RL Research Group 20250918

FLaRe: Achieving Masterful and Adaptive Robot Policies with Large-Scale RL Fine-Tuning

(ICRA 2025)

김재훈

Introduction

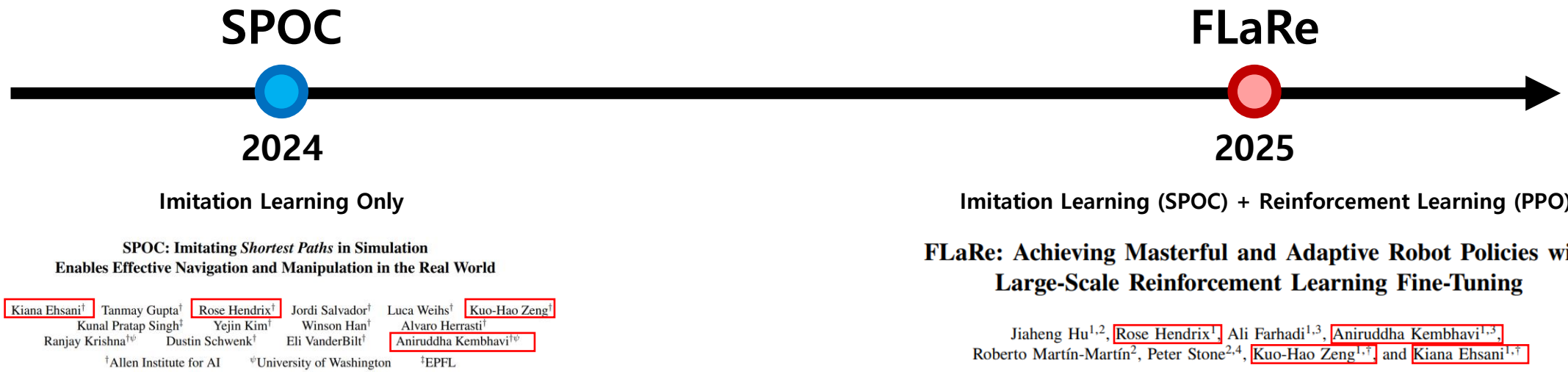
❖ Generalist Robot Policy 학습을 위해서는 강화학습(Reinforcement Learning; RL)이 필요함

- Large-scale multi-task 데이터로 모방학습(Behavior Cloning; BC) 기반 정책을 학습하였으나 일반화 성능이 떨어짐

  → 시뮬레이션 기반의 학습을 통한 compounding error 완화

  → Action sequence가 최종 목표 도달에 align되도록 튜닝

  - BC : 임의의 상태에서 학습한 최적의 행동을 예측하는 것이 목표
  - RL : 임의의 상태에서 최종 목표에 도달할 수 있도록 최적의 행동을 예측하는 것이 목표

