Recent Trends in VLA Models for Robotics

RILS LAB

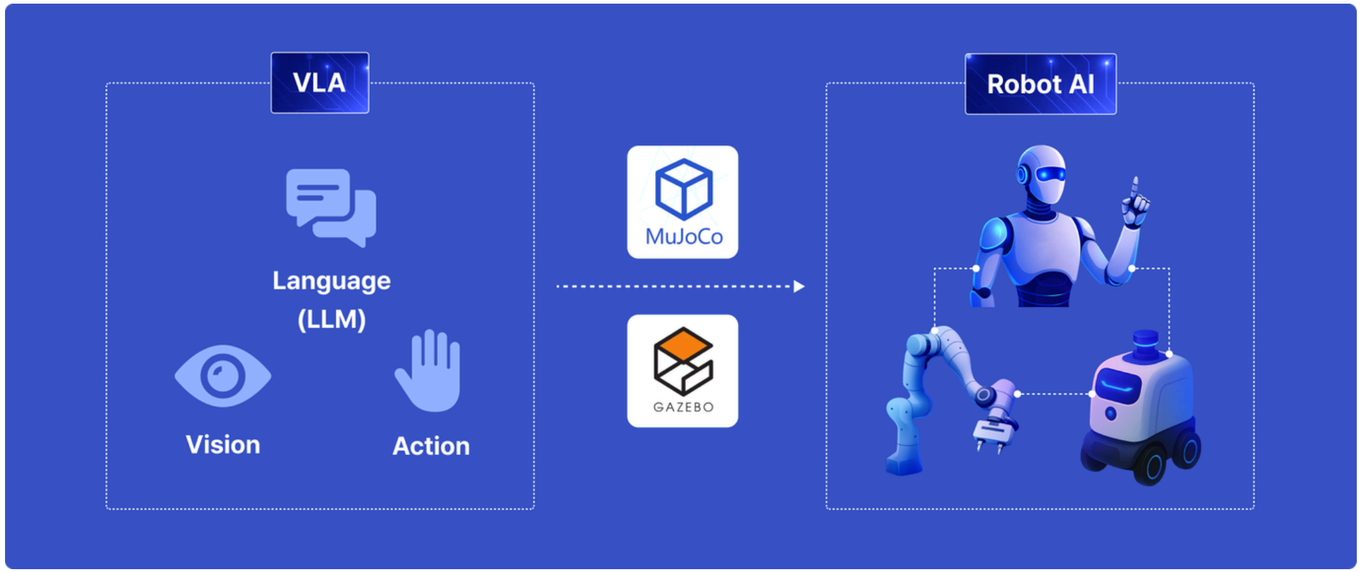
Robot Intelligence Learning & System

신승철

2026. 07. 23

Physical AI란?

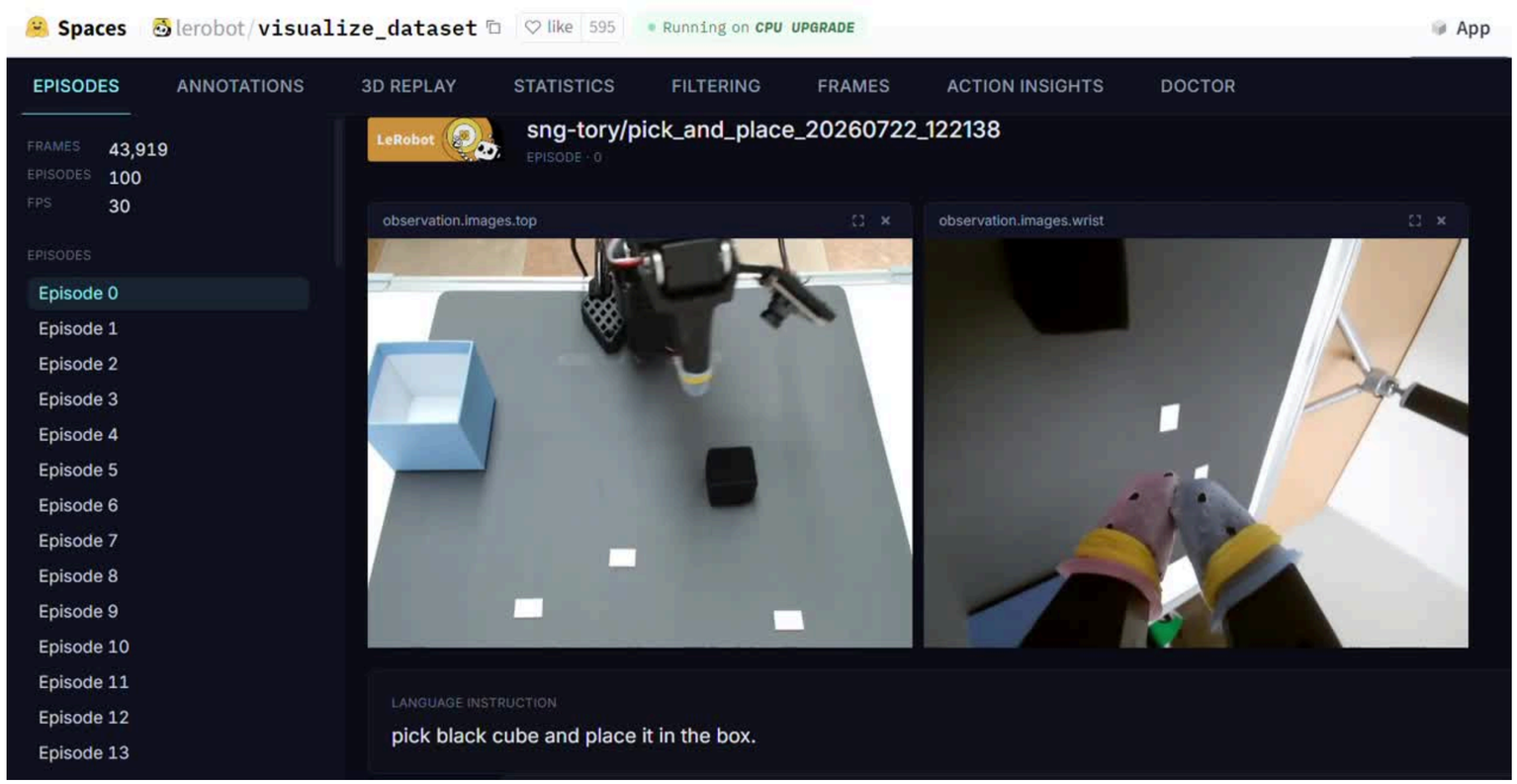
AI가 물체나 상황을 인지/판단 해주는 것을 넘어 인간을 대신해서 수행까지 하는 통합 기술



