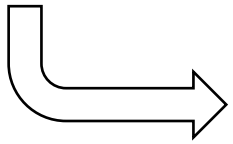
Generalization to New Actions in Reinforcement Learning

Generalization to New Actions in Reinforcement Learning

Ayush Jain [*1] **Andrew Szot** [*1] **Joseph J. Lim** [1]



Published as a conference paper at ICLR 2022

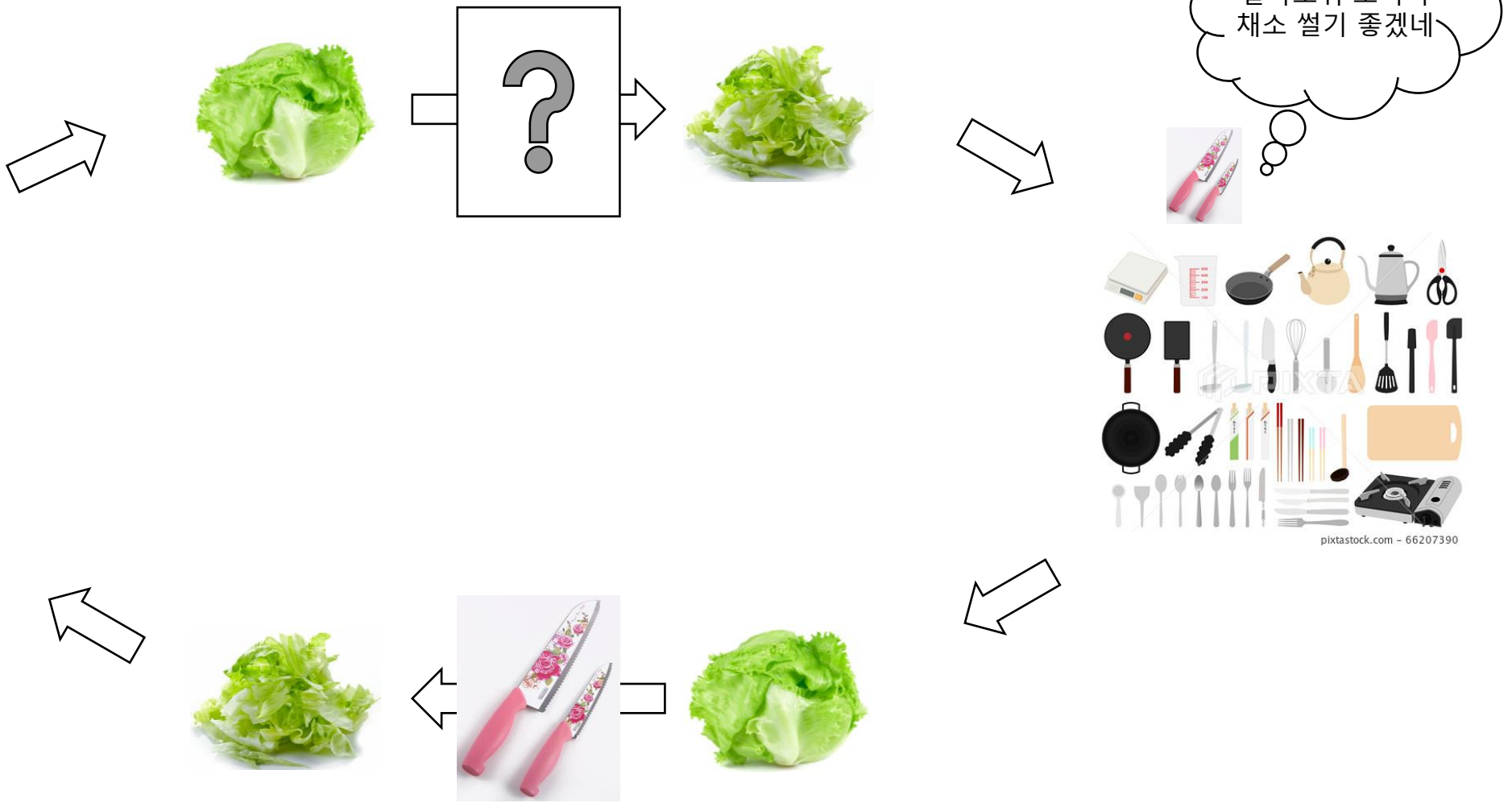
KNOW YOUR ACTION SET: LEARNING ACTION RELATIONS FOR REINFORCEMENT LEARNING

Ayush Jain[*1] **Norio Kosaka**[*2] **Kyung-Min Kim**[2 4] **Joseph J. Lim**[3† 4‡]
[1]University of Southern California (USC), [2]NAVER CLOVA,
[3]Korea Advanced Institute of Science and Technology (KAIST), [4]NAVER AI Lab

- KAIST 임재환 교수님 Cognitive Learning for Vision and Robotics Lab 연구
- ICML 2020 → ICRL 2022 후속 연구 진행

Introduction

- 인간이 salad 를 만들 때 도구를 선택하는 과정



Introduction

- 강화학습에서 이전에 보지 못했던 action 을 포함해 task 를 해결하는 문제는 중요한 숙제
- 본 논문에서는
  1. Robot 이 이전에 보지 못했던 toolkit 을 가지고 task 를 수행하거나
  2. 추천 시스템에서 새로운 products 에 대한 추천을 제안하거나
  3. Hierarchical reinforcement learning 에서 agent 가 새로 습득한 skill set 을 가지고 task 를 해결

  하는 등의 문제를 풀기 위해서 retraining 하는 과정 없이 action generalization 에 대한 방법 제시

Problem Formulation

- Agent 가 unseen actions 에 대해 retraining 없이 추론을 통한 toolkit 활용이 가능하도록 하는 것이 목적
- 문제는 discrete action spaces 를 가지는 기본적인 MDP 로 정의
- Generalization to new actions 은 학습과 평가 두 단계를 거침
  학습 단계에서는 주어진 action set 을 통해 학습
  평가 단계에서는 unseen actions 로 부터 샘플링 된 새로운 action set 을 가지고 평가

