Adapt On-The-Go: Behavior modulation for single-life robot deployment

Annie S. Chen

2026.08.03

Learning and Adaptation in Changing Environments

MuliLAB Journal Club
인공지능응용학과 허세민

01 Author & Journal

Annie S. Chen

팔로우

Google DeepMind, Stanford University
stanford.edu의 이메일 확인됨 - 홈페이지

machine learning robotics

제목	인용	연도
On the opportunities and risks of foundation models R Bommasani, DA Hudson, E Adeli, R Altman, S Arora, S von Arx, ... arXiv preprint arXiv:2108.07258	11710	2021
Just train twice: Improving group robustness without training group information EZ Liu*, B Haghgoo*, AS Chen*, A Raghunathan, PW Koh, S Sagawa, ... International Conference on Machine Learning, 6781-6792	856	2021
Surgical fine-tuning improves adaptation to distribution shifts Y Lee*, AS Chen*, F Tajwar, A Kumar, H Yao, P Liang, C Finn arXiv preprint arXiv:2210.11466	348	2022
Adapt on-the-go: Behavior modulation for single-life robot deployment AS Chen, G Chada, L Smith, A Sharma, Z Fu, S Levine, C Finn arXiv preprint arXiv:2311.01059	13	2023

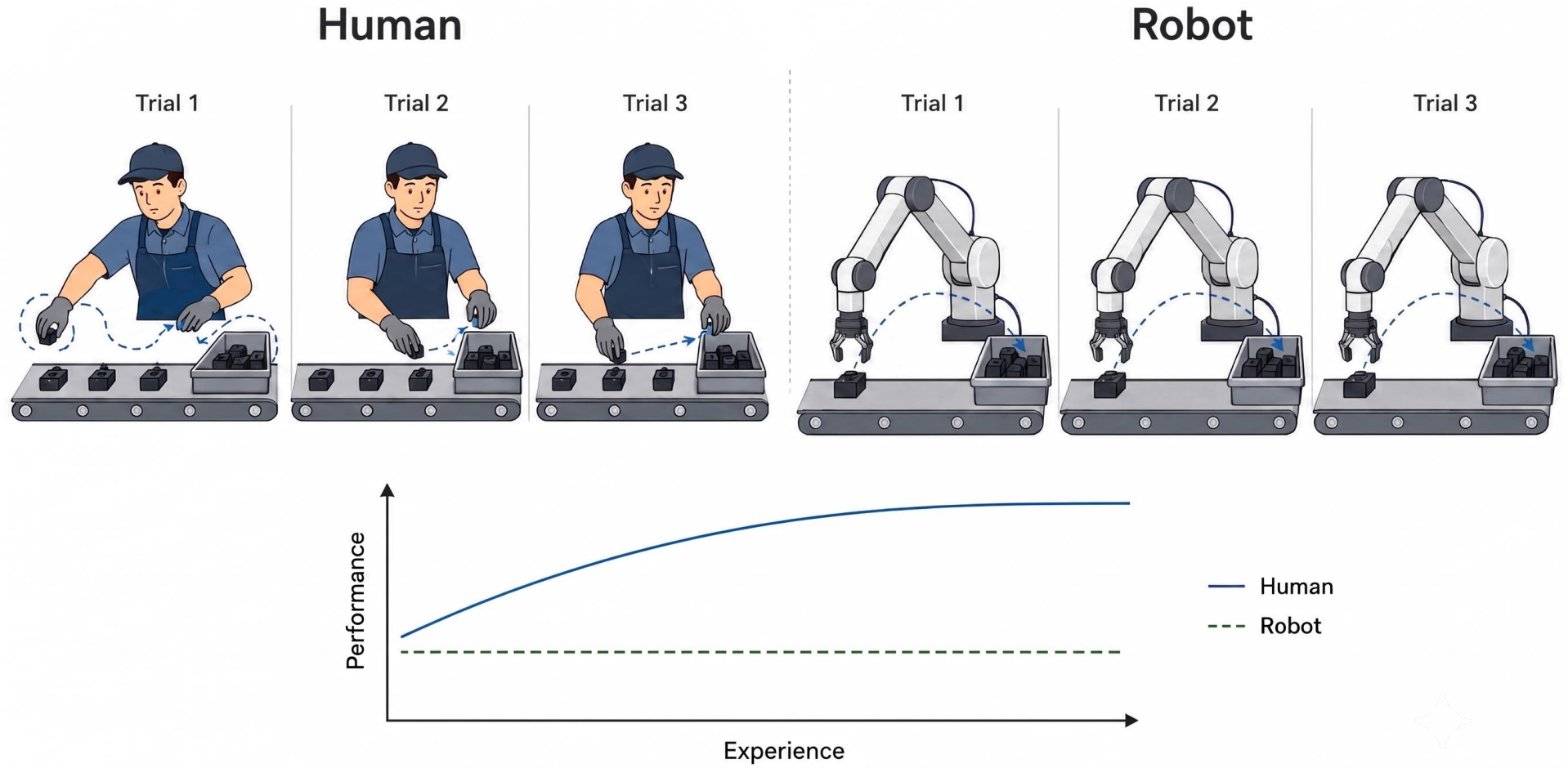
Overview

Introduction

01 **Competition** : Humans and robots improve in different ways.

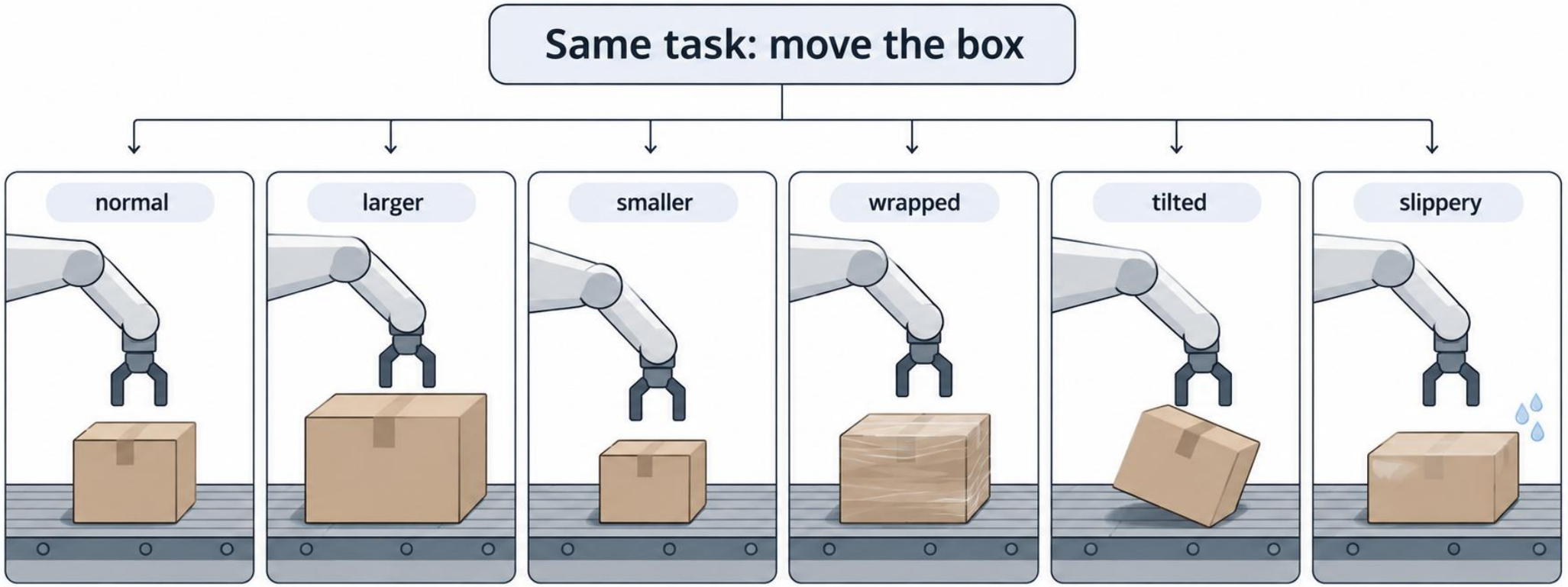
Who will be faster after repeated trials?

01 **Practice** : Repeated experience can improve behavior



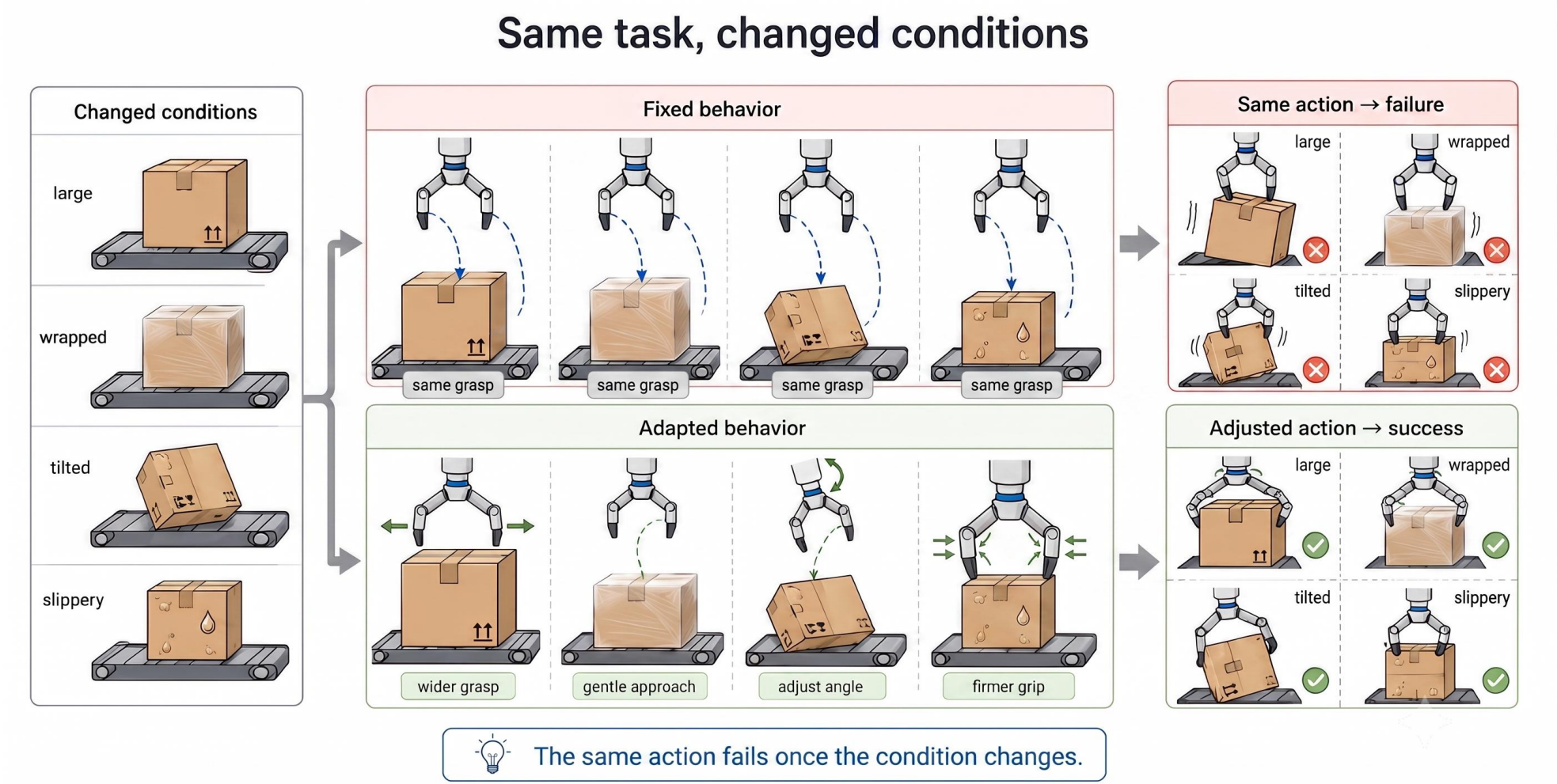
Experience can change future behavior.