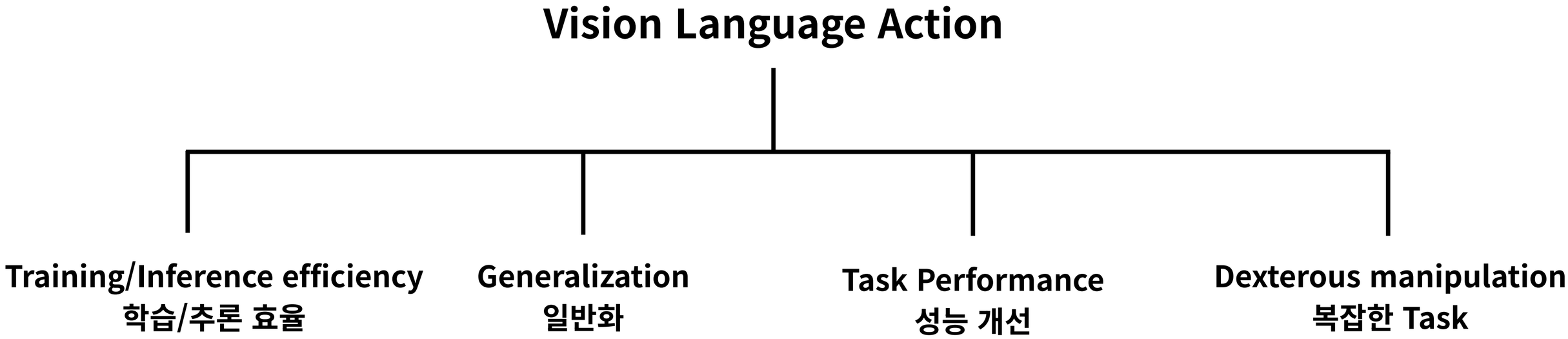
그 중에서도 상황이나 명령을 인지해서 로봇의 Action을 만들어 주는
Vision-Language-Action(VLA)는 Physical AI의 핵심 기술

저의 연구분야

VLA를 활용하여 인간의 팔을 대신해서 사용할 수 있는 로봇 arm 쪽을 연구중
그중에서도 인간의 action → robot의 action을 변환하는 motion retargeting 쪽을 연구



VLA 분야의 현재 연구 방향성



Training/Inference efficiency

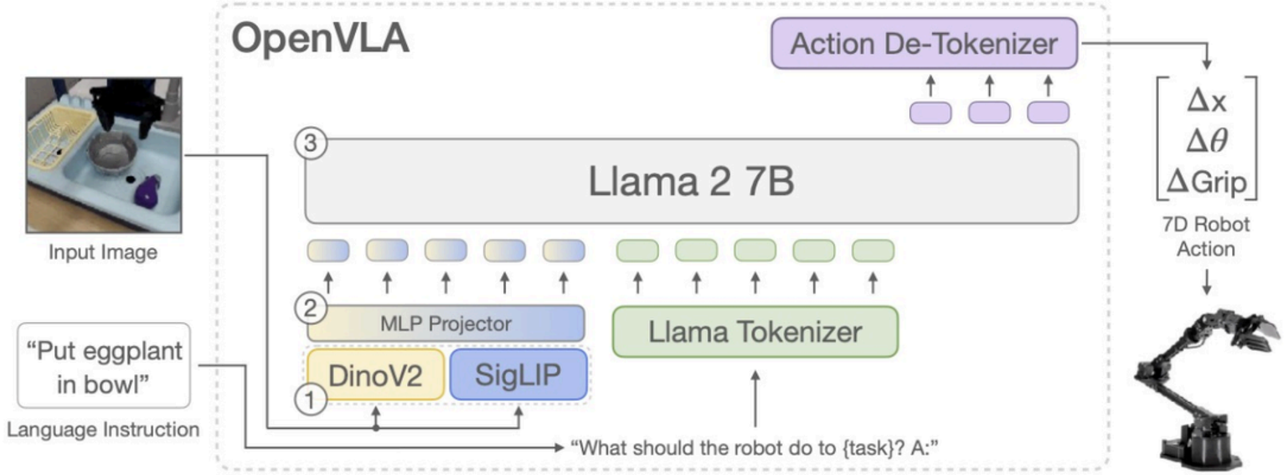
01

FAST: Efficient Action Tokenization for Vision-Language-Action Models

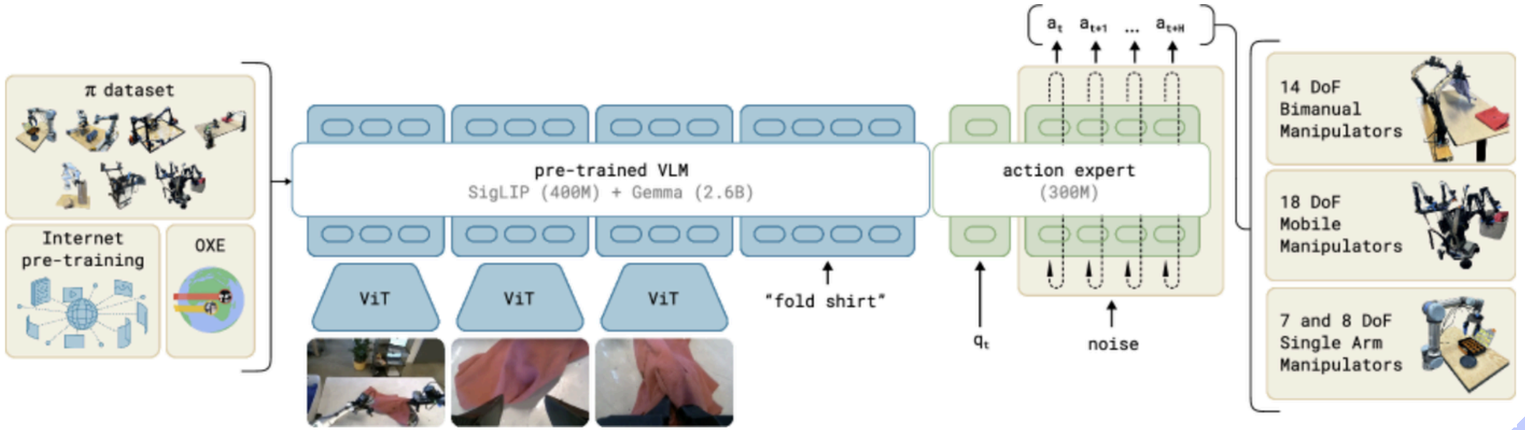
RSS / 2025

VLA action 생성 대표적인 2가지 방식

방식 1
Autoregressive VLA
Action을 토큰화하는 방식으로 학습하고 생성한다.



방식 2
Diffusion/Flow matching VLA
VLM의 지식을 넘겨받아 노이즈에서 부터
Action을 생성한다.



