NEURAL INFORMATION PROCESSING SYSTEMS

What can RL bring to VLA generalization?

An Empirical Study

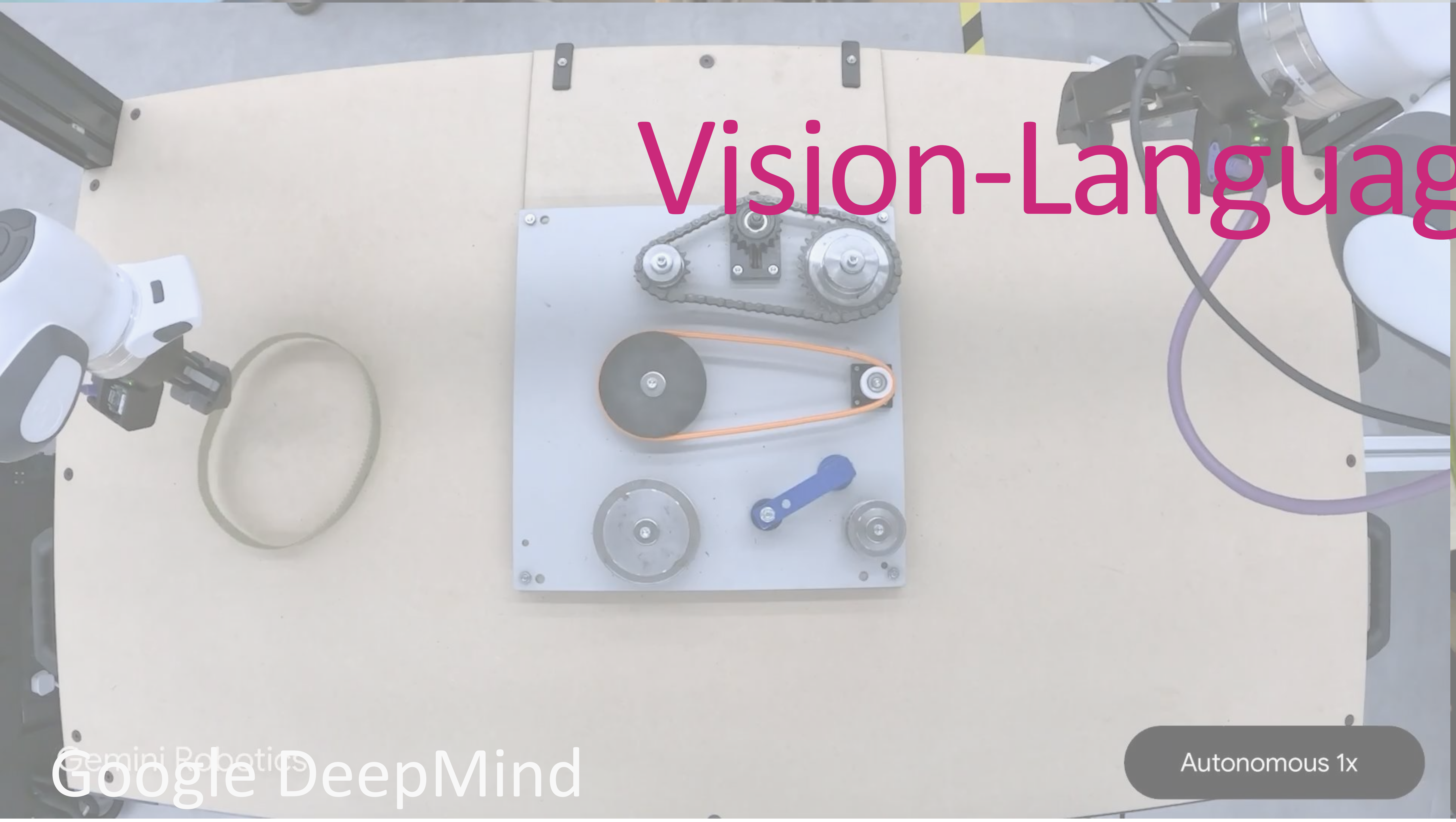
Jijia Liu*, Feng Gao*, Bingwen Wei,
Xinlei Chen, Qingmin Liao, Yi Wu, Chao Yu+, Yu Wang+