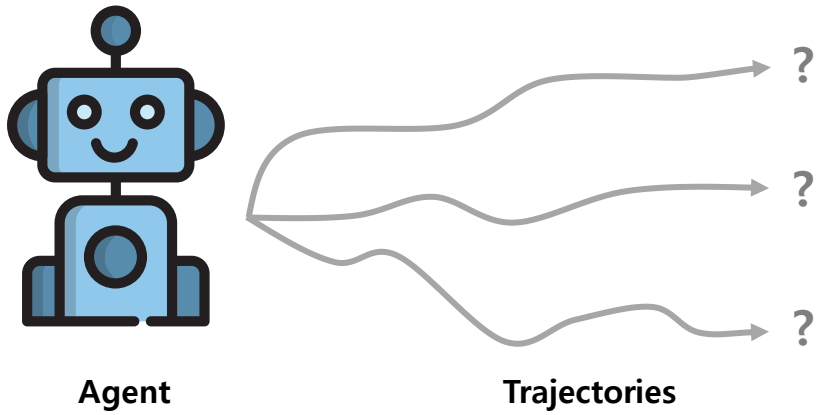


Introduction

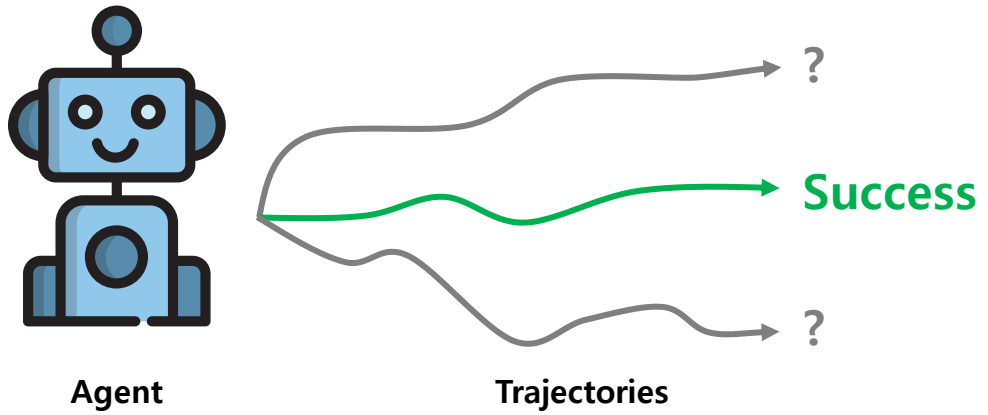
❖ Large-scale multi-task BC는 좋은 출발점

- 처음부터 trial-and-error를 통해서 학습하면 시간이 매우 오래 걸림 ← 순수 RL 방식의 한계점
- 따라서 large-scale network & data로 학습된 BC pretrained policy를 활용하면 보다 효율적인 학습이 가능

Training RL Policy



Training BC Policy



From scratch로 학습하기에는 비효율적
(특히 long-horizontal task의 경우)

Goal을 달성하는데 유용한 동작은 학습한 상태
(물론 앞서 언급했듯이 일반화 성능은 떨어짐)

Introduction

❖ FLaRe의 기여점 요약

- SoTA 모델(RL only - Poliformer w/ Dense Reward) 대비 최대 15배로 학습 시간을 단축함
- RL 기반 튜닝을 수행하여 일반화 성능 향상
- RL 튜닝을 거치면 사전학습된 정책도 새로운 로봇 형태에 쉽게 적용이 가능함을 확인

❖ FLaRe 보기 전에...

- SPOC로 학습된 모델을 사전학습 모델로 사용 (Sim-to-real 방식이나 실험 환경 역시 해당 논문의 세팅을 따라감)
- 단, 이 논문에서는 SPOC 구조를 약간 변경함 (LLaMA2 Decoder + DINOv2)
- RL 기반 fine-tuning을 할 때에는 sparse reward (0 or 1)만을 사용
- 새로운 아키텍처가 아닌 학습 프레임워크를 제안하는 논문

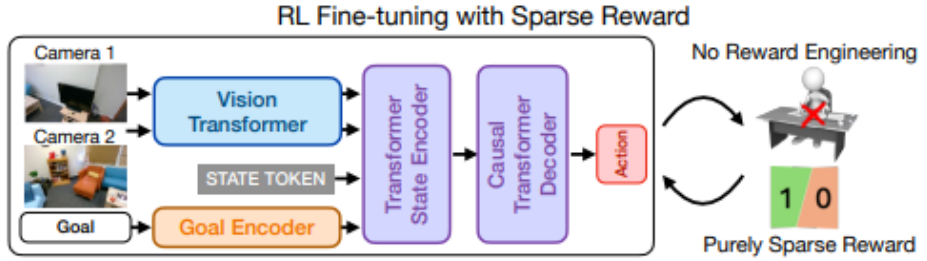


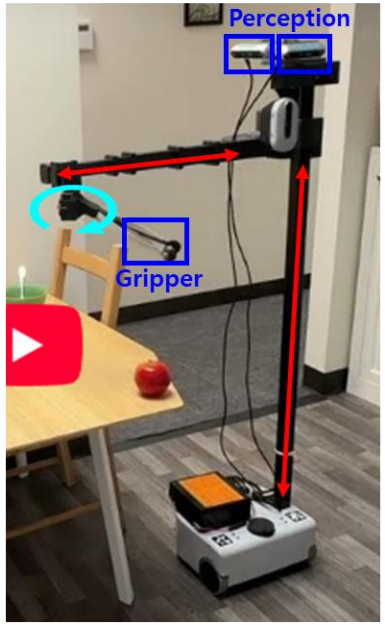
Fig. 3: FLaRe can efficiently fine-tune large transformer policies through large-scale Reinforcement Learning, using a sparse reward function that requires minimal human effort.

