Model-Based Reinforcement Learning with World Models

2025. 12. 05

정재우

Data Mining & Quality Analytics Lab.

Data Mining
Quality Analytics

발표자 소개

- **정재우(Jae woo Jung)**
  - 단국대학교 소프트웨어학과 학부 졸업 (2025.08)
  - 고려대학교 산업경영공학과 대학원 재학
  - 석사 과정 (2025.09 ~ present)

- **Research Interest**
  - Reinforcement Learning
  - Preference-based Reinforcement Learning

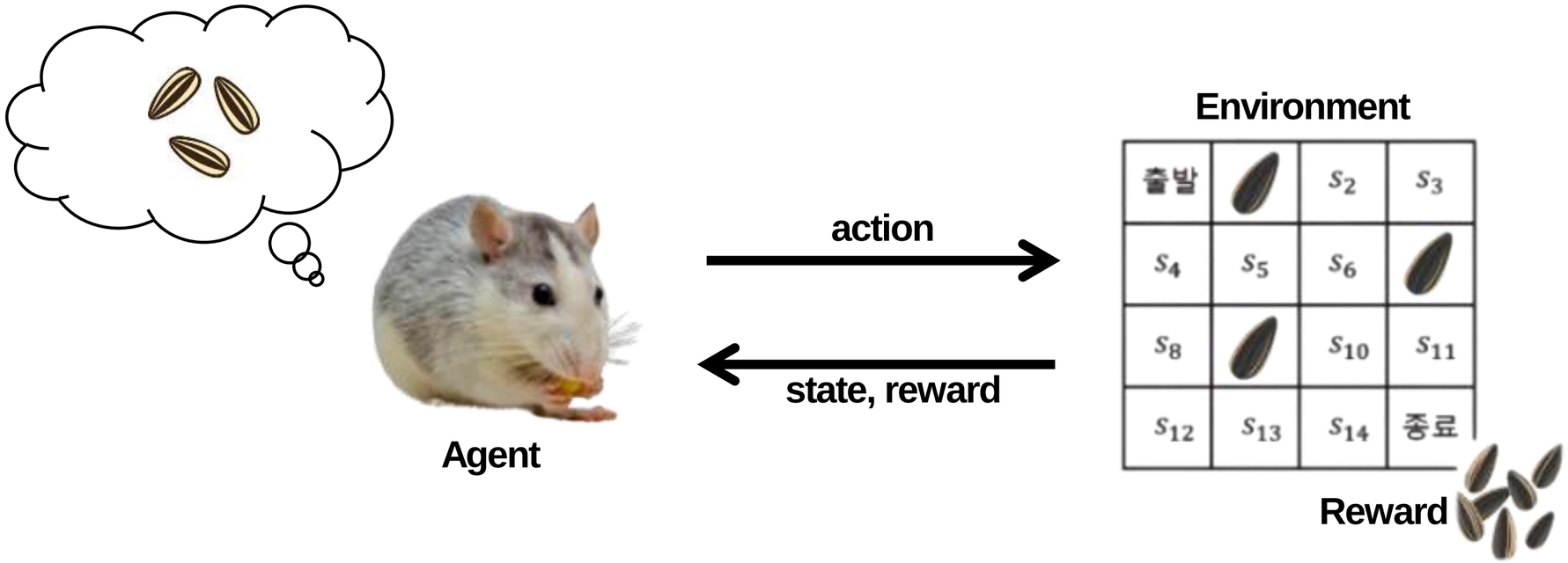
- **Contact**
  - **jjw0513@korea.ac.kr**

Contents

Introduction

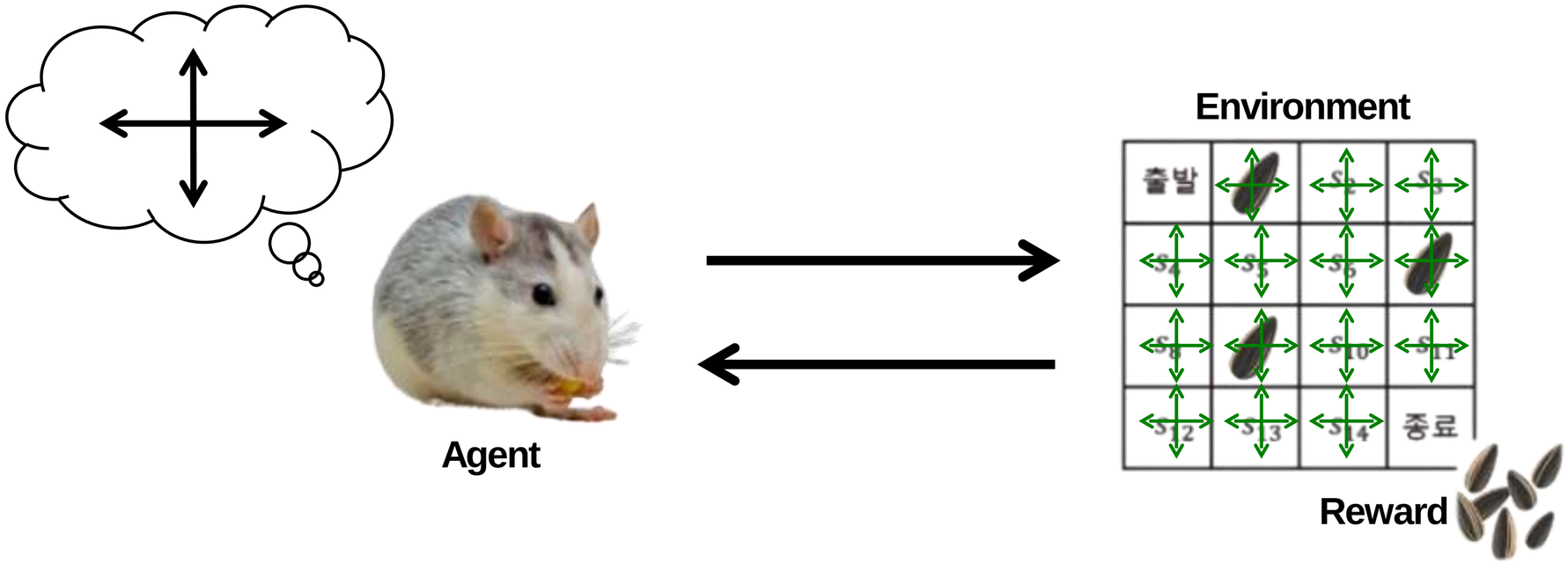
Reinforcement Learning

- **Reinforcement Learning**
  - 강화학습은 행동의 주체인 에이전트가 주어진 환경에서 시행착오를 겪으며 학습하는 방법.
  - 강화학습의 요소는 agent, state, action, reward



Introduction

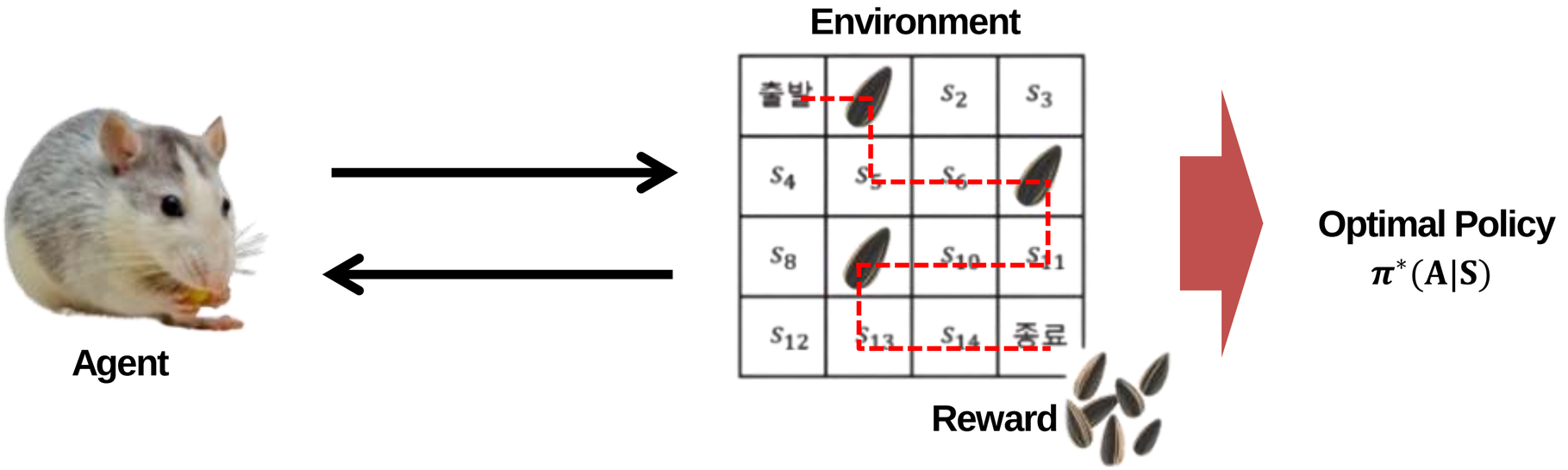
- **Reinforcement Learning**
  - 강화학습의 주요소는 agent, state, action, reward
  - 특정 상태에서 행동을 취하는 결정은 정책(policy)을 따름
  - 햄스터가 반복 경험을 통해 이득이 되는 행동을 취하고, 정책을 수립해감



Background

Reinforcement Learning

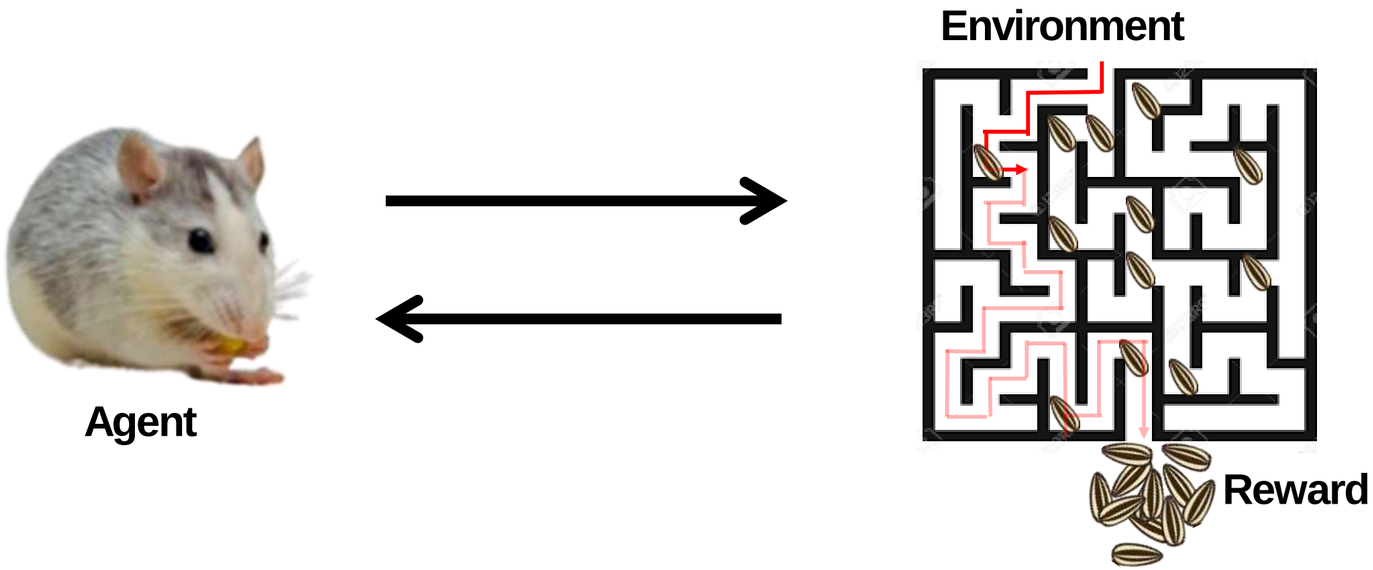
- **Reinforcement Learning**
  - 학습 목표 : 보상의 최대화
  - 보상의 최대화 = 최적의 행동 경로를 구하기



Background

Reinforcement Learning

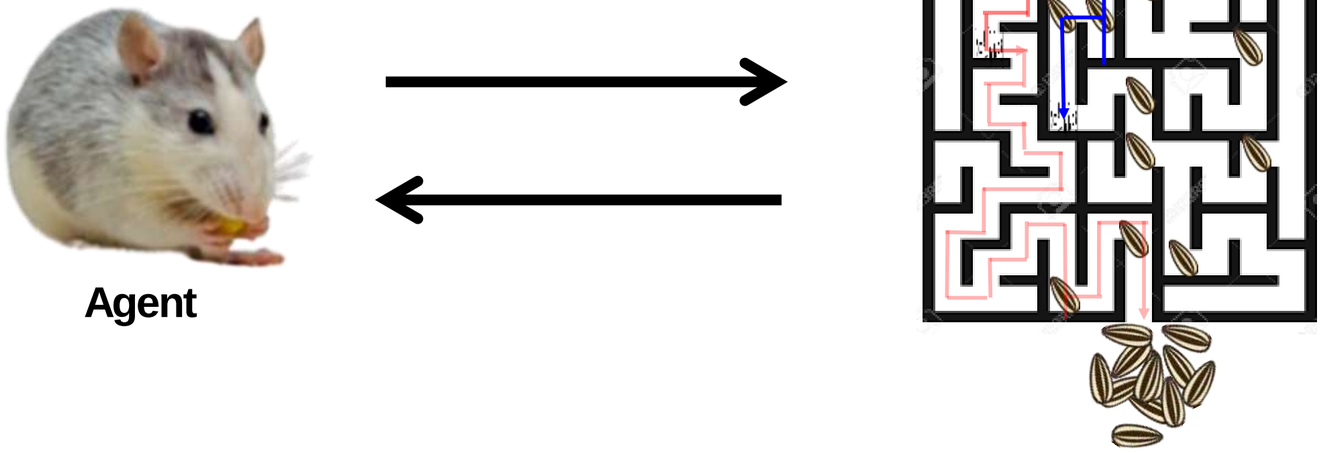
- **Reinforcement Learning**
  - 환경의 불확실성으로 정확한 누적 보상을 계산하는데 한계점이 존재
  - 즉각적 보상의 한계 : 단기적 보상에 집중하면 장기적 누적 보상을 극대화하지 못할 가능성 존재
  - Value function : 미래에 기대되는 보상들을 근사하여, 에이전트에게 감사된 보상 예측치를 제공
  - Value function 으로 상태나 행동에 따른 누적 기대 보상(가치)을 계산하여 평가 가능 → 정책 개선에 용이



Background

Reinforcement Learning

- **Reinforcement Learning**
  - 환경의 불확실성으로 정확한 누적 보상을 계산하는데 한계점이 존재
  - 즉각적 보상의 한계 : 단기적 보상에 집중하면 장기적 누적 보상을 극대화하지 못할 가능성 존재
  - Value function : 미래에 기대되는 보상들을 근사하여, 에이전트에게 감사된 보상 예측치를 제공
  - Value function 으로 상태나 행동에 따른 누적 기대 보상(가치)을 계산하여 평가 가능 → 정책 개선에 용이



$$Return: \ G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

$$State\ value\ function: V_\pi(s) = E_\pi[G_t | S_t = s]$$

$$Action\ value\ function: Q_\pi(s,a) = E_\pi[G_t | S_t = s, A_t = a]$$

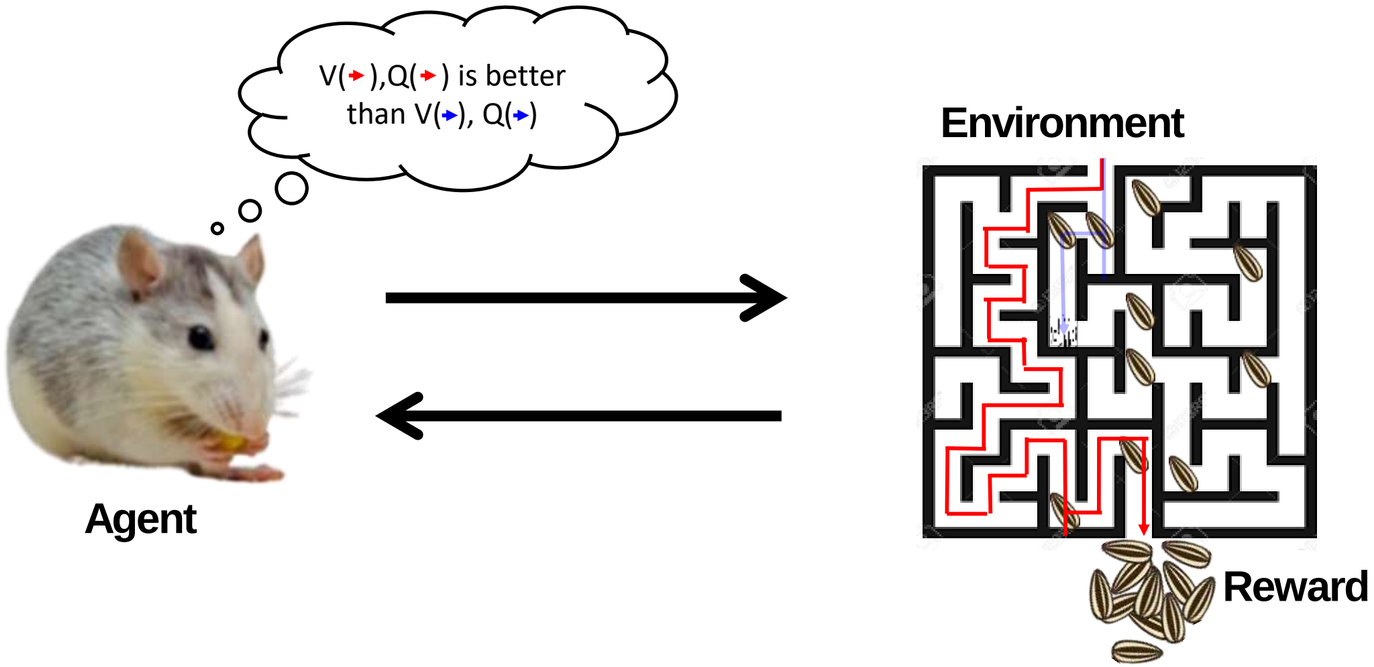
Bellman equation)

$$(1)\ V_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)[r + \gamma V_\pi(s')]$$
$$(2)\ Q_\pi(s,a) = \sum_{s',r} p(s',r|s,a)\left[r + \gamma \sum_{a'} \pi(a'|s') Q_\pi(s',a')\right]$$