01 **Variation** : Real deployment includes unexpected conditions.

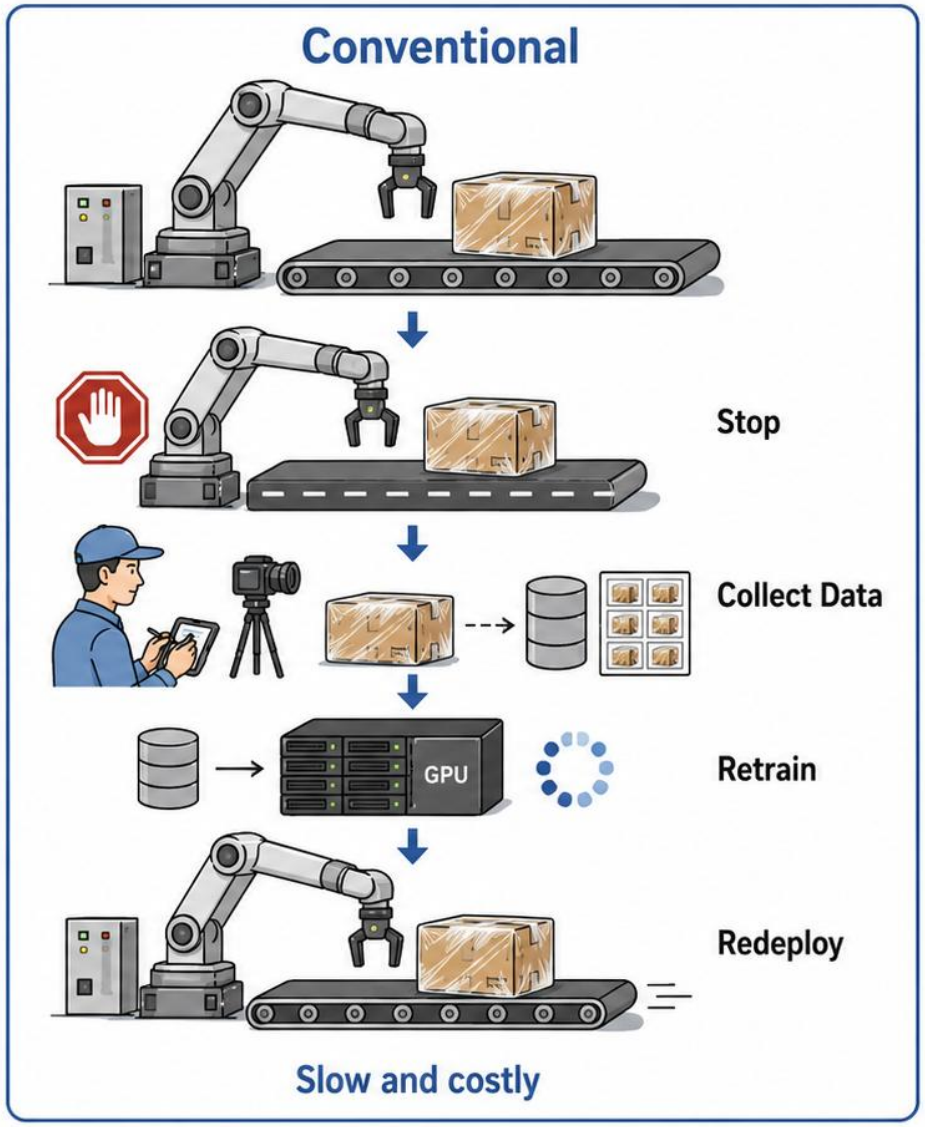


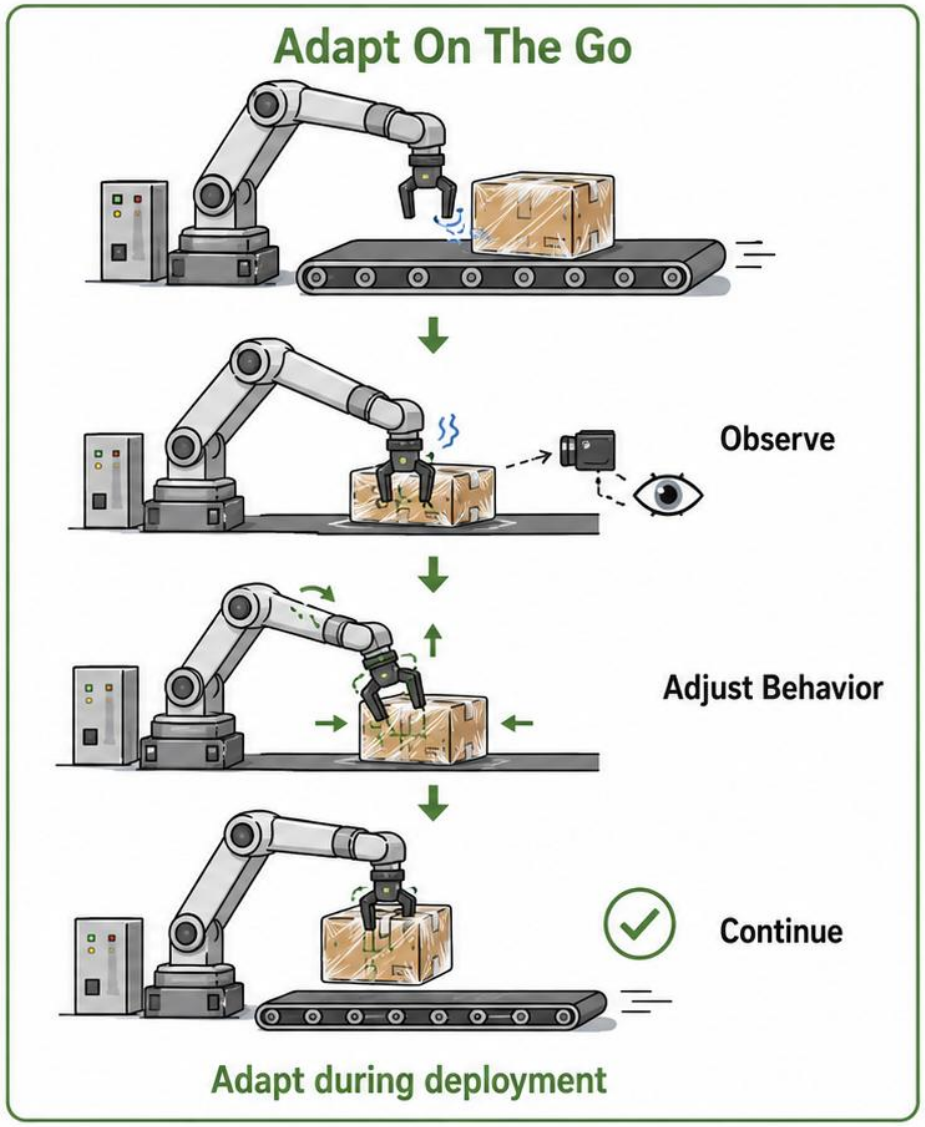
Unexpected situations often mean the same task under changed conditions.

01 **Challenge** : Robots must adapt without starting over.



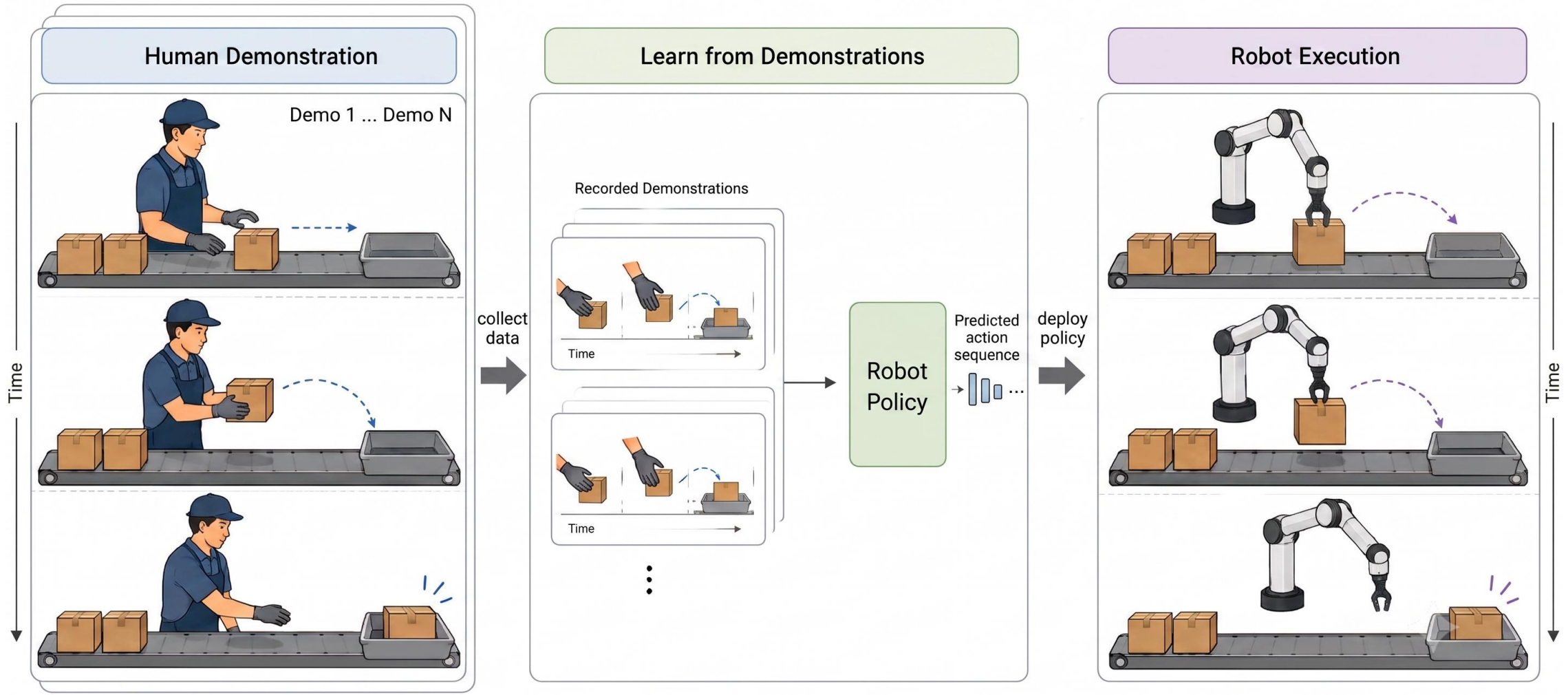
01 **Challenge** : Robots must adapt without starting over.