Approach

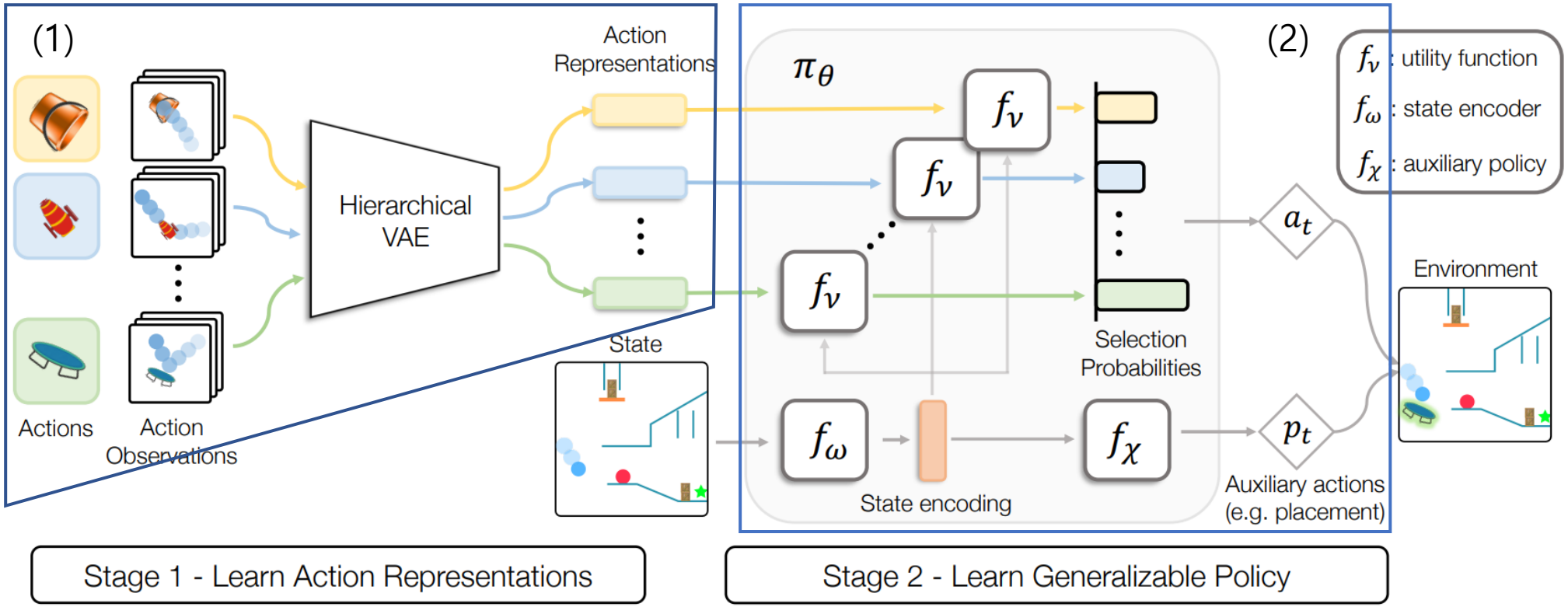
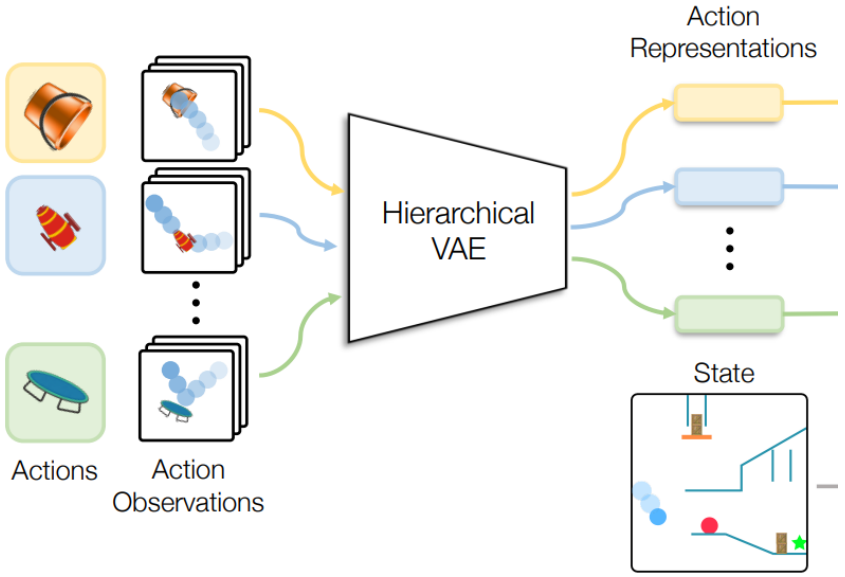


Figure 2. Two-stage framework for generalization to new actions through action representations. (1) For each available action, a hierarchical VAE module encodes the action observations into action representations and is trained with a reconstruction objective. (2) The policy π_θ encodes the state with state encoder $f_\omega(s)$ and pairs it with each action representation using the utility function f_ν. The utility scores are computed for each action and output to a categorical distribution. The auxiliary network takes the encoded state and outputs environment-specific auxiliary actions such as tool placement in CREATE. The policy architecture is trained with policy gradients.

Approach

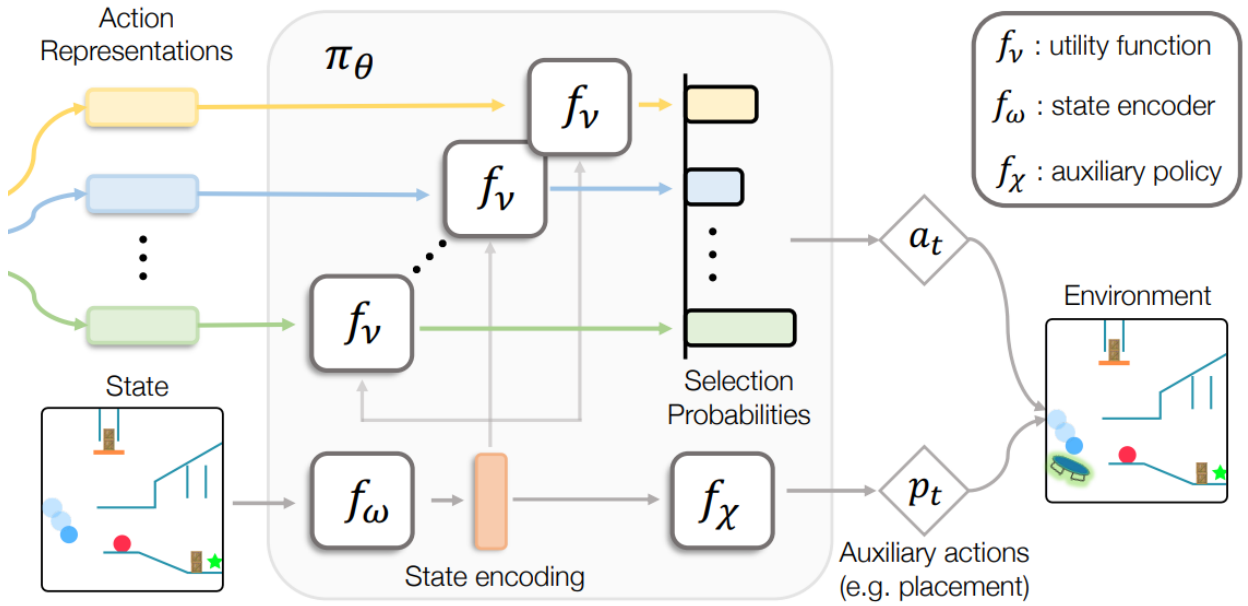
(1) Unsupervised Learning of Action Representation



- Set of action observations 을 가지고 action representation 하기 위해 unsupervised learning 진행
- Action observations 를 input 으로 representation latent space 를 학습하기 위해 VAE 중 action observations 의 sequential 한 특성 반영이 가능한 HVAE 적용
- HVAE : 기존 VAE sequential 한 정보나 다수 objects 에 대한 latent space distribution 까지 고려할 수 있도록 decoder 에 한 계층을 더 둔 구조

Approach

(2) Adaptable Policy Architecture



- State encoding 결과와 action representation 결과를 결합해(concat) utility network 결과 출력
- Utility network 출력 결과 score 는 softmax 를 통해 action 에 대한 probability 로 활용
- Tool 에 대한 placement 를 결정하기 위한 auxiliary policy 를 따로 구성