Background

Reinforcement Learning

- **Reinforcement Learning**
  - 환경의 불확실성으로 정확한 누적 보상을 계산하는데 한계점이 존재
  - 즉각적 보상의 한계 : 단기적 보상에 집중하면 장기적 누적 보상을 극대화하지 못할 가능성 존재
  - Value function : 미래에 기대되는 보상들을 근사하여, 에이전트에게 감사된 보상 예측치를 제공
  - Value function 으로 상태나 행동에 따른 누적 기대 보상(가치)을 계산하여 평가 가능 → 정책 개선에 용이



$$Return: \ G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

$$State\ value\ function: V_\pi(s) = E_\pi[G_t | S_t = s]$$

$$Action\ value\ function: Q_\pi(s,a) = E_\pi[G_t | S_t = s, A_t = a]$$

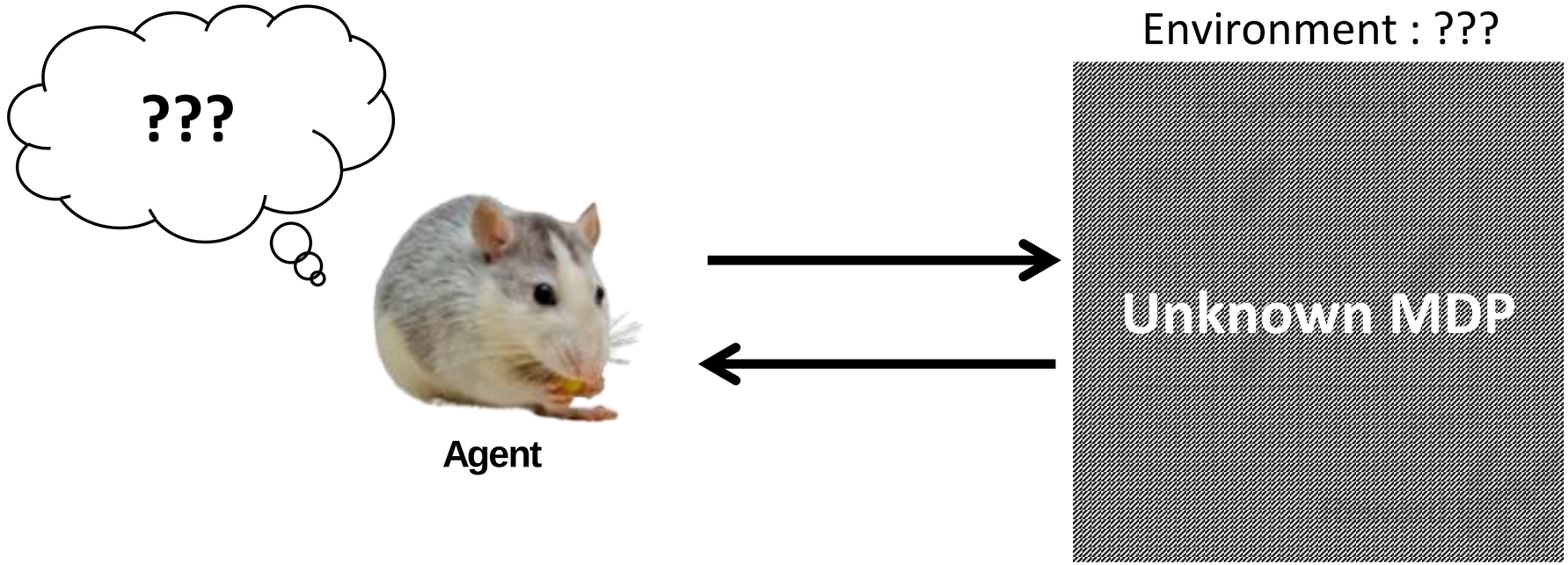
Bellman equation)

$$(1)\ V_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)[r + \gamma V_\pi(s')]$$
$$(2)\ Q_\pi(s,a) = \sum_{s',r} p(s',r|s,a)\left[r + \gamma \sum_{a'} \pi(a'|s') Q_\pi(s',a')\right]$$

Introduction

Reinforcement Learning

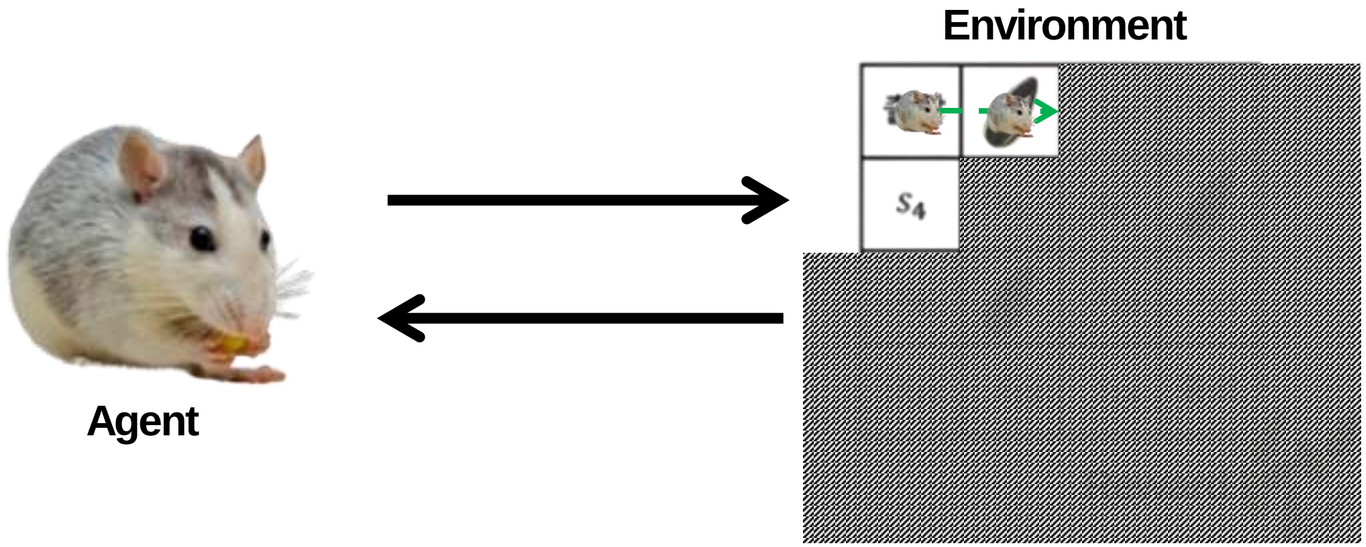
- **Model-Free RL**
  - 대부분의 강화학습 에이전트는 환경의 정보를 모름 → Unknown MDP
  - **에이전트는 알 수 없는 환경에서 무작위 행동을 취하며 초기 경험을 수집**



Introduction

Reinforcement Learning

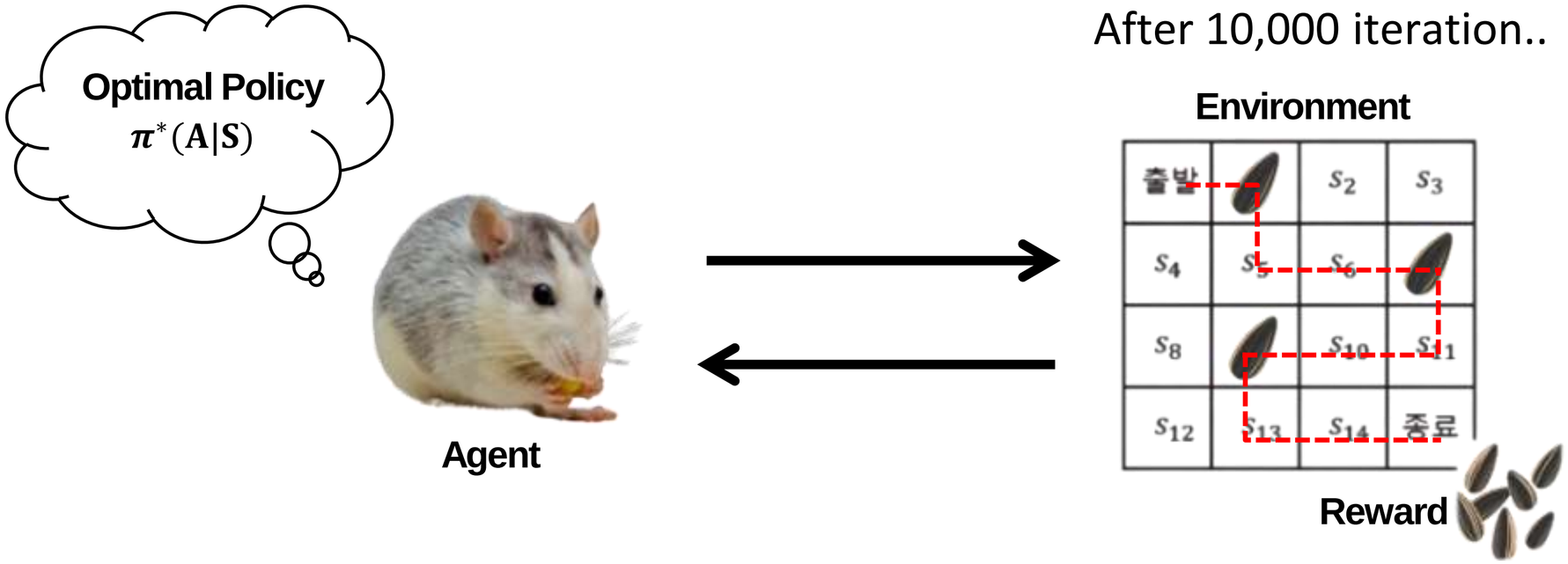
- **Model-Free RL**
  - 대부분의 강화학습 에이전트는 환경의 정보를 모름 -> Unknown MDP
  - **에이전트는 알 수 없는 환경에서 무작위 행동을 취하며 초기 경험을 수집**
  - **직접 환경에서 여러 행동을 취하며 경험을 누적**



Introduction

Reinforcement Learning

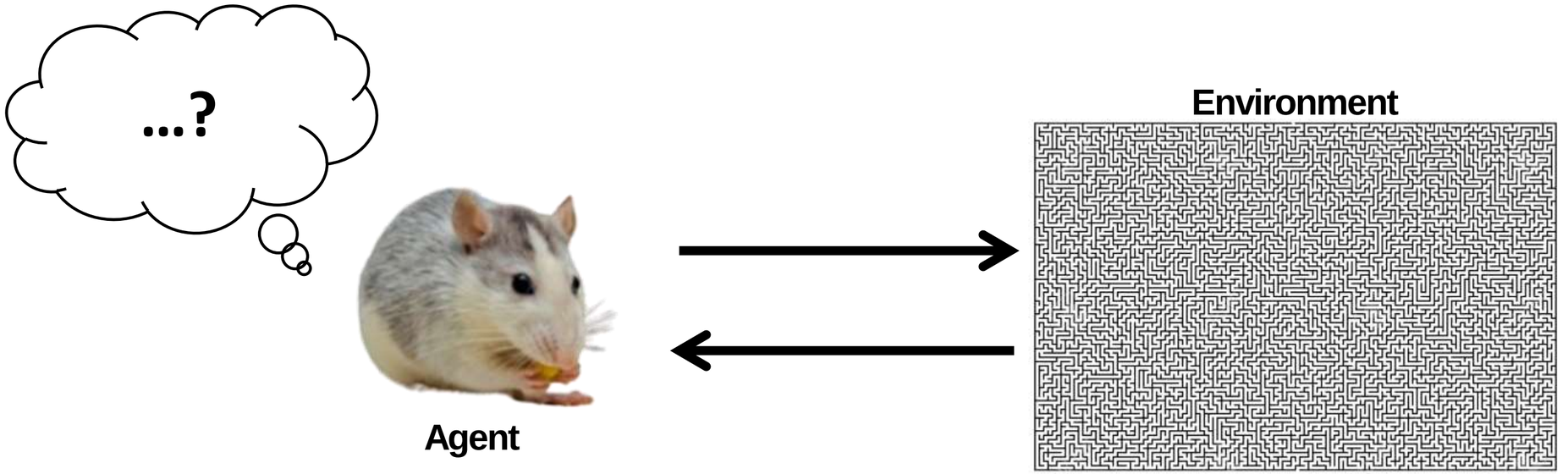
- **Model-Free RL**
  - 수많은 시도로 환경과 행동 간의 관계를 학습하며, 행동에 따른 기대 보상을 정확히 근사해감
  - 누적된 경험으로 최적의 정책을 달성 → 보상을 최대화하며 목표 달성
  - Model-Free RL 방법론



Introduction

Reinforcement Learning

- **Model-Free RL**
  - 하지만 환경이 매우 복잡하다면, 학습의 난이도 상승
  - **목표를 달성하거나 보상을 최대화하려면, 수없이 많은 환경과의 상호작용을 필요**
  - **학습 시간, 비용이 많이 들며 경험 데이터를 수많이 누적해야 한다는 단점 존재(데이터 효율성 저하)**

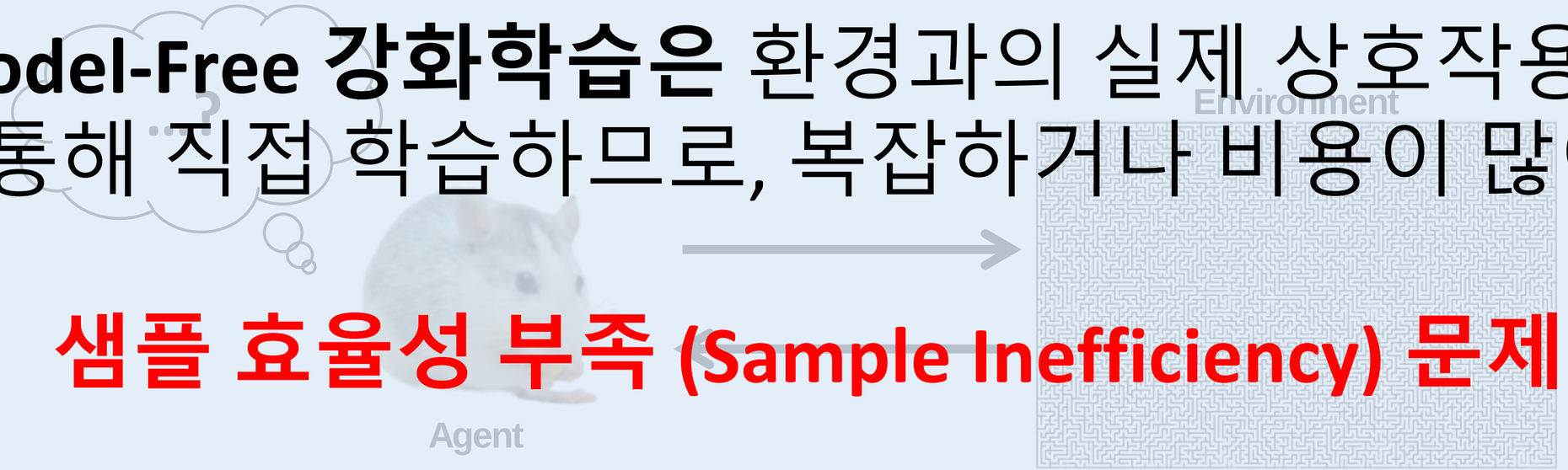


Introduction

Reinforcement Learning

- **Model-Free RL**
  - 하지만 환경이 매우 복잡하다면, 학습의 난이도 상승
  - 목표를 달성하거나 보상을 최대화하려면, 수없이 많은 환경과의 상호작용을 필요
  - 학습 시간, 비용이 많이 들며 경험 데이터를 수많이 누적해야 한다는 단점 존재(데이터 효율성 저하)

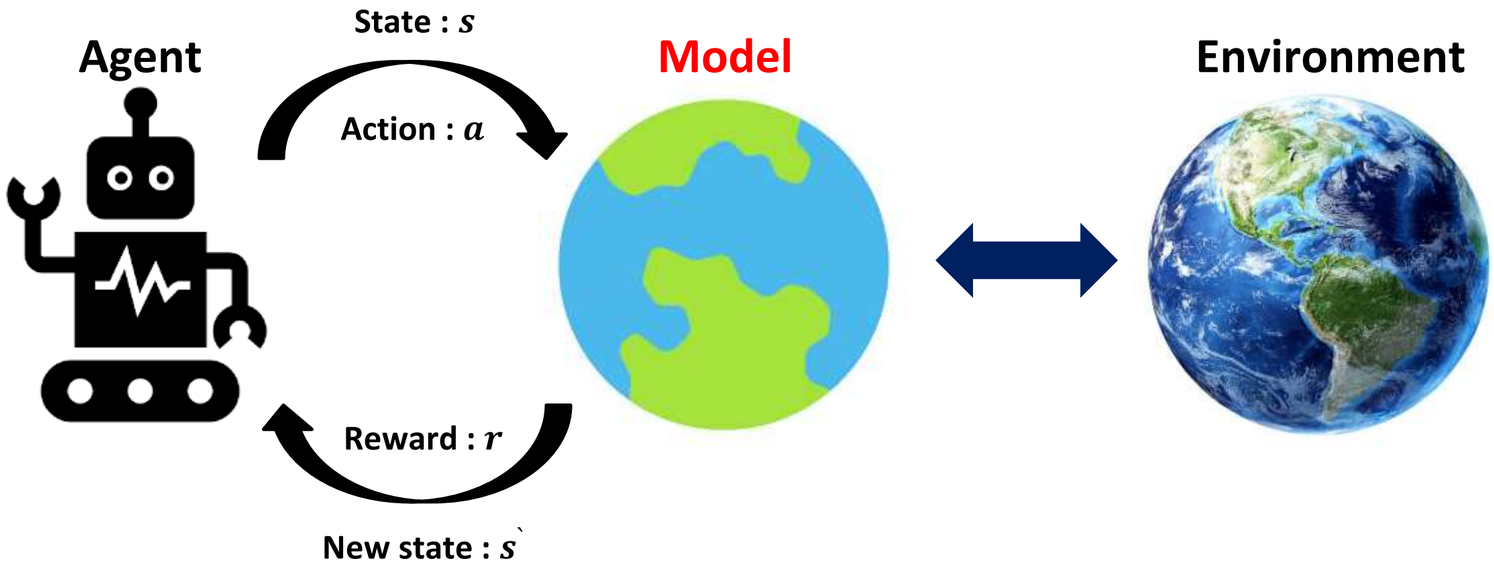
Model-Free 강화학습은 환경과의 실제 상호작용(경험)을 통해 직접 학습하므로, 복잡하거나 비용이 많이 드는



샘플 효율성 부족 (Sample Inefficiency) 문제 존재

Introduction

- **Model-Based Reinforcement Learning(MBRL)**
  - **MBRL은 환경의 동역학(dynamics)을 학습하여 Model을 기반으로 행동을 결정하는 접근 방법**
  - **“Model”이란 실제 환경에서의 동역학을 유사하게 모델링한 가상 환경**
  - 에이전트는 실제 환경에서 학습하는 대신, **Model**을 통해 다양한 시뮬레이션이 가능 -> **환경과 Model-Free 방법론보다 샘플 효율성 강건**
  - 어렵거나 많은 경험 데이터가 요구되는 상황에서 효율적이고 빠르게 학습

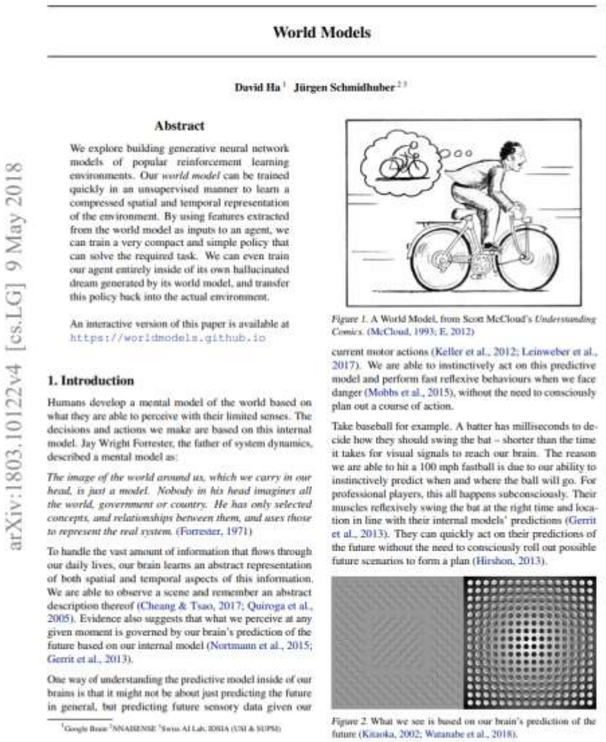


Model-Based RL

Methods

World Models

- **World Models(2018)**
  - **Model-Based RL 내 핵심 방법론으로 사용되는 모델**
  - **사람의 인지 작용을 묘사한 “World Model” 방법론 제시**
  - **실제 환경의 동역학 정보(dynamics)를 학습하여 가상의 환경을 모델링 (model)**



World Models

David Ha [1] Jürgen Schmidhuber [2 3]

Abstract

We explore building generative neural network models of popular reinforcement learning environments. Our *world model* can be trained quickly in an unsupervised manner to learn a compressed spatial and temporal representation of the environment. By using features extracted from the world model as inputs to an agent, we can train a very compact and simple policy that can solve the required task. We can even train our agent entirely inside of its own hallucinated dream generated by its world model, and transfer this policy back into the actual environment.