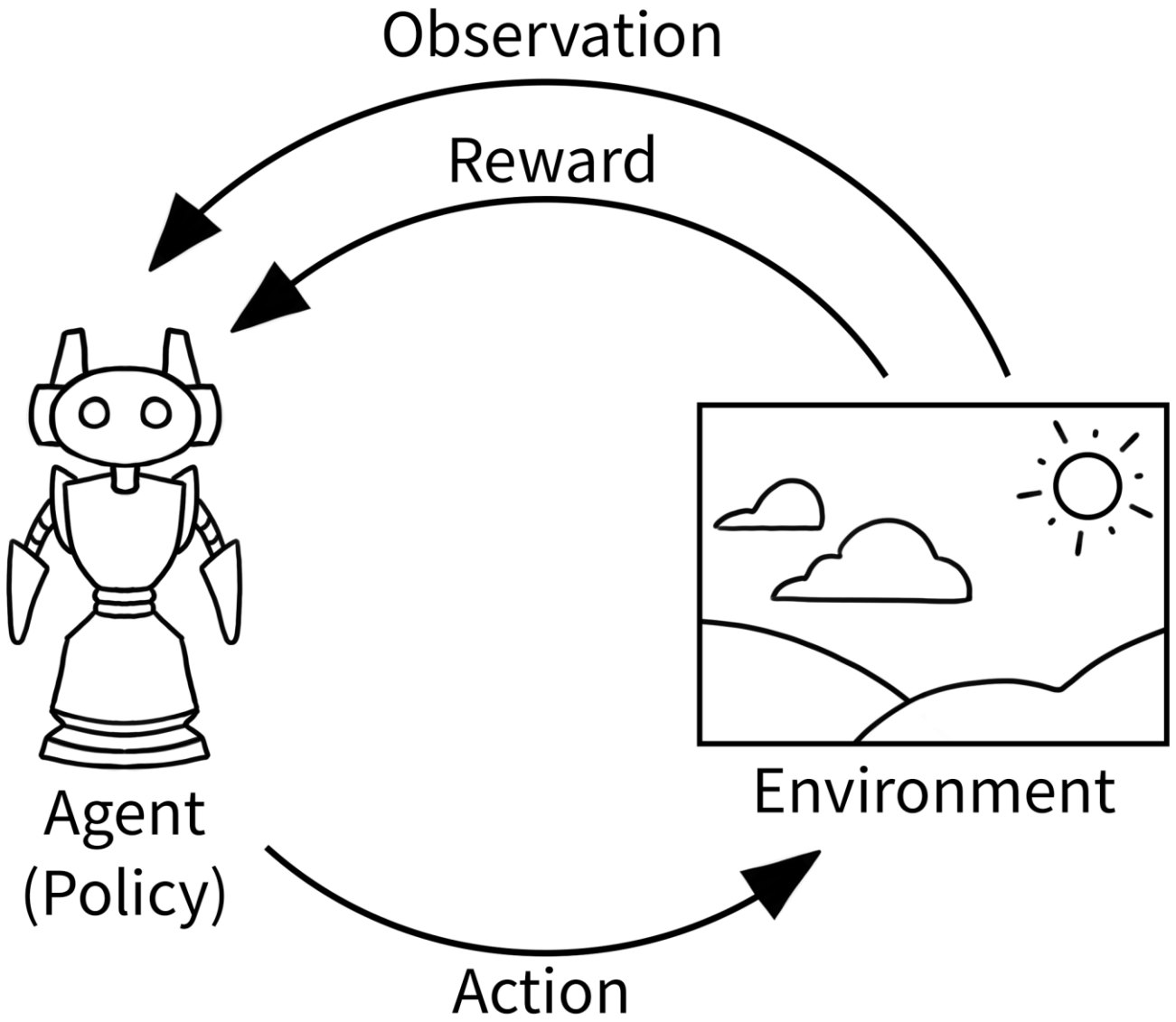
Foundations

02 **Imitation**: Robots often begin by learning from human demonstrations

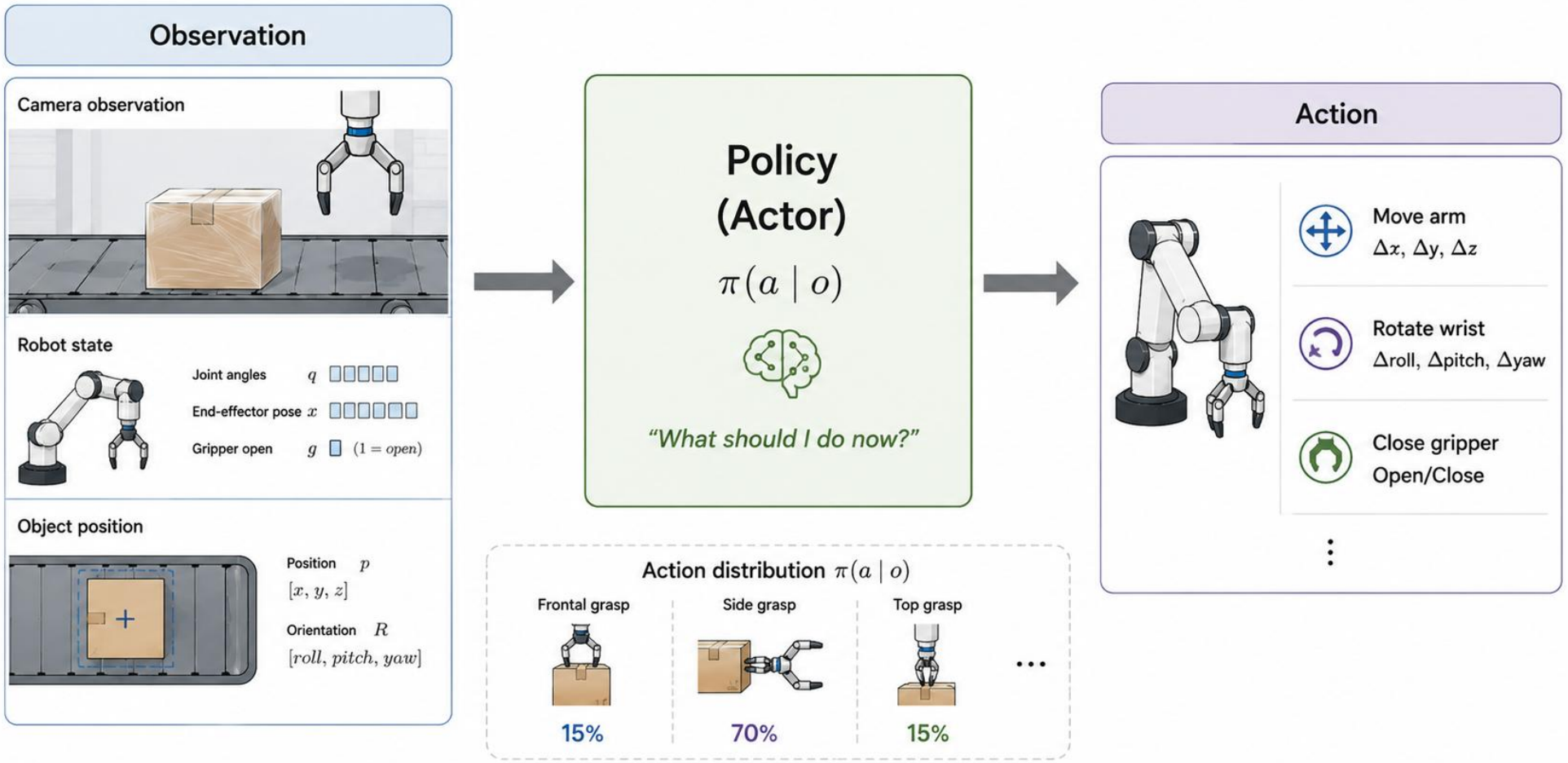


Robots often begin by learning from human demonstrations.

02 **Interaction** : The agent learns through action, feedback, and repeated updates.