Autoregressive VLA limitation

각 timestep, 각 dimension마다 모두 독립적으로 binning
각 timestep × 각 action dimension = 하나의 token

고주파 task에서 적용한다면?
예시) 50HZ, 14 dimension task

고주파 task에서는 새로운 정보가 거의 없어서
이전 토큰 복사 + 로봇 멈춤 현상 발생

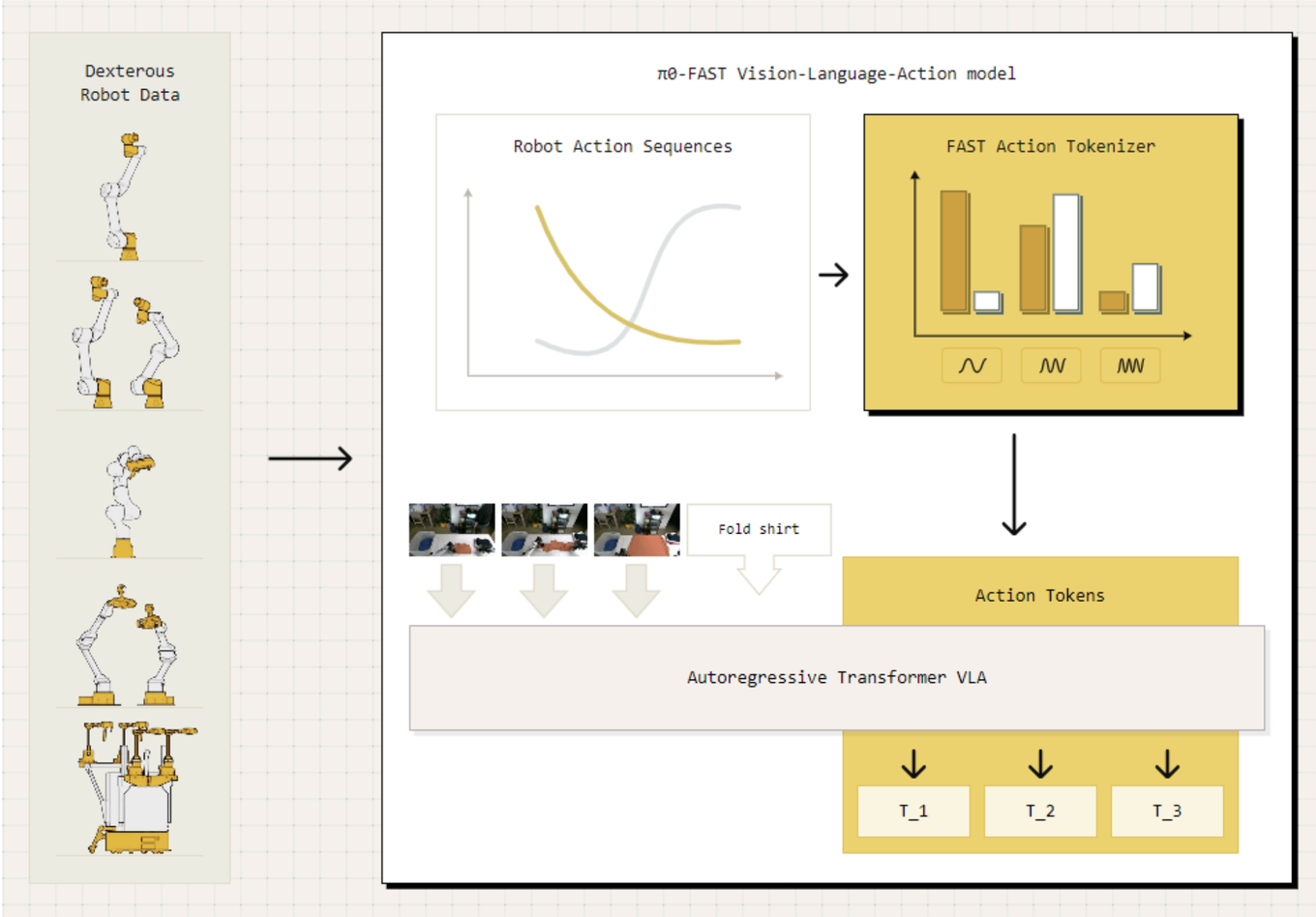
Main Idea

Autoregressive VLA가 action token화 하는 방식을 잘해주면 고주파, 섬세한 task 에서도 잘하지 않을까??

그렇게 된다면 학습이 오래걸리는 생성방식을 대체할 수 있지 않을까?

핵심 contribution

- **DCT/Quantization을 이용한 action 토큰화 방식**
- **BPE를 활용하여 적은 토큰으로 압축**



Main Idea

DCT/Quantization

시간 축 trajectory → frequency domain으로 변경 후 정수화 과정 진행

저주파 : 전체적인 움직임 형태, 고주파 : 빠른 진동이나 미세한 변화

예시)

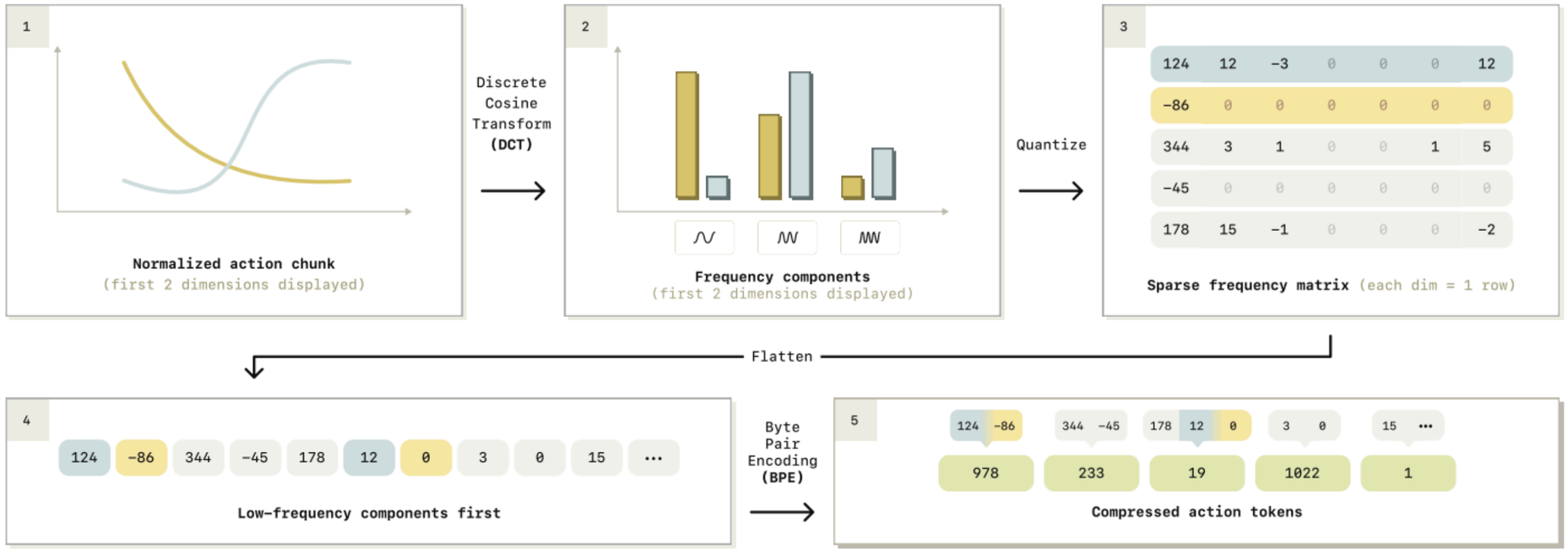
a1,a2,a3 → c0,c1,c2 (이때 c1은 저주파, c2는 고주파)