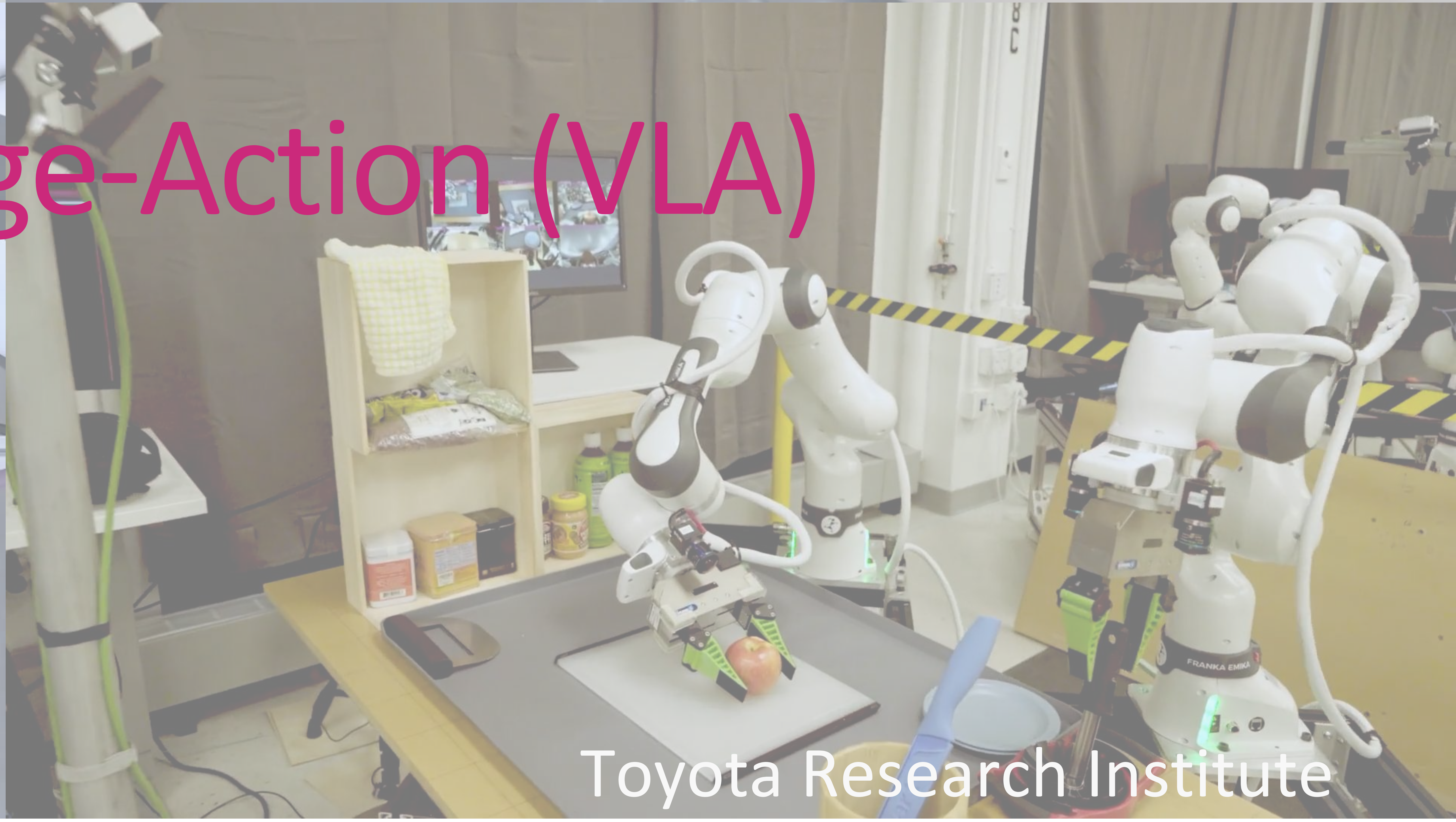
* Equal contribution   + Corresponding Authers

清華大學 Tsinghua University   上海期智研究院 SHANGHAI QI ZHI INSTITUTE   北京中关村学院 Zhongguancun Academy

Now dawns the era of generalist models

Vision-Language-Action (VLA)