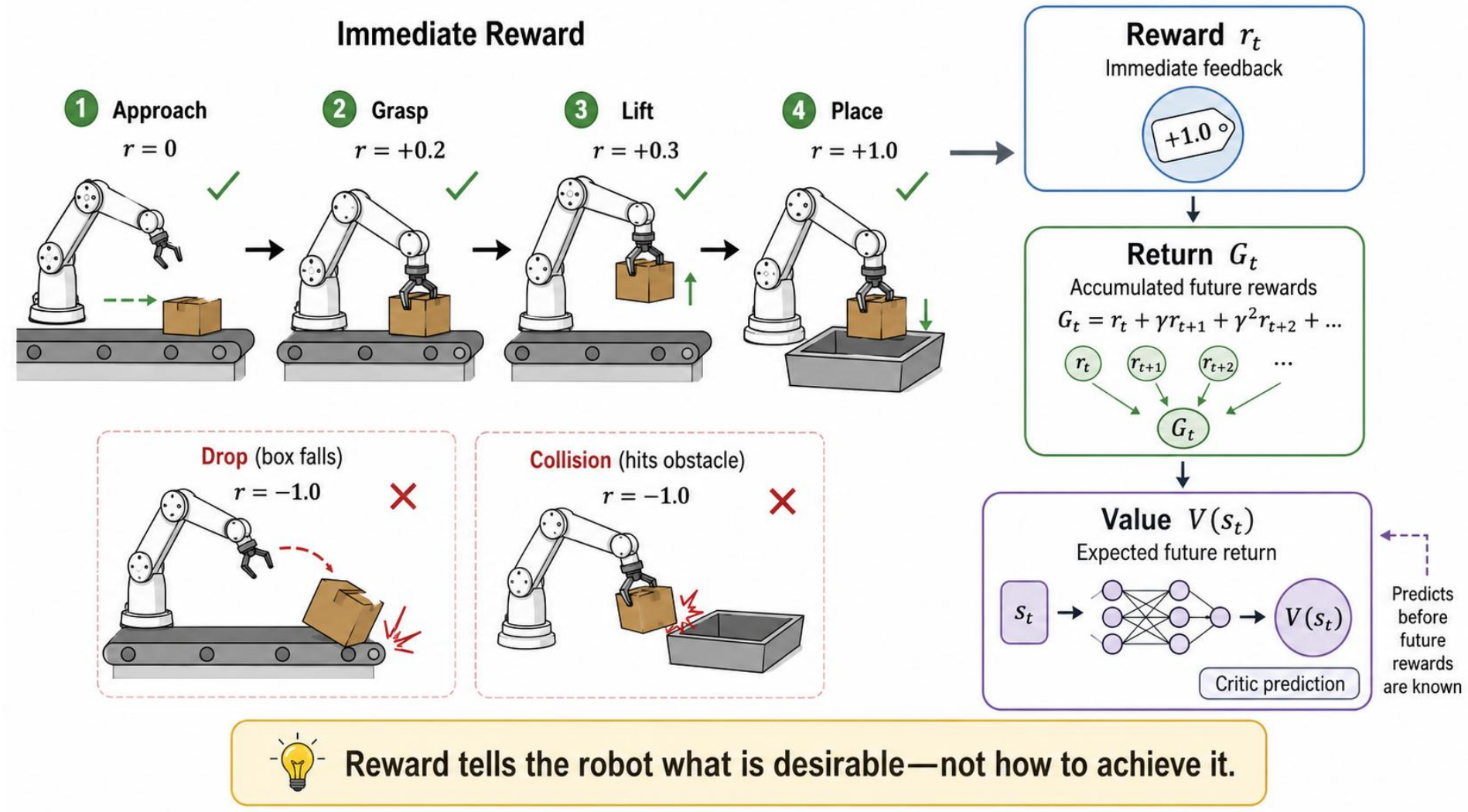


02 **Policy**: A policy maps the current observation to an action

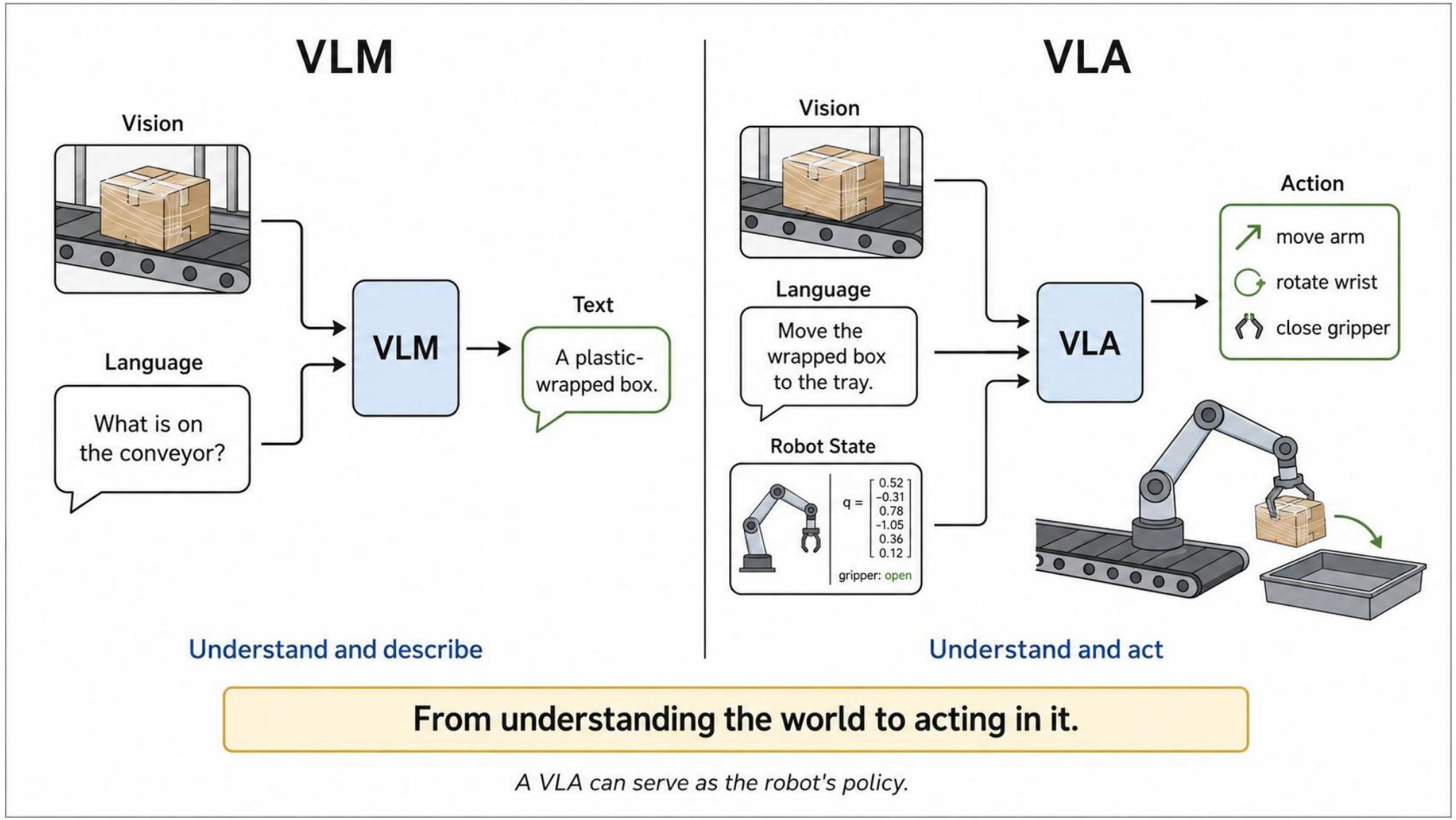


A policy maps the current observation to an action.

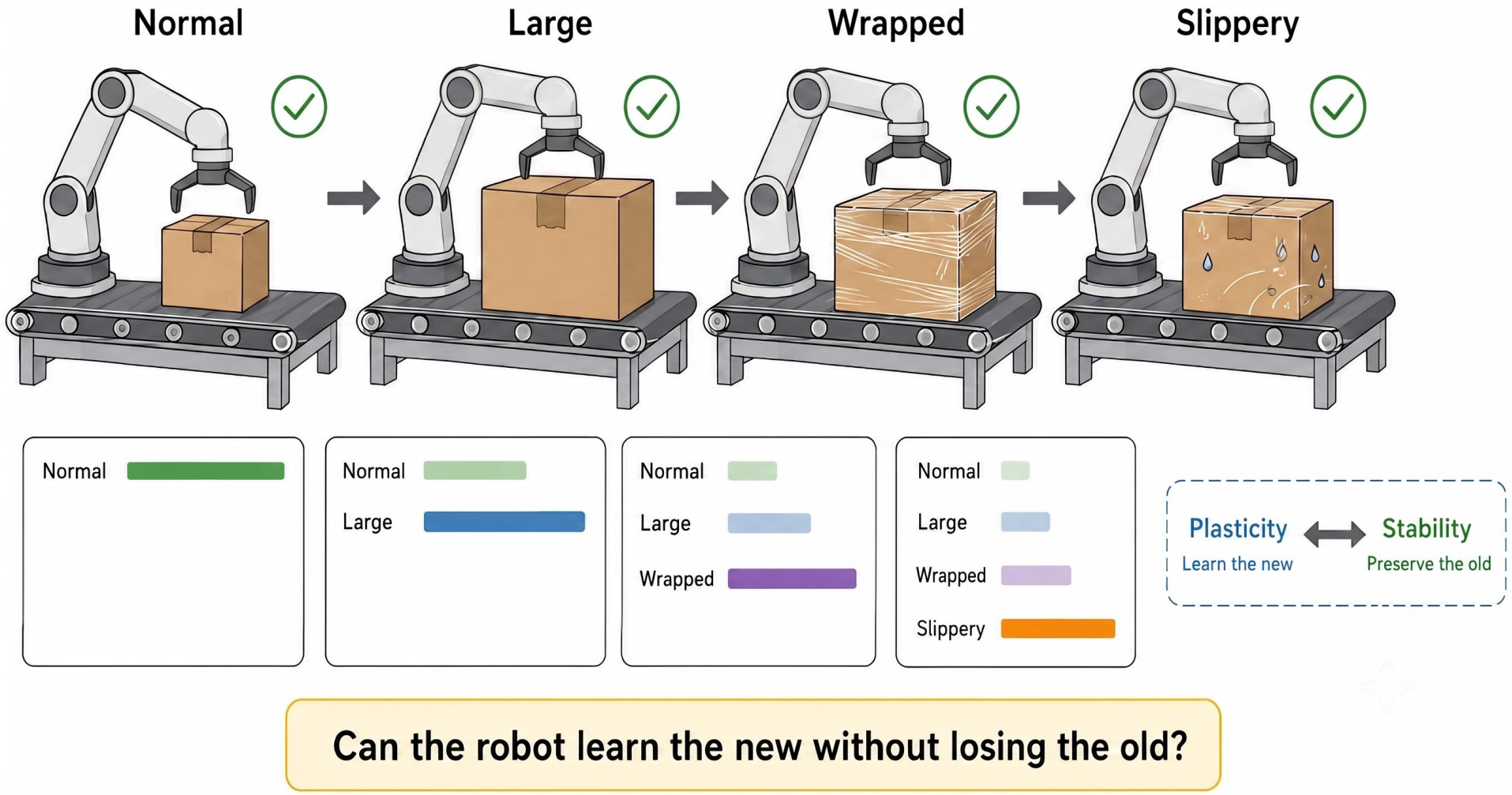
02 **Reward** : Reward evaluated outcomes, not individual actions



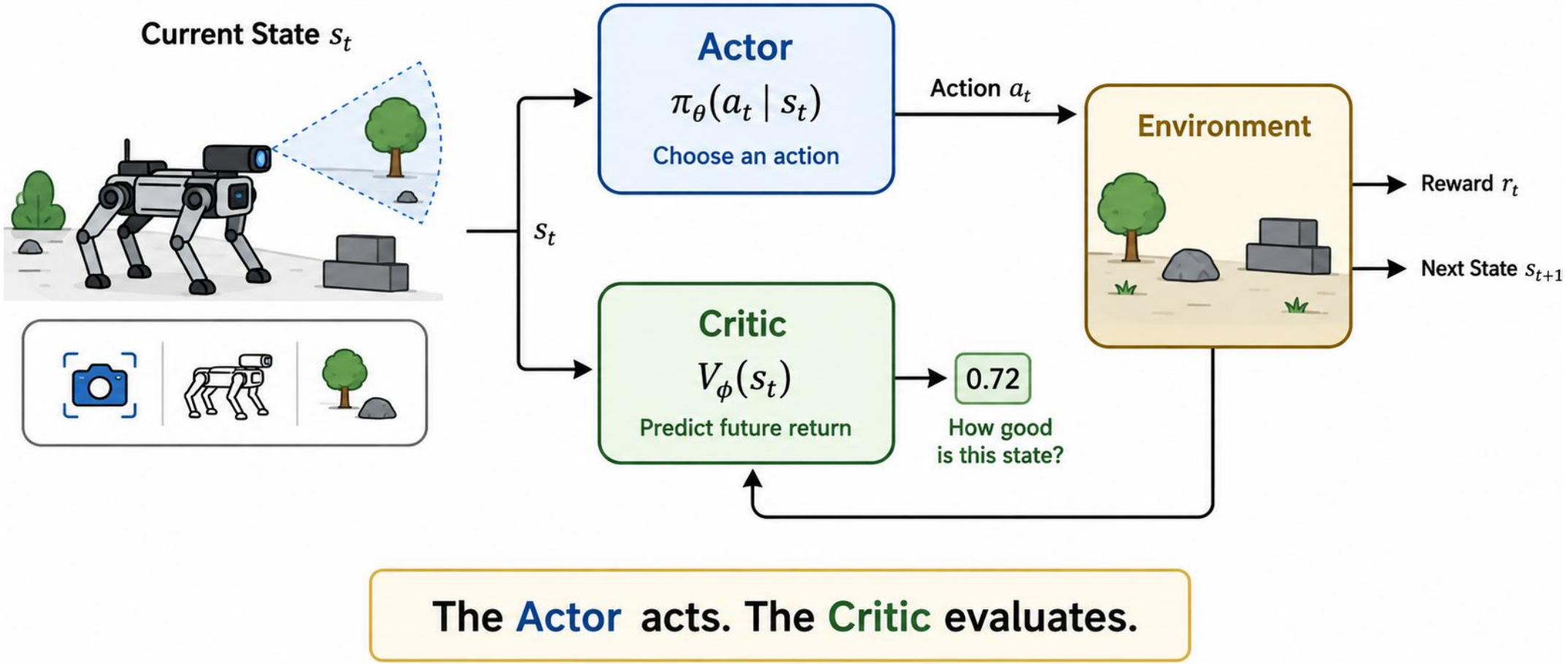
02 VLA: Reward evaluated outcomes, not individual actions



