RL Research Group 20251127

SimpleVLA-RL: Scaling VLA Training via Reinforcement Learning

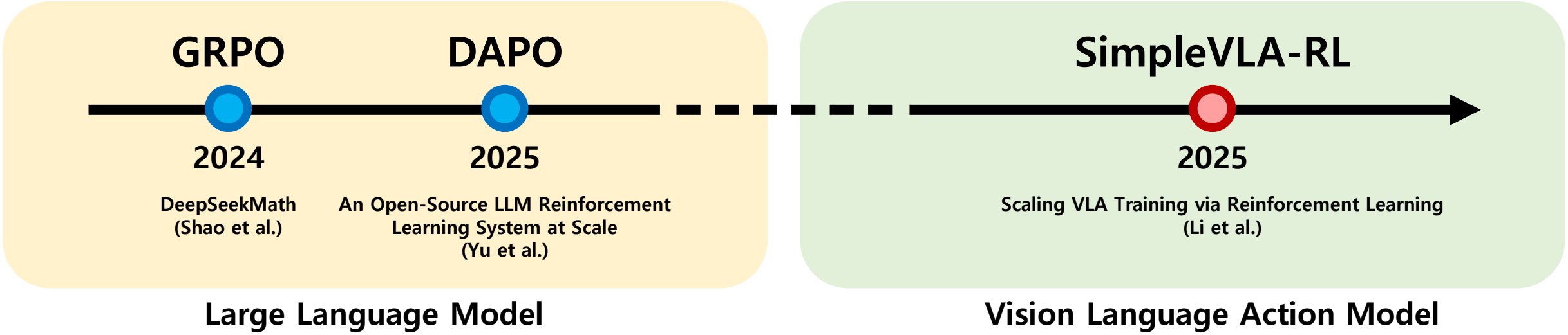
(ICLR 2026)

김재훈

Introduction

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

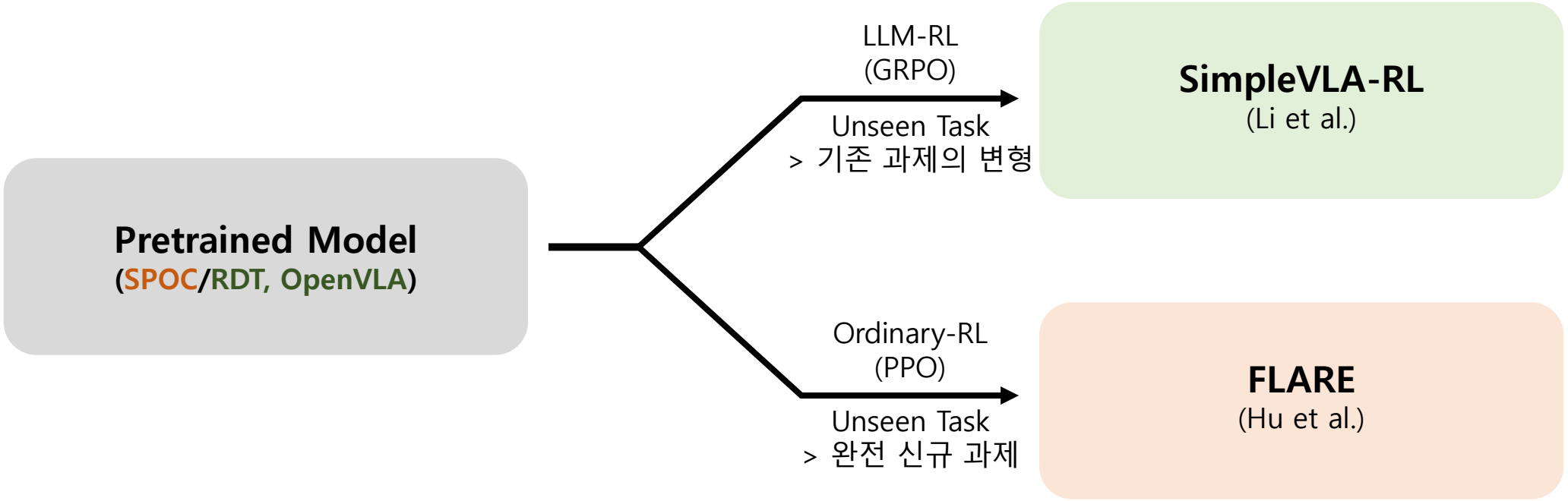
- 대규모 데이터로 사전 학습된 모델에 특정 과제(task)에 대한 고품질 조작 데이터를 지도학습(Supervised Fint-Tuning; SFT)
- 하지만 해당 방식은 처음보는 과제(unseen task)에 대한 일반화 성능(generalization performance)이 떨어짐
- 따라서 추가로 강화학습을 수행하여 일반화 성능을 높이는 방법론을 제안함