Specialist → **Generalist** → Specialized Generalist

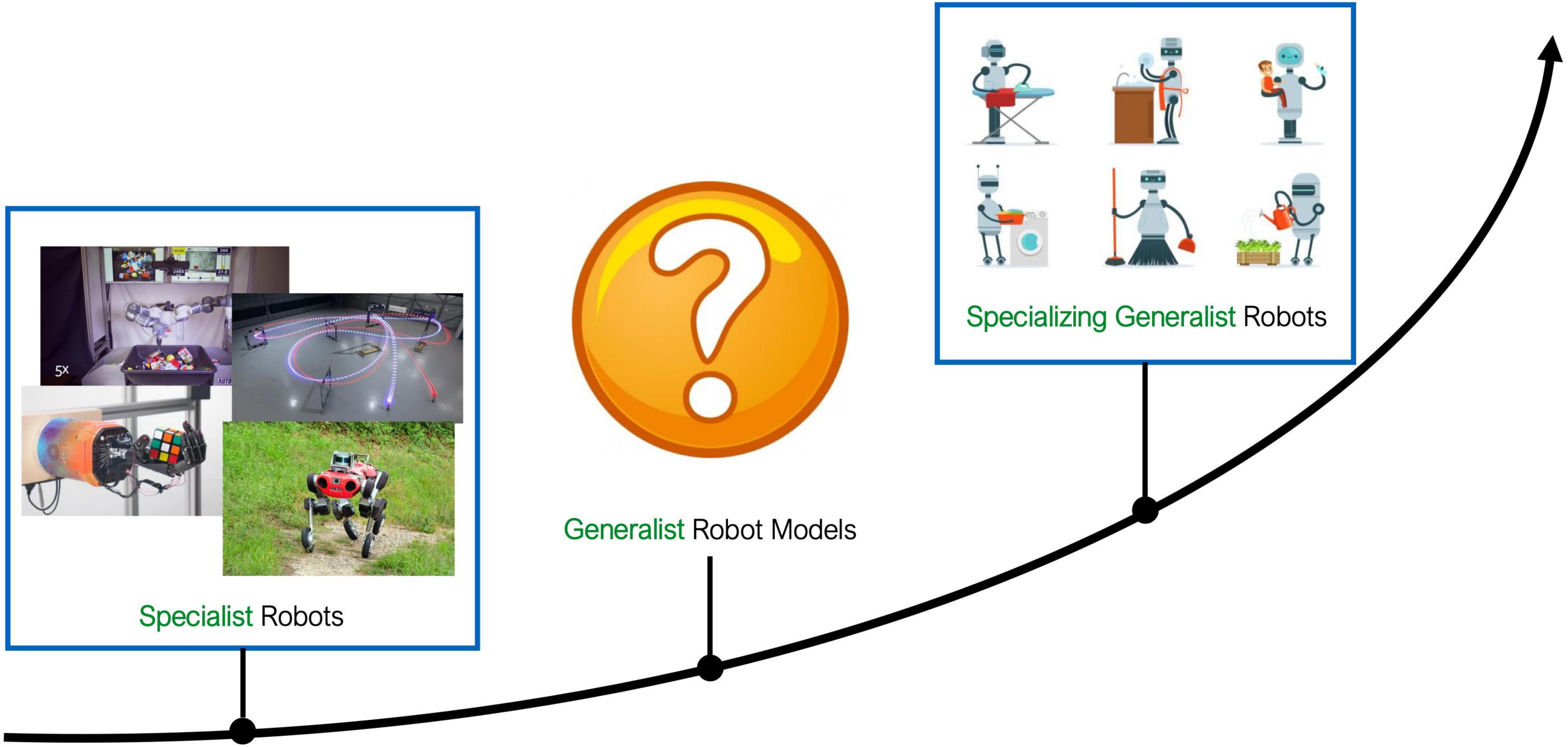