Approach

- Training
  - 학습 할 때 generalization 성능을 높이고 overfitting 을 방지하기 위해
    각 episode 마다 action 중 m 개의 랜덤한 action set 을 sampling 해서 진행
  - Entropy parameter 를 두어 exploration 이 많이 일어나도록 학습
  - Unseen + seen action 으로 이루어진 validation set 을 따로 두어 평가

Algorithm 1. Two-stage Training Framework

Inputs: Training actions $\mathbb{A}$, action observations $\mathbb{O}$
Randomly initialize HVAE and policy parameters
for epoch = 1, 2, . . . **do**
    Sample batch of action observations $\mathcal{O}_i \sim \mathbb{O}$
    Train HVAE parameters with gradient ascent on Eq. 2
end for
Infer action representations: $c_i = q_\phi^\mu(\mathcal{O}_i), \forall a_i \in \mathbb{A}$
for iteration = 1, 2, . . . **do**
    while episode not done **do**
        Subsample action set $\mathcal{A} \subset \mathbb{A}$ of size m
        Sample action $a_t \sim \pi_\theta(s, \mathcal{A})$ using Eq. 3
        $s_{t+1}, r_t \leftarrow \text{ENV}(s_t, a_t)$
        Store experience (s_t, a_t, s_{t+1}, r_t) in replay buffer
    end while
    Update and save policy θ using PPO on Eq. 4
end for
Select θ with best validation performance

Action representation 학습

Policy 학습

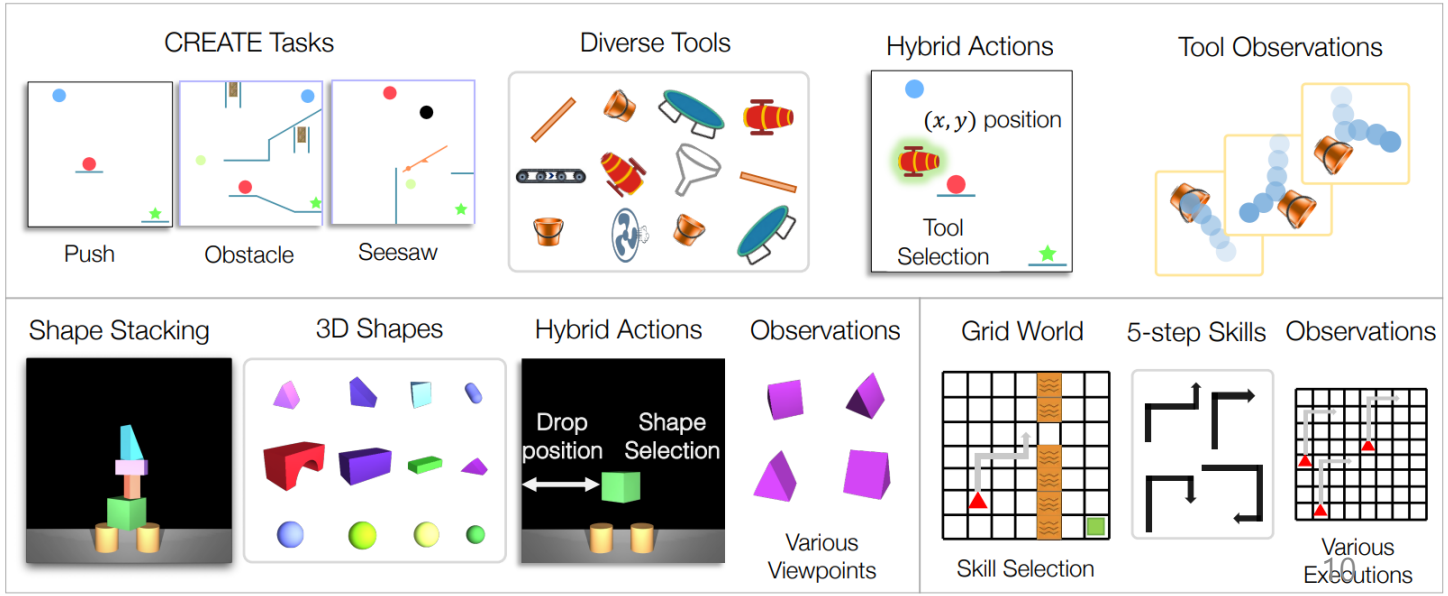
Environments

1. Grid World
   목적지에 도달하는 2D grid maze 환경에서 5 step move 결정 (action : left, right, up, down)

2. Recommender System
   User 에게 처음 보는 products 에 대해 click-through rate 가 높게끔 추천하는 환경
   여기서는 predefined action representation 사용

3. CREATE
   Physics-based 환경으로 tool 을 결정해 떨어지는 공을 목적지로 이동시키는 환경

4. Shape Stacking
   MujoCo-based 3D 탑 쌓기 환경



Experiments

- 실험 절차
  1. Task-independent 하게 모든 action 에 대해 action observations 수집
  2. Train, validation, test 를 위한 action 분리
  3. 수집한 action observations 로 HVAE 학습
  4. 학습 된 HVAE 를 통해 모든 actions 에 대한 representation 진행
  5. 각 episode 에서 actions 중 랜덤하게 sampling 해서 sub-set 을 만들어
     이를 가지고 policy 학습 진행
  6. Validation 혹은 test action set 에서 sampling 된 sub-set 을 가지고 평가 진행

