fig. 6]의 x축은 스텝이며, 이는 강화학습 알고리즘이 학습을 진행한 횟수이다. [Fig. 6]의 y축은 작업 성공률이며, 이는 3.2의 실험과 동일한 방식으로 얻었다. 순수 SAC의 작업 성공률이 가장 낮았고, IPD와 SRE를 모두 적용한 SEGD의 작업 성공률이 가장 높았다. IPD와 SRE 중 하나만 적용한 경우에는 작업의 특성에 따라 결과가 다르게 나타났다.

SAC의 경우 3가지 작업에서 모두 제일 낮은 작업 성공률을 달성하였다. 이는 SAC가 임의의 움직임을 통한 탐험으로 작업의 성공을 경험하기 어려워, 보상을 포함한 전이를 얻기 어렵기 때문이다. 이에 따라 펙인홀 및 집기-놓기 작업은 무작위 움직임으로는 성공하기 힘든 작업이므로, 작업 성공률이 0%로 나타났다. 그러나 서랍 열기 작업은 비교적 쉬운 작업에 속하므로, 임의의 움직임을 통하여서도 간혹 작업이 성공할 수 있다. 따라서 서랍 열기 작업의 경우에 SAC는 작업 성공률이 85%로 나타났다.

IPD나 SRE 중 하나만 적용한 경우는 SAC만 사용한 경우보다 높은 작업 성공률을 달성하였다. 이는 SAC와는 다르게, 작업의 성공으로 발생한 희소 보상으로 인해 평가자가 학습되었기 때문이다. 서랍 열기 및 펙인홀 실험에서는 IPD만을 적용한 경우와 SRE만을 적용한 경우가 성공률이 비슷하게 나타나지만, 집기-놓기 실험에서는 IPD만을 적용한 경우가 성공률이 더 높게 나타난다. 이러한 차이는 한 번의 잘못된 그리퍼의 조작이 작업 실패를 야기하는 집기-놓기 작업의 특성으로 인하여 발생한다. SAC+SRE의 경우, GP로 인하여 잘못된 행동이 자주 발생한다. 반면에, SAC+IPD의 경우에는 IPD로 인하여 BC의 행동에서 크게 벗어나는 행동을 하는 빈도가 적으므로 잘못된 행동이 적게 발생한다.

IPD와 SRE를 모두 적용한 SEGD는 모든 작업에서 가장 높은 작업 성공률을 달성하였다. SRE를 통하여 학습에 필요한 희소 보상을 충분히 수집할 수 있으며, IPD를 통하여 전문가의 시연과 유사한 행동을 기반으로 학습을 가속하기 때문이다. IPD의 학습 가속 효과는 네 알고리즘의 학습 그래프인 [Fig. 6]을 통하여 확인할 수 있다. [Fig. 6]의 (a)와 (b)에서 SVGD와 SAC+SRE는 IPD를 사용한 경우와 사용하지 않은 경우이다. 이 두 알고리즘에 대해 BC의 성능에 도달하는 데 걸린 스텝 수를 비교하면 각각 50%와 36% 줄어든 것을 확인할 수 있다.

4. 결 론

본 연구에서는 특별한 보상함수의 설계 없이 희소 보상만으로 주어진 작업을 학습하기 위해, 전문가 시연 기반의 물체 조작 알고리즘 SEGD를 제안하였다. SEGD는 기존의 SAC에 SRE와 IPD를 추가한 알고리즘이다. 여기서 SRE는 강화학습을 시작하기 전에 로봇을 직접 제어하여 강화학습에 필요한 사전 데이터를 수집하고, IPD는 전문가와 유사한 행동 후보들을 추출하여 탐험을 진행한다. 이로 인해 탐험 공간이 축소되어 학습 시간이 단축된다. SEGD의 성능을 검증하기 위해, 서랍 열기, 펙인홀 및 집기-놓기 작업에 대한 실험을 진행하였다. 이 실험에서 SEGD는 작업의 종류에 상관없이 96.5% 이상의 높은 성공률을 보임으로써 그 효용성을 입증하였다. SEGD는 보상함수를 설계할 필요가 없으며 안정적인 학습이 가능하므로, 다양한 분야에 활용될 수 있을 것으로 기대한다.