Hu, Jiaheng, et al. "Flare: Achieving masterful and adaptive robot policies with large-scale reinforcement learning fine-tuning." *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025.

Related works

❖ **SPOC: Imitating Shortest Paths in Simulation Enables Effective Navigation and Manipulation in the Real World (CVPR 2024)**

- 시뮬레이터에서 ground-truth information을 기반으로 수행하는 shortest path planner의 trajectory 데이터를 BC로 학습
- 자연어를 instruction으로 받는 transformer 구조의 모델 사용



Strech-RE1

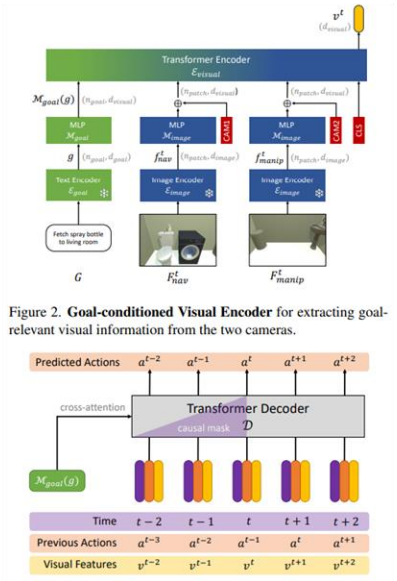
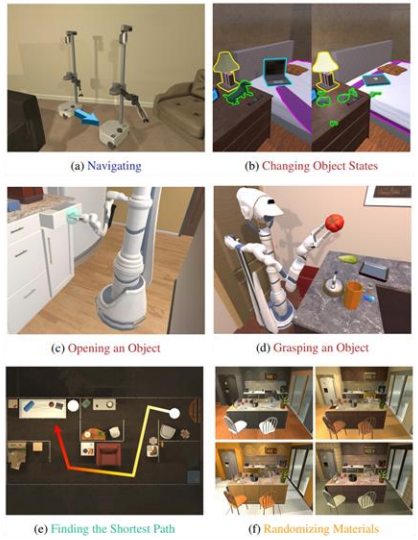


Figure 2. **Goal-conditioned Visual Encoder** for extracting goal-relevant visual information from the two cameras.

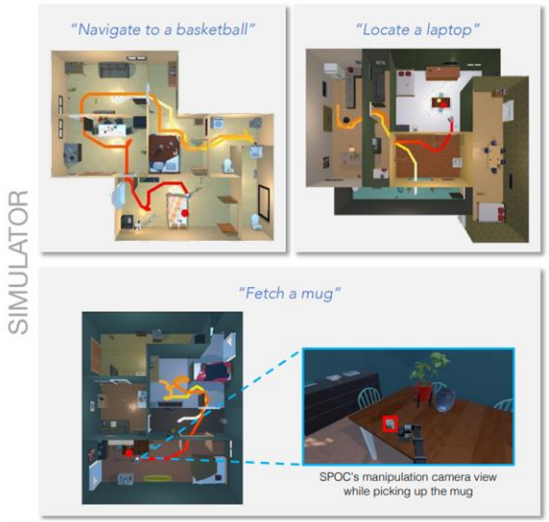
Figure 3. **Action Decoder** for predicting action at the current time step given the goal, current and past observations, and past actions.

Model



Train (Sim. AITHOR)

Test (Real World)



Test (Sim. CHORES)

Environment

https://docs.hello-robot.com/0.2/stretch-hardware-guides/docs/hardware_guide_re1/

Strech-RE1 소개 영상: https://www.youtube.com/watch?v=iEaapHNfEWA

Related works

❖ CHORES: Core HOusehold Robot EvaluationS

- 해당 논문에서 AI2THOR기반의 custom benchmark를 제안함

Task	Description & Example
ObjNav	Locate an object category: "find a mug"
PickUp	Pick up a specified object in agent line of sight: "pick up a mug"
Fetch	Find and pick up an object: "locate a mug and pick up that mug"
RoomVisit	Traverse the house. "Visit every room in this 5-room house. Indicate when you have seen a new room and when you are done."