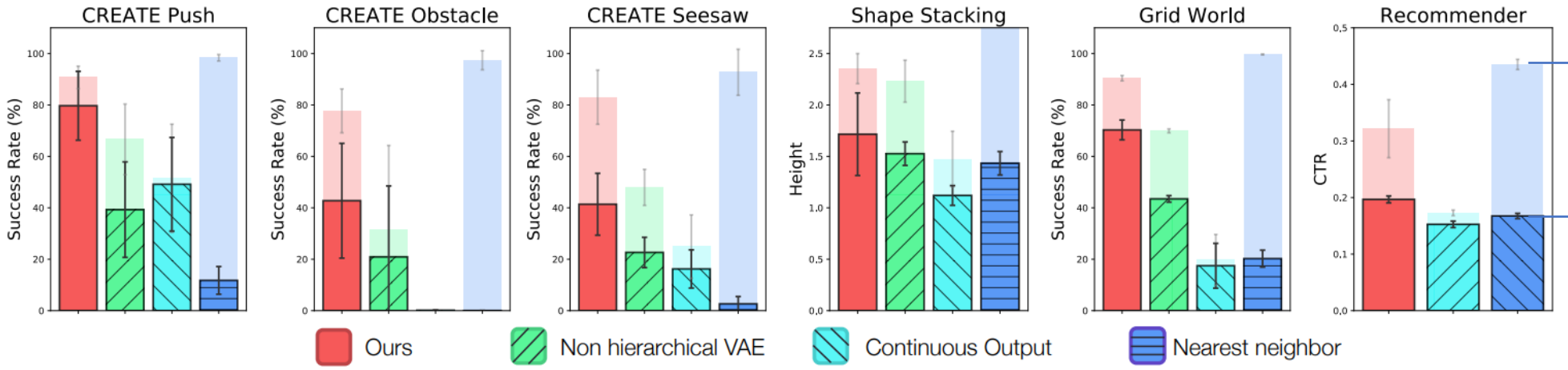
Experiments

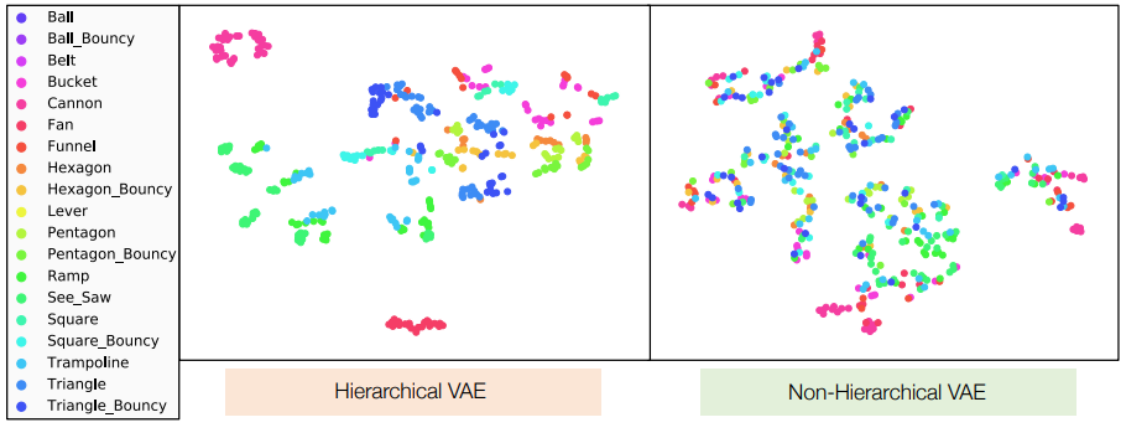
- Baselines
  - Non-hierarchical VAE : HVAE 와 VAE 비교
  - Continuous-output : policy output 을 continuous 로 변경해 비교
  - Nearest-Neighbor : discrete action policy 는 학습 된 상태에서
    new actions 에 대해 가장 비슷한 action 으로 선택하는 방법과 비교

- Ablations
  - Without subsampling : action set 에 대한 subsampling 없이 학습
  - Without entropy : entropy coefficient 를 zero 로 하고 학습 (no exploration)

Experiments

- Result - baselines

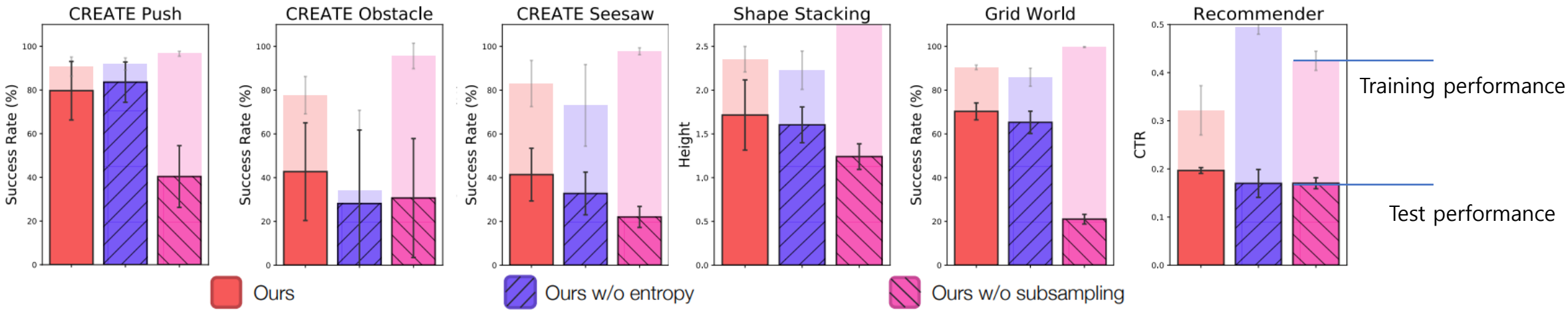




HVAE – VAE encoding 결과 t-SNE 분석
HVAE encoding 결과가 비슷한 종류의 tool 들을 잘 구분

Experiments

- Result - ablations

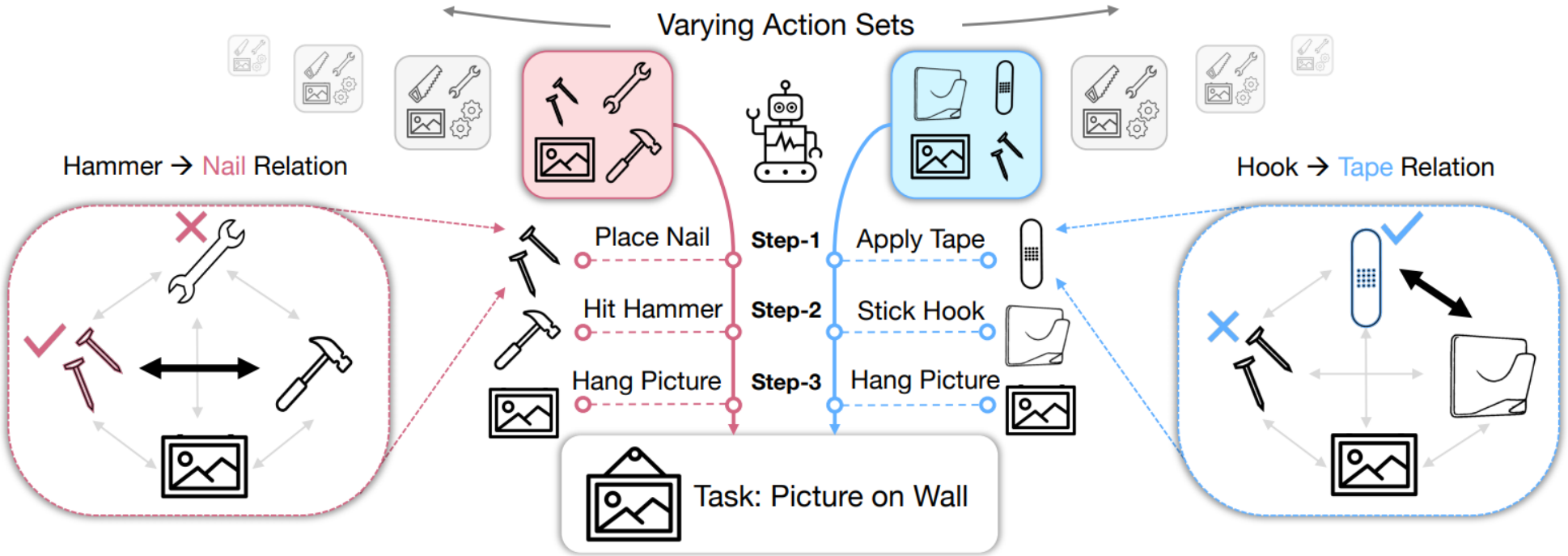


Entropy 를 제거했을 때, subsampling 을 하지 않았을 때
전체적인 성능 저하와 Training-Test 성능 간 gap 차이가 많이 발생

Know Your Action Set:

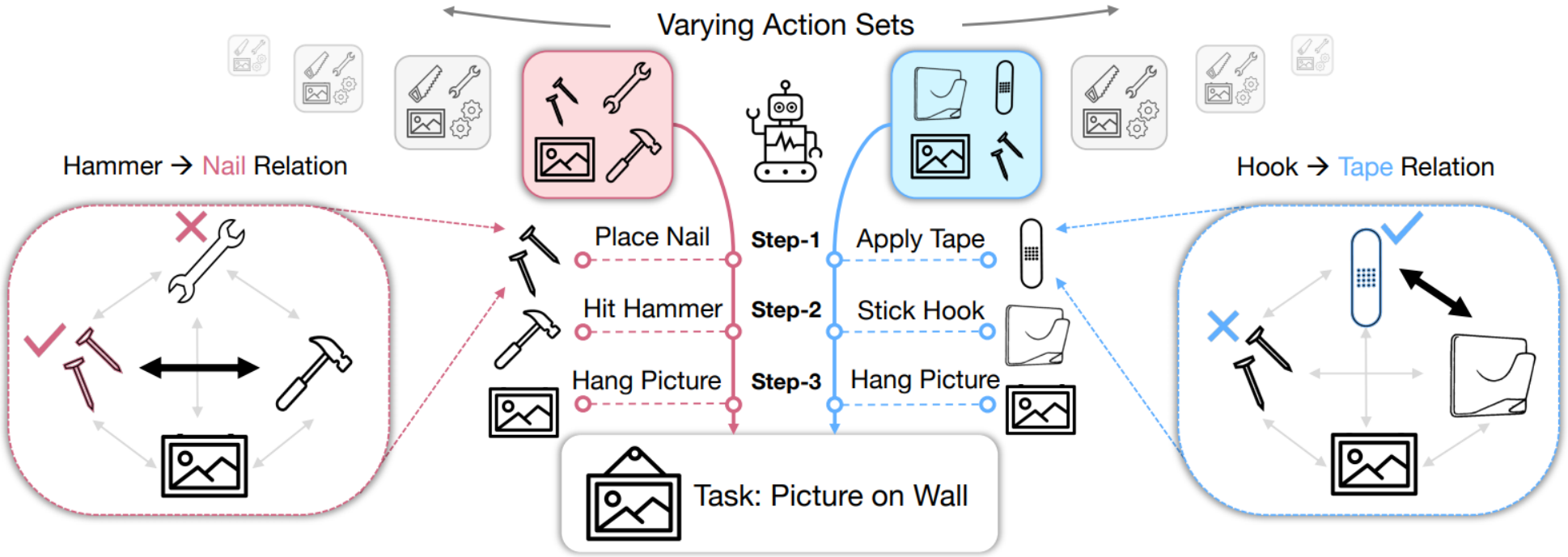
Learning Action Relations For Reinforcement Learning

Introduction



- 벽에 사진을 건다고 할 때, 못이 있는 경우 망치를 이용해 못을 벽에 박고 사진을 걸 수 있지만 못이 없을 경우 대안의 방식을 모색해야 함
- 이처럼 환경 뿐만 아니라 가능한 액션 또한 최적의 행동 결정에 영향을 주고 액션 간 상호의존성이 존재
- 본 논문에서는 Action 간 상호 의존 관계를 학습하고 representation 한 결과를 입력으로 각 action 에 대한 utility 값을 RL decision making 에 활용할 수 있는 구조 소개

Introduction



- 벽에 사진을 건다고 할 때, 못이 있는 경우 망치를 이용해 못을 벽에 박고 사진을 걸 수 있지만 못이 없을 경우 대안의 방식을 모색해야 함
- 이처럼 환경 뿐만 아니라 가능한 액션 또한 최적의 행동 결정에 영향을 주고 액션 간 상호의존성이 존재
- 본 논문에서는 Action 간 상호 의존 관계를 학습하고 representation 한 결과를 입력으로 각 action 에 대한 utility 값을 RL decision making 에 활용할 수 있는 구조 소개

Related Work

- List-wise action space
  - List-wise Approach : 추천 시스템에서 추천 아이템 리스트에 대한 ranking 을 계산하는 방식
  - 최근 Cascaded DQN 등 추천 시스템에서 action space 중 ranking 이 높은
    아이템 후보 리스트를 출력하고 이를 바탕으로 decision making 을 하는 식의
    추천 시스템 + RL 적용을 종종 연구
- Relational Reinforcement learning
  - 상호 관계성이 중요한 RL 문제(morphological control, multi-agent, physical construction)
    에 GNN 활용
  - 본 논문에서는 action 간 상호작용을 학습하기 위해 GAT 활용 제안
    GAT=Graph Neural network + Attention
    action 간 관계를 그래프 형태 구조로 보고 어떤 action 이 어떤 action 과 관계가 깊은지
    포커싱 하도록 학습

Problem Formulation

- 본 논문에서는 모든 Task instance 에서 매번 다른 action set 이 주어지고
  agent 가 어떤 action set 이 주어지던 최적 행동을 하도록 문제 설정
- The MDP is defined by a tuple $\{\mathcal{S}, \mathbb{A}, \mathcal{T}, \mathcal{R}, \gamma\}$

  S: state, T: transition, R: reward function, gamma: discount factor, A : countably infinite action set
- Varying Action Space 에 대한 Challenges
  - Using action representation: action representation C 를 input 으로 받아 network output 을
    RL 의 Q-value 나 probability distribution 에 대입할 수 있도록 policy framework 가 유연해야 함
  - Action set as part of state : action set 이 매번 달라지기 때문에 기존 state space 가 온전한 상태 정보를
    내포한다고 볼 수 없음 state 를 action representation 과 함께 재정의 $S' = \{s \circ \mathcal{C}_{\mathcal{A}} : s \in \mathcal{S}, \mathcal{A} \subset \mathbb{A}\}$
    따라서 policy framework 는 변하는 action set 에 대해 $s \circ \mathcal{C}_{\mathcal{A}}$ 를 입력으로 활용할 수 있어야 함
  - Interdependence of actions : 현재 최적 행동 선택은 미래의 가능한 행동에 의존적
    (못을 집는 행동은 망치가 사용 가능할 때 최적)
    따라서 최적 agent 는 현재 action 과 가능한 미래 action 사이 관계에 대해 명시적으로 모델링 되어있어야 함