An interactive version of this paper is available at https://worldmodels.github.io

1. Introduction

Humans develop a mental model of the world based on what they are able to perceive with their limited senses. The decisions and actions we make are based on this internal model. Jay Wright Forrester, the father of system dynamics, described a mental model as:

The image of the world around us, which we carry in our head, is just a model. Nobody in his head imagines all the world, government or country. He has only selected concepts, and relationships between them, and uses those to represent the real system. (Forrester, 1971)

To handle the vast amount of information that flows through our daily lives, our brain learns an abstract representation of both spatial and temporal aspects of this information. We are able to observe a scene and remember an abstract description thereof (Cheang & Tsao, 2017; Quiroga et al., 2005). Evidence also suggests that what we perceive at any given moment is governed by our brain's prediction of the future based on our internal model (Nortmann et al., 2015; Gerrit et al., 2013).

One way of understanding the predictive model inside of our brains is that it might not be about just predicting the future in general, but predicting future sensory data given our current motor actions (Keller et al., 2012; Leinweber et al., 2017). We are able to instinctively act on this predictive model and perform fast reflexive behaviours when we face danger (Mobbs et al., 2015), without the need to consciously plan out a course of action.

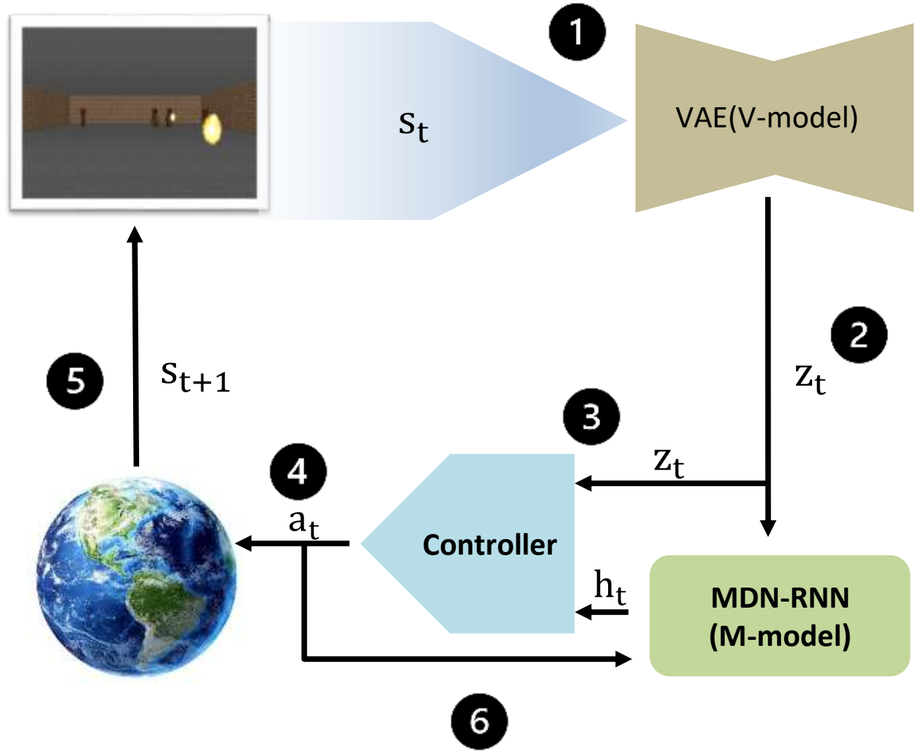
Take baseball for example. A batter has milliseconds to decide how they should swing the bat – shorter than the time it takes for visual signals to reach our brain. The reason we are able to hit a 100 mph fastball is due to our ability to instinctively predict when and where the ball will go. For professional players, this all happens subconsciously. Their muscles reflexively swing the bat at the right time and location in line with their internal models' predictions (Gerrit et al., 2013). They can quickly act on their predictions of the future without the need to consciously roll out possible future scenarios to form a plan (Hirshon, 2013).

[1]Google Brain [2]NNAISENSE [3]Swiss AI Lab, IDSIA (USI & SUPSI)

arXiv:1803.10122v4 [cs.LG] 9 May 2018

Figure 1. A World Model, from Scott McCloud's *Understanding Comics*. (McCloud, 1993; E, 2012)

Figure 2. What we see is based on our brain's prediction of the future (Kitaoka, 2002; Watanabe et al., 2018).



Methods

World Models

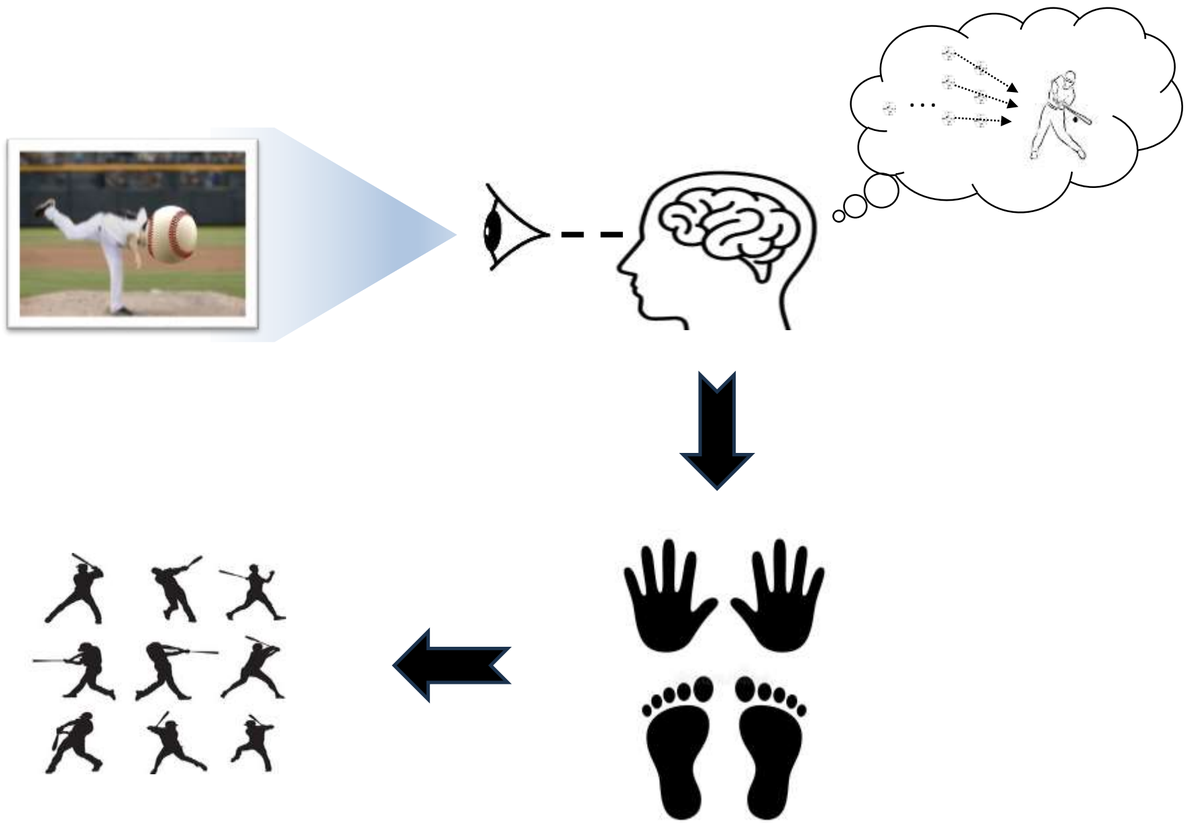
- **Motivation**
  - 인간은 과거의 축적된 추상적인 경험을 기반으로 판단을 내리는 **“인지적 추론”** 과정을 보유
  - **뇌는 시각적 정보를 압축 및 추상화하여 (무)의식적으로 행동을 명령**
  - **Ex) 타자가 무의식적으로 공의 궤적을 예측해 배팅을 취하는 과정**

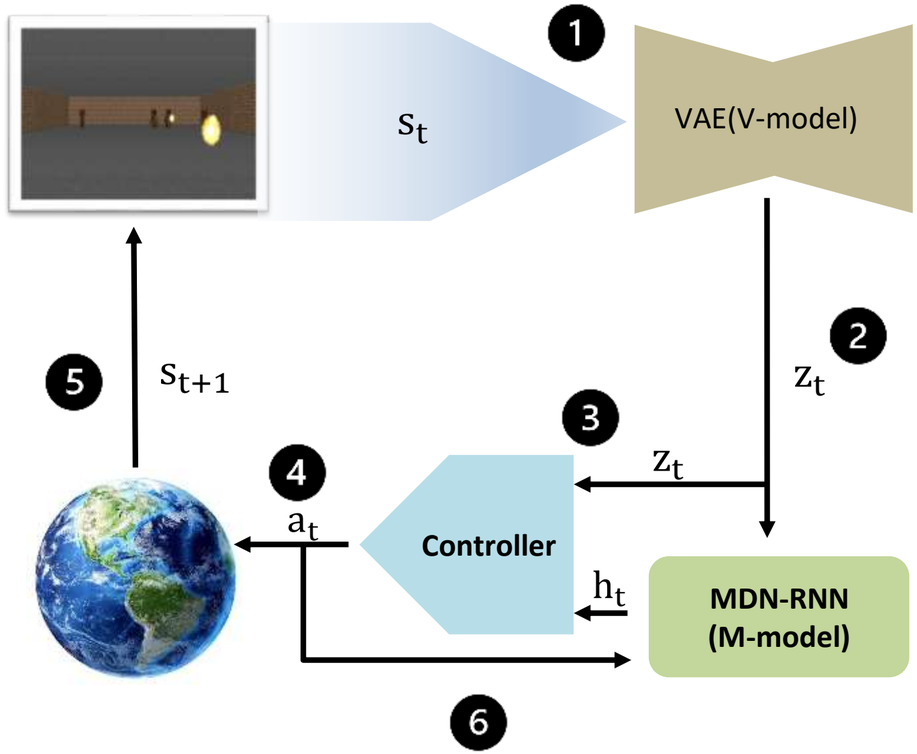
“우리가 머리 속에 지니고 있는 주변 세계의 이미지는 단지 모델에 불과하다.
인간은 오직 선택된 개념들과 주변 환경과의 **추상적인 관계** 만을 가지고 있으며, 이를 통해 실제 시스템을 표현한다.”
(Forrester, 1971)

Methods

World Models

- **World Models (V, M, C model)**
  - 시각적 데이터를 처리하는 기관 : V(vision) model
  - 과거를 기억하거나 다음 상태를 예측하는 기관 : M(memory) model
  - 위 환경의 정보로 행동을 취하는 기관 : C(controller) model

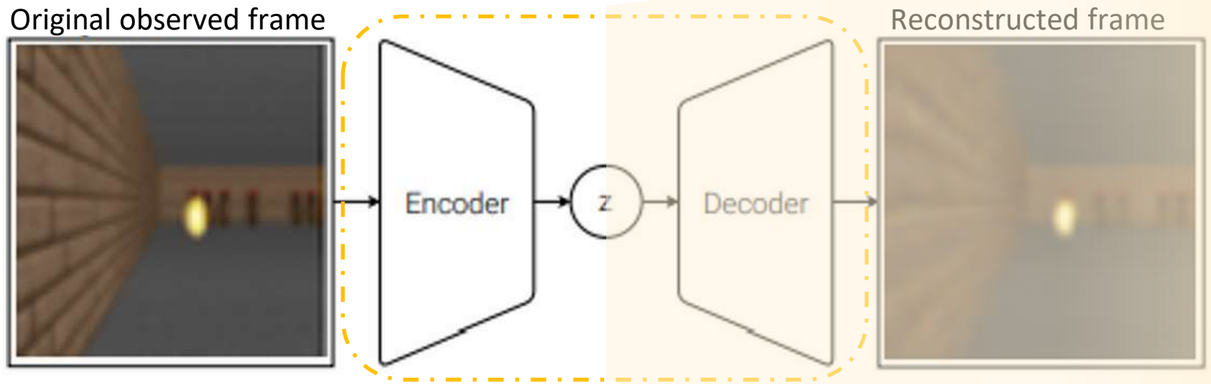




Methods

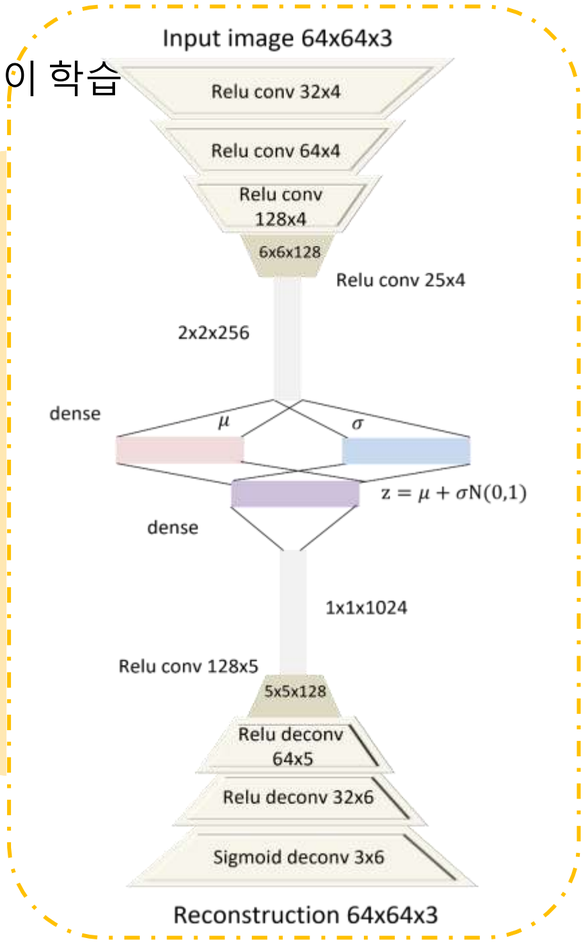
World Models

- **V(VAE, vision) Model**
  - 인간의 시각적 정보 처리에 해당
  - Time step마다 환경의 관측 데이터가 제공될 때, V 모델은 입력된 고차원 정보를 압축된 정보로 재구성
  - **ConVAE의 인코더로 입력 데이터의 중요한 특징을 학습 및 압축**
  - 정보 손실을 최소화하여 저차원의 샘플 데이터를 확보하고(z), 원본 데이터 특징을 잘 유지하는 것이 학습



$$q(z|x) = \mathcal{N}(z; \mu(x), \sigma^2(x)) \qquad p(x|z) = \mathcal{N}(x; \mu'(z), \sigma'^2(z))$$

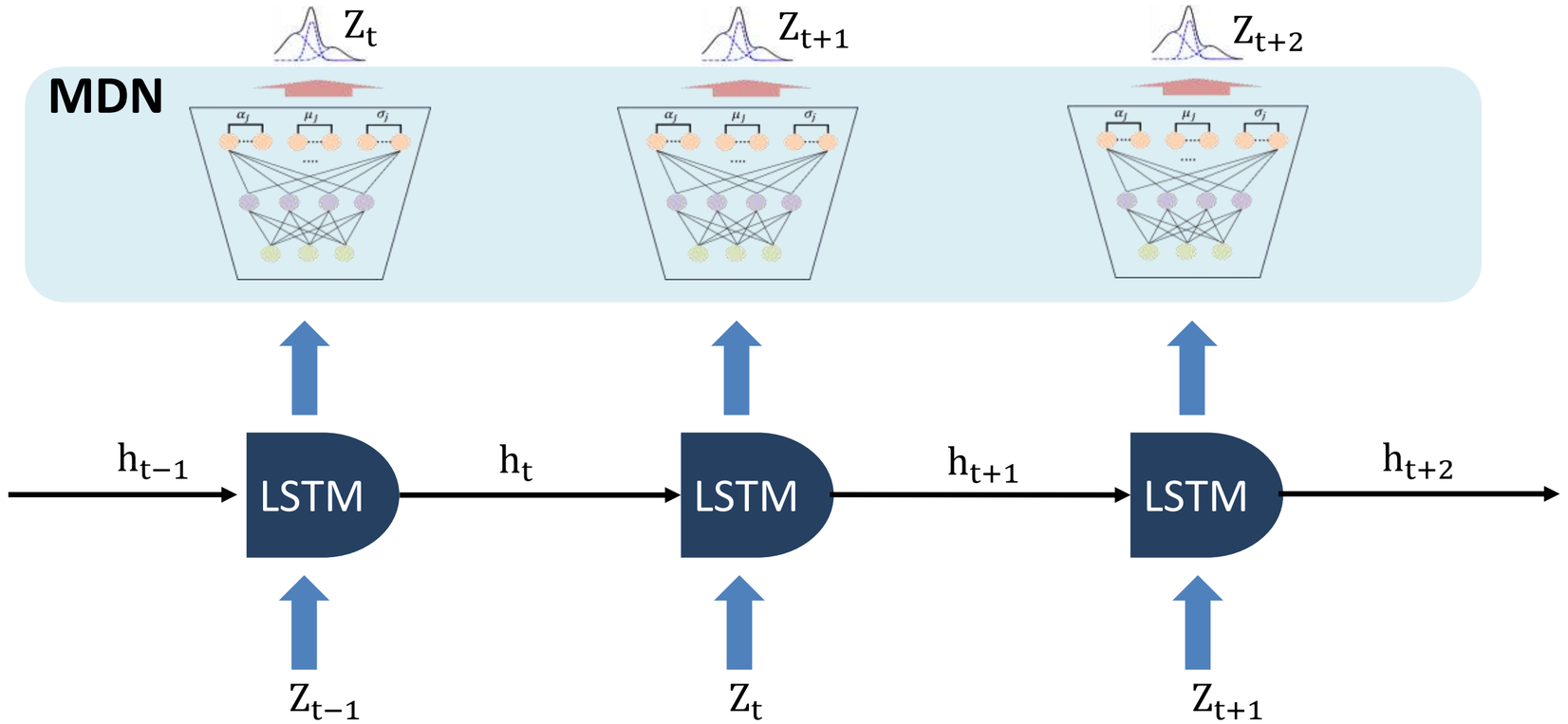
Objective Function : $\mathcal{L}_{VAE} = -\mathbb{E}_{q(z|x)}[\log p(x|z)] + D_{KL}(q(z|x)||p(z))$