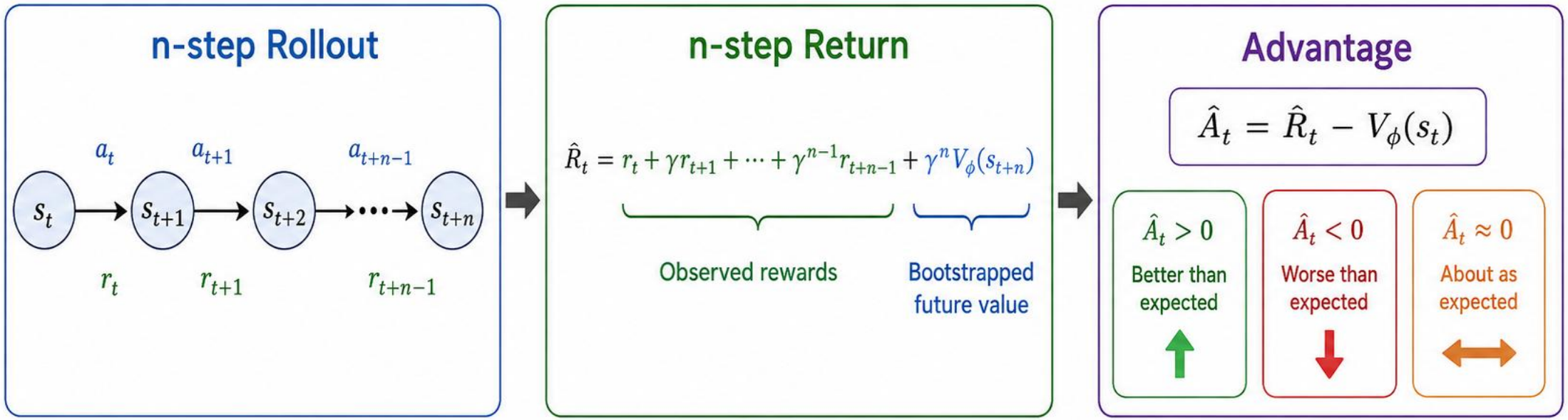
02 **Forgetting**: Adapting to a new condition can damage an old capability.



02 Actor-Critic : The actor chooses, while the critic evaluates what to expect.



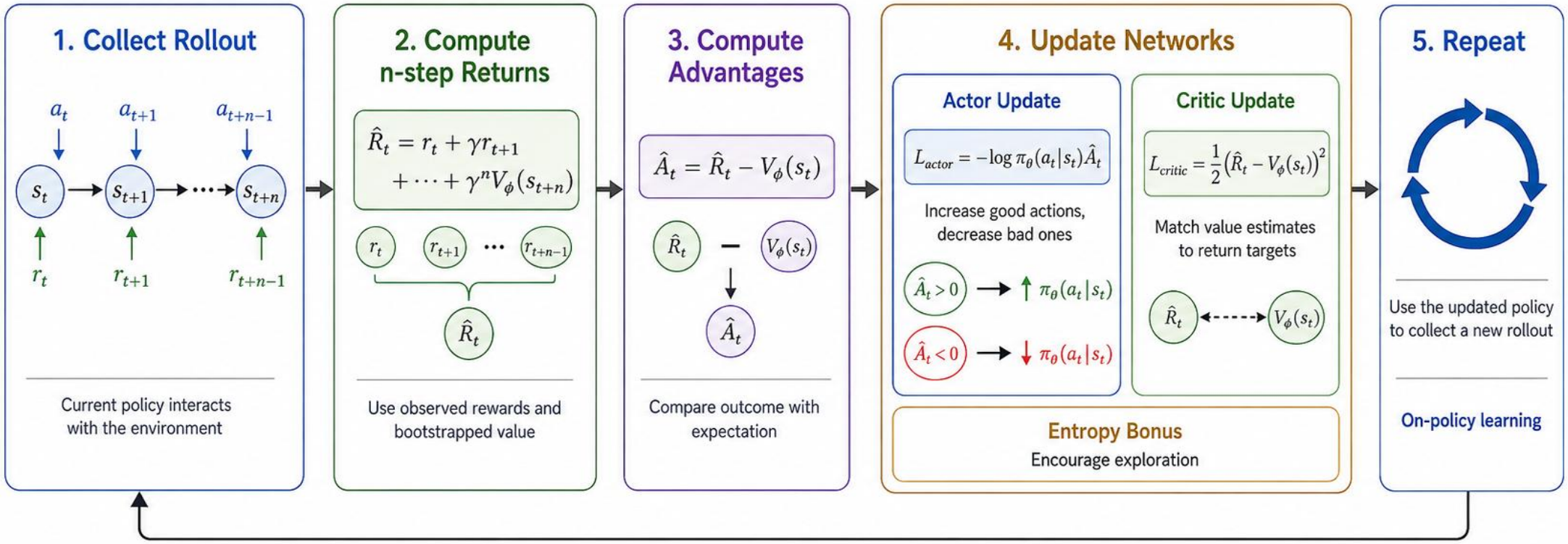
02 **Advantage** : A2C compares what happened with what the critic expected.



Advantage measures how surprising the outcome was.

It compares actual return with the Critic's expectation.

02 **A2C Update**: The same rollout updates both the policy and the value function.

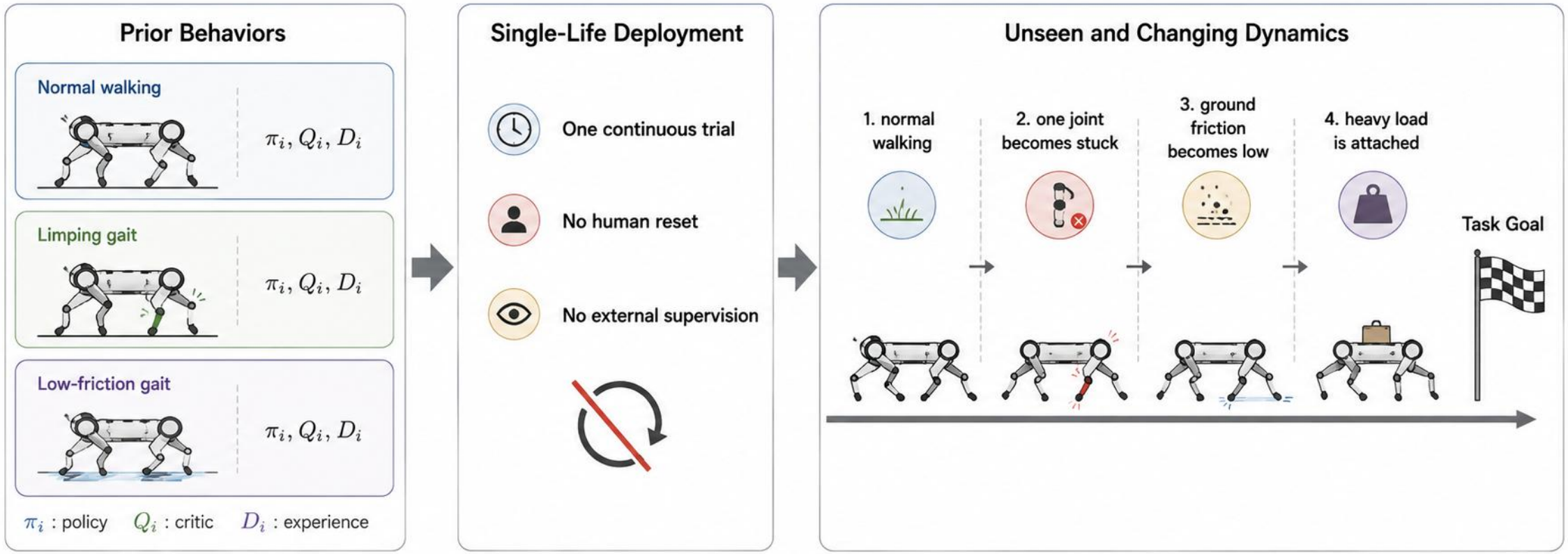


One rollout updates both the policy and the value function.

A2C learns from fresh on-policy experience.

Main papers

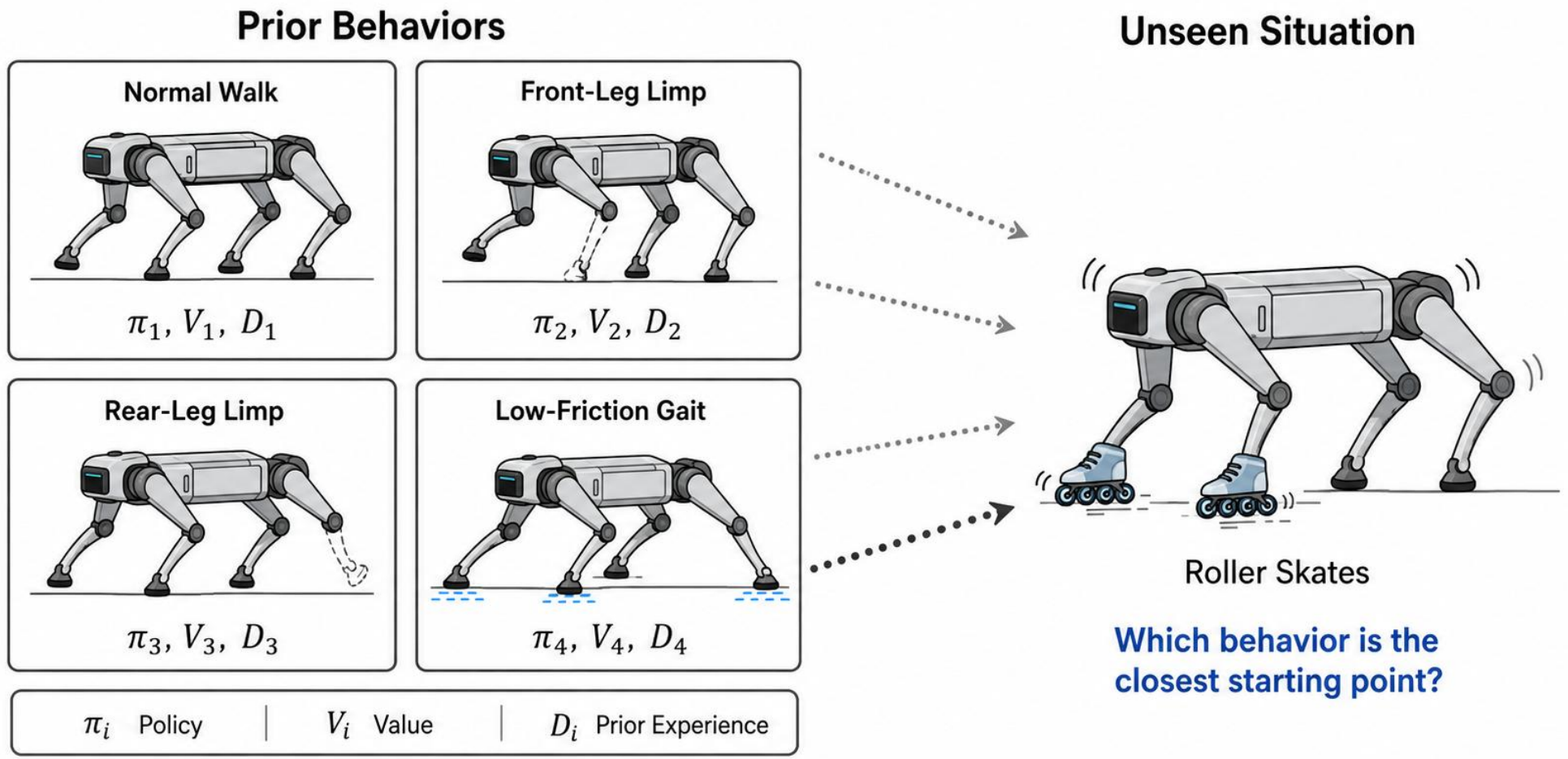
03 **Setting**: The robot must adapt within one uninterrupted deployment



Which behavior should the robot use right now?