BPE

DCT/정수화 진행 후 결과

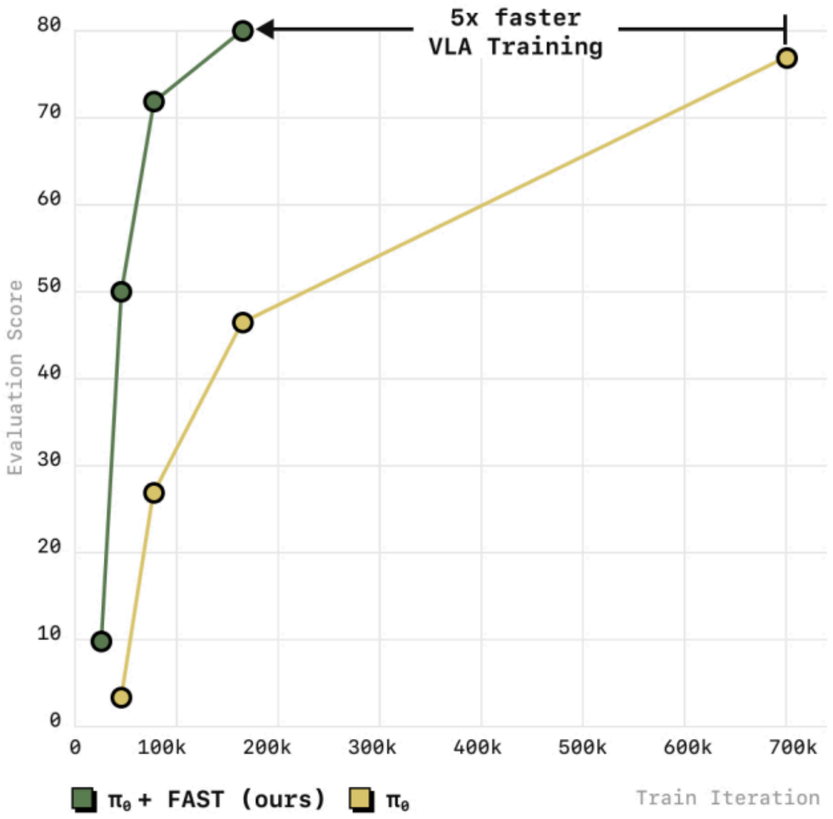
[6, -3, 1, 0, 0, 0, 0, 0, 2, -1, 0, 0, 0, ...]

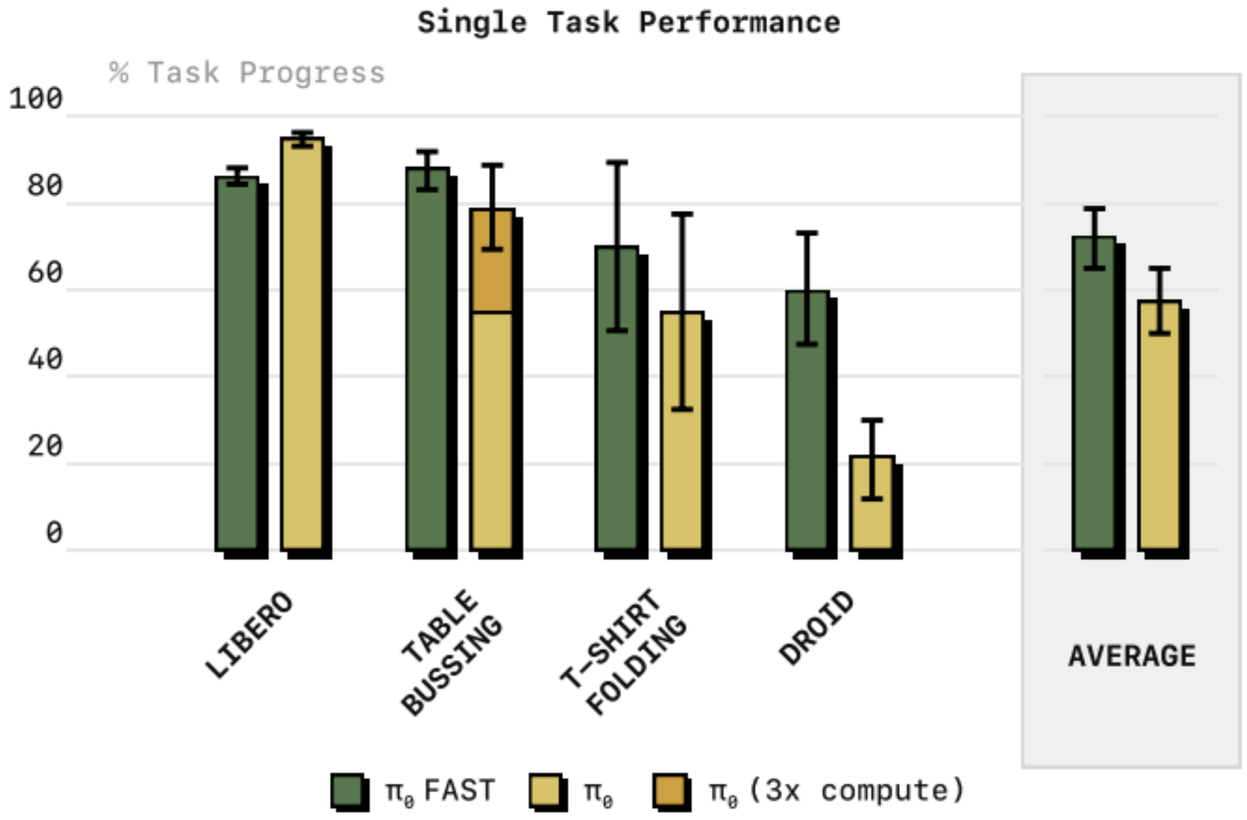
반복 패턴을 합쳐서 토큰화를 진행한다.