Table 1. CHORES tasks.

Task	Target Description & Example
ObjNav	Object's category: "vase"
ObjNavAfford	Object's possible uses: "a container that can best be used for holding fresh flowers decoratively"
ObjNavLocalRef	Object's nearby objects: "a vase near a tennis racket and a basketball"
ObjNavRelAttr	Object category comparative attribute: "the smallest vase in the bedroom"
ObjNavRoom	Object's room type: "vase in the living room"
ObjNavDesc	Open vocab instance description: "the brown vase painted orange with a bird on the side"
RoomNav	Type of room: "bedroom"

Table 2. CHORESNAV tasks. The full task specification also includes a navigation verb, such as "Search for a vase".

이따가 FLaRe와 비교할 성능 결과

Benchmark	Model	Training	ObjNav			PickUp			Fetch			RoomVisit			Avg
			Success	SEL	%Rooms	Success	SEL	%Rooms	Success	SEL	%Rooms	Success	SEL	%Rooms	Success
CHORES-S	EmbSigLIP* [38]	Single-task RL	36.5	24.5	42.2	71.9	52.9	30.3	0.0	0.0	50.5	16.5	11.9	44.6	31.2
	SPOC-1-task	Single-task IL	57.0	46.2	51.5	84.2	81.0	30.3	15.1	12.6	48.1	43.7	40.4	81.2	50.0
	SPOC	Multi-task IL	55.0	42.2	56.3	90.1	86.9	30.3	14.0	10.5	49.3	40.5	35.7	81.1	49.9
	SPOC w/ GT Det	Multi-task IL	85.0	61.4	58.7	91.2	87.9	30.3	47.3	35.6	61.6	36.7	33.7	79.3	65.0
CHORES-L	SPOC	Multi-task IL	33.7	25.1	53.7	75.1	69.1	31.5	10.6	8.1	42.9	35.0	33.2	77.8	38.6
	SPOC w/ GT Det	Multi-task IL	83.9	58.0	64.0	78.0	75.7	31.5	48.6	38.3	60.0	42.0	39.1	83.1	63.1

Table 3. **Training on single tasks, IL outperforms RL even with meticulous reward shaping.** EmbSigLIP refers to using the EmbCLIP [38] model with an upgrade to use the SIGLIP backbone since that hugely outperforms the ResNet-50 CLIP backbone (See Tab 5). Further, IL easily extends to multitask training without any performance degradation. Equipping the agent with detection massively boosts the success rate across all tasks except ROOMVISIT which does not require navigating to or manipulating objects.

Benchmark	ObjNav		ObjNavRoom		ObjNavRelAttr		ObjNavAfford	
	Success	%Rooms	Success	%Rooms	Success	%Rooms	Success	%Rooms
CHORESNAV-S	57.5	55.7	50.3	54.6	54.6	62.2	62.4	53.0
CHORESNAV-L	38.7	53.4	54.2	55.7	38.5	56.0	43.5	48.0

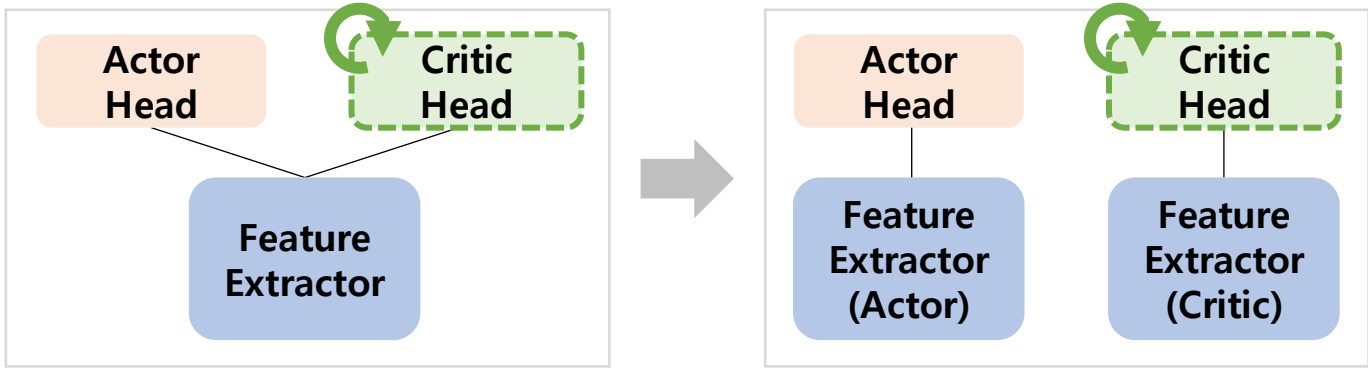
Benchmark	ObjNavLocalRef		ObjNavDesc		RoomNav		Avg
	Success	%Rooms	Success	%Rooms	Success	%Rooms	Success
CHORESNAV-S	45.1	51.5	30.6	49.9	74.5	48.1	53.6
CHORESNAV-L	44.5	58.7	30.5	56.8	67.5	49.9	45.3