Approach

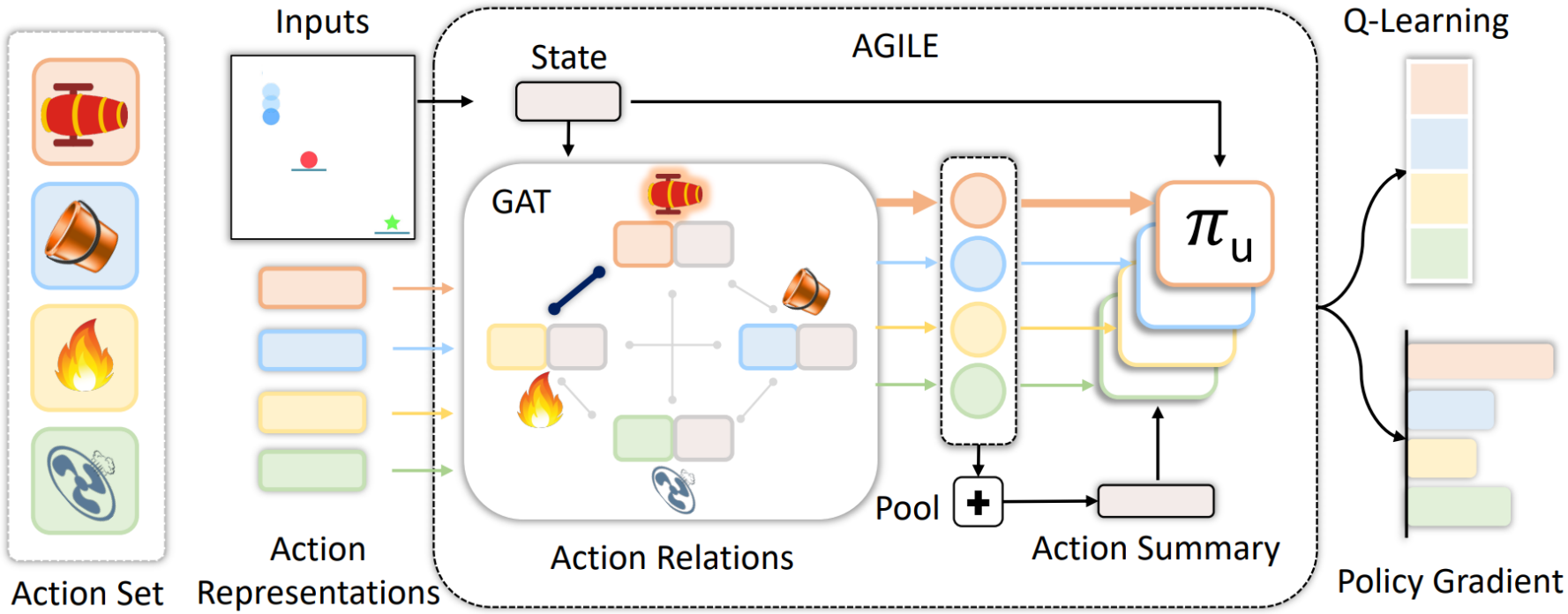


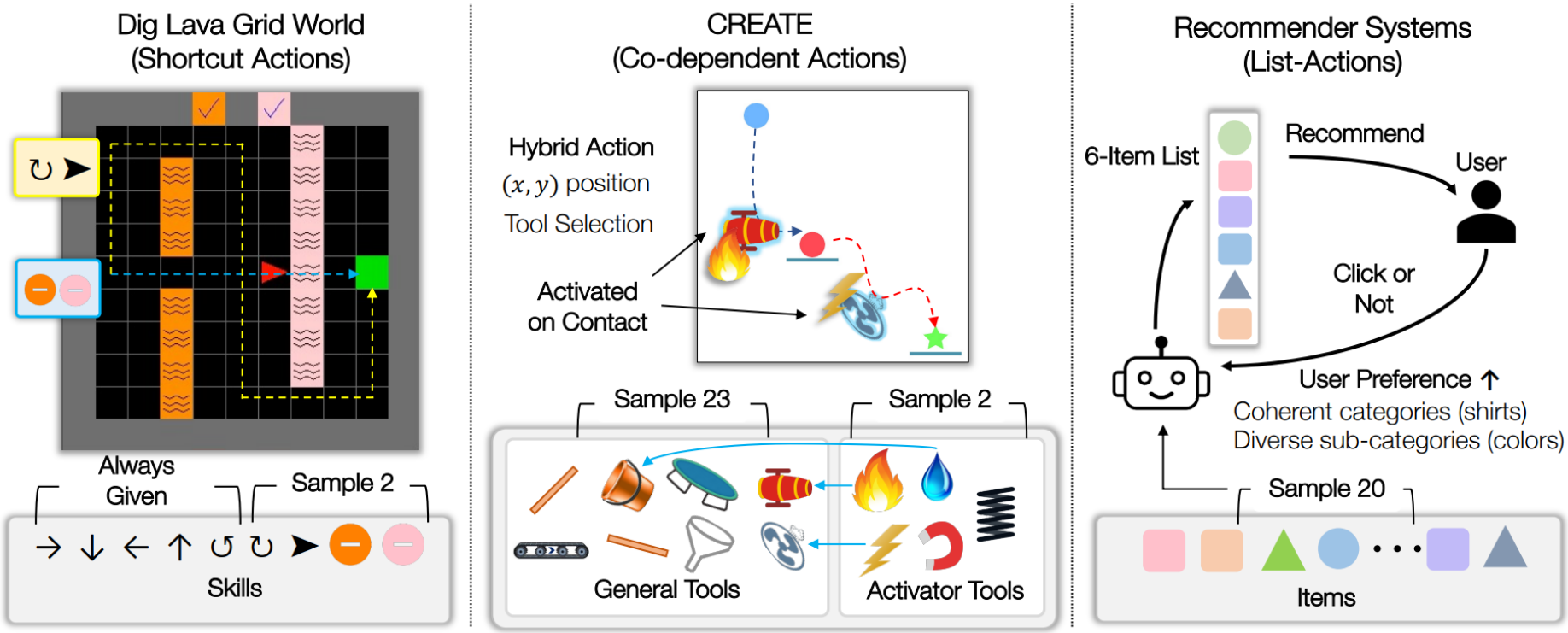
Figure 2: Given an action set, AGILE builds a complete graph where each node is composed of an action representation and the state encoding. A graph attention network (GAT) learns action relations by attending to other relevant actions for the current state. For example, the attention weight between the cannon and the fire is high because fire can activate the cannon. The GAT outputs more informed relational action representations than the original inputs. Finally, a utility network computes each action's value or selection probability in parallel using its relational action representation, the state, and a mean-pooled summary vector of all available actions' relational features.

Approach

- Action graph
  - 스크류드라이버가 가구 고칠 때는 관계가 있을 수 있지만 벽에 대해서는 무의미 하기 때문에
    특정 action 은 state 에 의존적
  - 따라서 본 논문에서는 action representation 뿐만 아니라 state 까지 고려한 action graph 구성
- Graph Attention Network
  - 가능한 action set 에서 가장 관련성이 높은 action 에 집중할 수 있도록 attention 기반 네트워크 활용
- Action set summary
  - GAT 에서 출력하는 정보는 가능한 action 과 action 간의 관계 정보
  - 현재 action set 을 state 와 함께 사용하기 위해 mean pooling 하여 summary 정보 활용
- Action utility
  - Relational action representation 을 RL 에 활용하기 위해 utility network 구조 적용
  - Relational action representation, state, action set summary 를 입력으로 utility score 를 출력하는 network 구성
  - 출력 Utility score 는 Q-value 나 probability distribution 에 사용 가능

Environments

- AGILE 구조를 3가지 varying action set scenario 로 검증 진행
  1. Goal-reaching 을 위한 간단한 actions scenario
  2. 상호 의존적 Tool reasoning
  3. List-action simulation 과 real-data recommender system