Result

속도 + 성능 측면 모두를 잡을 수 있게 됨





Generalization

02

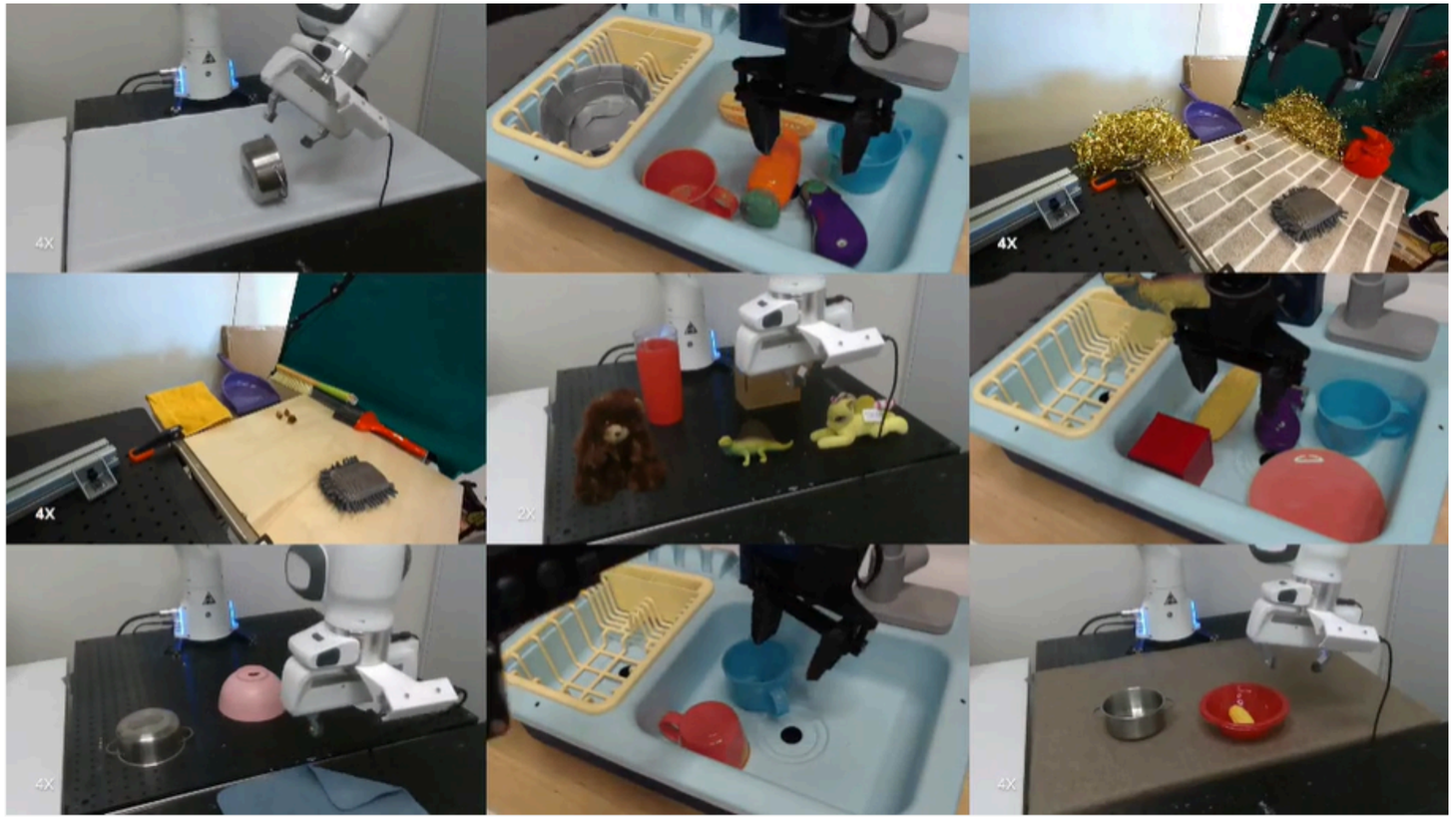
π0.5：a Vision-Language-Action Model with Open-World Generalization

CoRL / 2025

VLA limitation

기존 VLA 모델 한계
로봇 제어에서 좋은 성능을 보이지만,
대부분 학습 환경과 비슷한 조건에서만 평가

즉, 기존 모델은 학습에서 한 번도 보지 못한 환경에서도
복잡한 작업을 하는데 어려움을 느낌



Main Idea

실험 환경에서 벗어나서 새로운 환경에서도 복잡한 TASK를 할 때 일반화 성능을 보이는 모델을 만들어보자

핵심 contirbution

- 로봇 행동 + 이외 다양한 데이터 활용
- 계층적 VLA
- 이산 행동 토큰 + Flow matching 같이 활용

다양한 데이터 활용

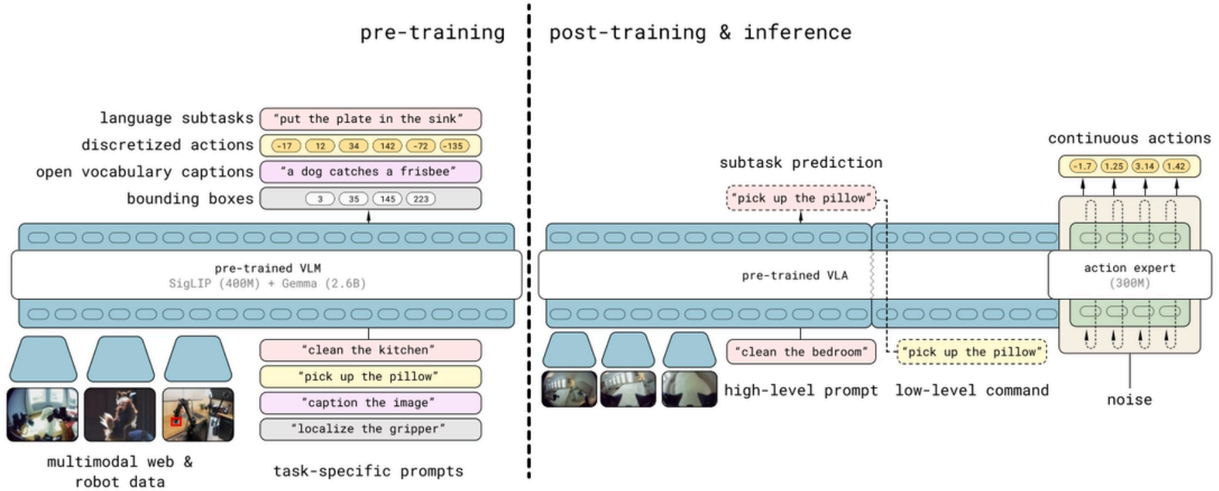
로봇 데이터 뿐만 아니라 Detection, Captioning, VQA 등의
다양한 데이터들도 함께 넣어서 훈련

저자의 생각
새로운 환경에 일반화하려면 목표 로봇의 직접 경험만으로는 부족하고,
다른 로봇·다른 작업·웹 지식·언어 지시에서 지식을 전이해야 한다. (co-training)