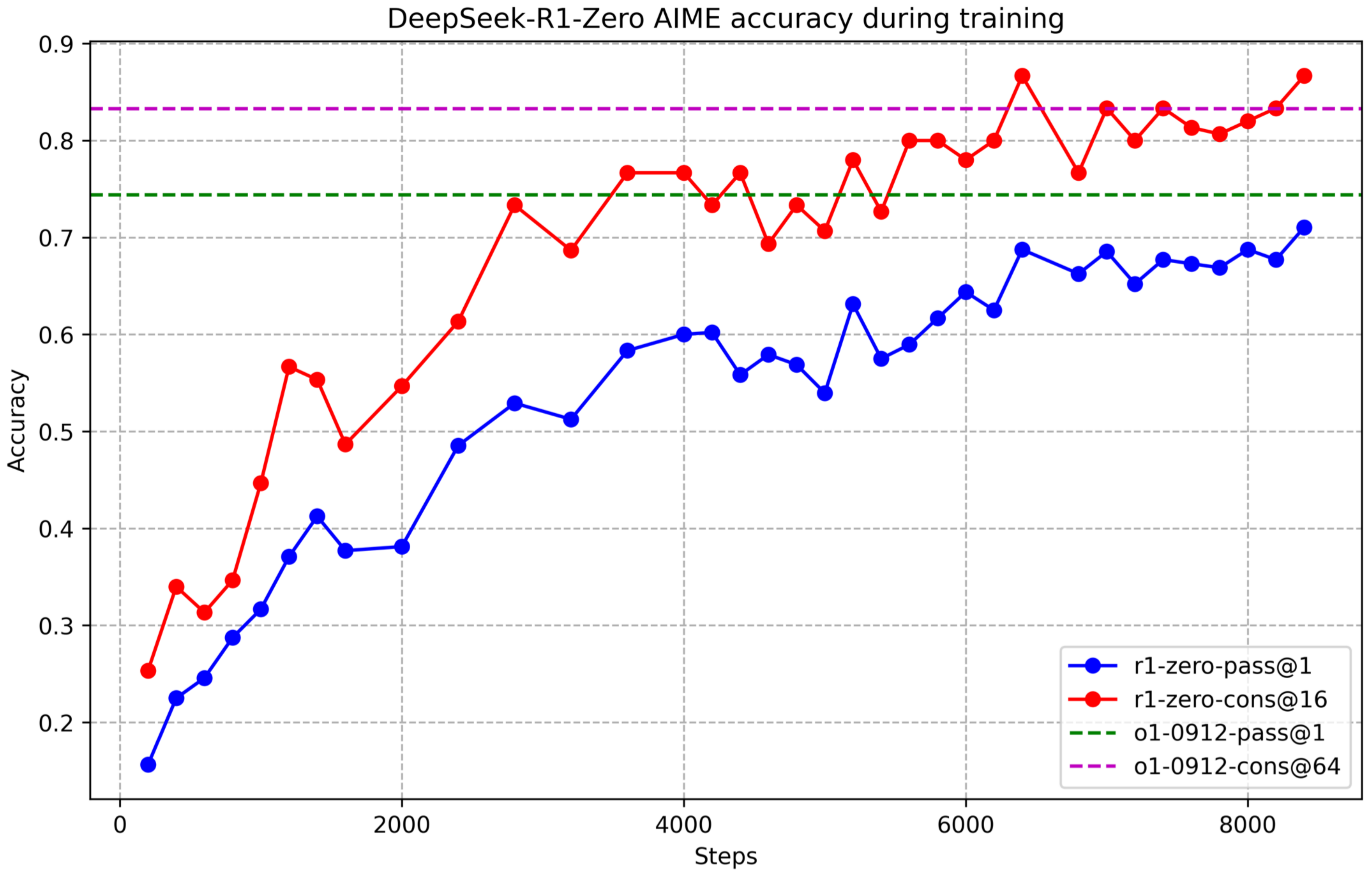
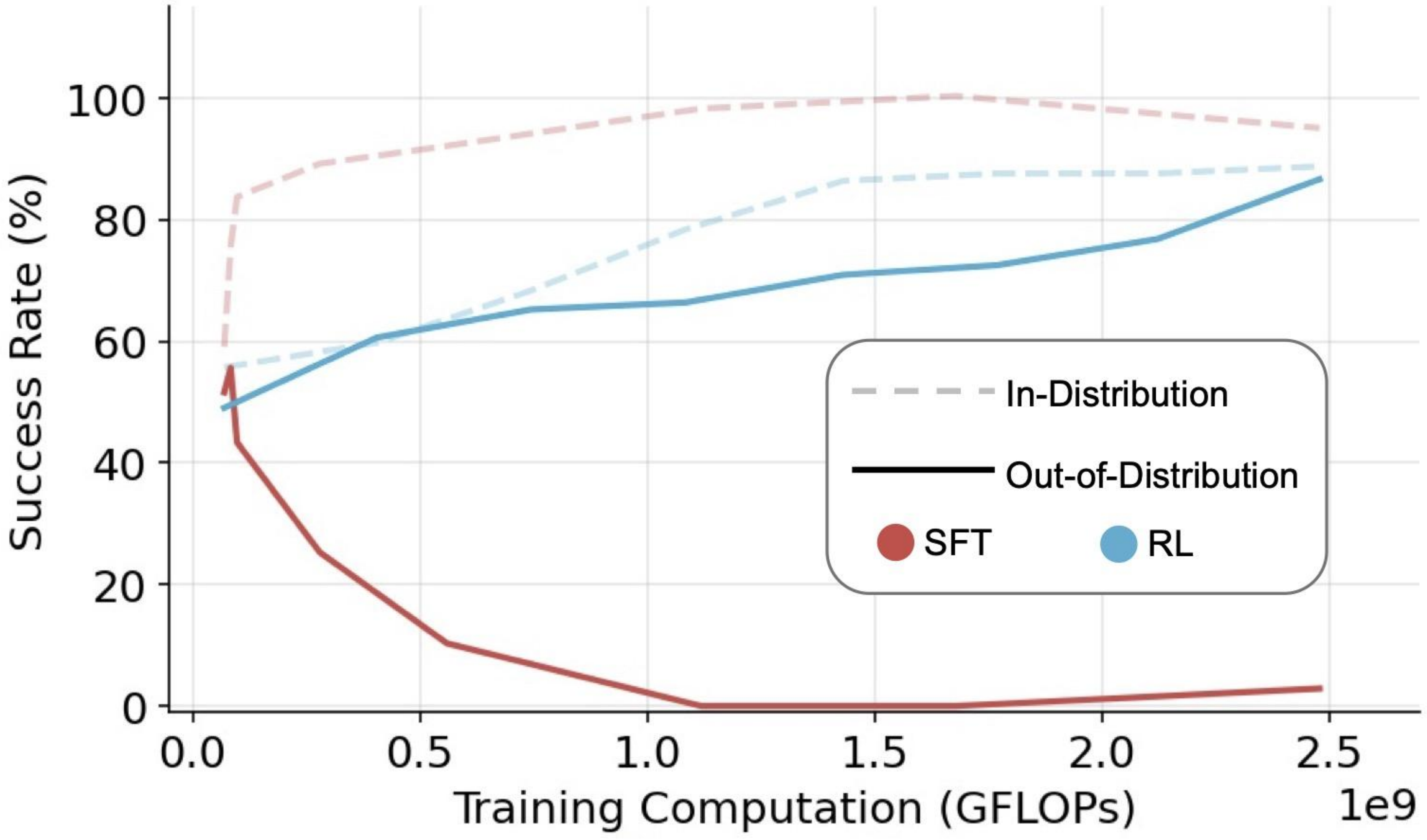