Introduction

❖ 지난 번에 공유한 연구(FLARE)와의 공통점/차이점

- [공통점] 지도학습(imitation learning, SFT)의 낮은 일반화 성능을 강화학습으로 해결해보자
- [SimpleVLA-RL] **LLM 특화된 강화학습** 알고리즘 사용(GRPO) / SFT로 학습 → **강화학습으로 추가 학습** / **탐험 장려** / Unseen **변형 과제**
- [FLARE] **일반적인 강화학습** 알고리즘 사용(PPO) / **오롯이 강화학습**으로 학습 / **탐험 방지** / Unseen **신규 과제**

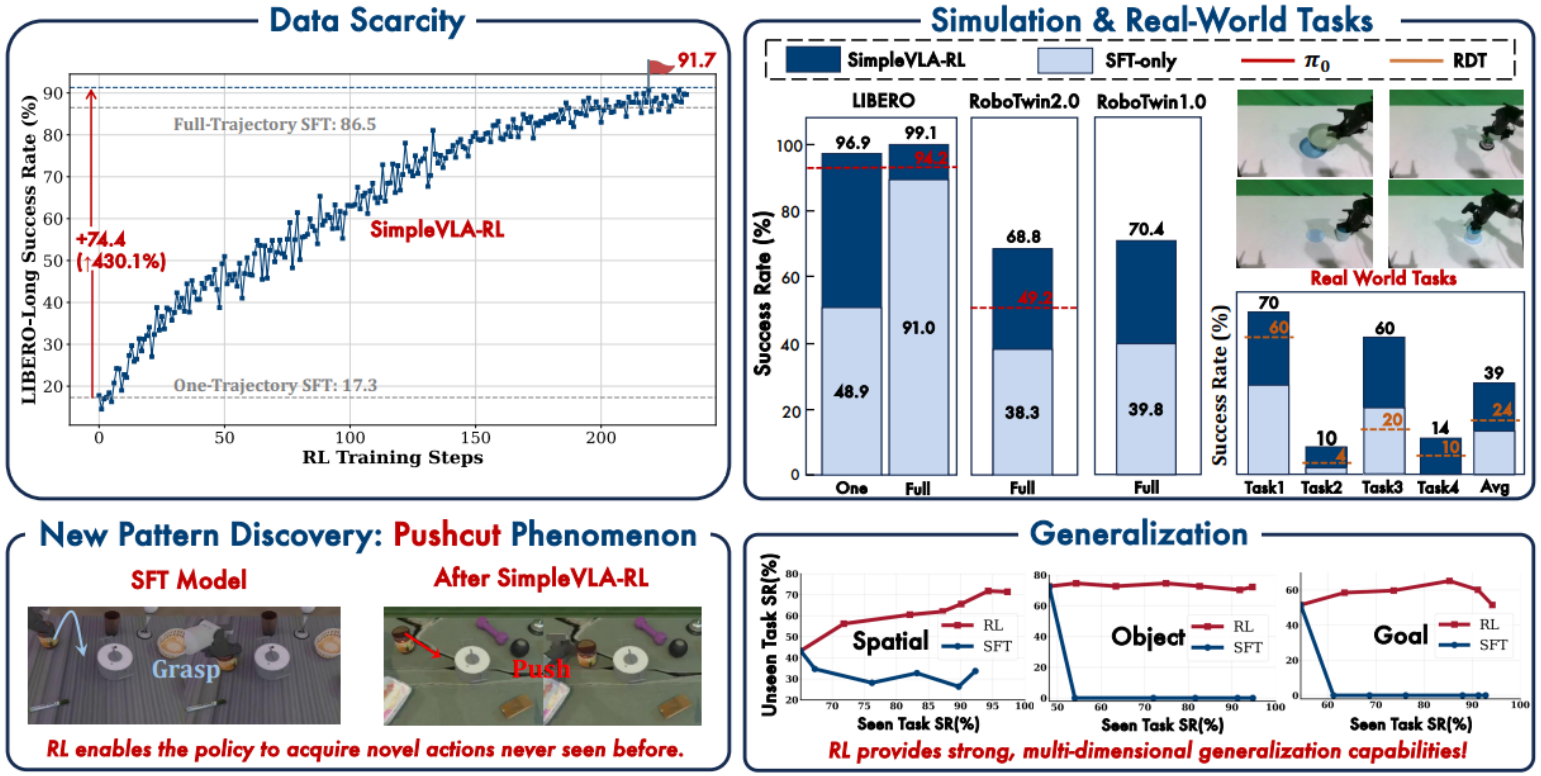


Hu, Jiaheng, et al. "Flare: Achieving masterful and adaptive robot policies with large-scale reinforcement learning fine-tuning." *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025.