Methods

World Models

- **M(MDN-RNN, memory) model**
  - 실세계와 환경의 정보는 시간에 따라 연속적으로 관측되므로, 다음 상태의 확률 분포를 예측하는 MDN-RNN을 사용
  - Action, observation, hidden state를 바탕으로 다음 시점의 잠재 벡터 확률 분포를 예측
  - 확률 분포를 예측함으로써, 환경의 불확실성과 미래에 대한 여러 예측을 수행 가능

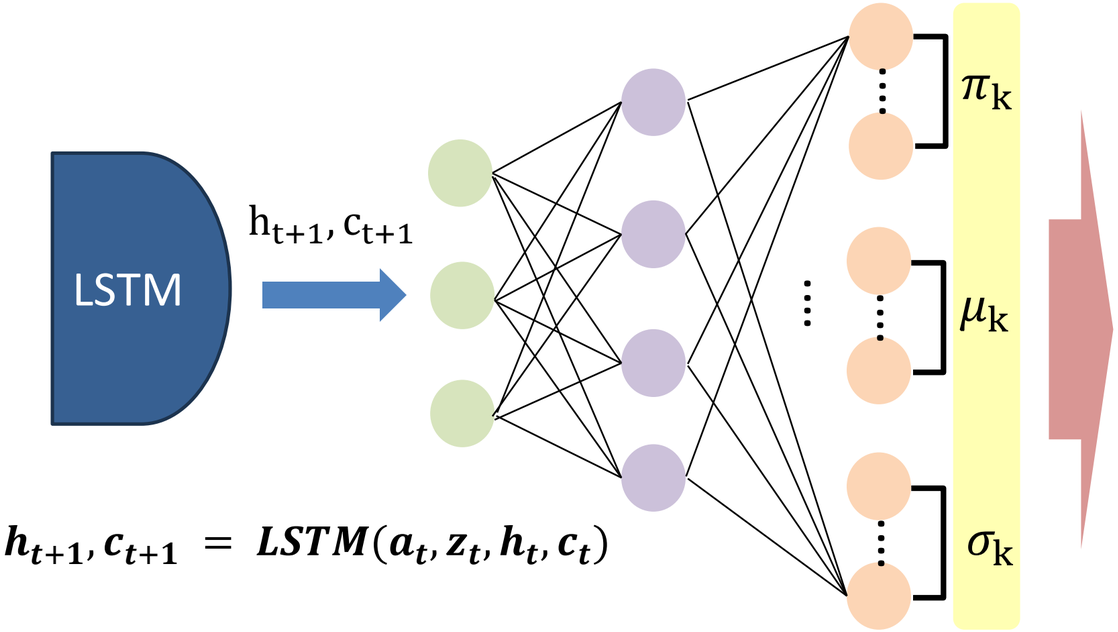


Methods

World Models

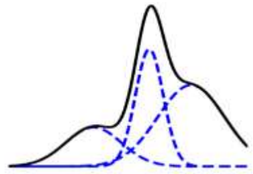
- **MDN-RNN**
  - MDN layer는 RNN의 출력인 h를 받아 여러 병렬적인 선형 레이어로 평균, 표준편차, 혼합 계수 파라미터를 예측
  - **실세계와 환경의 정보는 시간에 따라 연속적으로 관측되므로, 다음 상태의 확률 분포를 예측하는 MDN-RNN을 사용**
  - **V 모델로 잘 압축된 데이터 (z)를 활용하여 불확실한 미래 상황에 대비 가능 → "Dream"과 연결**

$$\pi_k(h_{t+1}), \mu_k(h_{t+1}), \sigma_k(h_{t+1}) = MDN(h_{t+1})$$



$$h_{t+1}, c_{t+1} = LSTM(a_t, z_t, h_t, c_t)$$

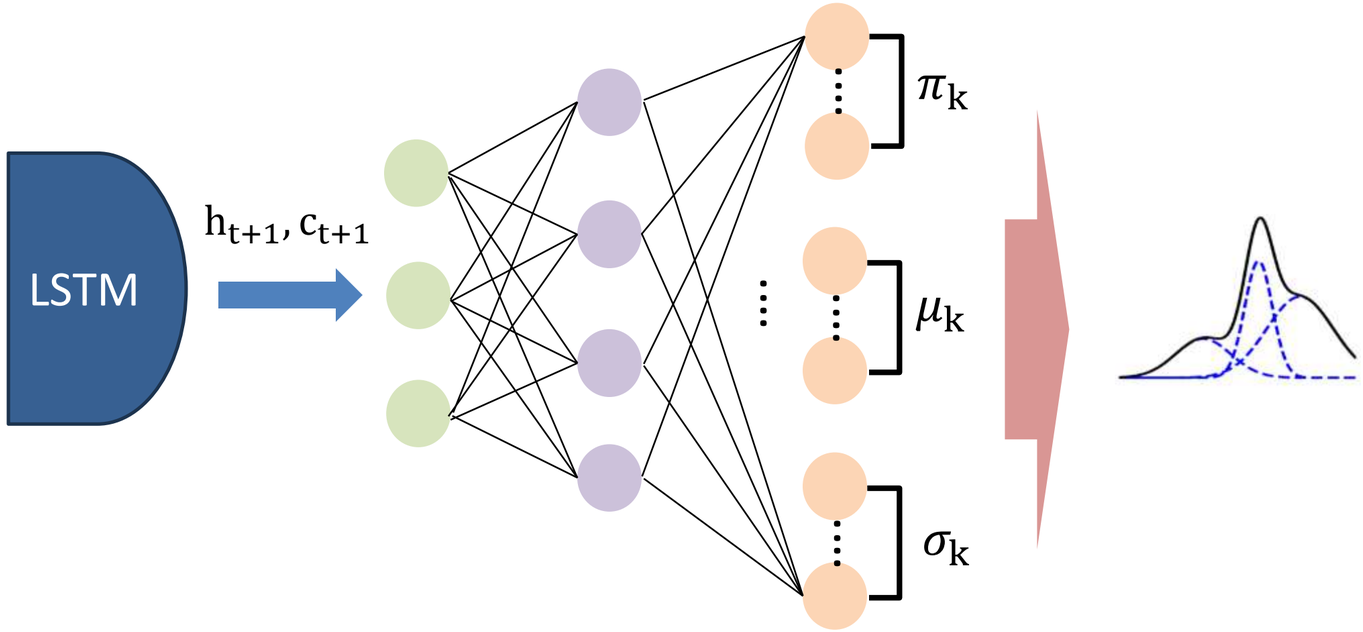
$$P(z_{t+1}|a_t, z_t, h_t) = \sum_{k=1}^{K} \pi_k(h_t) N(\mu_k|h_t, diag(\sigma_k(h_t)^2))$$

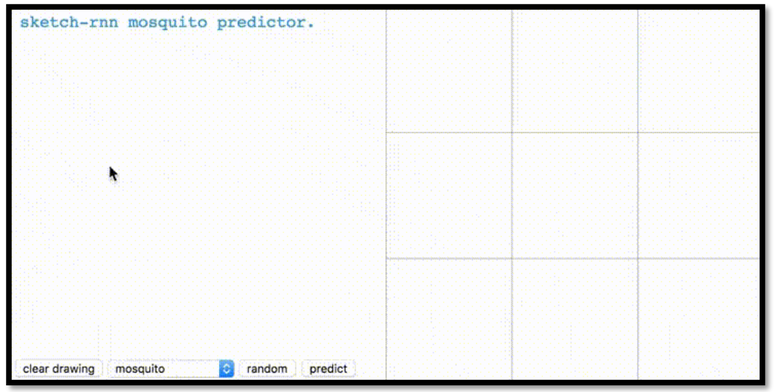


Methods

World Models

- **MDN-RNN**
  - **실세계와 환경의 정보는 시간에 따라 연속적으로 관측되므로, 다음 상태의 확률 분포를 예측하는 MDN-RNN을 사용**
  - **V 모델로 잘 압축된 데이터 (z)를 활용하여 불확실한 미래 상황에 대비 가능**
  - Sketch-RNN 실험에서 MDN-RNN 시퀀스 생성 능력 확인 : 입력에 대해 다양한 결과 생성
  - MDN-RNN은 다음 시점의 잠재 벡터를 예측하여 환경의 미래 상태 시퀀스를 모델링 → **불확실한 환경의 특성을 고려하여 여러 상황 대비**



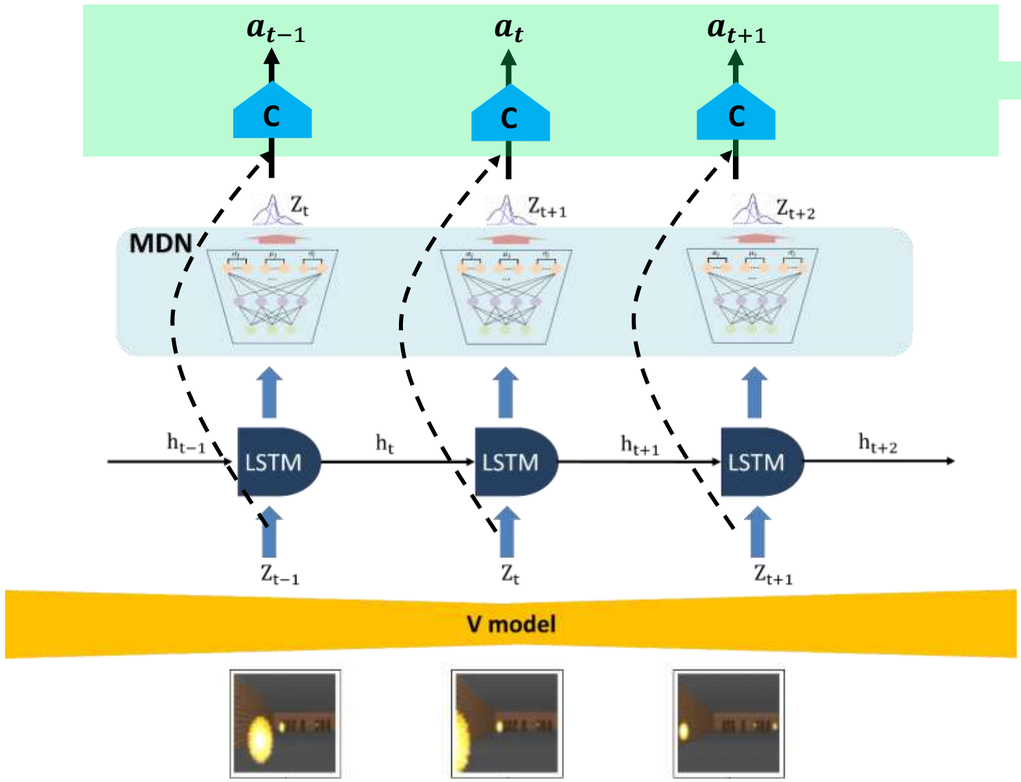


Sketch-RNN

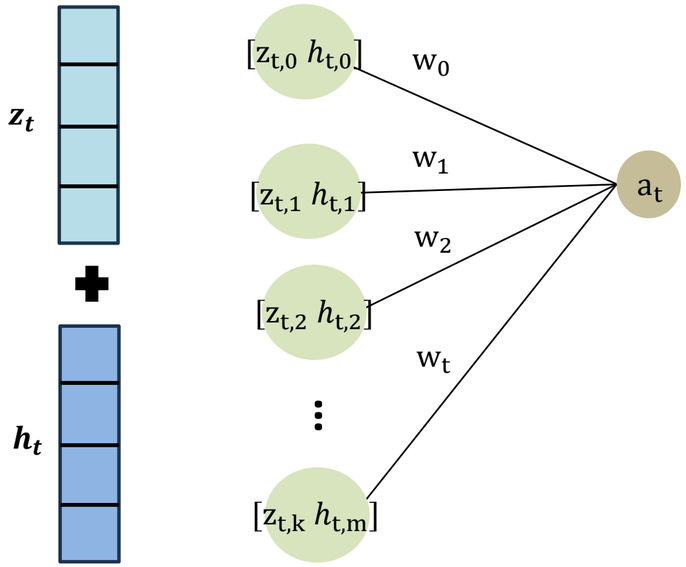
Methods

World Models

- **C(controller) model**
  - 인지적 추론을 바탕으로 결정된 행동을 취하는 부분
  - 모델링된 환경의 정보를 바탕으로, 에이전트가 실제 환경에서 행동을 취하고 다음 환경 정보를 얻는 모델
  - **C 모델은 linear layer로 구성된 fully connected network이며 V, M 모델에 비해 간단한 네트워크 형성**



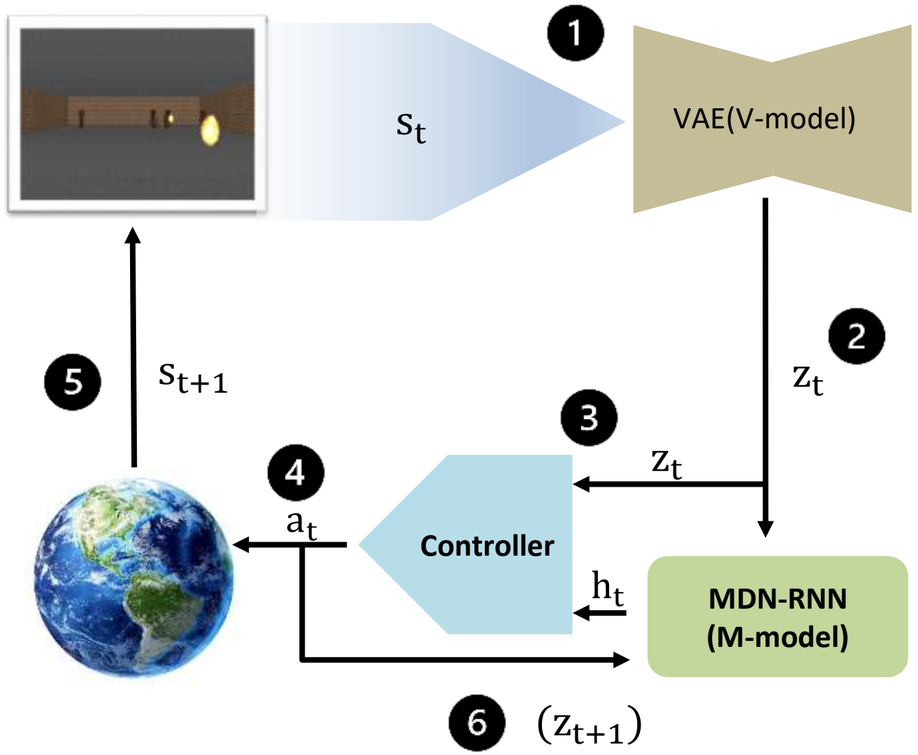
$$a_t = W_c[z_t \; h_t] + b_C$$



Methods

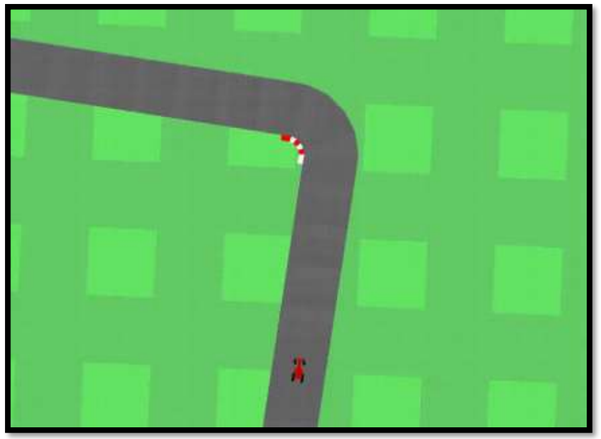
World Models

- **World Models (V, M, C model)**
  - 시각적 데이터를 처리하는 기관 : V(vision) model
  - 과거를 기억하거나 다음 상태를 예측하는 기관 : M(memory) model
  - 위 환경의 정보로 행동을 취하는 기관 : C(controller) model



Experiments

- **Experiments**
  - CarRacing : 매 시도마다 새로운 트랙이 생성되고 속도, 가속 등의 연속적인 행동을 취하는 환경
  - VizDoom : 연속적인 3D 환경에서 가능한 오래 살아남는 것이 목표인 환경

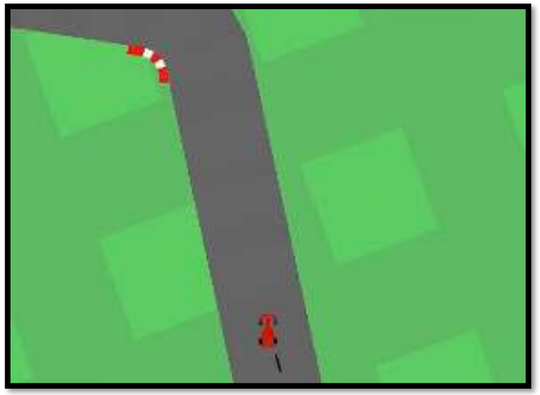


(a) CarRacing-v0

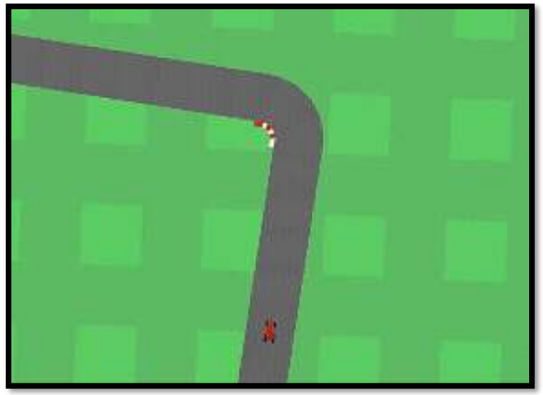
(b) VizDoom : Take Cover

Experiments

- **Experiments**
  - 기존 Model-Free 방법론과 World Model 간 성능 비교 분석 진행
  - V model only는 World Model의 Vision model만 사용한 결과
  - Full World Model을 사용했을 때, 가장 높은 점수 획득
  - 인지적 추론을 담당하는 M model의 영향력을 확인
  - World Model 접근 방식이 고차원 픽셀 입력에서 환경의 동역학 표현을 학습해 강화학습 문제 해결



(a) V model only

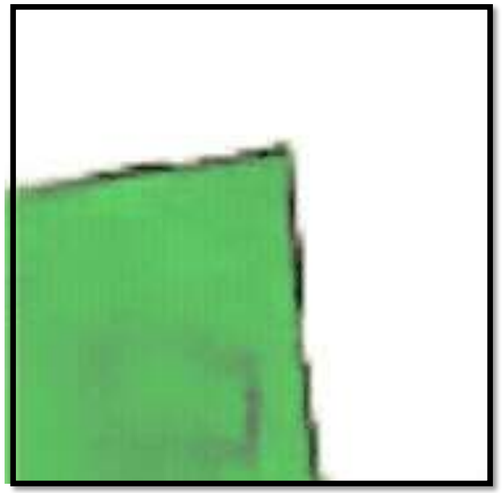


(b) Full World Model

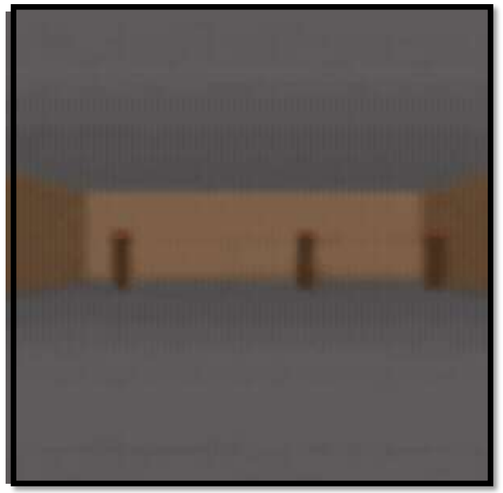
Method	AVG. Score
DQN	343 ± 18
A3C(continuous)	591 ± 45
A3C(discrete)	652 ± 10
CEOBILLIONAIRE	838 ± 11
V Model only	632 ± 251
V Model(with hidden layer)	788 ± 141
Full World Model	**906 ± 21**