Image source: Prof. Yuke Zhu, "Pathway to Generalist Robots: Scaling Law, Data Flywheel, and Humanlike Embodiment."

RL is helpful for LLMs and VLMs

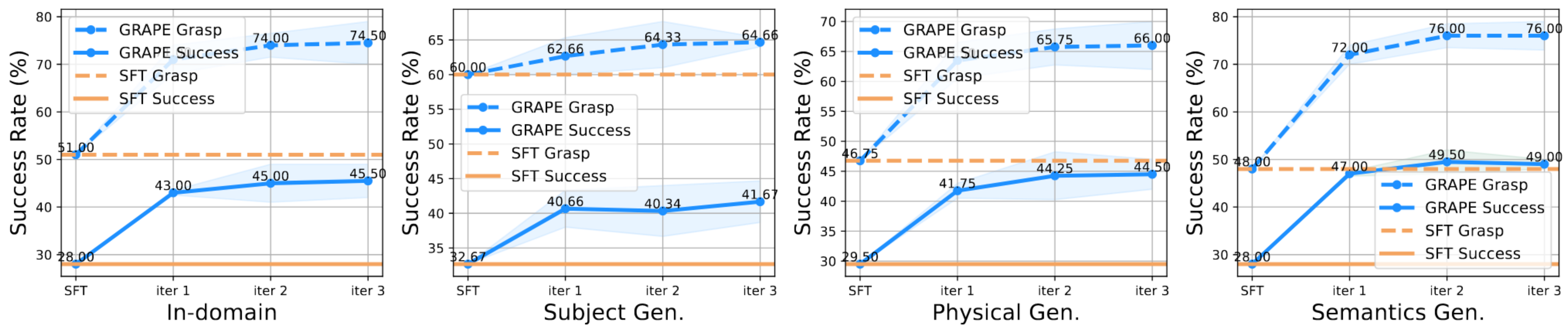




RL can help long reasoning and OOD generalization

[DeepSeek R1] [Chu et al. SFT Memorizes, RL Generalizes]

RL can also improve VLA models

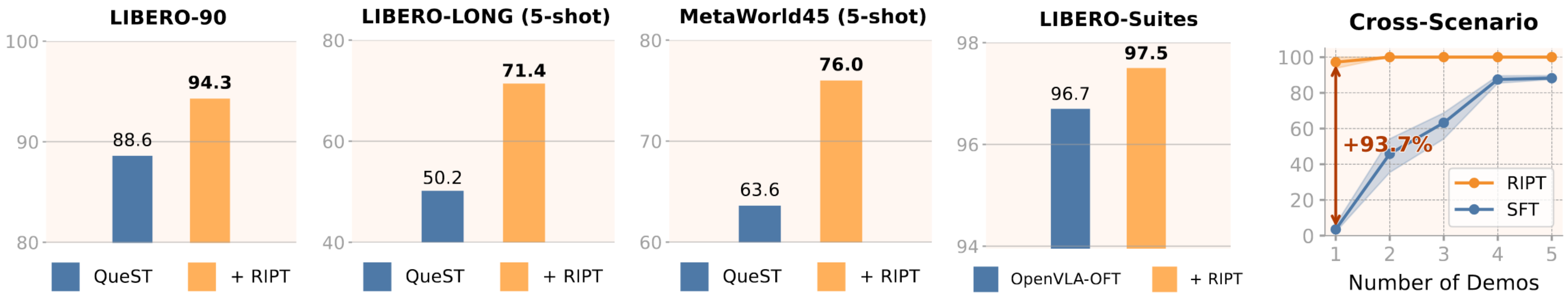


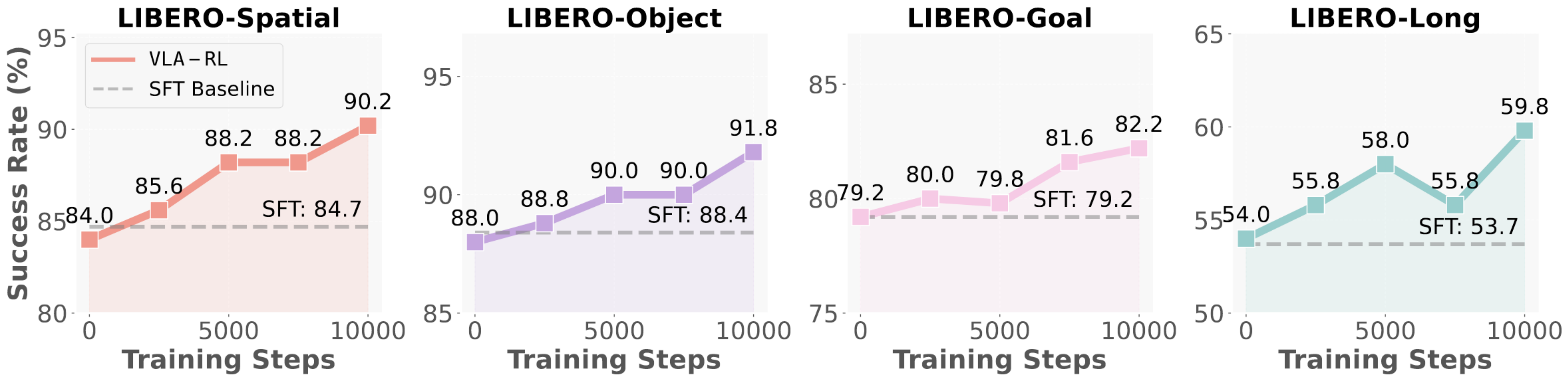
[Zhang et al. GRAPE]