Table 8. CHORESNAV results to evaluate SPOC's ability to handle diverse target specifications for navigation.

https://docs.hello-robot.com/0.2/stretch-hardware-guides/docs/hardware_guide_re1/

Strech-RE1 소개 영상: https://www.youtube.com/watch?v=iEaapHNfEWA

FLaRe



❖ C. Stabilize RL Fine-tuning

- **Using On-Policy Algorithms**

  - Off-policy는 sample efficient할 수 있으나…

    1) 하이퍼파라미터에 민감하고 2) 학습이 덜 stabl한 경향이 있음

  - Pretrained BC Policy을 초기값으로 사용하기 때문에 sample efficiency는 어느 정도 해소함 → 따라서 PPO 알고리즘으로 RL 튜닝 수행

- **Taking Small Update Steps**

  - 사전 학습에 사용한 learning rate 보다 더 낮은 값을 사용해야함

- **Disabling Entropy Bonus**

  - PPO 알고리즘에서 에이전트의 탐험을 유도하는 entropy bonus를 끄고 튜닝 진행
  - 해당 값이 들어오면 오히려 학습을 왜곡하는 현상이 발생함
  - [개인 해석] 탐험은 초기 상태에서 빠르게 벗어나는게 목표인데 사전학습된 정책은 초반부터 좋은 정책을 수행하기 때문에 학습이 왜곡

- **Disabling Feature Sharing**

  - 일반적으로 actor-critic 모델에서 feature extractor는 공유하도록 구성
  - 주어진 상황은 BC 방식으로 사전학습된 모델 → Critic head를 새로 추가함 (random init.) → 따라서 함께 학습 시 재앙적 망각 발생 가능
  - 따라서 아예 둘을 분리하고 학습을 진행하는 것이 효과적

FLaRe

❖ Experiment

- Q1. IL+RL, IL only, RL only의 SoTA 모델과의 성능비교
- Q2. FLaRe로 학습했을 때 unseen task에 대해서도 잘 작동하는가?
- Q3. FLaRe로 학습한 정책이 real-world에서도 잘 작동하는가?
- Q4. FLaRe 방법론이 new robot embodiments/behavior에 효율적으로 적용될 수 있나?
- Q5. FLaRe에 사용된 stabilization 테크닉이 실제로 성능에 영향을 미쳤나?