계층적 VLA

LLM에서 Reasoning을 하는 것처럼 한번의 추론을 거치고
액션을 생성해내는 구조로 복잡한 task를 해결하려는 시도

이미지 + 로봇 상태 + 전체 명령 → 고수준 하위 작업 → 저수준 Action Chunk



이산 행동 토큰 + Flow matching

FAST 방식의 장단점
학습할때는 속도가 빠르나
추론 시에 AR 방식으로 느림

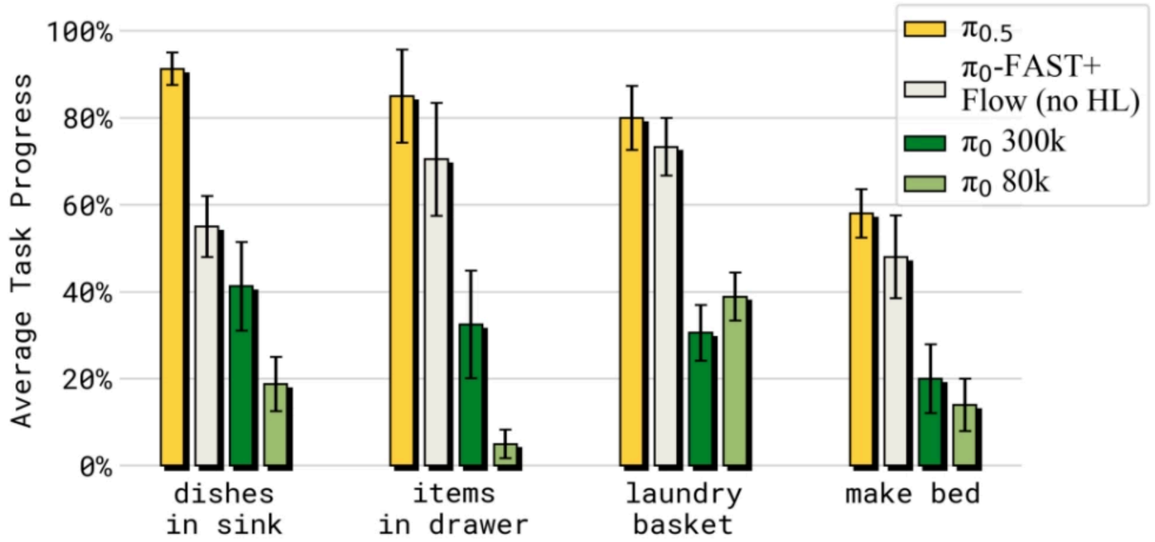
pi0 방식의 장단점
학습시에 속도는 느리나
학습시에 적은 step으로 denoising하여 속도가 빠름

pre-training 시에 FAST 방식 적용
post-training/Inference 시에 pi0의 Flow matching 방식 적용
이를 통해 학습 효율은 FAST에서 얻고, 추론 속도와 정밀성은 Flow Matching에서 얻는다.

Result

학습시에는 보지 못하는 환경에서 test 해본 결과

오래학습한 pi0 보다도 더 안정적인 일반화 성능을 보여준다.



03

Do You Know Where Your Camera Is? View-Invariant Policy Learning with Camera Conditioning

ICRA / 2026

VLA limitation

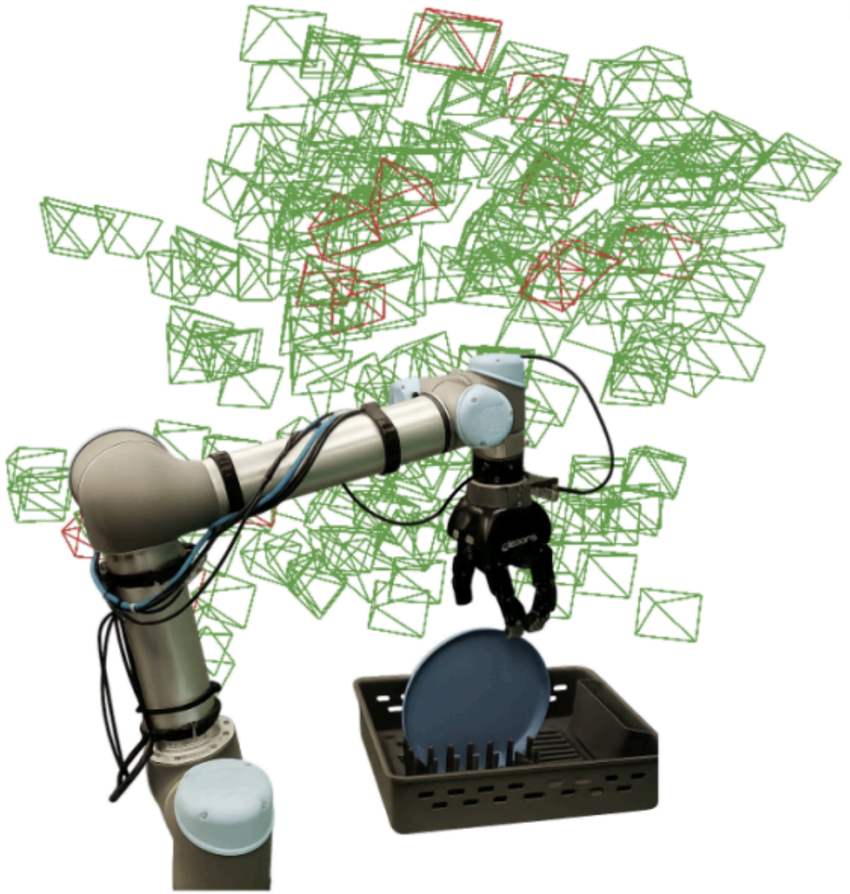
기존 VLA 방식
RGB Image → Vision Encoder → Policy → Action

문제
RGB 영상 하나만 보고

- 물체 위치도 이해해야 하고
- 로봇 위치도 이해해야 하고
- 카메라가 어디에 있는지도 추론

즉 카메라 포즈 추정 + Manipulation 두 가지를 동시에 학습

그렇기 때문에 고정된 환경에서 조금만 변하면 성능이 급격히 떨어짐



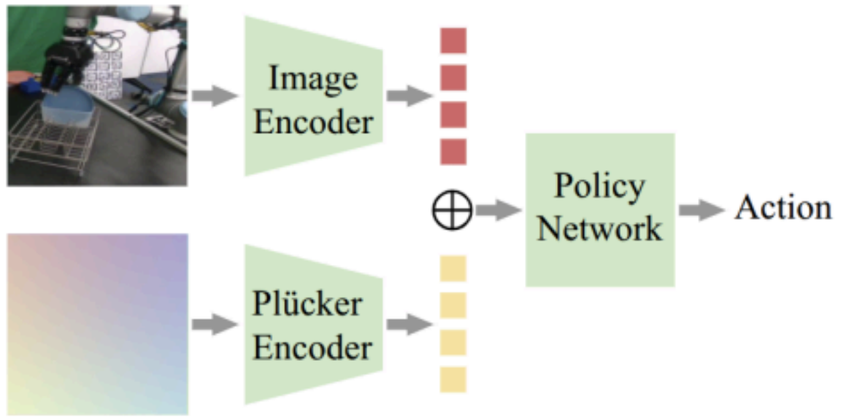
Main Idea

카메라 위치 정보를 미리 알려준다면
위치에 따른 성능이 robust해지지 않을까?