Introduction

❖ SimpleVLA-RL의 기여점 요약

- SoTA 모델(RDT, OpenVLA-OFT)에 대하여 full-sample SFT 대비 10~15% 성능 향상 (1-sample SFT의 경우 최대 약 74%까지 향상)
- RL 기반 튜닝을 수행하여 일반화 성능 향상 (Unseen task: Spatial, Object, Goal)



Example)

Spatial: pick up the black bowl **between the plate and ramekin**

Object: pick up the **cream cheese** and place it in the basket

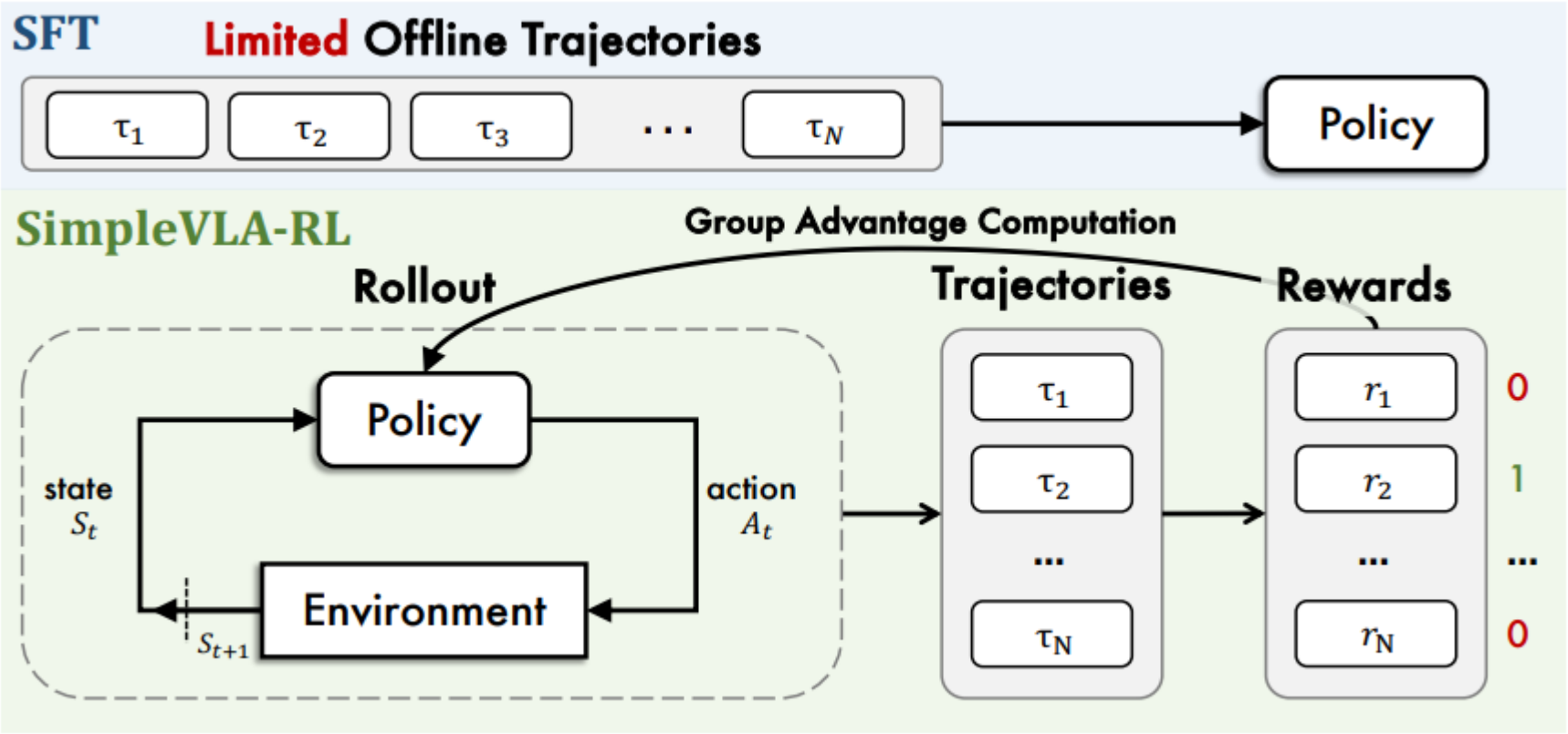
Goal: put the wine bottle **on top of the cabinet**

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.