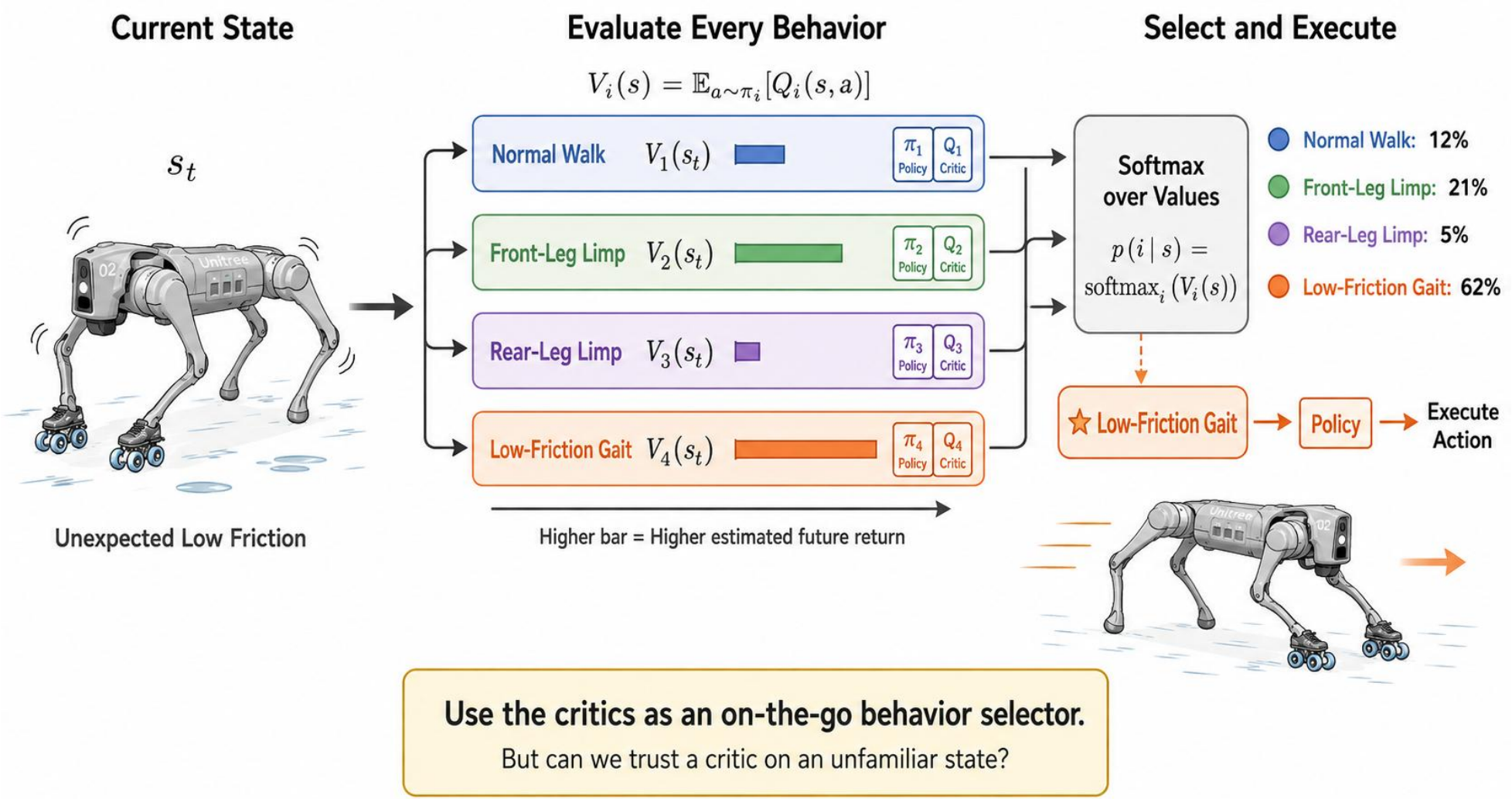
P_{target} may change over time

03 **Repertoire** : Diverse prior behaviors provide multiple starting points

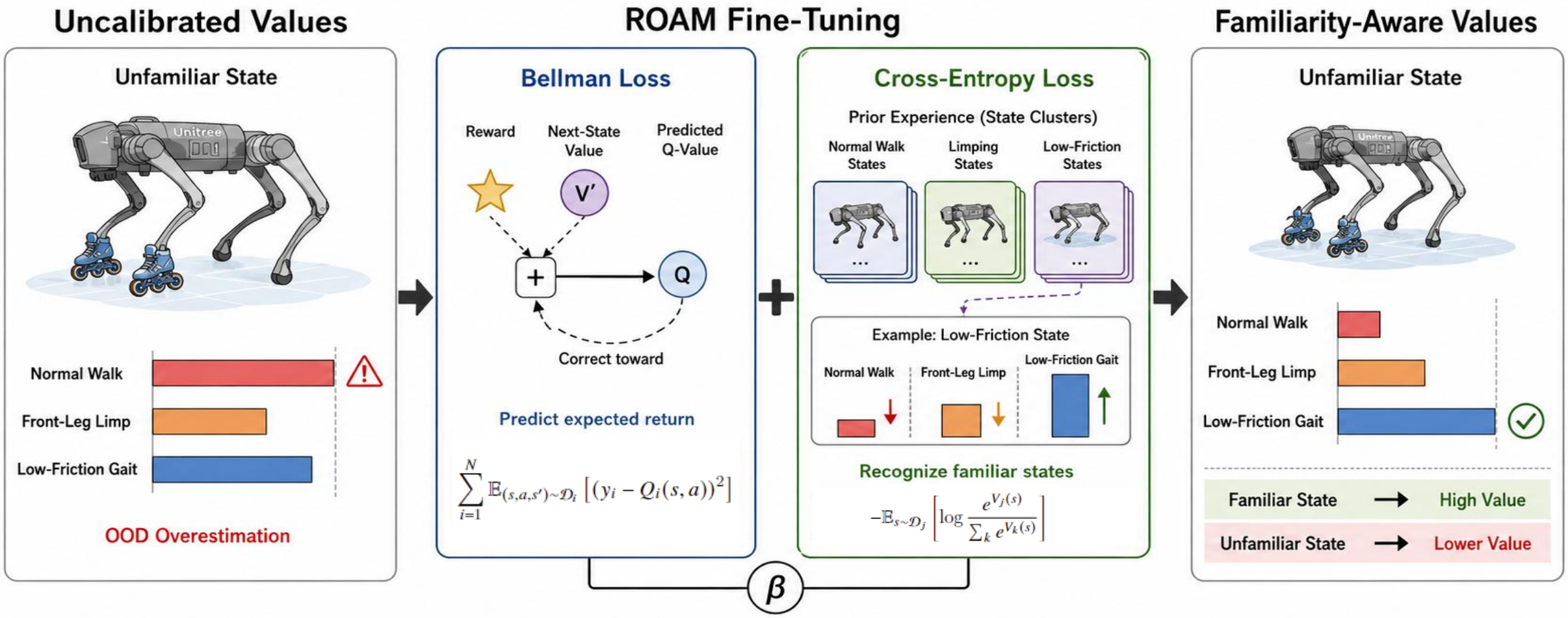


Do not adapt from zero. Start from a useful behavior.

03 **Selection** : Value functions estimate which behavior is useful now



03 **Loss**: Values should reflect both reward and familiarity.



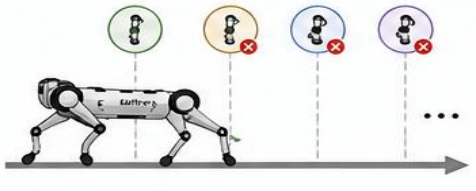
Bellman predicts usefulness. Cross-entropy calibrates familiarity.

More reliable values enable better behavior selection.

03 **Evaluation**: Can RAOM adapt efficiently to unseen dynamics?

Simulation

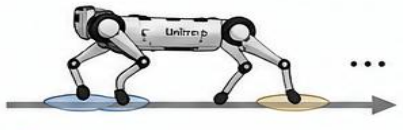
1 Dynamic Stiffness



Joint change every 100 steps

Walk 10 m

2 Dynamic Friction



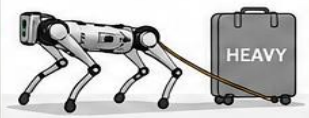
Friction change every 50 steps

Walk 10 m

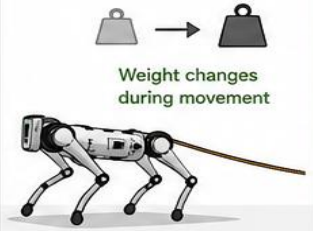
Metric: Steps to Completion ↓

Real-World Go1

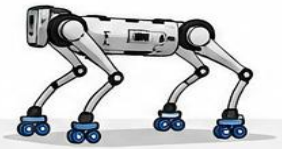
1 Heavy Luggage



2 Dynamic Load



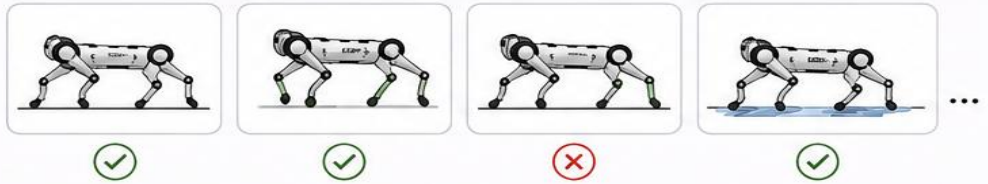
3 Roller Skates



Completion Time ↓ Falls / Readjustments ↓

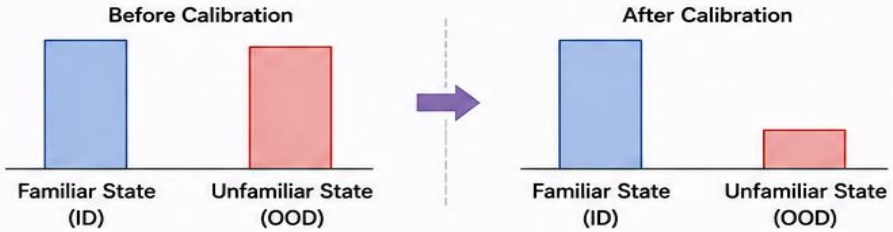
Why Does ROAM Work?