arXiv:2411.19309

[Tan et al. RIPT-VLA]

arXiv:2505.17016



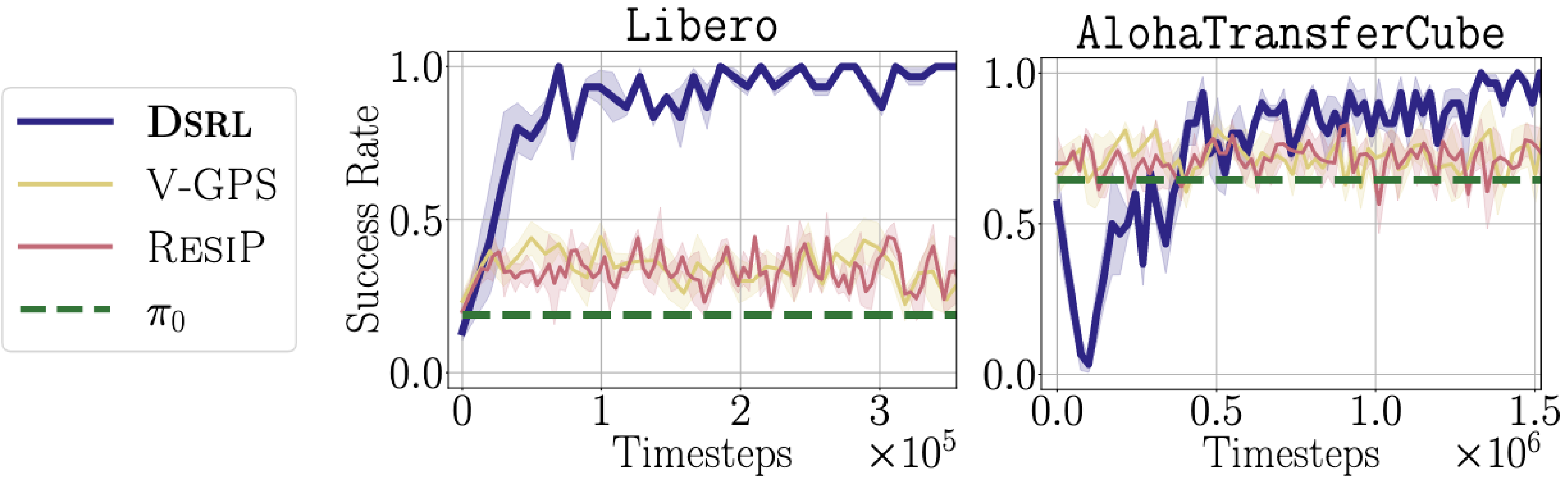


[Lu et al. VLA-RL]

arXiv:2505.18719

[Wagenmaker et al. DSRL]