Experiments

- **Training inside of the Dream**
  - Dream : World Model의 M model은 다음 상태를 예측할 수 있고, 예측한 상태를 다음 상태로 삼아 가상의 시나리오를 생성하는 과정
  - 실제 상태가 아닌 예측 상태를 다음 상태로 설정하여 가상의 환경에서 에이전트가 정책을 실행
  - **실제 행동을 적용하기 전, 가상의 환경에서 시뮬레이션을 통해 학습 가능**



Racing inside of World Model`s Dream

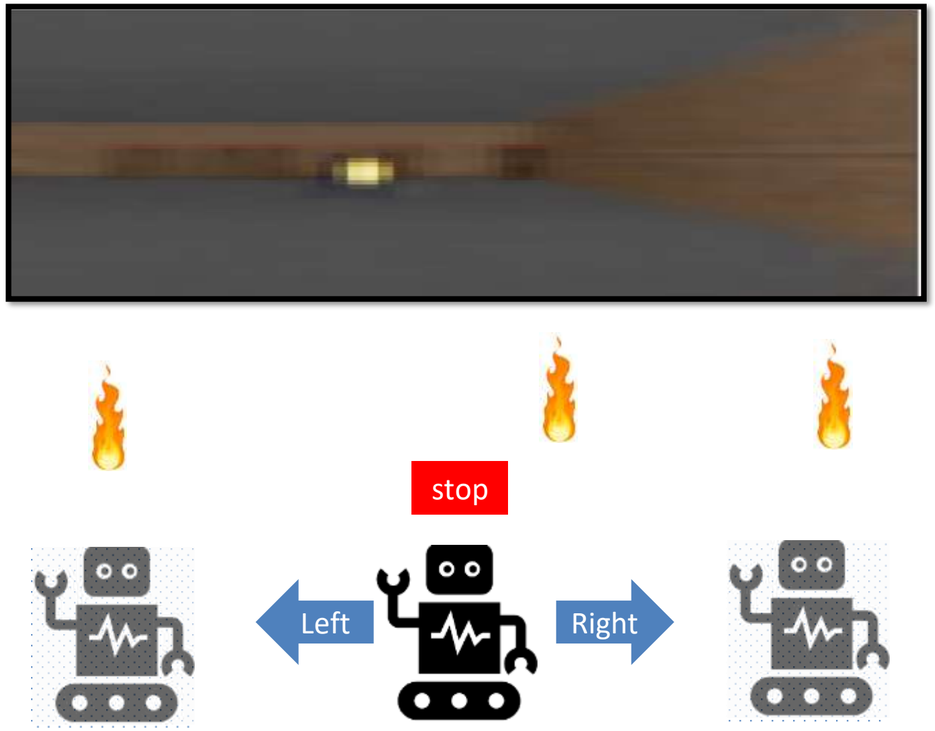


VizDoom inside of World Model`s Dream

Experiments

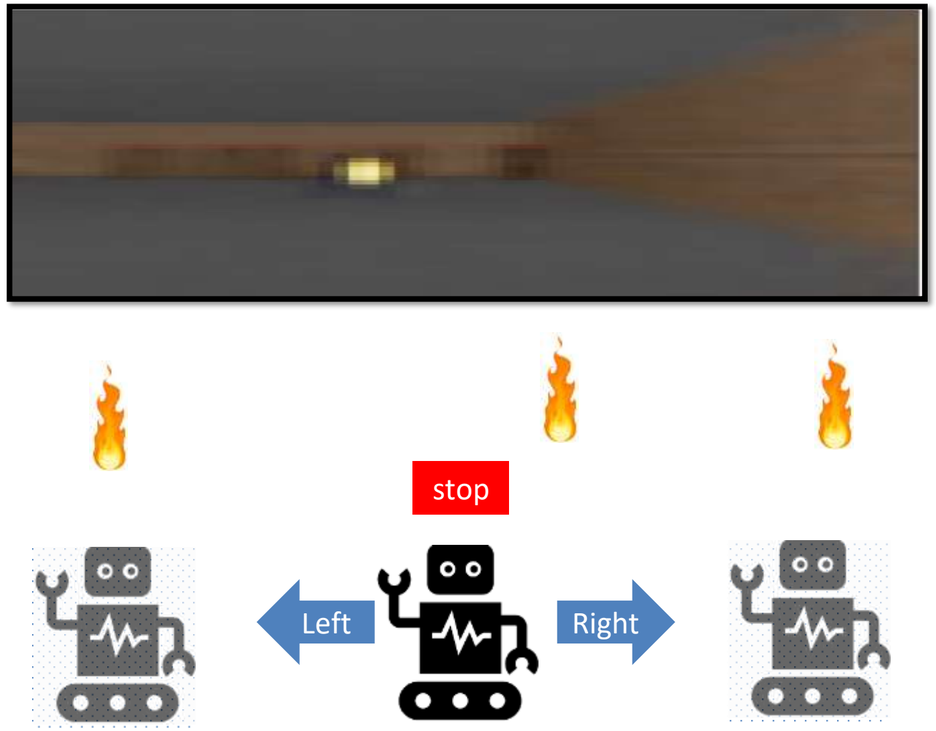
World Models

- **Learning Inside of a Dream**
  - 에이전트가 World Model이 생성한 가상의 Dream 환경에 내에서 학습한 후, 학습된 정책이 실제 환경에서 효과적인지 실험
  - 에이전트는 가상(Dream) 환경에서 약 900 time step 점수를 달성
  - M model은 에이전트와 행동과 환경의 동역학(파이어볼, 적, 벽 등)을 최대한 실제와 비슷하게 학습 가능



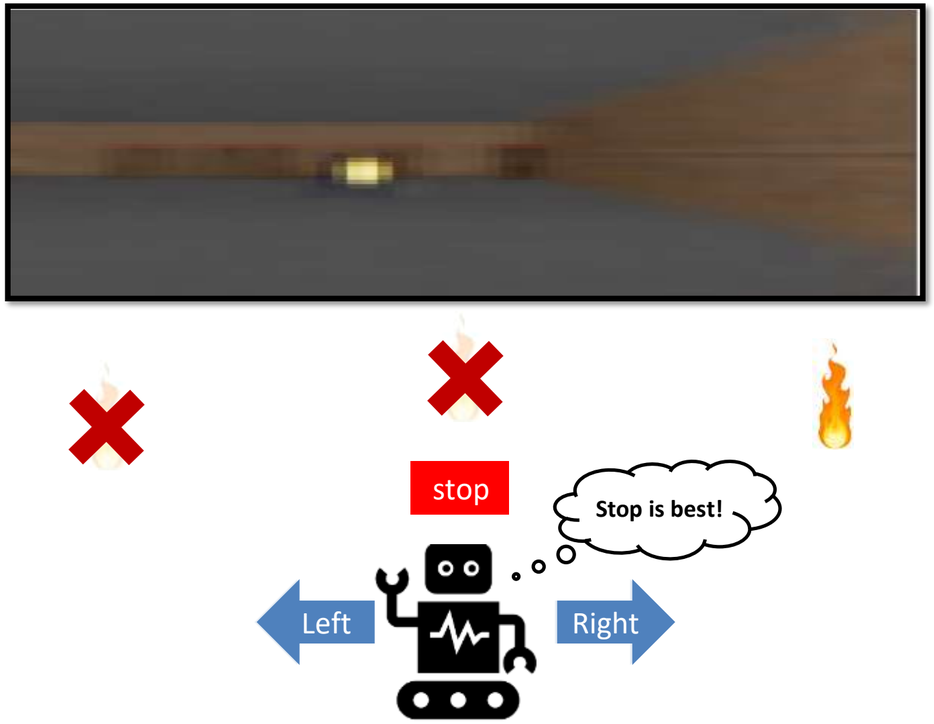
Experiments

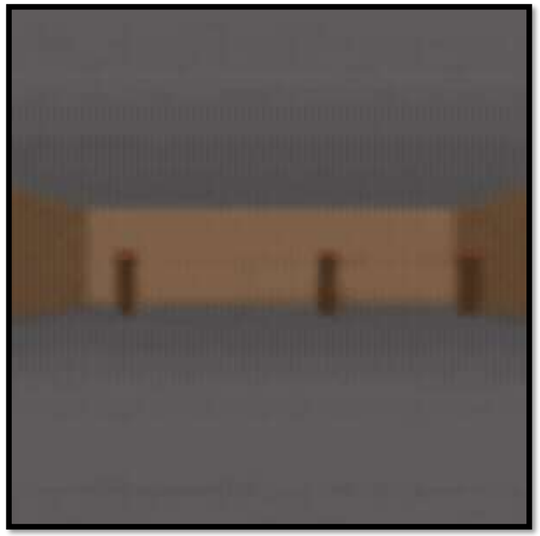
- **Experiments inside of the Dreams**
  - 가상 환경 내부에서 훈련된 에이전트를 가져와서 원래 VizDoom 시나리오에서 성능을 테스트
  - **가상 환경에서 훈련된 정책으로 실제 환경에 적용했을 때, 100회 연속 실험에서 약 1,100 time step (>750) 생존 시간 기록**



Experiments

- **Experiments inside of the Dreams**
  - 하지만 학습된 World Model은 실제 환경이 아니므로, 복잡도와 표현이 떨어질 수 있음
  - 또한 model이 불덩이를 생성하지 않거나 도중에 사라지도록 환경을 모델링하는 경우 존재 -> 잘못된 정책 학습 수행
  - 난이도를 조절하거나, 다양한 상황을 학습할 수 있도록 하는 방법 필요



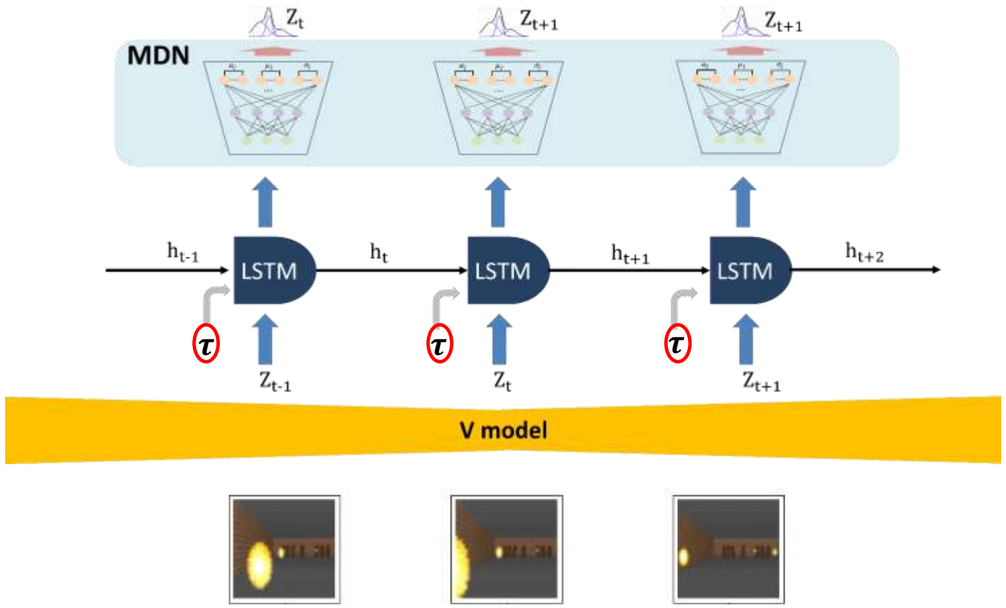


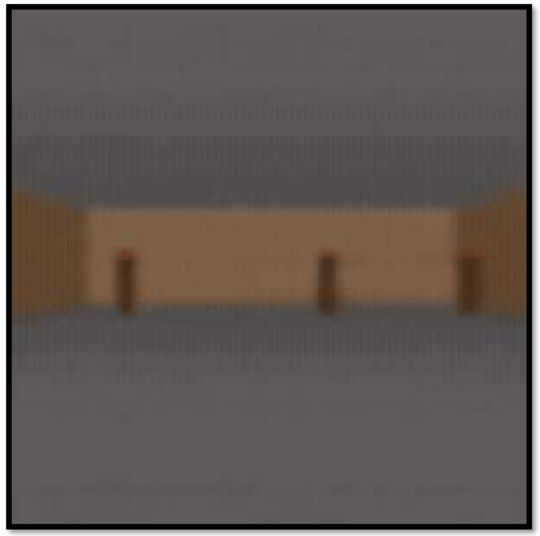
VizDoom with adversarial policy in Dream

Experiments

World Models

- **Experiments inside of the Dreams**
  - **하지만 학습된 World Model은 실제 환경이 아니므로, 복잡도와 표현이 떨어질 수 있음**
  - **MDN-RNN의 출력 확률 분포에서 샘플링 할 때, '불확실성' 또는 '무작위성'의 정도를 조절하기 위해 Temperature parameter τ를 사용**
  - **τ 를 높임으로써 가상 환경을 더 예측 불가능하고 상황으로 유도 → 더 어렵고 다양한 환경에서 훈련한 에이전트는 실제 환경에서 더 강인할 것**





VizDoom with adversarial policy in Dream

Experiments

- **Experiments inside of the Dreams**
  - MDN-RNN의 출력 확률 분포에서 샘플링할 때, '불확실성' 또는 '무작위성'의 정도를 조절하기 위해 Temperature parameter τ를 사용
  - τ를 높임으로써 가상 환경을 더 예측 불가능하고 상황으로 유도 -> 더 어렵고 다양한 환경에서 훈련한 에이전트는 실제 환경에서 더 강인할 것
  - 여러 τ를 적용하고 가상 환경에서 정책 학습 후, 실제 환경에서 정책 수행 결과
  - 실제 환경의 점수는 temperature parameter가 높을 경우 향상

Dream with τ(0.5)

Dream with τ(3.0)

Temperature	Virtual Score	Actual Score
0.10	**2086 ± 140**	193 ± 58
0.50	2060 ± 277	196 ± 50
1.00	1145 ± 690	868 ± 511
1.15	918 ± 546	**1092 ± 556**
1.30	732 ± 269	753 ± 139
Random Policy	N/A	210 ± 108
Gym Leader	**N/A**	820 ± 58

Experiments

World Models

- Experiments inside of the Dreams
  - MDN-RNN의 출력 확률 분포에서 샘플링할 때, '불확실성' 또는 '무작위성'의 정도를 조절하기 위해 Temperature parameter τ를 사용
  - τ를 높임으로써 가상 환경을 더 예측 불가능하고 상황으로 유도 -> 더 어렵고 다양한 환경에서 훈련한 에이전트는 실제 환경에서 더 강인할 것
  - 여러 τ를 적용하여 가상 환경에서 정책 학습 후 실제 환경에서 성능 측정 결과
  - 실제 환경의 점수는 temperature parameter가 높을 경우 향상

Limitation of World Models (2018)

1) Derivative-Free 기반 업데이트를 수행하므로 정책의 그라디언트 사용 불가, 최적의 정책 매개변수를 위한 업데이트 방향 부재

2) 복잡한 환경에서 연속적이고 정교한 행동에 대한 정책을 기대하기 어렵고, 무작위적 행동 탐색에 의존 (효율성 저하)

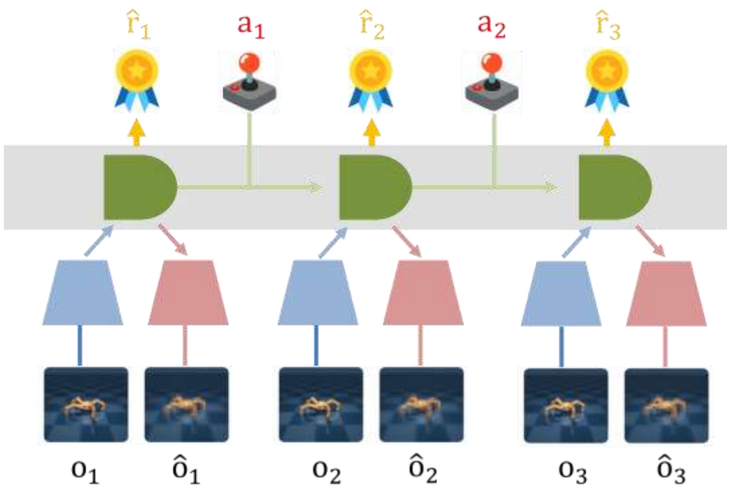
3) 더 먼 미래를 예측하지 못하고 짧은 Dream만 가능

Temperature	Virtual Score	Actual Score
0.10	2086 ± 140	196 ± 58
0.50	2060 ± 277	196 ± 50
1.00	1145 ± 690	868 ± 511
1.15	1467 ± 946	1092 ± 556
1.30	732 ± 269	753 ± 139
Random Policy	N/A	210 ± 108
Gym Leader	N/A	820 ± 58

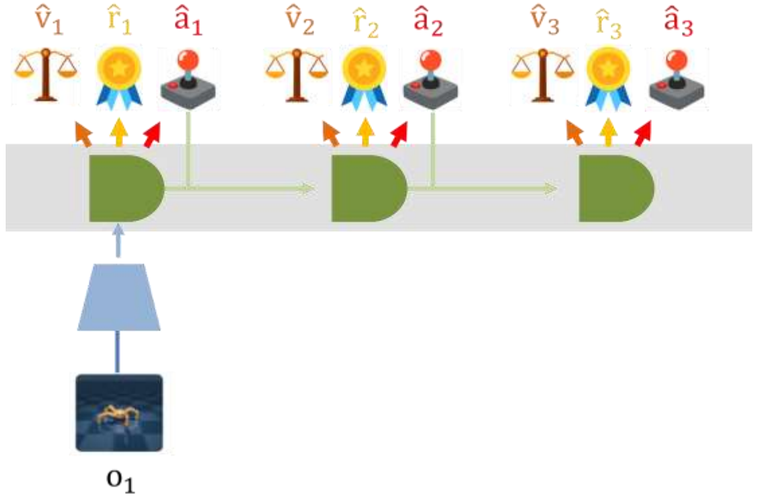
Methods

Dreamer

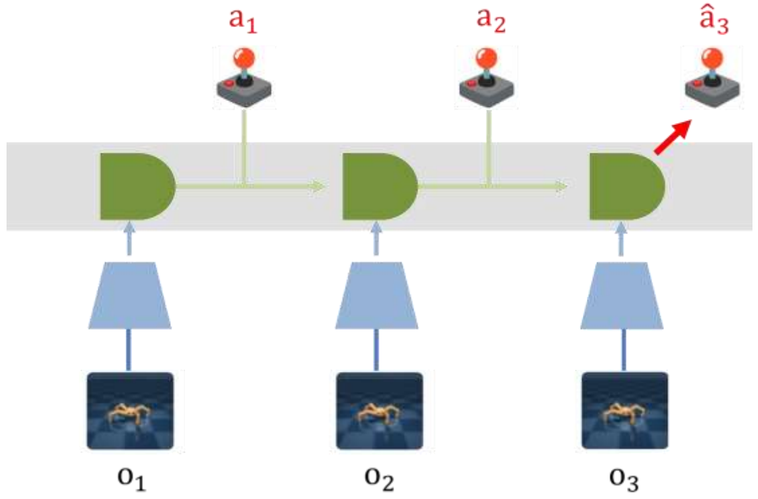
- **Dream to Control : Learning Behaviors by Latent Imagination**
  - 학습된 월드 모델의 잠재 공간에서 그라디언트를 활용하여 장기적인 행동을 학습하는 강력한 강화 학습 에이전트 “Dreamer” 제시
  - 잠재 공간 상 미래에 대한 더 먼 미래를 예측 및 학습 (Predicting multiple future steps)
  - 환경의 동역학 학습 → Imagination을 통한 정책 학습 → 실제 환경에서 정책 수행 및 경험 수집