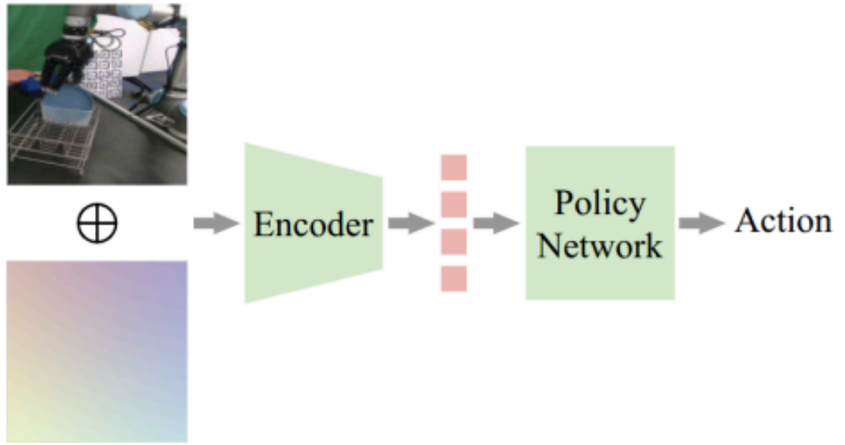
- **Plücker Ray-map 제안**

단순히 카메라 위치 정보를 제공하는 것이 아닌
각 픽셀은 공간의 어디 방향을 보는지를 계산하여
640×480×6 Ray map 생성

- **새로운 Viewpoint Generalization Benchmark 구축**



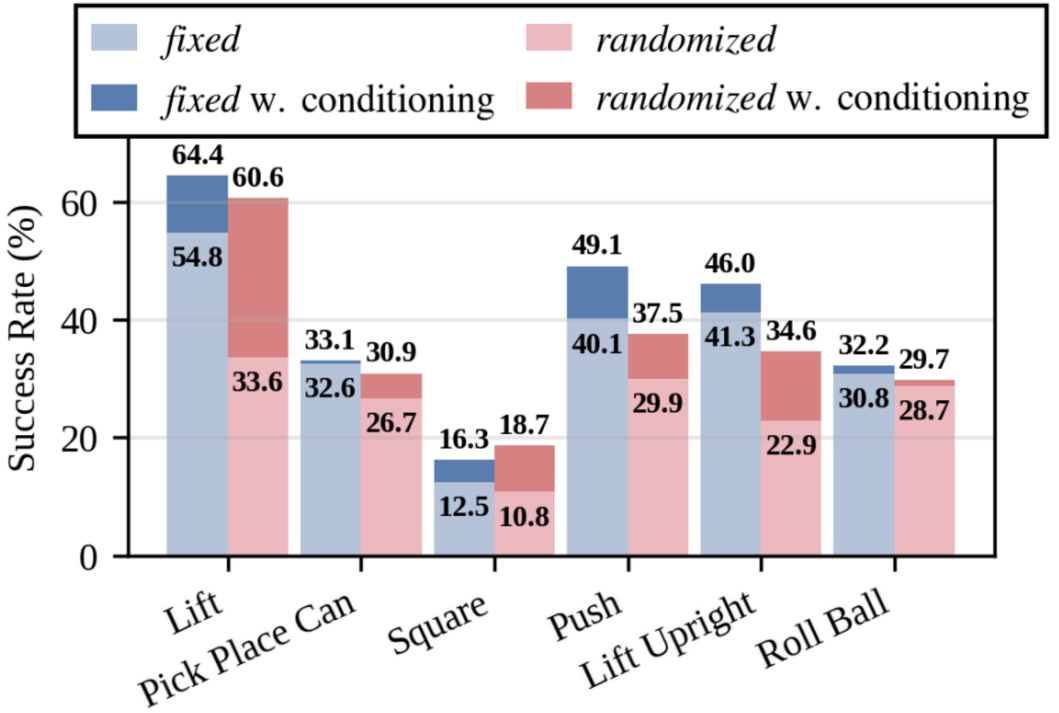
(a) Policy with a pretrained encoder



(b) Policy without a pretrained encoder

Result

ACT, Diffusion Policy, SmolVLA를 포함한
다양한 정책에 별도의 구조 변경 없이 적용 가능하며
시점 변화에 대한 일반화 성능을 일관되게 향상



Method	Lift	Pick Place Can	Assembly Square	Push	Lift Upright	Roll Ball
ACT	33.6	26.7	10.8	29.9	22.9	28.7
ACT w. conditioning	60.6 (+27.0)	30.9 (+4.2)	18.7 (+7.9)	37.5 (+7.6)	34.6 (+11.7)	29.7 (+1.0)
DP	29.1	23.1	2.0	20.0	9.5	19.9
DP w. conditioning	51.1 (+22.0)	39.3 (+16.2)	2.4 (+0.4)	30.3 (+10.3)	20.7 (+11.2)	23.7 (+3.8)
SmolVLA	19.6	56.0	22.0	39.0	23.6	27.6
SmolVLA w. Conditioning	54.4 (+34.8)	70.0 (+14.0)	26.4 (+4.4)	43.8 (+4.8)	33.4 (+9.8)	30.4 (+2.8)

Task Performance

04

π ∗ 0.6 : a VLA That Learns From Experience

RSS / 2026

VLA limitation

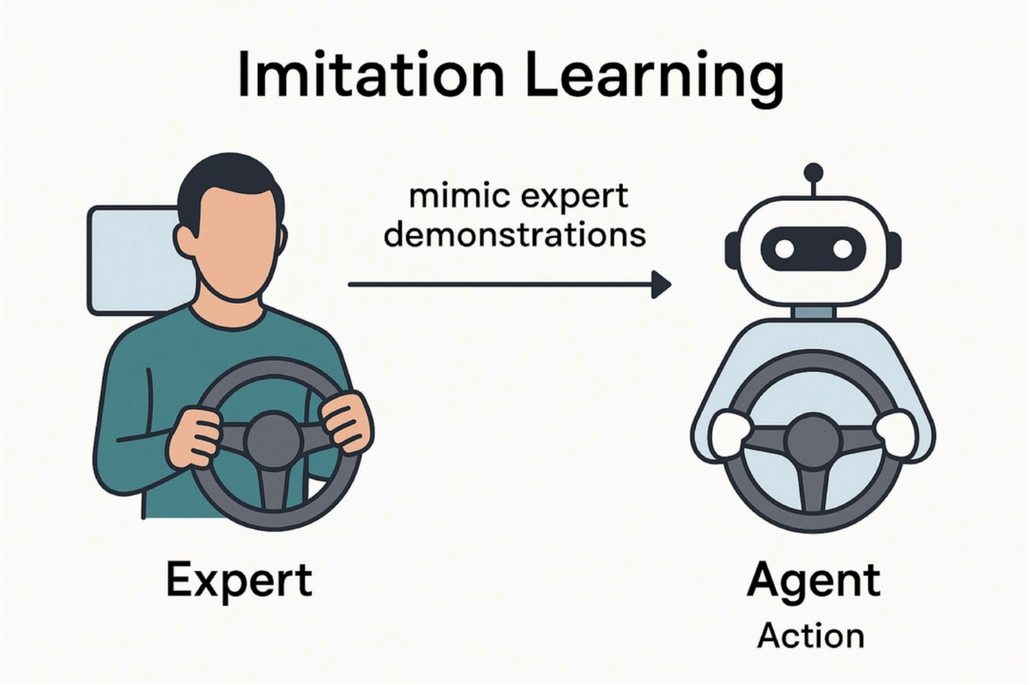
기존 VLA(π_0, $\pi_{0.5}$, $\pi_{0.6}$)는 모두 기본적으로 Imitation Learning 기반

Human Demonstration → Supervised Learning → Robot Policy
사람이 보여준 행동은 잘 따라한다.