SimpleVLA-RL

❖ Overall training process

- 특정 과제에 대한 데이터로 모델을 지도학습 기반으로 사전학습(Supervised Fine-Tuning; SFT)
- GRPO를 사용하여 SFT 기반의 사전학습 모델을 튜닝(SimpleVLA-RL)



SimpleVLA-RL

❖ Pseudo-code (adapting from LLM to VLA)

- LLM이 문장을 생성 → 주어진 query에 대한 답변을 생성(한 번의 상호작용)
- VLA는 trajectory를 생성 → 환경과 여러 번 상호작용(for loop)하면서 일련의 행동을 생성

```
def rollout(policy, dataset, number_sample=8, max_steps=None):
  rollout_dataset = []
  for batch in dataset:
     batch = batch.repeat(number_sample)
-    # LLM generates diverse outputs using random sampling
-    outputs = policy.generate(batch, temperature=1.0)
-    rollout_dataset.append((batch, outputs))
+    # Parallel env initialization and interaction
+    envs = env_process_pool.submit(batch.initialize)
+    states = env_process_pool.submit(envs.setup)
+    for t in range(max_steps):
+       # VLA generates diverse trajectories using temperature
   sampling on action tokens
+       actions = policy.generate(states, temperature=1.0)
+       rollout_dataset.append({f"{e.name}_step_{t}": (s,a) for e
   ,s,a in zip(envs,states,actions)})
+       states, dones = env_process_pool.submit(envs.step,
   actions)
+       # Remove completed tasks
+       active = [(e,s) for e,s,d in zip(envs,states,dones) if
   not d]
+       if not active:
+          break
+       envs, states = zip(*active)
  return rollout_dataset
```

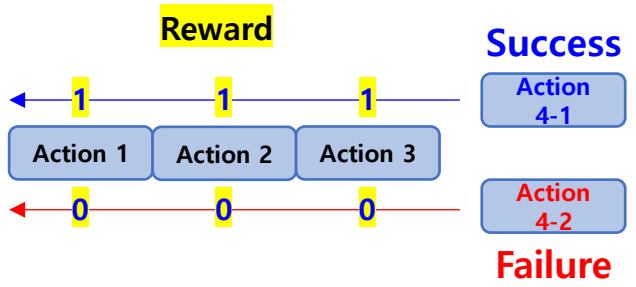
→ LLM에서의 rollout

→ 병렬 수행 구문

→ 최대 step까지 진행하여 step이 누적된 trajector를 저장

→ VLA에서의 rollout

SimpleVLA-RL



❖ Reward modeling

- 성공/실패 여부에 따라서 해당 에피소드의 보상은 모두 동일하게 제공
- GRPO에서는 trajectory 단위로 가치 평가(advantage 계산)를 수행하기 때문에 통일하는 듯?

3.2. Outcome Reward Modeling

이진 보상 함수

SimpleVLA-RL employs a straightforward binary reward function for RL training. Unlike traditional RL approaches that require carefully crafted reward functions [Hadfield-Menell et al., 2017; Knox et al., 2023; Booth et al., 2023], we follow DeepSeek-R1's approach by assigning trajectory-level rewards of either 0 or 1 based solely on task completion. When the VLA model successfully completes a task, the entire trajectory is assigned a reward of 1; otherwise, it receives a reward of 0. For gradient computation, these trajectory-level rewards are uniformly propagated to the individual action tokens. Consequently, all tokens within successful trajectories are assigned a reward of 1, whereas those in unsuccessful trajectories are assigned a reward of 0. Our reward function is:

Trajectory 단위로 reward를 부여

굉장히 단순한 credit assignment
→ 성공하면 모든 action token에 reward 1을 부여

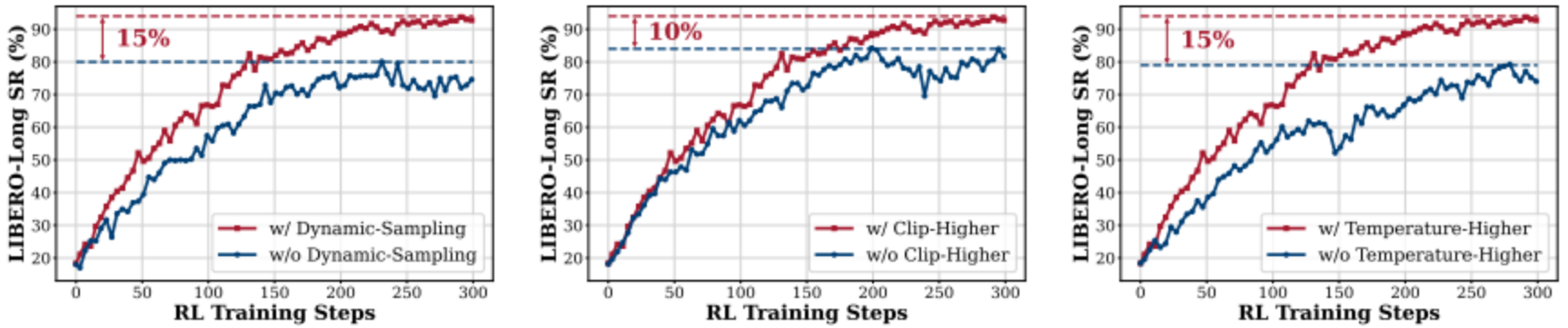
$$R(a_{i,t} \mid s_{i,t}) = \begin{cases} 1, & \text{is\_successful}[\text{traj}_i(a_i, s_i)], \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

This simple outcome-level reward is simple yet effective: scalable, broadly applicable across environments, and free from complex process-based design [Wu et al., 2021]. By focusing solely on task completion, it avoids the non-transferability issues typical of task-specific rewards.

SimpleVLA-RL

❖ 핵심 테크닉

- Dynamic sampling
- Clip higher
- Higher rollout temperature



(a) Dynamic Sampling **(b)** Clip Higher **(c)** Higher Rollout Temperature

Figure 3 | The effectiveness of three key enhancements: dynamic sampling, higher rollout temperature, and clip higher.

SimpleVLA-RL

❖ 학습 테크닉 1 – Dynamic Sampling

- GRPO 처럼 advantag를 계산하는 경우 모든 trajectory의 reward가 동일하면 기울기가 0이 나와 학습을 방해
- 이를 방지하기 위하여 모든 trajectory의 reward가 동일한 그룹은 제거

Dynamic Sampling Critic-free RL algorithms suffer from vanishing gradients when trajectories are assigned the same rewards. For example, GRPO computes advantages using group-relative normalization, comparing each response's reward to the mean and standard deviation of rewards within its group of sampled outputs. When all trajectories share identical rewards, their advantage estimation becomes zero, resulting in null gradients and causing unstable training dynamics.

문제상황:

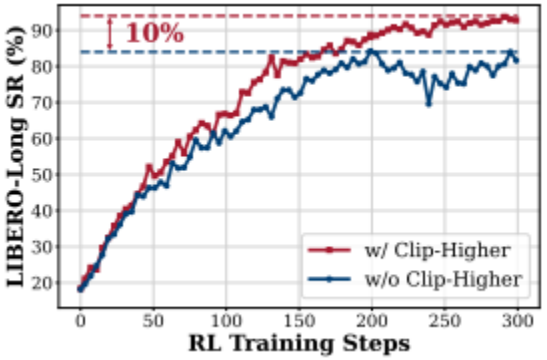
We address this challenge through Dynamic Sampling [Yu et al., 2025; Cui et al., 2025a], a method that has been proven effective in LLM RL [Team et al., 2025b; Yu et al., 2025; Cui et al., 2025a; Shi et al., 2025]. During rollout, we exclude groups in which all trajectories either succeed or fail. Sampling proceeds until the batch consists solely of groups with mixed outcomes, which can be formally expressed as:

해결방법:

$$0 < \left|\{\text{traj}_i(a_i, s_i) \mid \text{is\_successful}[\text{traj}_i(a_i, s_i)]\}\right| < G. \tag{10}$$

This ensures non-zero advantage estimates and stable gradient flow throughout training.