arXiv:2506.15799



Task	π_0	DSRL
Turn on toaster	5/20	**18/20**
Put spoon on plate	15/20	**19/20**

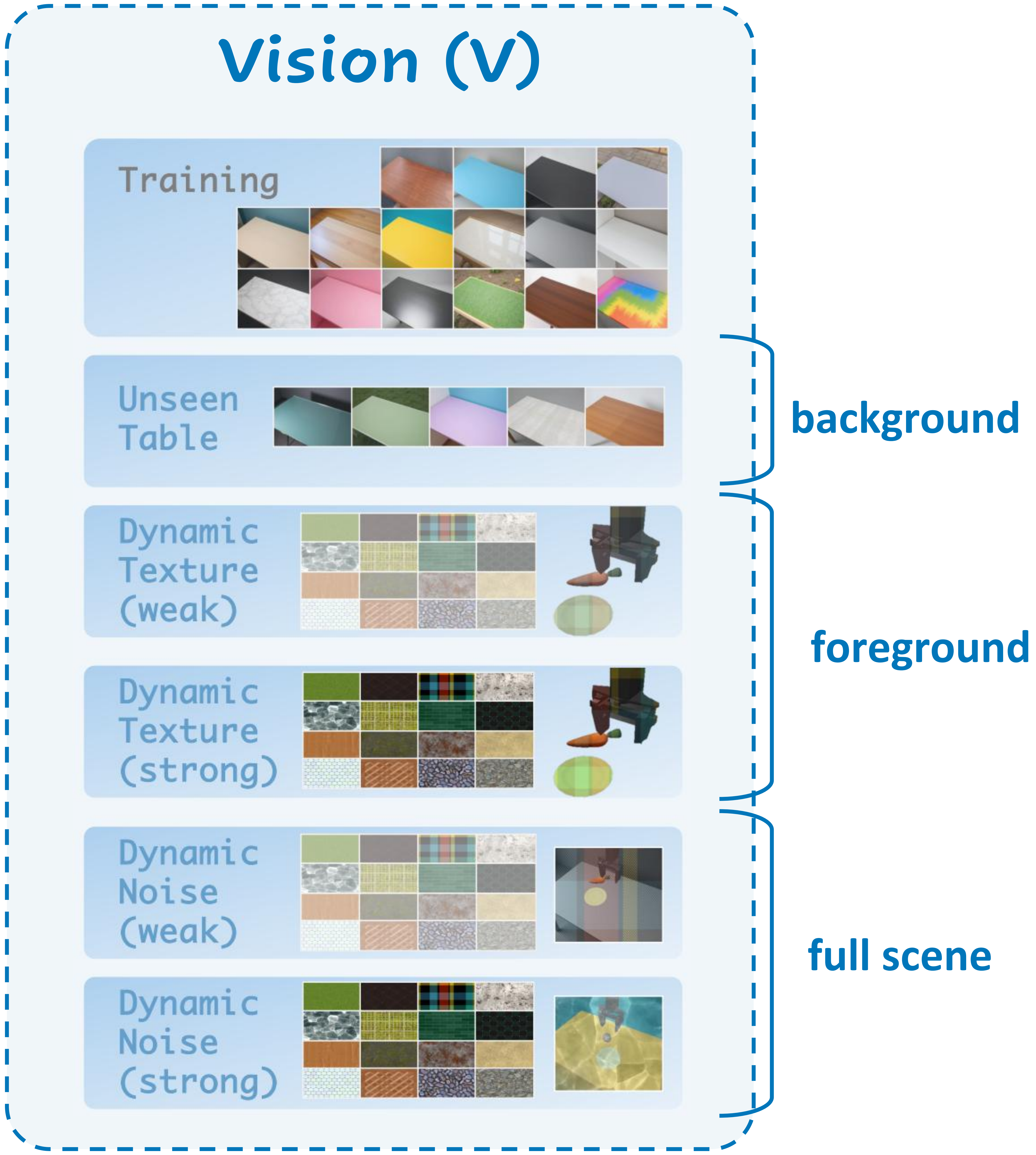
But which facets of VLA stand to gain the most from RL?

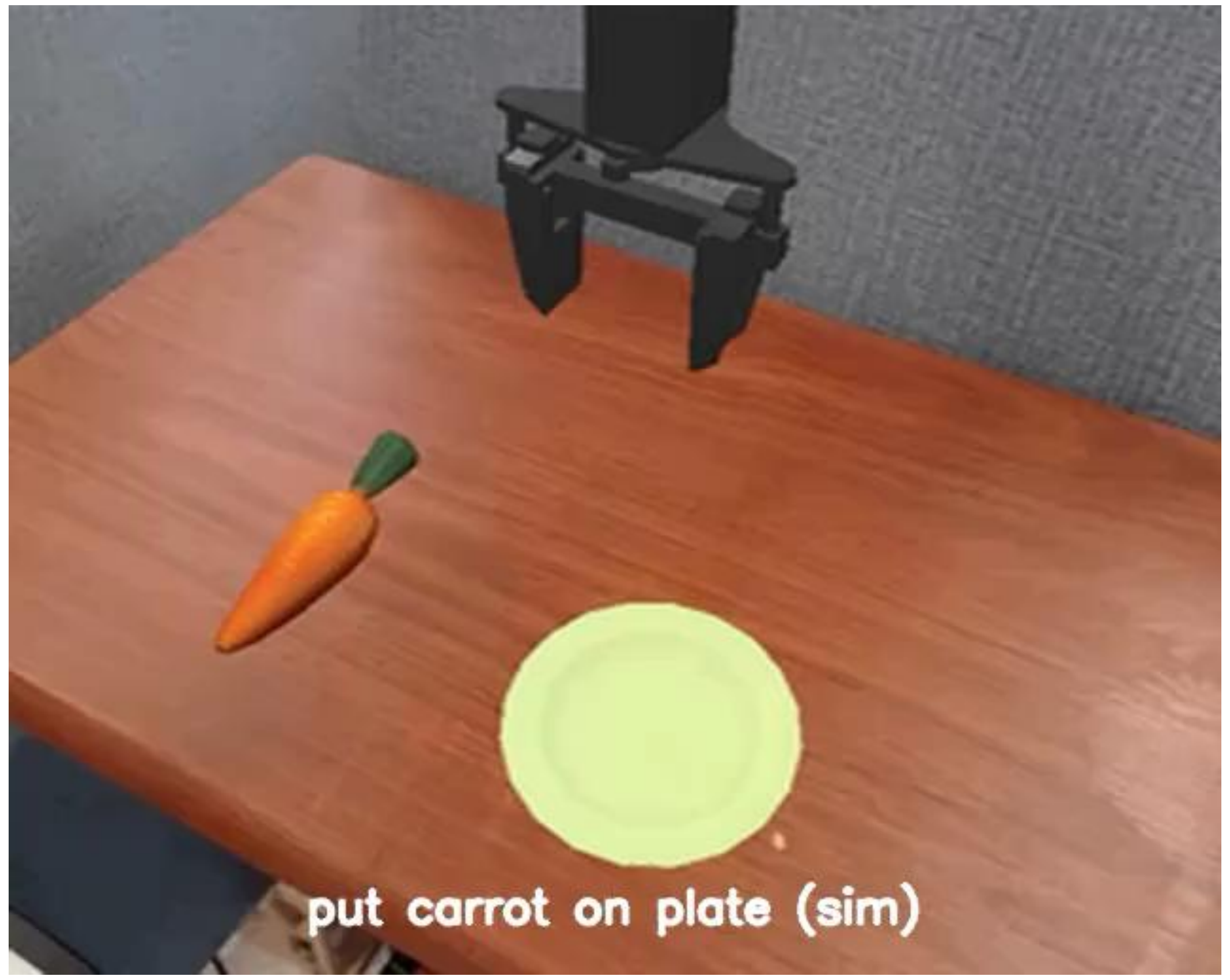
Vision (V)