문제점

- 실제 배치 적용중 발생하는 새로운 실패를 배우지 못함
- Demonstration보다 더 빠르고 효율적인 행동을 배우기 어려움
- 실패 데이터는 대부분 버려짐

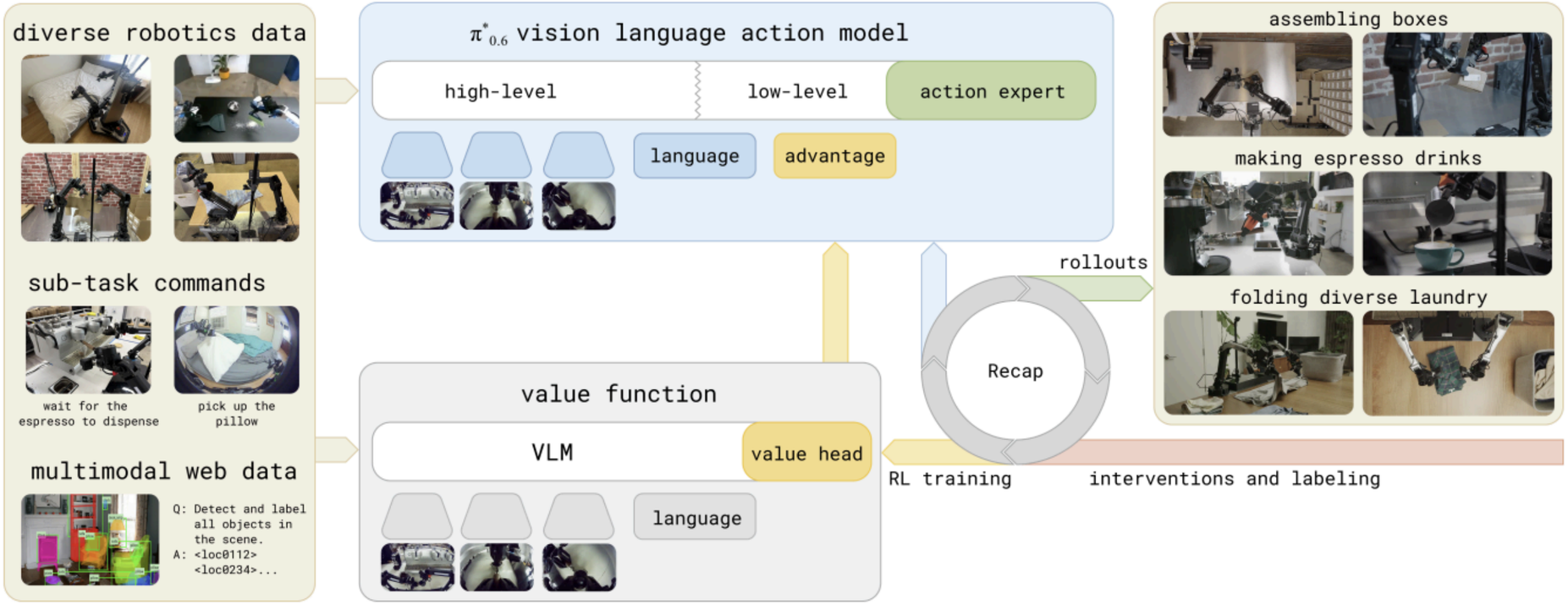
실제 경험으로부터 성장하지 못하는 것이 가장 큰 문제



Main Idea

실제 로봇 경험까지 모두 학습에 이용된다면 계속 실패/성공 수행 과정속에서도 성능을 올릴 수 있지 않을까?

핵심 Contribution
RECAP이라는 RL Framework를 제안



Process

실제 로봇 실행 및 데이터 수집 : 성공 혹은 실패 영상 모두 수집

로봇 수행중에 사람이 조정해서 성공으로 변환하는 데이터도 수집, 또한 성공인지 실패인지 사람이 라벨링

↓

Value Function을 학습 및 advantage 계산

각 장면이 얼마나 좋은 상태인지 판단 후 advantage 계산, 느리게 성공한 것고 advantage가 낮게 측정

↓

Advantage를 positive/negative로 변환 후 입력 conditioning으로 넣어줌

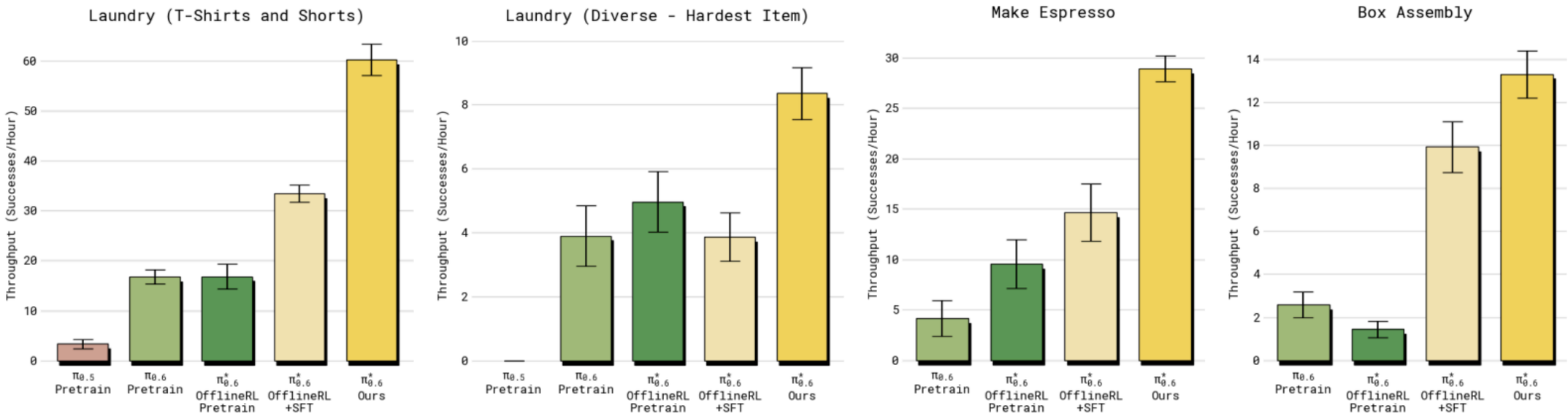
모든 행동을 유지한 채 조건을 붙임

↓

학습 후 실제 실행 시에는 입력에 positivie만을 넣어서 추론

Result

시간 대비 성공률이 가장 높은 것을 확인할 수 있음.



Thank you

Question & Disscusion

RILS LAB