(a) Learn dynamics from experience



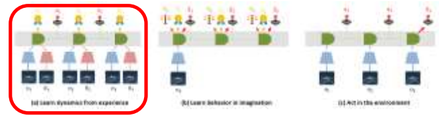
(b) Learn behavior in imagination



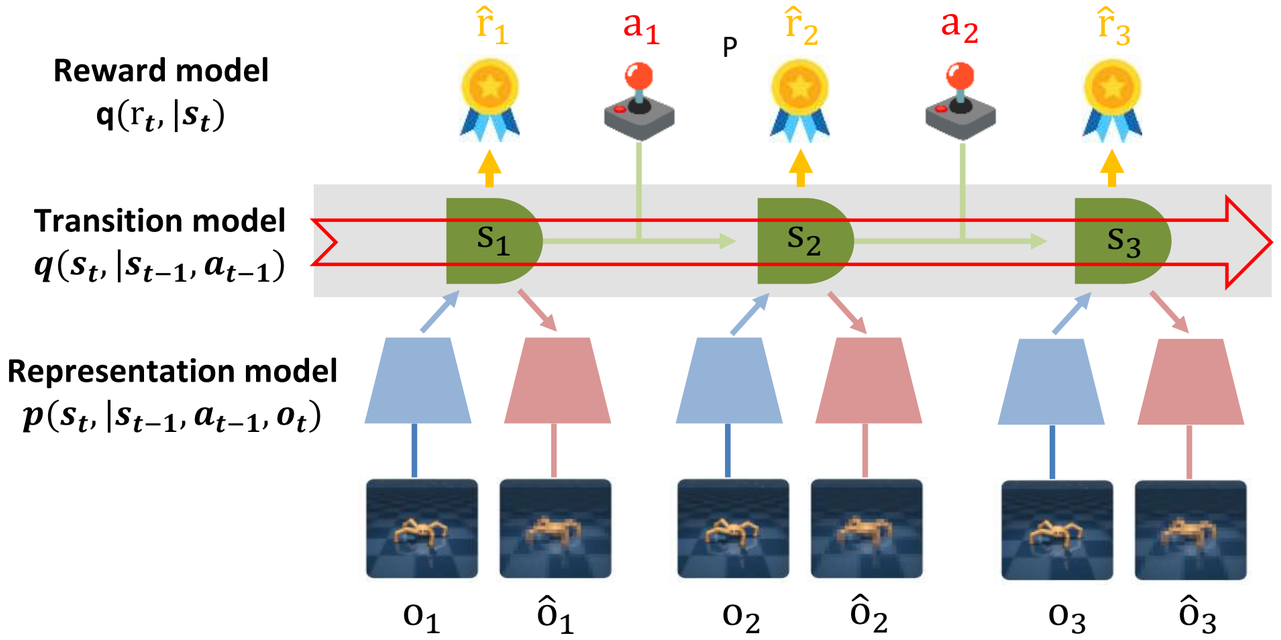
(c) Act in the environment

Methods

Dreamer

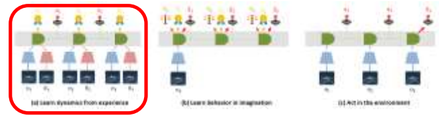


- **Dynamics Learning (Learning World Model)**
  - 에이전트가 환경에서 수집한 초기 경험 데이터로 환경의 동역학을 학습
  - **에이전트가 가상 환경 시뮬레이터(World Model)로 정확한 정책을 학습하도록 유도**
  - Representation, Transition, Reward, Observation model 학습



Methods

Dreamer



- **Recurrent State Space Model(RSSM)**
  - **Dreamer가 사용하는 World Model은 RSSM이란 구조로 학습**
  - RSSM은 고차원 관측을 처리하며 시간적으로 변화하는 시스템의 잠재 상태를 추론 및 예측
  - 위 잠재 상태를 통해 미리 미래를 예측하여 에이전트의 가상 경험을 가능케 하는 확률적 모델
  - World Model 중 V, M model의 연장선이며 시각 정보를 압축 및 기억하고 미래를 상상 하는 '뇌 ' 의 역할

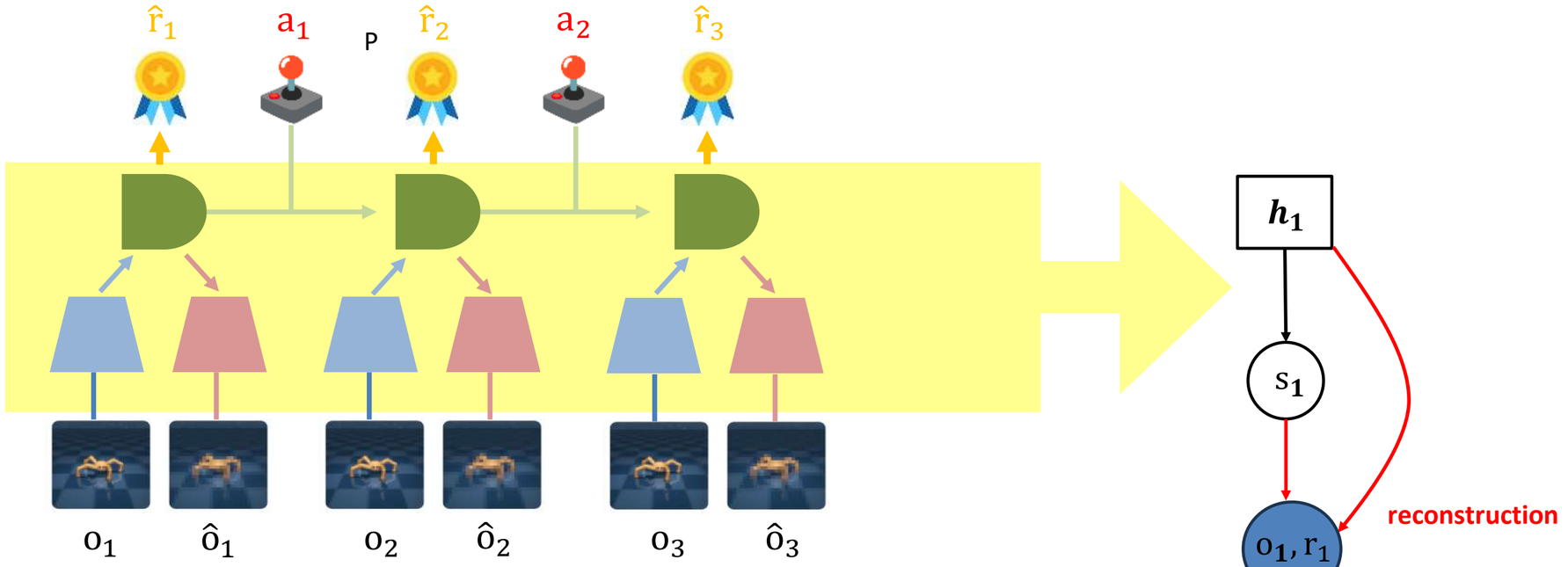
Deterministic model : $h_t = f(h_{t-1}, s_{t-1}, a_{t-1})$

Stochastic model : $s_t \sim p(s_t|h_t)$

Inference model : $s_t \sim p(s_t|h_t, o_t)$

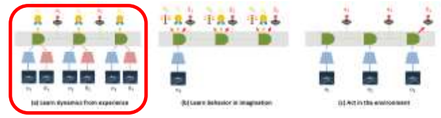
Observation model : $o_t \sim p(o_t|h_t, s_t)$

Reward model : $r_t \sim p(r_t|h_t, s_t)$



Methods

Dreamer



- **Recurrent State Space Model(RSSM)**
  - **Dreamer가 사용하는 World Model은 RSSM이란 구조로 학습**
  - RSSM은 고차원 관측을 처리하며 시간적으로 변화하는 시스템의 잠재 상태를 추론 및 예측
  - 위 잠재 상태를 통해 미리 미래를 예측하여 에이전트의 가상 경험을 가능케 하는 확률적 모델
  - World Model 중 V, M model의 연장선이며 시각 정보를 압축 및 기억하고 미래를 상상 하는 '뇌 ' 의 역할

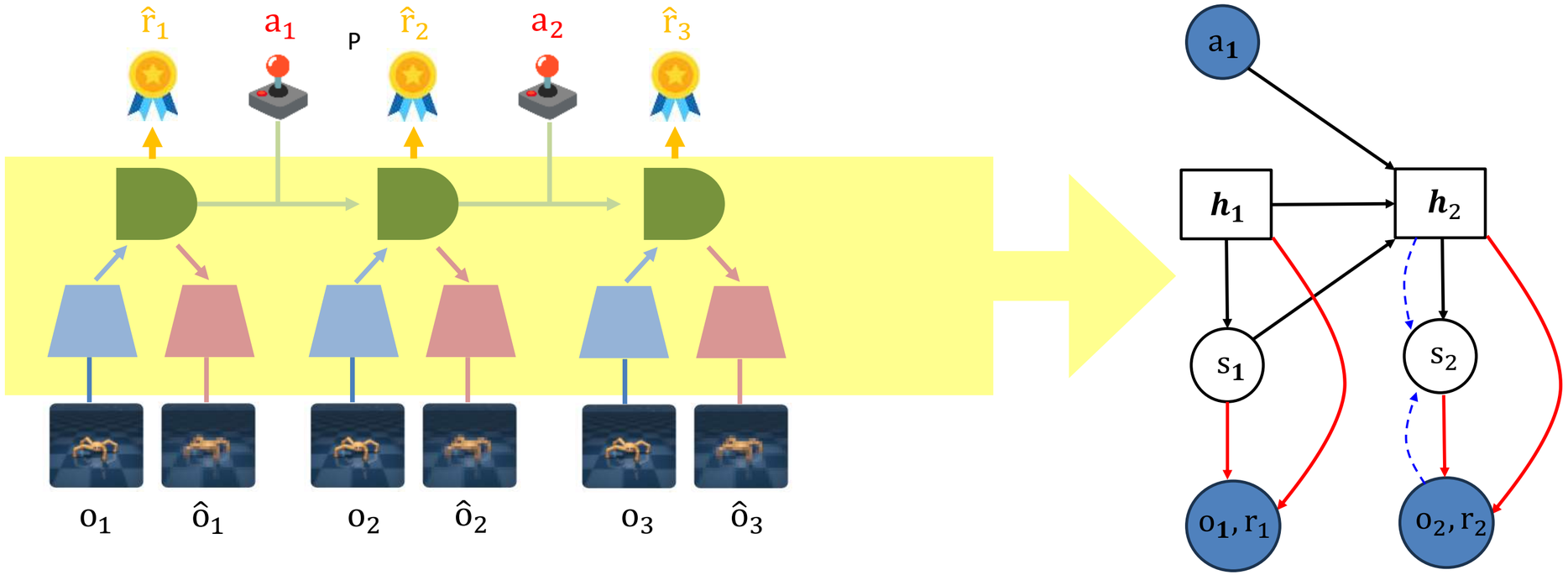
Deterministic model : $h_t = f(h_{t-1}, s_{t-1}, a_{t-1})$

Stochastic model : $s_t \sim p(s_t|h_t)$

Inference model : $s_t \sim p(s_t|h_t, o_t)$

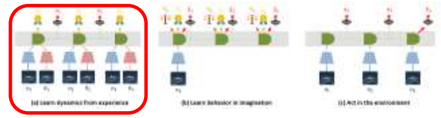
Observation model : $o_t \sim p(o_t|h_t, s_t)$

Reward model : $r_t \sim p(r_t|h_t, s_t)$



Methods

Dreamer



- **Recurrent State Space Model(RSSM)**
  - **Dreamer가 사용하는 World Model은 RSSM이란 구조로 학습**
  - RSSM은 고차원 관측을 처리하며 시간적으로 변화하는 시스템의 잠재 상태를 추론 및 예측
  - 위 잠재 상태를 통해 미리 미래를 예측하여 에이전트의 가상 경험을 가능케 하는 확률적 모델
  - World Model 중 V, M model의 연장선이며 시각 정보를 압축 및 기억하고 미래를 상상 하는 '뇌 ' 의 역할

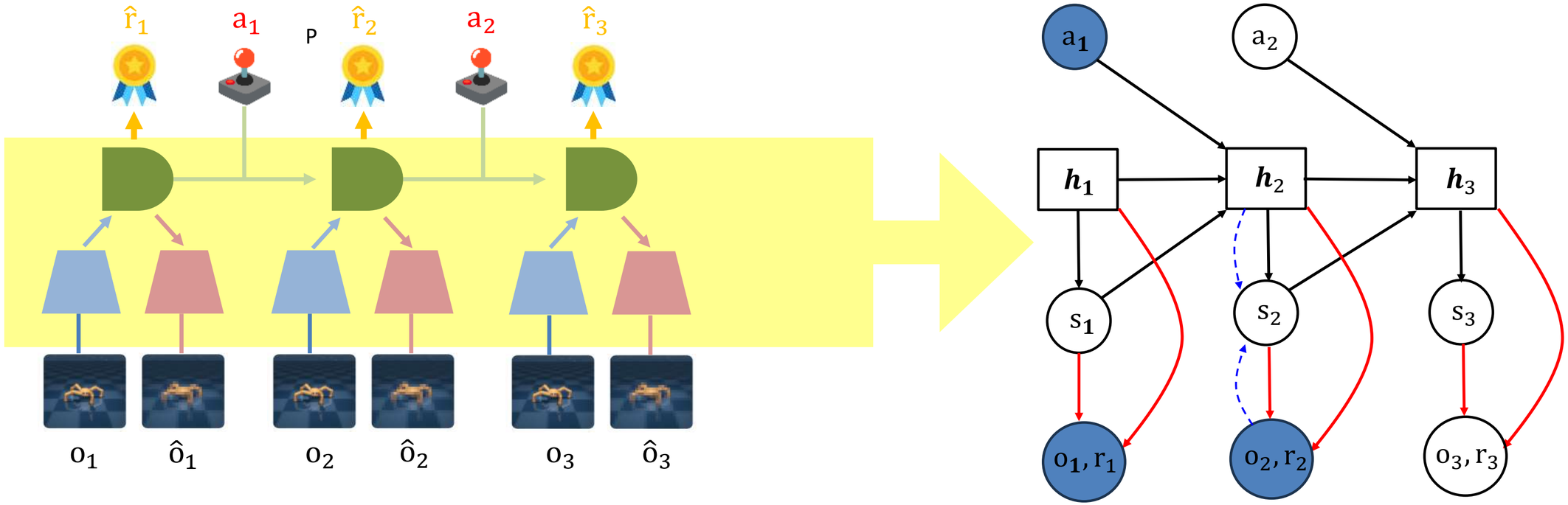
Deterministic model : $h_t = f(h_{t-1}, s_{t-1}, a_{t-1})$

Stochastic model : $s_t \sim p(s_t|h_t)$

Inference model : $s_t \sim p(s_t|h_t, o_t)$

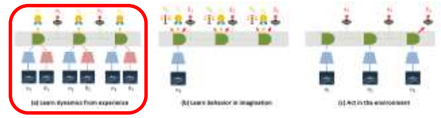
Observation model : $o_t \sim p(o_t|h_t, s_t)$

Reward model : $r_t \sim p(r_t|h_t, s_t)$



Methods

Dreamer



- **Dreamer with RSSM**
  - **Dreamer가 사용하는 World Model은 RSSM이란 구조로 학습**
  - 결정론적/확률적 변수를 동시에 다루는 이유
  - 결정론적 상태 $\boldsymbol{h}$의 역할 : 장기적인 기억이나 추세를 파악
  - 확률론적 상태 $\boldsymbol{s}$ 의 역할 : 연속적인 환경을 처리하며 환경의 내제된 불확실성을 모델링
  - 예측 가능한 정보($\boldsymbol{h}$)와 불확실한 정보($\boldsymbol{s}$) 동시에 사용함으로써 에이전트는 여러 미래를 경험 가능

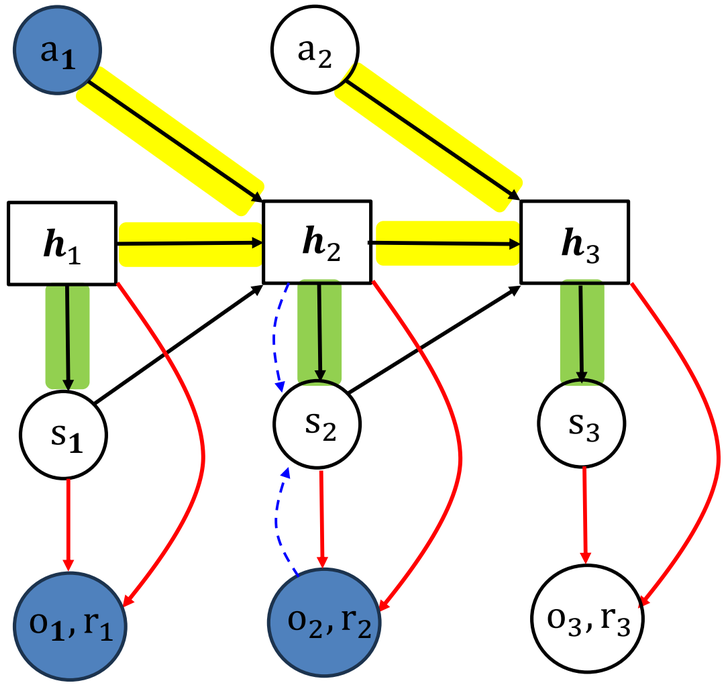
Deterministic model : $\boldsymbol{h_t = f(h_{t-1}, s_{t-1}, a_{t-1})}$