1 Behavior Selection Accuracy



Relevant behaviors Irrelevant behaviors

2 ID vs OOD Values



The experiments test both performance and mechanism.

RLPD · RMA · HLC · ROAM-NoCE · ROAM

03 **Simulation**: ROAM completes the task with fewer interactions

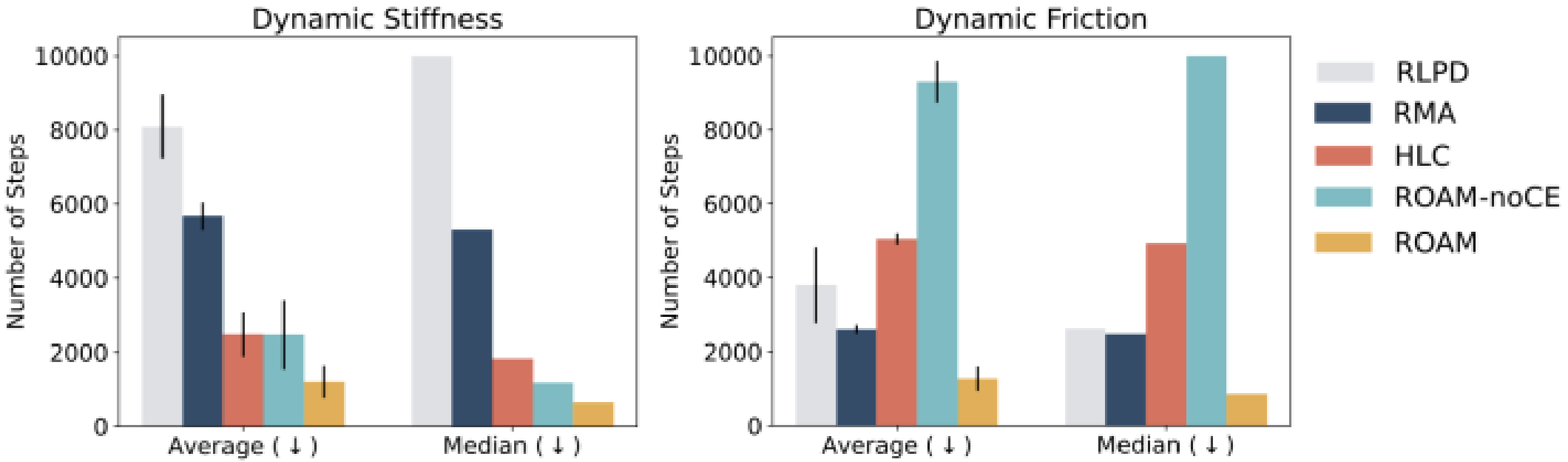


Figure 3: **Results on the simulated Go1 robot.** In both evaluation settings, ROAM is over 2x as efficient as all comparisons in terms of both average and median number of steps taken to complete the task.

03 **Analysis**: Cross-entropy improves relevant behavior selection.

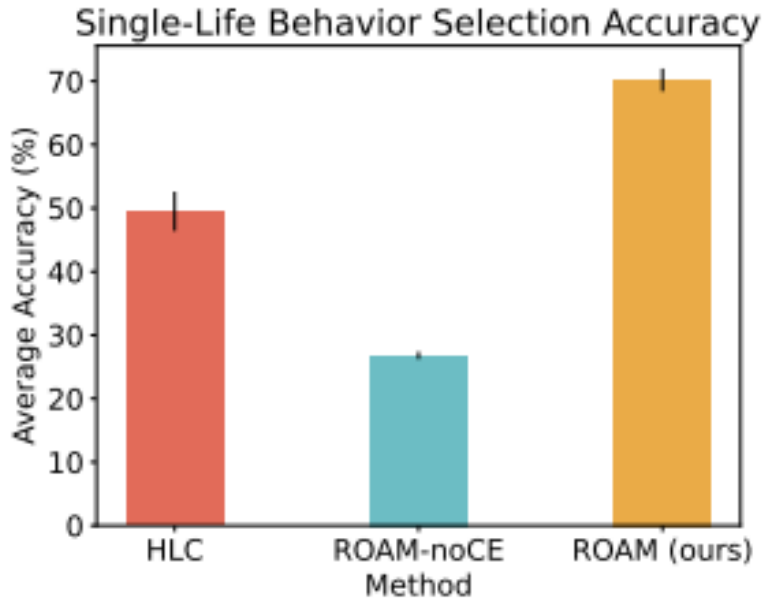


Figure 4: **Single-life Behavior Selection Accuracy.** The percent of steps where different methods select a relevant behavior for the current situation. ROAM is able to choose a relevant behavior significantly more often on average when adapting to test-time situations than HLC and ROAM-noCE.

		Before FT	After FT
Changing Stiffness	Avg ID Values	779.2	778.7
	Avg OOD Values	746.4	542.9
Changing Frictions	Avg ID Values	1243.4	1247.6
	Avg OOD Values	1217.6	1192.8

Figure 5: **Effect of the Cross-Entropy Loss on ID and OOD values.** The average values of the different behaviors in states visited by that behavior (ID) vs states visited by other behaviors (OOD), before and after fine-tuning with the additional cross-entropy term for ROAM. ROAM is able to maintain high values of the behavior in ID states, while decreasing the value of the behavior in OOD states.

03 **Real-World**: ROAM adapts to loads and roller skates on a real robot.

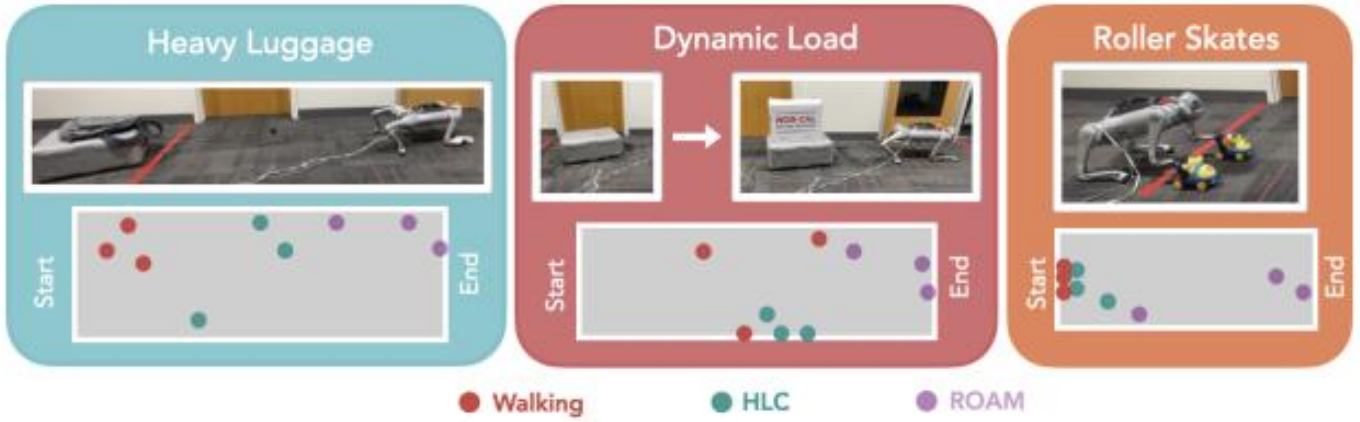


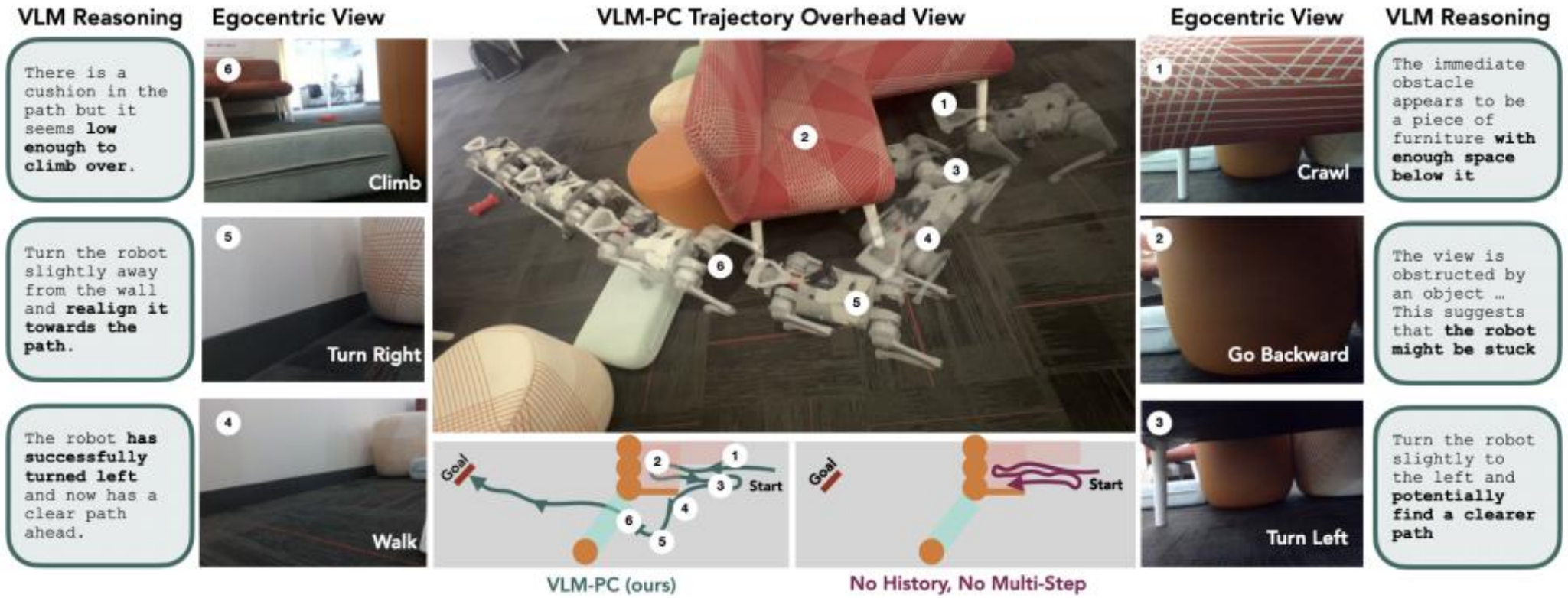
Figure 6: **Real-world single-life tasks.** We evaluate on: (1) pulling a load of heavy luggage 6.2 kg (13.6 lb), (2) pulling luggage where the weight dynamically changes between 2.36 kg (5.2 lb) and 4.2 kg (9.2 lb), and (3) moving forward with roller skates on the robot's front two feet. For each trial, for each method, we also show the locations of the robot before its first fall or readjustment (Red is Walking, Blue is HLC, and Purple is ROAM).