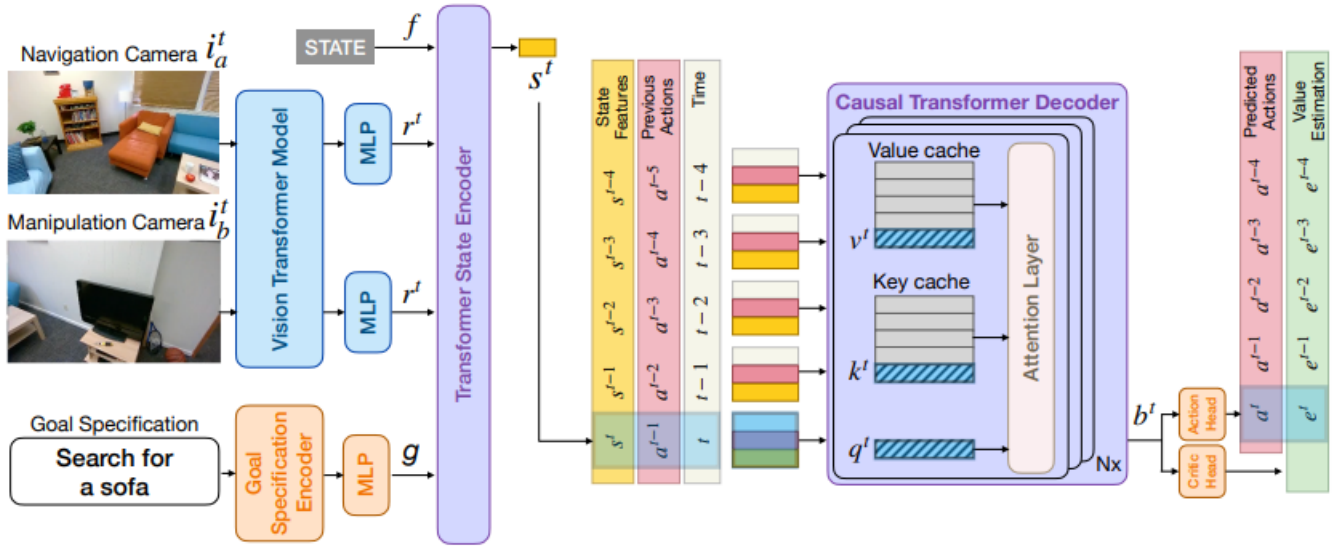


Fig. 7: A visualization of the network architecture of the transformer-based SPOC model that FLaRe fine-tunes upon.

FLaRe

Task	Description & Example	Max Steps
ObjectNav	Locate an object category: “find a mug”	600
PickUp	Pick up a specified object in agent line of sight: “pick up a mug”	600
Fetch	Find and pick up an object: “locate a mug and pick up that mug”	600
RoomVisit	Traverse the house. “Visit every room in this 5-room house. Indicate when you have seen a new room and when you are done.”	1000

❖ Q1. IL+RL, IL only, RL only의 SoTA 모델과의 성능비교

- **CHORES-small** 벤치마크로 성능 비교를 수행
- Fair/Unfair comparison으로 구분 → Sparse reward를 쓰는 경우/ ground-truth 정보로 알 수 있는 dense reward를 쓰는 경우
- Baseline들은 FLaRe보다 더 많은 학습(step)을 수행함 → 그럼에도 FLaRe의 성능이 더 압도적으로 높음
  - Fair comparison (IL+RL, RL Only [1])에서 ObjectNav와 RoomVisit은 3배 / Fetch와 Pickup은 2배
  - Unfair comparison (RL only [2, 3])은 정책 성능이 수렴할 때까지 계속 학습을 수행하고 최고 점수로 비교
  - Unfair comparison (RL only [2, 3])에서 Poliformer(Dense)는 ObjectNav를 300M Step 학습 (FLaRe의 15배)
  - SPOC (IL Only)은 학습하는 데이터셋의 400 epoch까지 학습하는 것으로 확인 (https://github.com/JiahengHu/FLaRe)

TABLE I: Success and Episode-length weighted Success (SEL) against baseline methods on the CHORES [7] benchmark. Baselines with privileged information are *marked in blue*. FLaRe significantly outperforms the previous SoTA methods.

Success (SEL)	IL+RL: Sparse Reward			IL Only	RL Only		
	FLaRe (Ours)	PIRLNav	JSRL	SPOC	Poliformer - Sparse	Poliformer - Dense	EmbSigLIP - Dense
ObjectNav	85.0 **(67.6)**	20.0 (7.0)	21.0 (15.6)	55.0 (42.2)	14.5 (10.4)	**85.5** (61.2)	36.5 (24.5)
Fetch	**66.9 (54.7)**	0.0 (0.0)	2.9 (2.8)	14.0 (10.5)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
PickUp	**91.8 (90.4)**	0.0 (0.0)	50.9 (47.7)	90.1 (86.9)	0.0 (0.0)	90.1 (88.7)	71.9 (52.9)
RoomVisit	**70.4 (67.1)**	12.5 (11.0)	19.0 (18.6)	40.5 (35.7)	12.5 (12.5)	12.5 (10.9)	16.5 (11.9)

2025/04/03 김재훈 Jump-Start Reinforcement Learning 1

FLaRe

❖ Q2. FLaRe로 학습했을 때 unseen task에 대해서도 잘 작동하는가?