Stochastic model : $\boldsymbol{s_t \sim p(s_t|h_t)}$

Inference model : $\boldsymbol{s_t \sim p(s_t|h_t, o_t)}$

Observation model : $\boldsymbol{o_t \sim p(o_t|h_t, s_t)}$

Reward model : $\boldsymbol{r_t \sim p(r_t|h_t, s_t)}$



$$\boldsymbol{h_t = f(h_{t-1}, s_{t-1}, a_{t-1})}$$

$$\boldsymbol{s_t \sim p(s_t|h_t)}$$

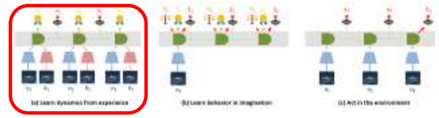
$$\boldsymbol{s_t \sim q(s_t|h_t, o_t)}$$

$$\boldsymbol{o_t \sim p(o_t|h_t, s_t)}$$

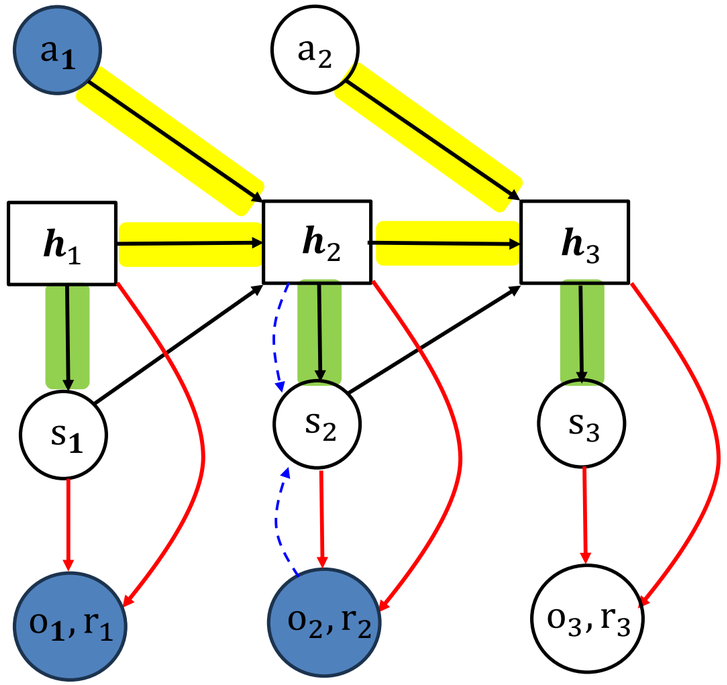
$$\boldsymbol{r_t \sim p(r_t|h_t, s_t)}$$

Methods

Dreamer



- **Dreamer with RSSM**
  - **Dreamer가 사용하는 World Model은 RSSM이란 구조로 학습**
  - 결정론적/확률적 변수를 동시에 다루는 이유
  - 결정론적 상태 $\boldsymbol{h}$의 역할 : 장기적인 기억이나 추세를 파악
  - 확률론적 상태 $\boldsymbol{s}$ 의 역할 : 연속적인 환경을 처리하며 환경의 내제된 불확실성을 모델링
  - 예측 가능한 정보($\boldsymbol{h}$)와 불확실한 정보($\boldsymbol{s}$) 동시에 사용함으로써 에이전트는 여러 미래를 경험 가능



$$\boldsymbol{h_t = f(h_{t-1}, s_{t-1}, a_{t-1})}$$

$$\boldsymbol{s_t \sim p(s_t|h_t)}$$

$$\boldsymbol{s_t \sim q(s_t|h_t, o_t)}$$

$$\boldsymbol{o_t \sim p(o_t|h_t, s_t)}$$

$$\boldsymbol{r_t \sim p(r_t|h_t, s_t)}$$

Dreamer notation

Deterministic model : $\boldsymbol{h_t = GRU(h_{t-1}, s_{t-1}, a_{t-1})}$

Transition model : $\boldsymbol{s_t \sim q_\theta(s_t|s_{t-1}, a_{t-1})}$

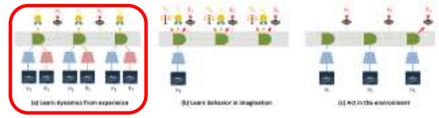
Representation model : $\boldsymbol{s_t \sim p_\theta(s_t|s_{t-1}, a_{t-1}, o_t)}$

Observation model : $\boldsymbol{o_t \sim q_\theta(o_t|s_t)}$

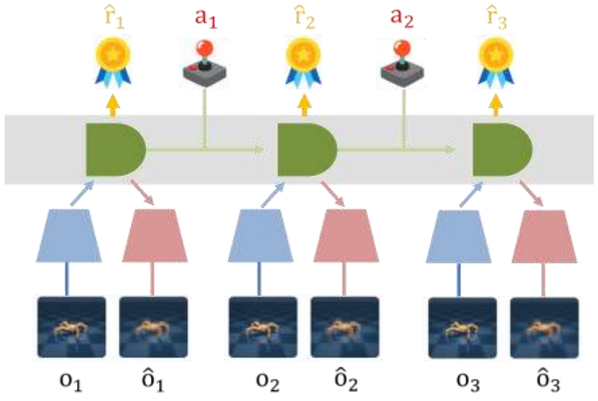
Reward model : $\boldsymbol{r_t \sim q_\theta(r_t|s_t)}$

Methods

Dreamer



- **Dreamer with RSSM**
  - **Dreamer의 dynamics learning은 Reconstruction Objective function $\mathcal{J}_{\mathcal{REC}}$ 을 기반으로 업데이트**
  - **목적 함수를 최대화하는 목표를 설정 -> $\mathcal{J}_{\mathcal{O}}^{t}, \mathcal{J}_{\mathcal{R}}^{t}, \mathcal{J}_{\mathcal{D}}^{t}$** 각 term log- likelihood 를 최대화
  - **KL-Divergence로 이루어진 $\mathcal{J}_{\mathcal{D}}^{t}$**는 representation model과 transition model 간 분포를 가깝게 만들 목적 -> prior/posterior 분포가 일관성을 갖도록



Deterministic state : $\boldsymbol{h_t = GRU(h_{t-1}, s_{t-1}, a_{t-1})}$

Transition model : $\boldsymbol{s_t \sim q_\theta(s_t | s_{t-1}, a_{t-1})}$

Representation model : $\boldsymbol{s_t \sim p_\theta(s_t | s_{t-1}, a_{t-1}, o_t)}$

Observation model : $\boldsymbol{o_t \sim q_\theta(o_t | s_t)}$

Reward model : $\boldsymbol{r_t \sim q_\theta(r_t | s_t)}$

(1) $\mathcal{J}_{\mathcal{REC}} \doteq E_p\left(\sum_t \left(\mathcal{J}_{\mathcal{O}}^{t} + \mathcal{J}_{\mathcal{R}}^{t} + \mathcal{J}_{\mathcal{D}}^{t}\right)\right) + \text{const}$

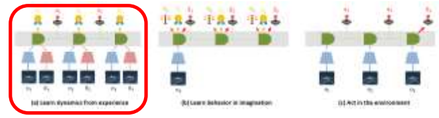
② ③ ④

(2) $\mathcal{J}_{\mathcal{O}}^{t} \doteq \ln q\left(o_t | s_t\right)$ **(3)** $\mathcal{J}_{\mathcal{R}}^{t} \doteq \ln q\left(r_t | s_t\right)$

(4) $\mathcal{J}_{\mathcal{D}}^{t} \doteq -\beta\, \mathrm{KL}\left(p(s_t | s_{t-1}, a_{t-1}, o_t) \,\|\, q(s_t | s_{t-1}, a_{t-1})\right))$

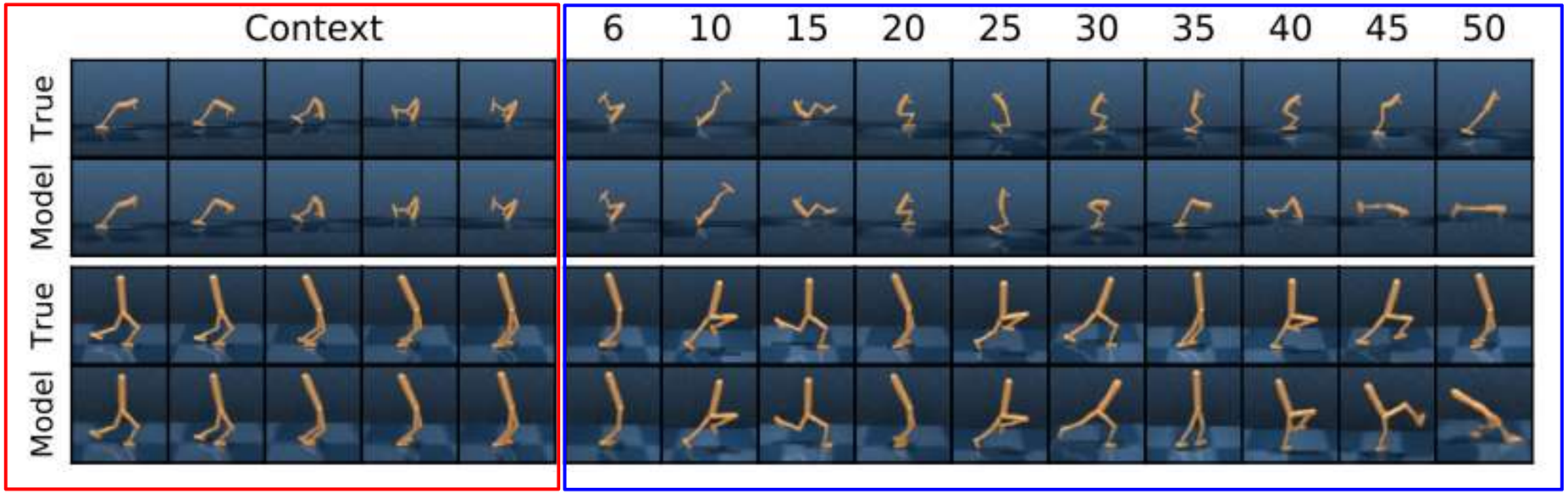
Methods

Dreamer



- **Effects of RSSM**
  - Dreamer는 학습된 World Model의 잠재 공간에서 가상의 행동을 상상 및 학습
  - **World Model이 실제 환경의 미래 상태와 유사한 예측을 수행해야 최적의 정책 도출 가능**
  - **RSSM 시각화 과정을 통해, 실제 환경의 시각적 변화를 효과적으로 포착하고 장기간에 걸쳐 일관된 예측 수행을 확인**

Context(5) 이후, 예측한 미래 상태 및 행동에 대한 더 먼 미래의 결과 예측

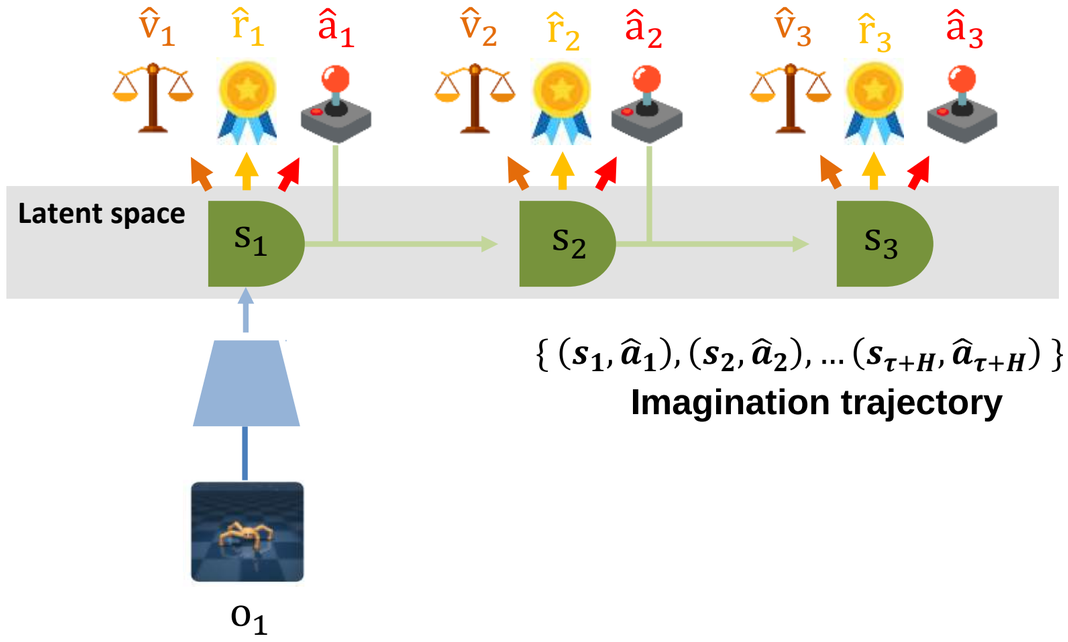


World Model이 예측을 시작하기 위해 사용한 초기 5개 이미지(경험)

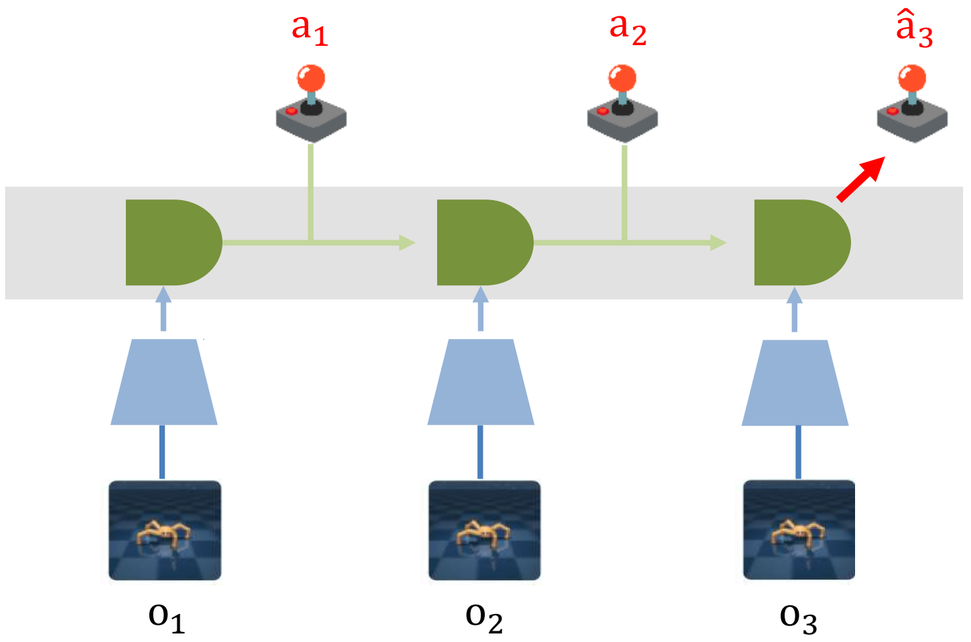
Methods

Dreamer

- **Learning behaviors by latent imagination**
  - **<u>Imagination</u> :** 학습된 World Model 내 잠재 가상 공간내에서 미래 행동의 결과를 미리 예측하고 정책을 최적화하는 과정
  - Dynamics learning에서 학습된 transition, reward model과 초기 policy model의 예측을 따르며 진행 → **Imagination trajectory 생성**
  - **잠재 공간에서 생성된 Imagination trajectory는 추후 에이전트 컴포넌트 업데이트에 사용되는 경험**



(b) Learn behavior in imagination

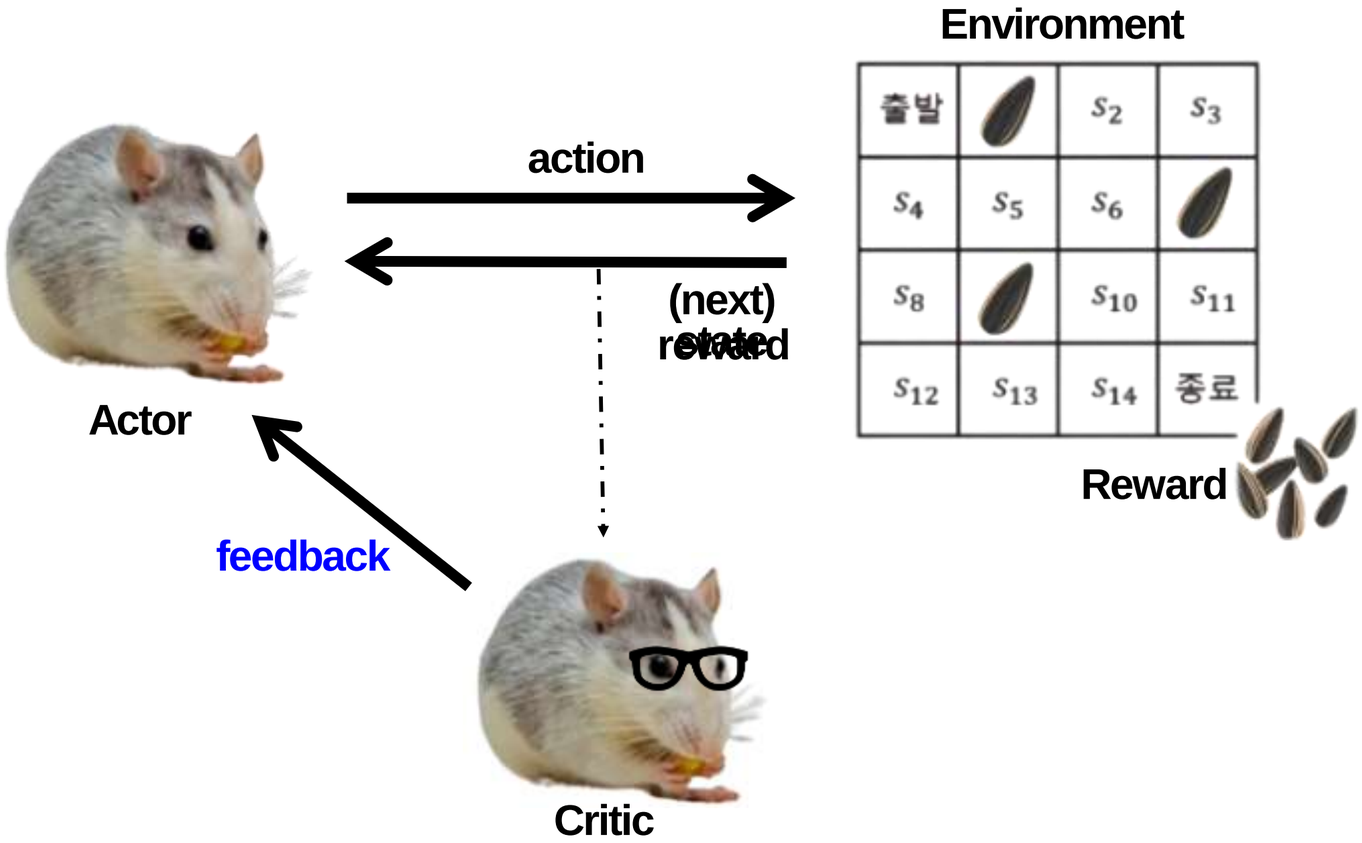


(c) Act in the environment

Methods

Dreamer

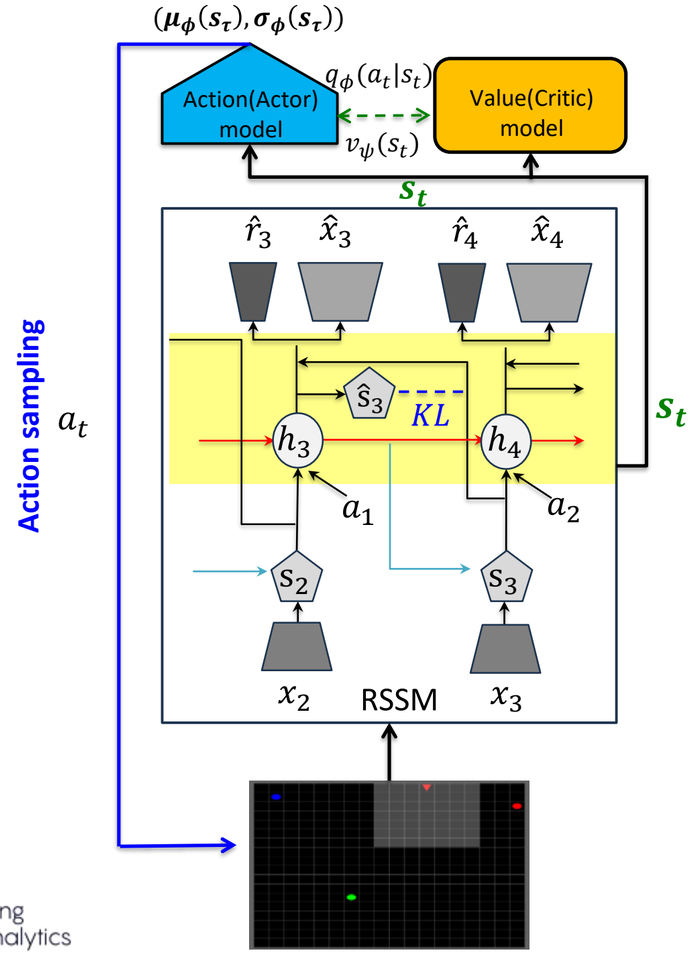
- **Learning behaviors by latent imagination**
  - Dreamer의 에이전트 및 정책 학습은 강화학습의 **Actor-Critic network** 구조 사용
  - 환경에서 Actor가 환경을 취하면, Critic은 가치 함수로 Actor의 행동(정책)을 평가
  - Actor는 행동에 대한 평가 정보를 기반으로 정책 업데이트 반복