SimpleVLA-RL



❖ 학습 테크닉 2 – Clipping Higher

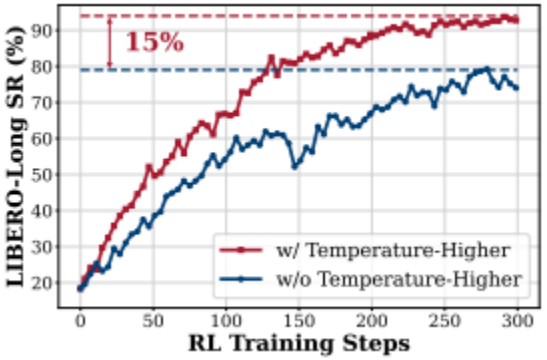
- 성공한 경험 방향으로 점차 탐험을 수행하기 위함
- Clipping이 너무 강하면 오히려 모델 업데이트가 잘 되지 않아서 성공하는 방향으로 다시 접근이 어려울 수 있음

문제상황:

Clipping Higher PPO and GRPO employ clipping over the importance sampling ratio to restrict the trust region [Schulman et al., 2015] and enhance RL stability [Shao et al., 2024; Schulman et al., 2017]. However, the upper clipping threshold restricts the probability increase of low-probability tokens, thereby potentially constraining exploration. Following DAPO [Yu et al., 2025], we modify the clipping range in the GRPO training objective from [0.8, 1.2] to [0.8, 1.28].

해결방법:

SimpleVLA-RL



❖ 학습 테크닉 3 – Higher Rollout Temperature

- 행동 다양성을 확보하기 위함 → 이것도 탐험을 활성화하기 위한 수단 중 하나로 보임

발견: 적용:

Higher Rollout Temperature Recent works on LLM RL adjusting the rollout temperature to promote exploration have been widely shown to be effective, with sampling at higher temperatures yielding particularly notable improvements [Liu et al., 2025c; An et al., 2025; Liao et al., 2025]. To encourage the VLA model to generate more diverse trajectories during the rollout phase, we increase the sampling temperature from 1.0 to 1.6. As shown in Figure 3, these modifications led to notable improvements.

SimpleVLA-RL

❖ RL objective function

- On-policy이지만 importance sampling을 쓰는 이유는 미니 배치로 업데이트를 진행하기 때문 (학습 step마다 조금씩 업데이트)
- 따라서 학습 안정성을 위해서 클리핑을 통해 행동을 선택하는 확률분포가 서서히 변하도록 함
- 탐험을 크게 방해하는 KL-divergence term은 삭제

기존 (LLM)

$$J_{\text{GRPO}}(\theta) = \mathbb{E}_{s_0 \sim \mathcal{D}, \{\tau_i\} \sim \pi_{\theta_{\text{old}}}} \left[\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|\tau_i|} \sum_{t=1}^{|\tau_i|} \min \left(r_{i,t}(\theta) \hat{A}_i, \text{clip}(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon) \hat{A}_i \right) - \beta D_{\text{KL}}(\pi_\theta || \pi_{\text{ref}}) \right]$$

제안 (SimpleVLA-RL)

$$\mathcal{J}(\theta) = \mathbb{E}_{s_0 \sim \mathcal{D}, \{a_t\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot|s_t)} \left[\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|a_i|} \sum_{t=1}^{|a_i|} \min \left(r_{i,t}(\theta) \hat{A}_i,\ \text{clip}\left(r_{i,t}(\theta), 1-\varepsilon_{low}, 1+\varepsilon_{high}\right) \hat{A}_i \right) \right]$$

$$\text{s.t.} \quad 0 < \left| \{ \text{traj}_i(a_i, s_i) \mid \text{is\_successful}[\text{traj}_i(a_i, s_i)] \} \right| < G,$$

SimpleVLA-RL

❖ Experiment

- 기본 모델은 OpenVLA-OFT를 사용하였으며 여기에 SFT와 SimpleVLA-RL 방법론을 적용하여 성능을 비교함
- 최대 상호작용 횟수 -→ LIBERO: 512 step / RoboTwin1.0&2.0: Task마다 다르며 200, 400, 800 step