Environments

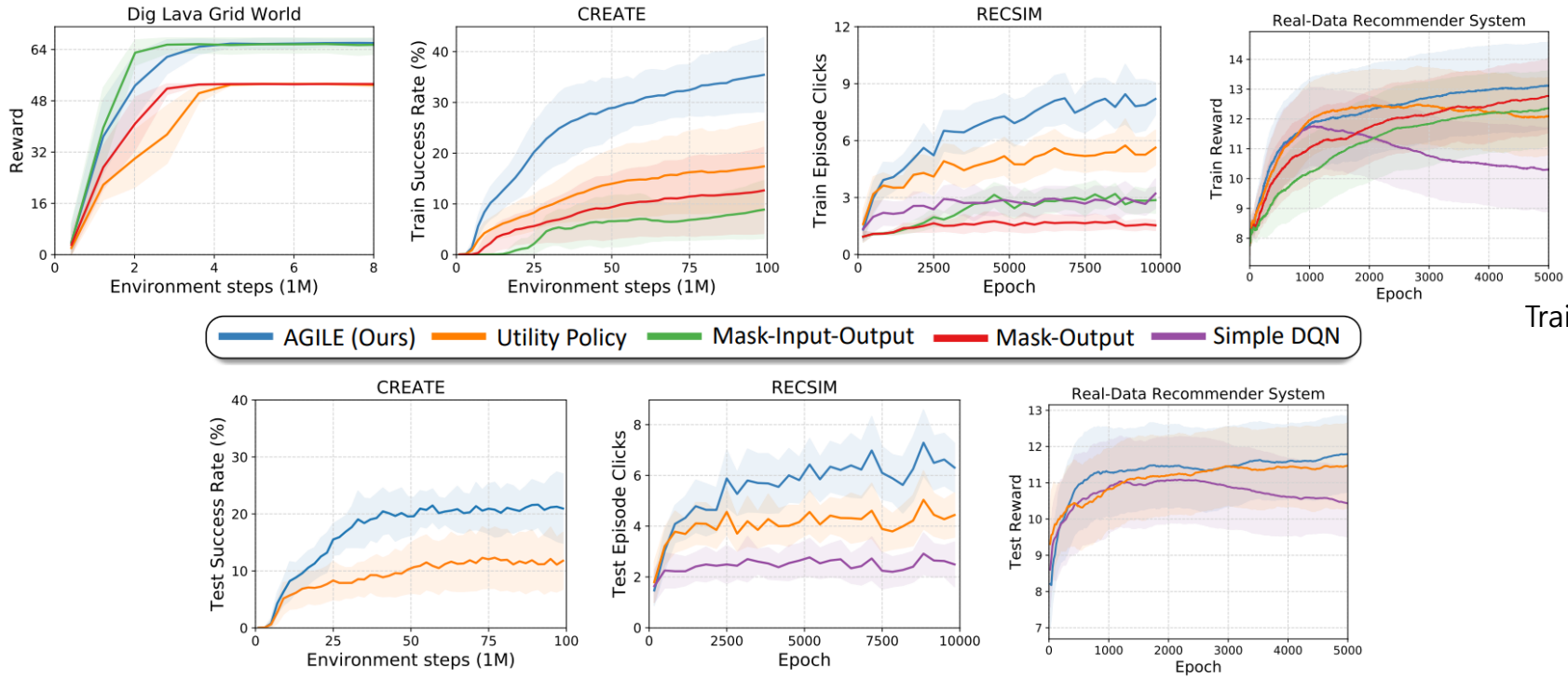
1. DIG LAVA GRID NAVIGATION
   - 2D grid 환경에서 goal 에 도달하는 것이 목표인 환경, PPO 활용
   - 4 방향 이동, 오른쪽 회전 action 은 항상 주어지고
     왼쪽 회전, 용암 파기 등 특수 스킬 4개 중 2개를 sampling 해서 학습
2. CHAIN REACTION TOOL ENVIRONMENT : CREATE
   - General tool 과 이걸 활성화 시킬 수 있는 activator tool 을 이용해 공을 바닥까지
     이동시키는 것이 목표, PPO 활용
   - 대포, 선풍기 등의 general tool 에서 23 개 샘플링, 전기 불 등의 activator tools 에서
     2개 샘플링 해서 학습
3. RECOMMENDER SYSTEMS
   - 가장 일반적인 varying action space RL 문제
   - RecSim : 구글에서 2019 발표한 list-wise recommendation task 활용, CDQN 활용
     250 개 아이템 중 에피소드 마다 20개 아이템 샘플링 후 agent 는 이 중 6개를 매 step 추천
   - Real-data Recommender system : LINE 온라인 광고 추천 서비스를 상대로 실험, CDQN 활용
     User 클릭 수 및 추천 리스트의 CPR 값으로 reward 구성

Experiments

- 실험을 통해 다음과 같은 사항 확인
  1. 이전 fixed action set 을 가정한 연구 대비 AGILE 이 효과적인지?
  2. Relational action representation이 action set summary, action utility score 계산에 얼마나 효과적인지?
  3. AGILE 의 attention 이 의미있는 결과를 보이는지?
  4. AGILE 에서 graph network 에 attention 이 필요한지?
  5. State-dependent action 관계를 학습하는 것이 일반적인 varying action space 문제를 해결하는데 중요한 요소인지?

Experiments

- Varying action space 에서 AGILE 성능 검증
  - Baseline 실험으로 기존 알고리즘 대비 AGILE 성능 비교
    - Mask-Output : 고정된 action space 를 가정하고 Q network 나 policy output 에서 불가능한 action masking
    - Mask-Input-Output : Mask-Output 에 더해, 각 action 이 활용 가능한지 아닌지 binary 정보를 입력 단에서 활용
    - Utility-Policy : action representation 과 각 action 에 대한 utility policy 를 활용하여 unseen action 을 다루는 네트워크