	Heavy Luggage		Dynamic Luggage Load		Roller Skates	
	Avg. Time (s) ↓	Falls ↓	Avg. Time (s) ↓	Falls ↓	Avg. Time (s) ↓	Falls ↓
Walking	45.3	2.3	32	1	NC	NC
HLC	42.7	3	28.3	1.3	62.3	2.7
ROAM (ours)	**25.7**	**0.7**	**24.3**	**0.3**	**27.3**	**1**

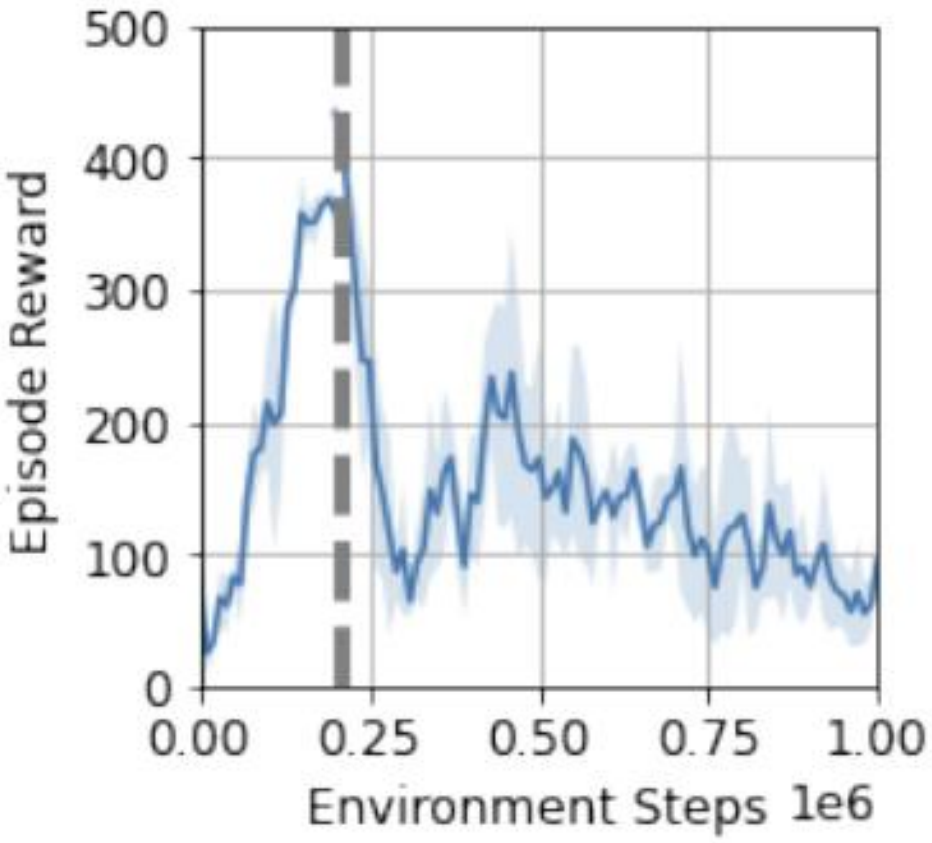
Table 1: **Results on the real Go1 robot on 3 different tasks**: On all 3 tasks, across 3 trials for each method, ROAM significantly outperforms both comparisons in terms of both average wall clock time (s) and number of falls or readjustments needed to complete the task in a single life. NC (no complete) indicates that the task was not able to be successfully completed with the given method.

Beyond adaptation

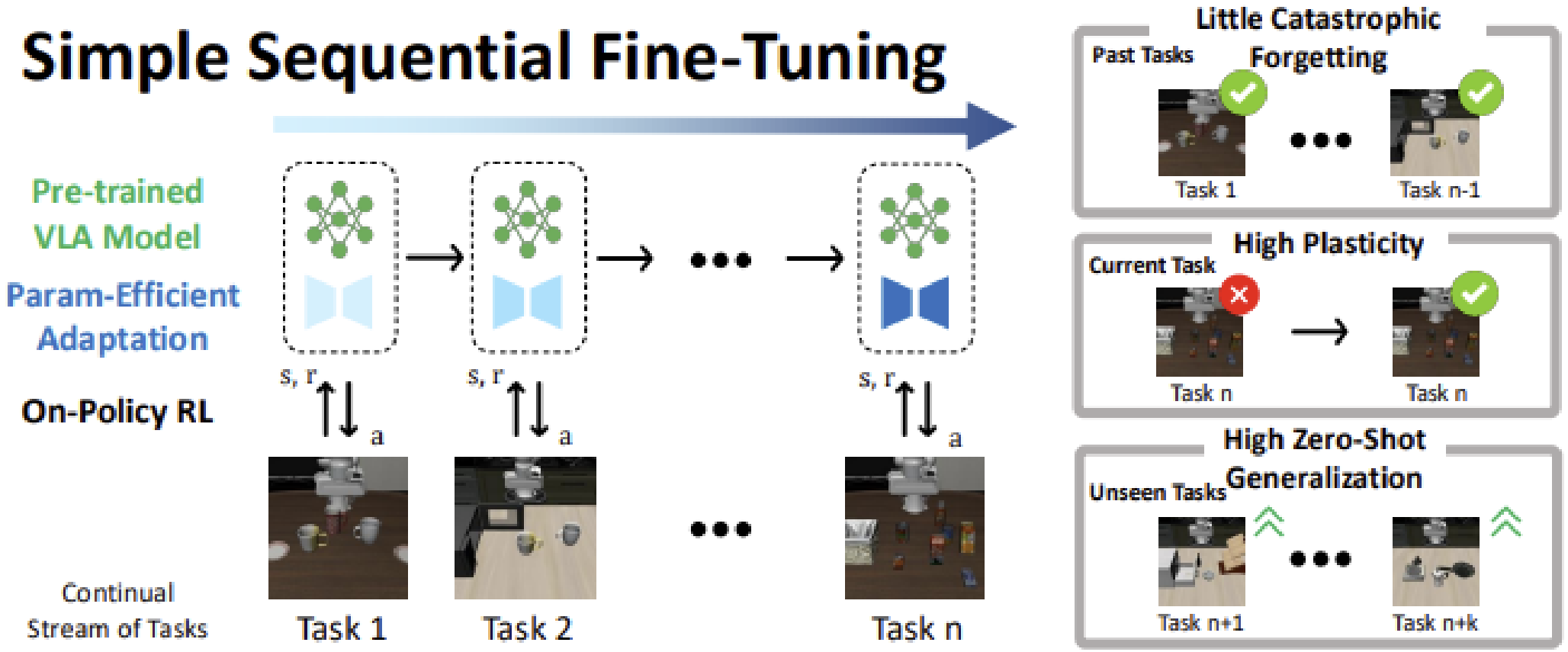
04 **Reasoning** : From value-based selection to semantic replanning



04 **Forgetting** : Fast adaptation does not guarantee long-term retention

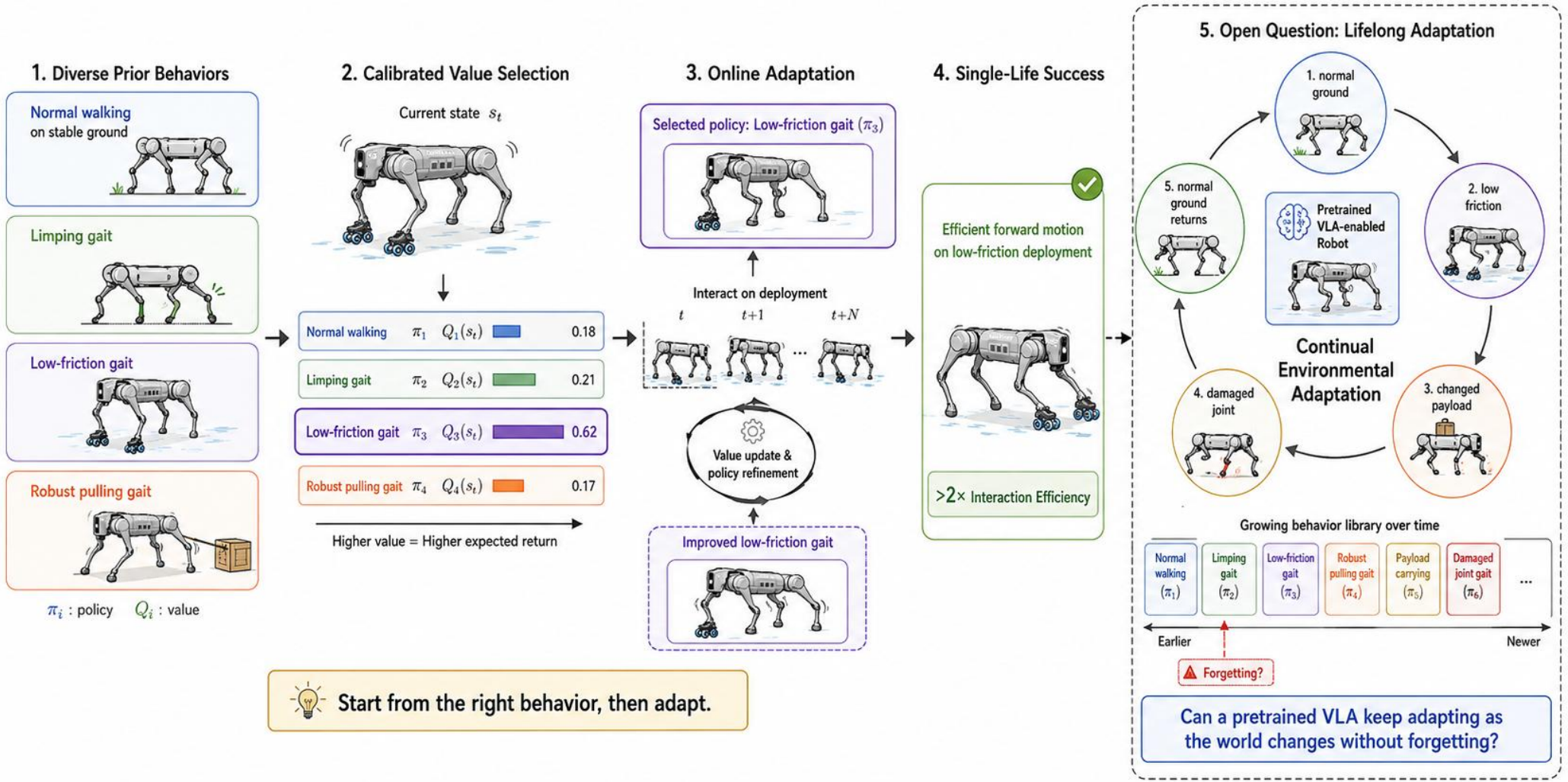


04 Continual VLA : Large pretrained VLAs may forget less than expected



Conclusion

05 **Conclusion** : ROAM adapts on the go – but lifelong adaptation remains open



Thank You

2026.08.03

MuliLAB Journal Club
인공지능응용학과 허세민