Model	LIBERO				
	Spatial	Object	Goal	Long	Avg
Octo	78.9	85.7	84.6	51.1	75.1
OpenVLA	84.7	88.4	79.2	53.7	76.5
Nora	92.2	95.4	89.4	74.6	87.9
π_0 + FAST	96.4	96.8	88.6	60.2	85.5
π_0	96.8	98.8	95.8	85.2	94.2
UniVLA	96.5	96.8	95.6	92.0	95.2
OpenVLA-OFT	91.6	95.3	90.6	86.5	91.0
w/ ours	**99.4**	**99.1**	**99.2**	**98.5**	**99.1**
Δ	+7.8	+3.8	+8.6	+12.0	+8.1

Model	RoboTwin1.0				Avg
	Hammer Beat	Block Handover	Blocks Stack	Shoe Place	
DP	0.0	12.0	7.1	4.3	5.9
DP3	64.7	84.3	24.0	59.3	58.1
OpenVLA-OFT	67.2	61.6	7.1	23.4	39.8
w/ ours	**92.6**	**89.6**	**40.2**	**59.3**	**70.4**
Δ	+25.4	+28.0	+33.1	+35.9	+30.6

RoboTwin 2.0	Short Horizon Tasks (100-130 Steps)				
Model	Lift Pot	Beat Hammer Block	Pick Dual Bottles	Place Phone Stand	Avg
π_0	51.0	59.0	50.0	22.0	45.5
RDT	45.0	22.0	18.0	13.0	24.5
OpenVLA-OFT	10.1	28.1	29.7	17.1	21.3
w/ ours	**64.1**	**87.5**	**68.3**	**39.6**	**64.9**
Δ	+54.0	+59.4	+38.6	+22.5	+43.6
Medium Horizon Tasks (150-230 Steps)					
Model	Move Can Pot	Place A2B Left	Place Empty Cup	Handover Mic	Avg
π_0	41.0	38.0	60.0	96.0	58.8
RDT	33.0	21.0	42.0	95.0	47.8
OpenVLA-OFT	28.1	37.5	77.3	45.3	47.1
w/ ours	**61.2**	**45.3**	**94.2**	**89.2**	**72.5**
Δ	+33.1	+7.8	+16.9	+43.9	+25.4
Long (280-320 Steps) & Extra Long Horizon Tasks (450-650 Steps)					
Model	Handover Block	Stack Bowls Two	Blocks Rank Rgb	Put Bottles Dustbin	Avg
π_0	39.0	53.0	45.0	36.0	43.3
RDT	26.0	42.0	17.0	26.0	27.8
OpenVLA-OFT	33.1	40.6	70.2	42.2	46.5
w/ ours	**57.8**	**75.8**	**81.3**	**60.9**	**69.0**
Δ	+24.7	+35.2	+11.1	+18.7	+22.4
Overall Avg	**RDT:** 33.3	π_0: 49.2	**OpenVLA-OFT:** 38.3	**w/ ours:** 68.8	+30.5

SimpleVLA-RL

❖ Analysis

- Overcoming data scarcity : 얼마나 극단적으로 SFT 데이터를 줄여도 좋은 성능을 달성할 수 있는가? (기존 VLA는 데이터 양에 의존적)
- Generalization analysis : 처음보는 공간, 물체, 그리고 과제에 대해서 얼마나 강건한 성능을 보이는가?
- Real-world experiments : SimpleVLA-RL로 학습된 모델은 sim-to-real transfer가 얼마나 잘 되는가? (RL 기반의 학습이라서 하는 듯)

SimpleVLA-RL

❖ Overcoming data scarcity

- SFT에서 demonstration 데이터를 하나만 사용하였을 때(One-Trajectory SFT)의 성능 비교
- 너무 당연한 결과 같은데 RL을 쓰는 효과를 극명하게 보여주려고 한 실험인 듯...

Table 5 | Comparisons between One-Trajectory and Full-Trajectory SFT on LIBERO.

Model	LIBERO				
	Spatial	Object	Goal	Long	Avg
One-Trajectory SFT					
OpenVLA-OFT	63.6	54.9	59.6	17.3	48.9
w/ ours	**98.2**	**98.7**	**98.8**	**91.7**	**96.9**
Δ	+34.6	+43.8	+39.2	+74.4	+48.0
Full-Trajectory SFT					
OpenVLA-OFT	91.6	95.3	90.6	86.5	91.0
w/ ours	**99.4**	**99.1**	**99.2**	**98.5**	**99.1**
Δ	+7.8	+3.8	+8.6	+12.0	+8.1