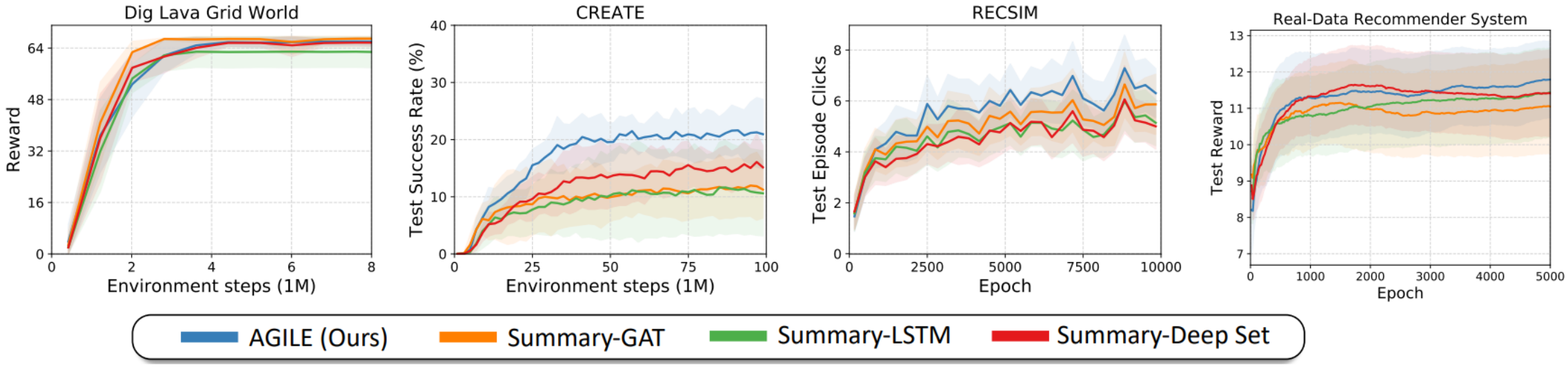


Train with assume fixed action spaces

Test with unseen actions

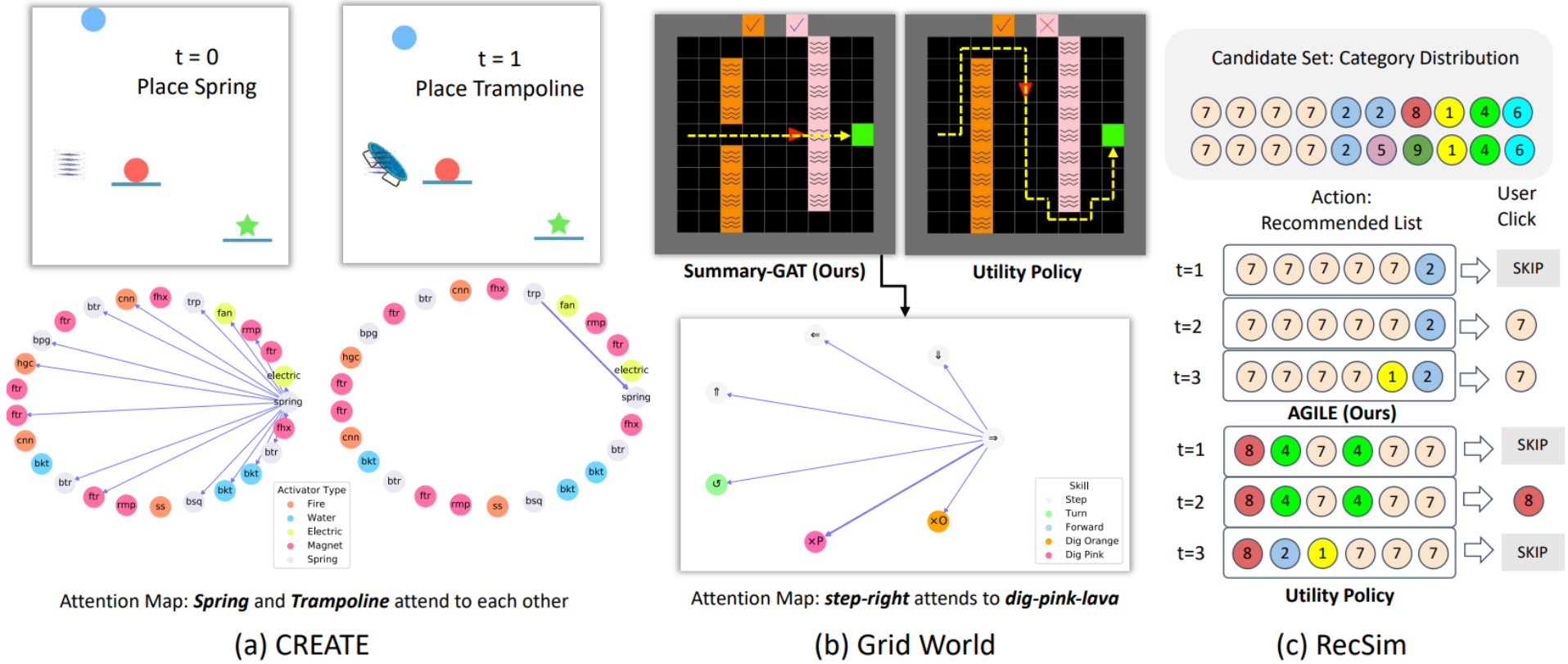
Experiments

- Varying action space 에서 AGILE 성능 검증
  - Ablation 실험으로 relational action feature 와 summary 효과 비교
    - Summary-LSTM : GAT summary 대신 LSTM summary 활용
    - Summary-Deep Set : GAT summary 대신 Deep set 이라는 summary 아키텍처 사용
    - Summary-GAT : GAT summary 을 action set summary 에만 활용 (utility network 의 입력으로 제외)



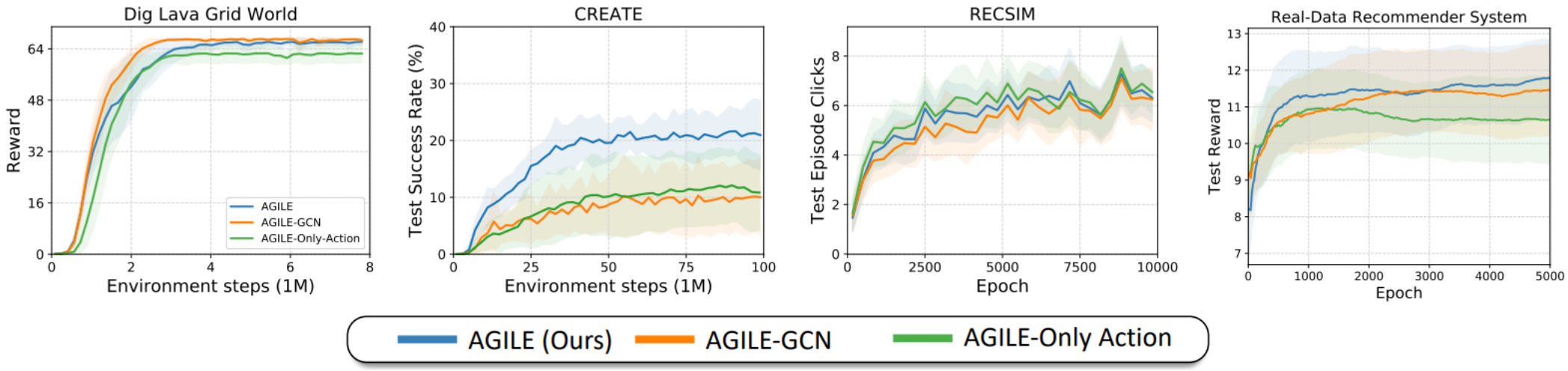
Experiments

- Attention 이 action relation 학습에 도움이 되는지 검증
  - Attention map 확인 시 trampoline 과 스프링 activator 에 대한 attention 이 높은 것 확인
  - Utility policy 대비 Summary-GAT 가 최적 policy 로 행동하는 것 확인
  - Utility policy 대비 AGILE 이 CPR 을 높이는(좀 더 공통 카테고리 아이템 추천) 방향 확인



Experiments

- Graph Attention Network 효용성 검증
  - GAT 를 GCN 으로 변경했을 때 실험 결과 비교



- 간단한 관계 학습은 GCN으로도 가능한 것을 보임(Dig, RECSIM),
  하지만 action 간 관계성이 중요한 CREATE, Real Rec에서는 성능이 떨어지는 것 확인
- GAT 입력으로 state 제외한 action representation 만 주었을 때 (AGILE-Only Action) 성능이 떨어짐
  RecSim 에서 CPR 은 user state 와 관계 없이 common category 가 중요