References

[1] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," *2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, 2017, DOI: 10.1109/ICRA.2017.7989385.

[2] A. Y. Ng and S. Russell, "Algorithms for inverse reinforcement learning," *17th International Conference on Machine Learning (ICML)*, 2000, [Online], https://ai.stanford.edu/~ang/papers/icml00-irl.pdf.

[3] G. Schoettler, A. Nair, J. Luo, J. Luo, S. Bahl, J. A. Ojea, E. Solowjow, and S. Levine, "Deep reinforcement learning for industrial insertion tasks with visual inputs and natural rewards," *International Conference on Intelligent Robots and Systems*, 2019, [Online], https://arxiv.org/pdf/1906.05841.pdf.

[4] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM SIGGRAPH*, 2018, [Online], https://dl.acm.org/doi/pdf/10.1145/3197517.3201311.

[5] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations," *arXiv:1709.10087v2 [cs.LG]*, 2018, [Online], https://arxiv.org/pdf/1709.10087.pdf.

[6] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, A. Sendonaris, G. Dulac-Arnold, I. Osband, J. Agapiou, J. Z. Leibo, and A. Gruslys, "Learning from demonstrations for real world reinforcement learning," *arXiv preprint arXiv:1704.03732*, 2017, [Online], https://openreview.net/forum?id=5AvKPhXBsz.

[7] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD, Australia, 2017, DOI: 10.1109/ICRA.2018.8463162.

[8] A. Singh, L. Yang, K. Hartikainen, C. Finn, and S. Levine, "End-toEnd Robotic Reinforcement Learning without Reward Engineering," *arXiv:1904.07854 [cs.LG]*, 2019. [Online], https://arxiv.org/pdf/1904.07854.pdf.

[9] P. Sermanet, K. Xu, and S. Levine, "Unsupervised perceptual rewards for imitation learning," *arXiv:1612.06699v3 [cs.CV]*, 2017, [Online], https://arxiv.org/pdf/1612.06699.pdf.

[10] L. Smith, N. Dhawan, M. Zhang, P. Abbeel, and S. Levine, "AVID: Learning Multi-Stage Tasks via Pixel-Level Translation of Human Videos," *arXiv:1912.04443v3 [cs.RO]*, 2019, [Online], https://arxiv.org/pdf/1912.04443.pdf.

[11] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," *Machine Learning Research*, pp. 1861-1870, 2018, [Online], https://proceedings.mlr.press/v80/haarnoja18b.

[12] S. Schaal, A. Ijspeert, and A. Billard, "Computational approaches to motor learning by imitation," *Philosophical Transactions: Biological Sciences*, 2003, DOI: 10.1098/rstb.2002.1258.

[13] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, "Understanding disentangling in beta-vae," *arXiv:1804.03599v1 [stat.ML]*, 2017, [Online], https://arxiv.org/pdf/1804.03599.pdf.

[14] A. Kumar, J. Fu, M. Soh, G. Tucker, and S. Levine, "Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction," *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 2019, [Online], https://proceedings.neurips.cc/paper/2019/hash/c2073ffa77b5357a498057413bb09d3a-Abstract.html.

김 두 준

2020 고려대학교 기계공학과(학사)

2020~현재 고려대학교 기계공학과 (석사과정)

관심분야: 로봇 제어, 모방학습

조 현 준

2016 고려대학교 기계공학과(학사)

2016~현재 고려대학교 기계공학과 (석박사통합과정)

관심분야: 로봇 제어, 딥 러닝

송 재 복

1983 서울대학교 기계공학과(공학사)

1985 서울대학교 기계공학과(공학석사)

1992 Mechanical Engineering, MIT, Cambridge(공학박사)

1993~현재 고려대학교 기계공학부 정교수

관심분야: 로봇의 설계 및 제어, 협동로봇, 중력보상 로봇, AI 기반 로봇 매니퓰레이션