SimpleVLA-RL

❖ Generalization analysis

- 처음 보는 상황 (spatial, object, goal)에 대해서 얼마나 잘 동작하는지를 평가
- 우선 One-Trajectory SFT로 학습한 후 → 추가로 450개의 demo를 SFT로 수행하거나 RL을 수행하는 방식으로 학습 및 비교
- SFT는 단순히 암기하는 경향이 있어서 일반화 성능이 떨어짐. / 반면 RL은 목적을 달성하기 위한 다양한 시도를 통해서 일반화 성능을 확보

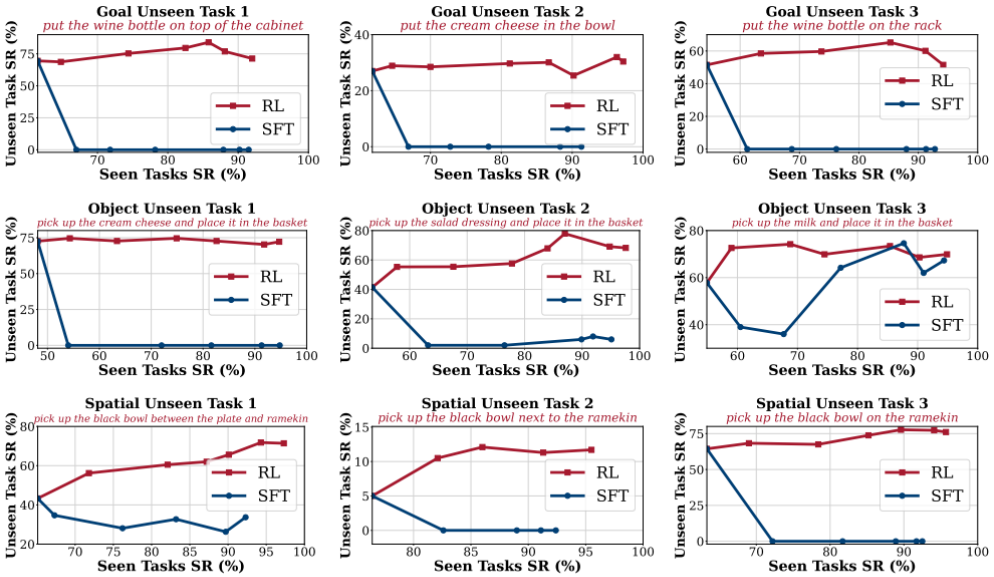


Figure 4 | Generalization Analysis on LIBERO: Goal Unseen **(Top)**, Object Unseen **(Middle)**, Spatial Unseen **(Bottom)**.

새로운 지시어를 잘 이해하는가?

물체의 특성이 바뀌어도 잘 작동하는가?

물체의 위치나 배치가 바뀌어도 잘 작동하는가?

SimpleVLA-RL

❖ **Real-world experiments**

- 현실 데이터 없이 오직 시뮬레이션 상으로만 학습을 수행
- 50번의 검증을 수행한 평균 점수를 비교

Table 6 | Real-world experiment (sim2real) results.

	Stack Bowls	Place Empty Cup	Pick Bottle	Click Bell	Avg
RDT	60.0	4.0	10.0	20.0	23.5
OpenVLA-OFT	38.0	2.0	0.0	30.0	17.5
w/ ours	70.0	10.0	14.0	60.0	38.5
Δ	+32.0	+8.0	+14.0	+30.0	+21.0

SimpleVLA-RL

❖ Discussions – Failure Modes of SimpleVLA-RL

- Demo 기반의 SFT가 없이는 RL을 사용하여도 성능 향상이 없음
- SFT의 초기 성능에 의존하는 경향이 있음
  - SFT 성능이 좋아야 RL을 활용했을 때 성능 향상 폭이 큼

Table 7 | Impact of initial model capability on **SimpleVLA-RL** performance.

	RoboTwin2.0					
	Move Can Pot	**Place A2B Lift**	**Place A2B Right**	**Place Phone Stand**	**Pick Dual Bottles**	**Avg**
0 trajs SFT	0	0	0	0	0	0
+RL	0	0	0	0	0	0
100 trajs SFT	9.4	7.8	7.8	10.1	1.2	7.3
+RL	51.6	25.0	27.2	18.8	4.3	25.4
Δ	+42.2	+17.2	+19.4	+8.7	+3.1	+18.1
1000 trajs SFT	28.1	37.5	28.7	17.1	29.7	28.2
+RL	61.2	45.3	37.5	39.6	68.3	50.4
Δ	+33.1	+7.8	+8.8	+22.5	+38.6	+22.2