- **CHORESNav-small** 벤치마크로 성능 비교를 수행
- 학습에서 본 적이 없는 task를 수행하는 unseen 상황을 구성
- 다만, BC Policy의 경우 unseen task에 대한 expert demo.를 1M Frame 학습
- FLaRe로 학습하는 경우에는 unseen task를 사전학습 때부터 전혀 학습하지 않음

Task	Target Description & Example
ObjNav	Object's category: "vase"
ObjNavAfford	Object's possible uses: "a container that can best be used for holding fresh flowers decoratively"
ObjNavLocalRef	Object's nearby objects: "a vase near a tennis racket and a basketball"
ObjNavRelAttr	Object category comparative attribute: "the smallest vase in the bedroom"
ObjNavRoom	Object's room type: "vase in the living room"
ObjNavDesc	Open vocab instance description: "the brown vase painted orange with a bird on the side"
RoomNav	Type of room: "bedroom"

ObjNavAffor:	"Find something I can sit on"
ObjNavRelAttr:	"Find the largest apple"
RoomNav:	"Go to the kitchen"

Unseen Tasks

TABLE II: FLaRe can fine-tune for tasks that are never seen by the base model, and achieve state-of-the-art performance. Baselines with privileged information are *marked in blue*.

Success (SEL)	FLaRe (ours)	Poliformer (Sp)	SPOC++	Poliformer (De)
ObjNavRelAttr	**71.0 (63.6)**	6.7 (6.7)	54.5 (44.6)	36.1 (32.4)
RoomNav	**91.6 (85.6)**	57.0 (51.8)	74.5 (59.9)	75.0 (62.4)
ObjNavAfford	**79.7 (70.6)**	35.5 (29.4)	62.4 (50.6)	53.8 (43.1)

Ehsani, Kiana, et al. "Spoc: Imitating shortest paths in simulation enables effective navigation and manipulation in the real world." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024.

FLaRe

❖ Q3. FLaRe로 학습한 정책이 real-world에서도 잘 작동하는가?

- Real-world 적용을 위한 별도의 fine-tuning을 수행하지 않고도 잘 작동하는 것을 확인

TABLE III: Real-world results (total of 46 tasks). For manipulation tasks, we report both full success (policy and heuristic grasping) and policy success (proximity) following [7].

Success Rate	FLaRe (ours)	SPOC	Poliformer (Dense)
ObjectNav	**94.4**	50.0	83.3
Fetch	**66.7** (55.6)	33.3 (11.1)	X
PickUp	**86.7** (66.7)	66.7 (46.7)	X
RoomVisit	**75.0**	50.0	X

공개된 성능 없음

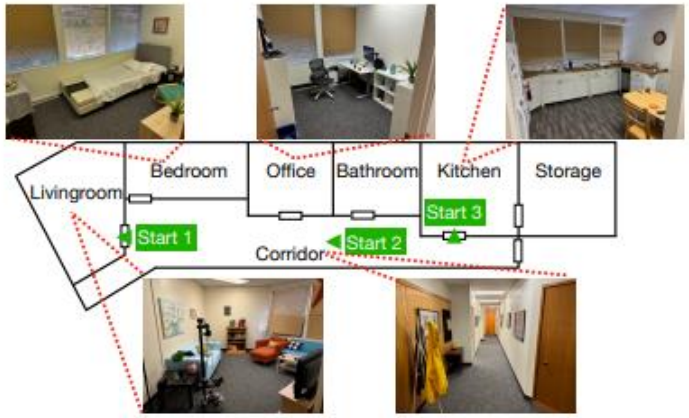
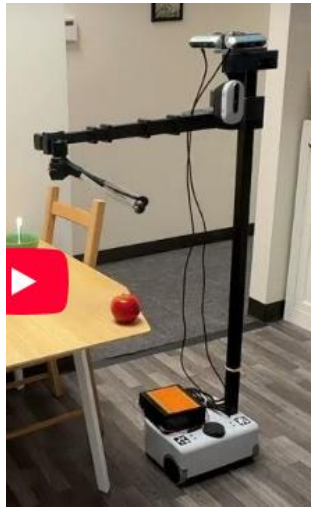