Methods

Dreamer

- **Learning behaviors by latent imagination**
  - **Imagination은 학습된 World Model 내 잠재 가상 공간내에서 미래 행동의 결과를 미리 예측하고 정책을 최적화하는 과정**
  - **Dynamics learning에서 학습된 transition, reward model과 초기 policy model의 예측을 따르며 진행 → Imagination trajectory 생성**



Action model: $a_\tau \sim q_\phi(a_\tau|s_\tau)$

Value model: $v_\psi(s_\tau) \approx E_{q(.|s_\tau)}\left(\sum_{\tau'=t}^{t+H} \gamma^{\tau'-t} r_{\tau'}\right)$

Action sampling: $a_\tau = tanh\left(\mu_\phi(s_\tau) + \sigma_\phi(s_\tau)\epsilon\right),\ \epsilon \sim Normal(\mathbf{0}, \boldsymbol{I}).$

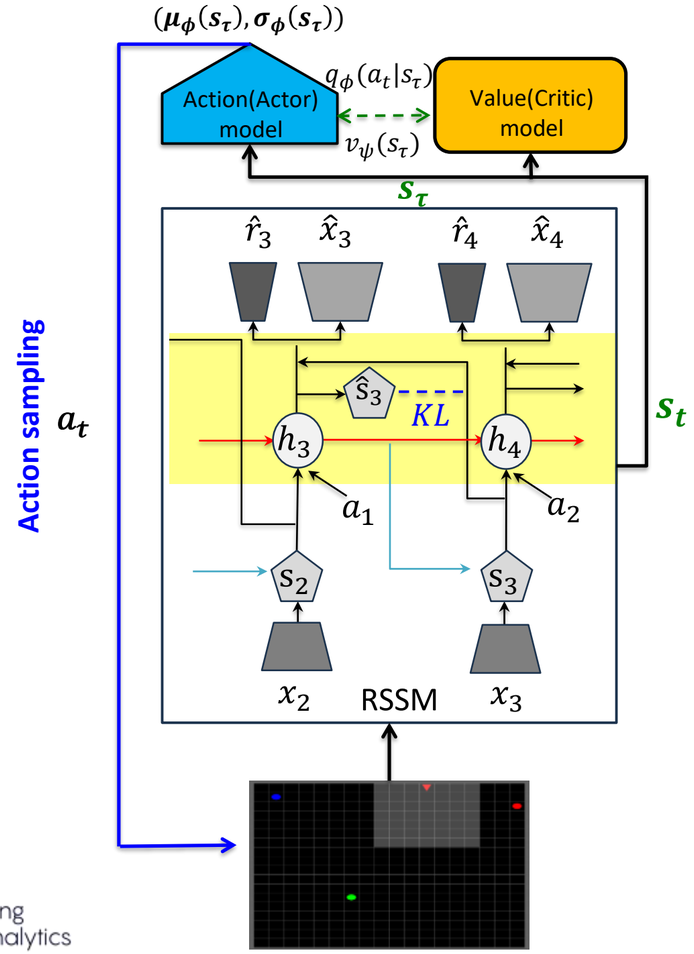
Objective of Action model :

$$(1)\ \max_\phi E_{q_\theta, q_\phi}\left(\sum_{\tau=t}^{t+H} V_\lambda(s_\tau)\right) \quad (2)\ E_{q_\theta, q_\phi}\left(\frac{1}{2} \| v_\psi(s_\tau) - V_\lambda(s_\tau) \|^2\right) \quad < V_\lambda : \text{Value estimator} >$$

Methods

Dreamer

- **Learning behaviors by latent imagination**
  - **Imagination은 학습된 World Model 내 잠재 가상 공간내에서 미래 행동의 결과를 미리 예측하고 정책을 최적화하는 과정**
  - **Dynamics learning에서 학습된 transition, reward model과 초기 policy model의 예측을 따르며 진행 → Imagination trajectory 생성**



Action model: $a_\tau \sim q_\phi(a_\tau|s_\tau)$

Value model: $v_\psi(s_\tau) \approx E_{q_\theta,q_\phi}\left(\sum_{\tau'=t}^{t+H} \gamma^{\tau'-t} r_{\tau'}\right)$

Action sampling: $a_\tau = tanh\left(\mu_\phi(s_\tau) + \sigma_\phi(s_\tau)\epsilon\right),\ \epsilon \sim Normal(0, I).$

Objective of Action model :

$$(1)\max_\phi E_{q_\theta,q\phi}\left(\sum_{\tau=t}^{t+H} V_\lambda(s_\tau)\right)(2)E_{q_\theta,q_\phi}\left(\frac{1}{2}\parallel v_\psi(s_\tau) - V_\lambda(s_\tau)\parallel^2\right)$$

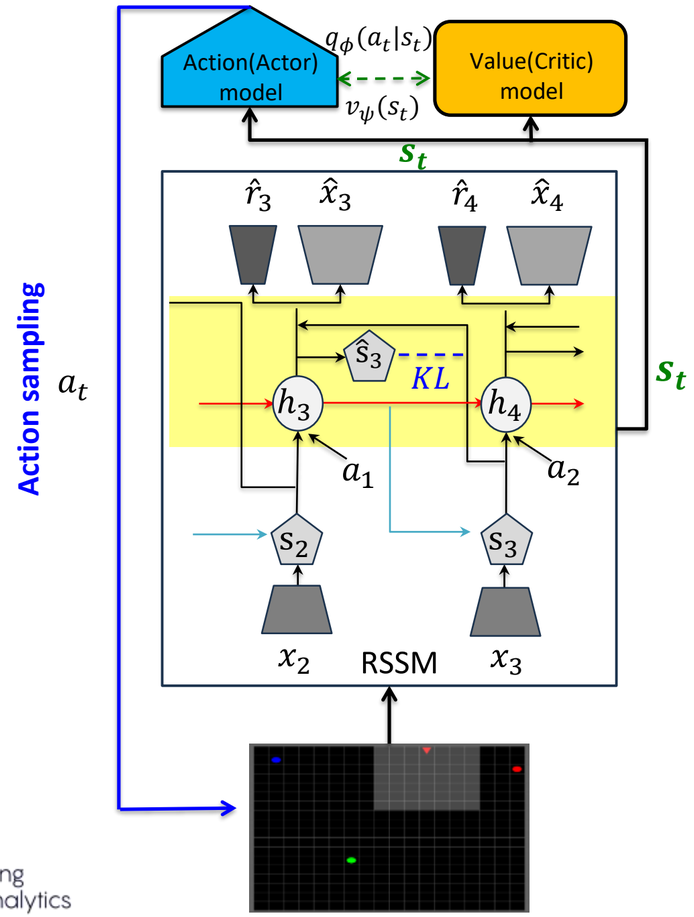
$$V_N^k(s_\tau) \doteq E_{q_\theta,q_\phi}\left(\sum_{n=\tau}^{h-1}\gamma^{n-\tau} r_n + \gamma^{h-\tau} v_\psi(s_h)\right) \quad \text{with} \quad h = \min(\tau + k, t + H)$$

$$V_\lambda(s_\tau) \doteq (1-\lambda)\sum_{n=1}^{H-1}\lambda^{n-1} V_N^n(s_\tau) + \lambda^{H-1} V_N^H(s_\tau),$$

Methods

Dreamer

- **Learning behaviors by latent imagination**
  - **Imagination은 학습된 World Model 내 잠재 가상 공간내에서 미래 행동의 결과를 미리 예측하고 정책을 최적화하는 과정**
  - **Dynamics learning에서 학습된 transition, reward model과 초기 policy model의 예측을 따르며 진행 → Imagination trajectory 생성**



Action model: $a_\tau \sim q_\phi(a_\tau|s_\tau)$

Value model: $v_\psi(s_\tau) \approx E_{q_\theta,q_\phi}\left(\sum_{\tau'=t}^{t+H} \gamma^{\tau'-t} r_{\tau'}\right)$

Action sampling: $a_\tau = tanh\left(\mu_\phi(s_\tau) + \sigma_\phi(s_\tau)\epsilon\right),\ \epsilon \sim Normal(0, I).$

Objective of Action model :

$$(1)\ \max_\phi E_{q_\theta,q_\phi}\left(\sum_{\tau=t}^{t+H} V_\lambda(s_\tau)\right) \quad (2)\ E_{q_\theta,q_\phi}\left(\frac{1}{2}\parallel v_\psi(s_\tau) - V_\lambda(s_\tau)\parallel^2\right) \quad < V_\lambda : \text{Value estimator} >$$

Update $\phi \leftarrow \phi + \alpha\nabla_\varphi \sum_{\tau=t}^{t+H} V_\lambda(s_\tau)$ Update $\psi \leftarrow \psi - \alpha\nabla_\psi \sum_{\tau=t}^{t+H} \frac{1}{2}\left\|v_\psi(s_\tau) - V_\lambda(s_\tau)\right\|^2$

Experiments

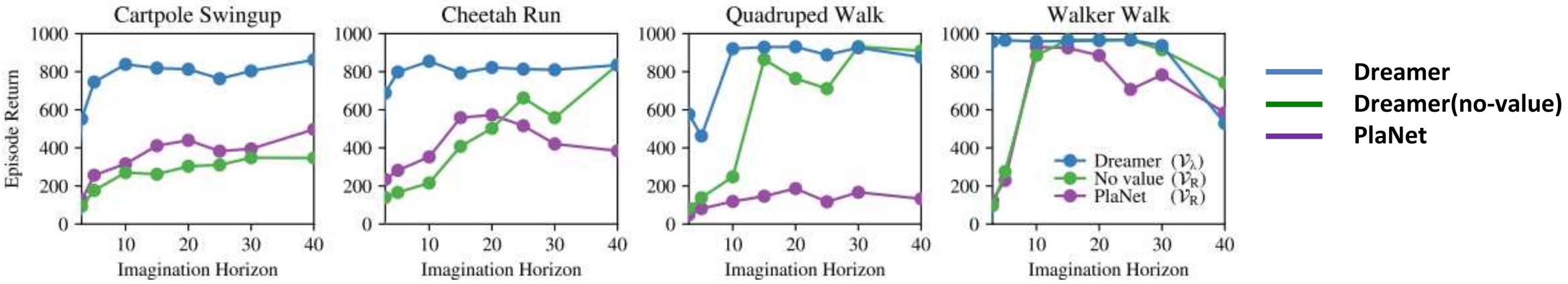
- **Experiments**
  - Dreamer의 성능을 평가하기 위해 20개의 DMC 환경에서 시각적 제어 task를 실험
  - 장기 시퀀스(Long-hrizon), 연속 액션, 이산 액션 등의 상황에 대한 해결 능력 평가

DMC Env

Experiments

Dreamer

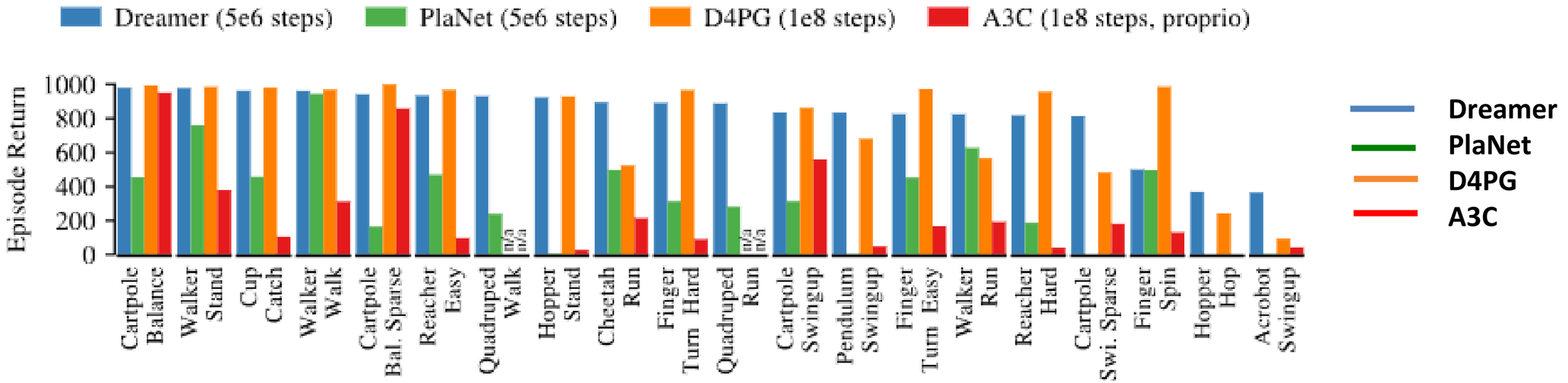
- **Experiments : Imagination horizons**
  - Dreamer의 핵심적 아이디어인 상상 예측 범위(Imagination Horizon, H)가 에이전트의 성능에 미치는 영향 측정
  - Imagination Horizon : 에이전트가 가상 환경에서 미래 행동을 계획(planning)할 때, H-time step 이후 까지 예측
  - **실험을 통해 Dreamer가 Imagination horizon length (H)의 길이에 영향을 덜 받는 경향을 확인**
  - **Dreamer의 value model이 Imagination 과정에서 직접 얻을 수 없는 보상까지 고려하여 정책에 반영 → 장기적 목표 달성**



(a) walker-walk **(b) cheetah-run** **(c) door-open**

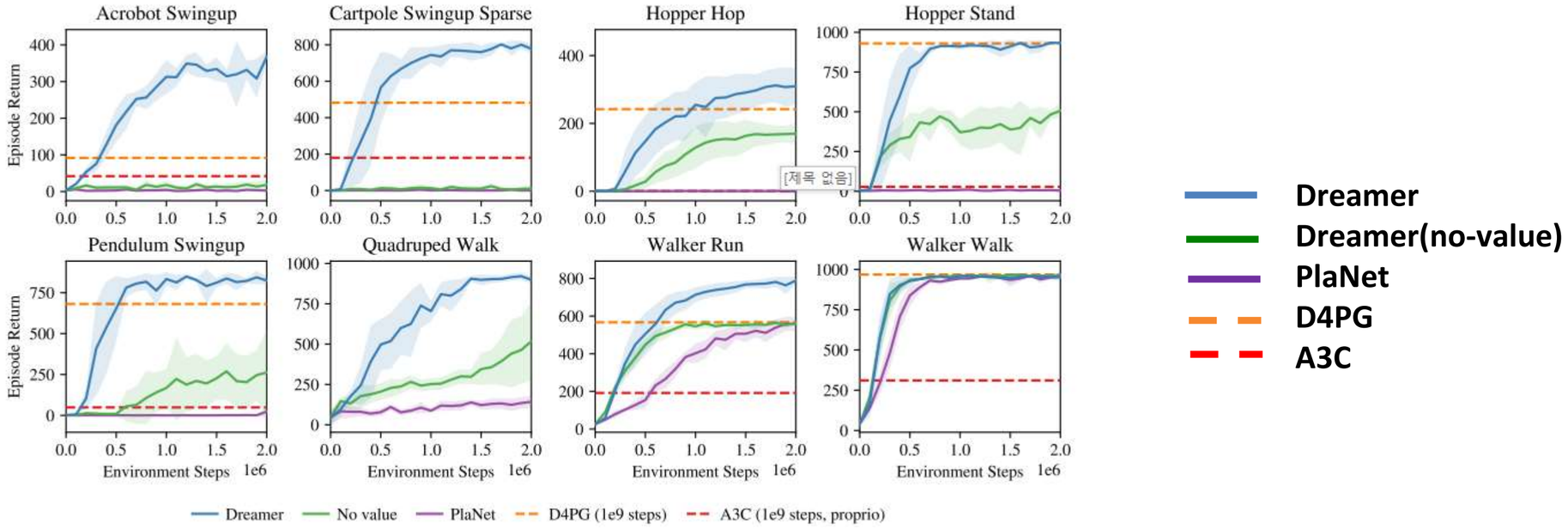
Experiments

- **Experiments : Performamce comparison to existing methods**
  - Dreamer와 기존 RL 에이전트 간 비교 실험 수행 (M-B : 2/M-F : 2)
  - **Dreamer는 Model-Free에 비해 훨씬 적은 step으로 동등하거나 우수한 성능 달성**
  - **Dreamer는 잠재 공간에서 Actor-Critic으로 action, value model을 학습하여 World Model 계열인 Planet을 능가**
  - **Dreamer의 World Model과 behavior 학습은 latent imagination을 통해 연속적, 장기적 task에 대한 정책을 강건히 학습**



Experiments

- **Experiments(Visual control task)**
  - 관측 정보를 image로 입력 받아 환경을 제어하는 task
  - 총 7개의 환경에서 다른 방법론보다 높은 보상을 획득하거나 최상위 성능 기록 → **복잡한 시각 제어 작업에 특화**
  - **대부분의 경우 Dreamer는 더 적은 상호작용 스텝 내에서 Model-Free의 최종 성능과 유사하거나 능가하는 경향**
  - **Dreamer(no-value) 버전은 Dreamer(full)보다 현저히 낮은 성능 → 잠재 공간 상 value model을 사용하며 보상 예측 및 정책 최적화의 여부**



Conclusion

Summary

01

Model-Based Reinforcement Learning

- World Model (2018) : 인간의 인지적 추론 과정을 모방한 환경 모델링 방법론 제안
- Dreamer (2020) : 기존 World Model을 발전시키고 장기적 예측 및 잠재 동역학 학습

02

World Models (2018)

- 인지적 추론을 모방한 V, M, C model로 이루어진 가상 환경 모델링 방법론
- **Dream이란 과정을 통해 과거의 이력을 기반으로 미래 상태를 먼저 예측하여 정책 학습 가능**

03

Dreamer-v1 (2020)

- World Model의 한계점을 해결하고 Dream 과정을 발전시켜 더 먼 미래까지 예측 및 행동 학습 가능
- RSSM을 사용하여 결정론적/확률론적 잠재 상태 모두 고려 → **결정적인 이벤트 정보를 보존하며 미래의 불확실성에 대비**
- Action model을 사용함으로써 그라디언트 기반 업데이트를 수행하고 Value model로 더 넓은 범위의 보상을 예측

04

World Model 방법론들은 신용할당(Credit Assignment Problem, CAP)에 효과적

- CAP : 획득한 보상의 원인인 행동을 추적하지 못하거나, 학습에 결정적인 이벤트를 파악하지 못하는 문제점
- World Model 계열 방법론들은 잠재 공간에서 압축된 데이터로 여러 미래를 예측하고 검증하므로 매우 효과적으로 알려짐

Thank You

Data Mining
Quality Analytics