Fig. 5: The real-world layout that we tested upon

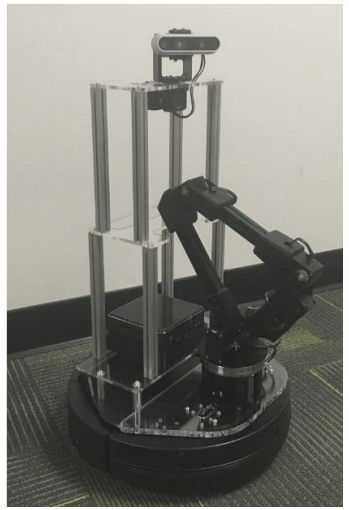
❖ Q4. FLaRe 방법론이 new robot embodiments/behavior에 효율적으로 적용될 수 있나?

- Stretch-RE1에서 사전학습한 뒤 LoCoBot에서 FLaRe로 RL fine-tuning을 수행
  - ObjectNav Task를 수행
  - Invalid action은 마스킹 & 그 중 두 action 차원은 카메라 컨트롤로 재정의

New Embodiment	Success Rate ↑	SEL ↑
FLaRe	**72.0**	**47.2**
Poliformer zero-shot[2]	57.5	30.1
Poliformer (Sparse Reward)	44.0	29.7



Stretch-RE1

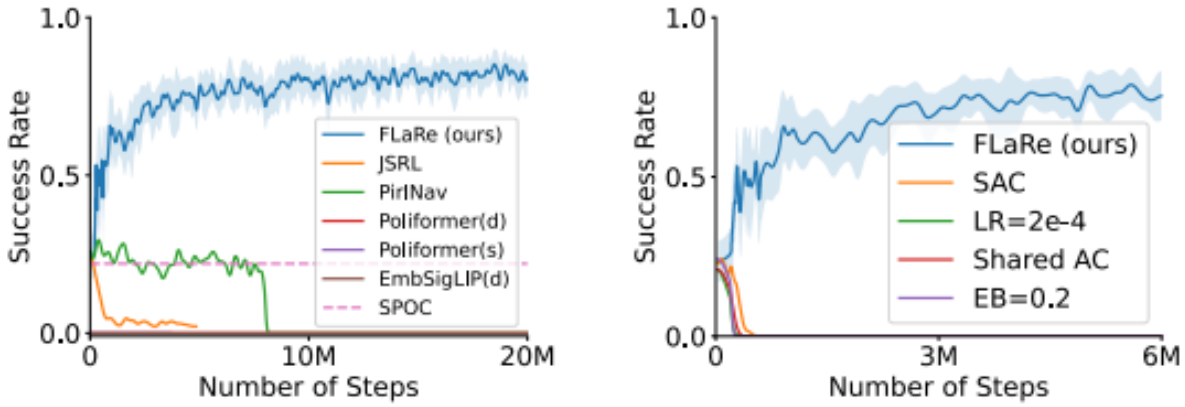


LoCoBot

FLaRe

❖ **Q5. FLaRe에 사용된 stabilization 테크닉이 실제로 성능에 영향을 미쳤나?**

- 1) PPO → SAC 방식으로 바꾼 경우, 2) LR를 10배로 키워서 학습한 경우, 3) Shared feature extractor를 쓴 경우, 4) Entropy Bonus를 쓴 경우
- 네 가지 중 한 가지만 바꾸더라도 학습이 완전 망가져버리는 결과를 보여줌



(a) Baselines (b) Ablations

Fig. 6: Baseline performances and ablation studies on the Fetch task. FLaRe is the only method that can achieve good performance on this challenging task.

FLaRe

❖ Limitation

- FLaRe의 주요 제약은 fine-tunin이 시뮬레이션 환경에 의존한다는 점

  - 시뮬레이션이 제대로 지원하지 않는 영역들(ex. 액체나 부드러운 물체 등)을 다루는 과제들은 여전히 적용하기 어려움