Semantics (L)

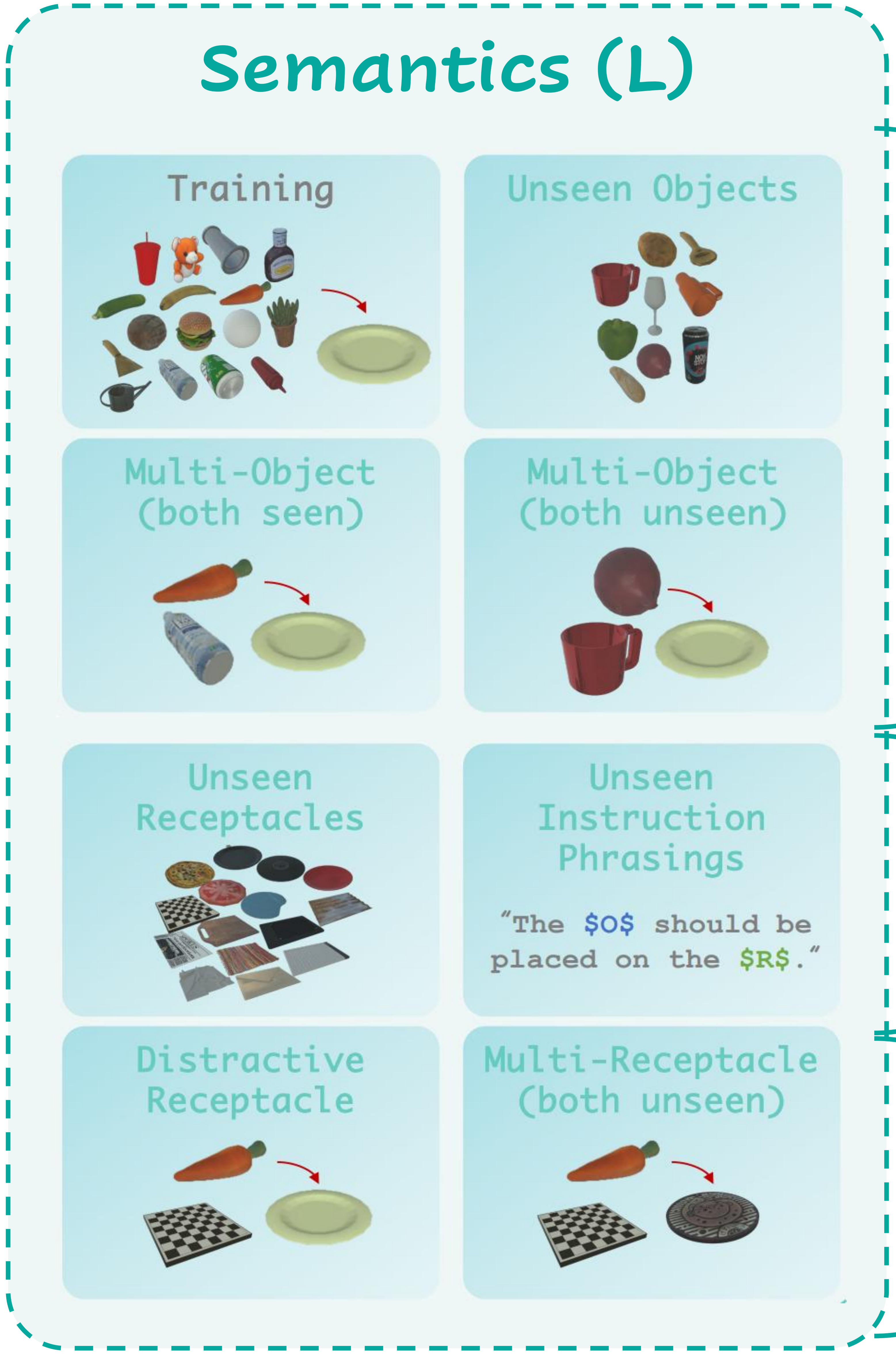
Execution (A)

Facets of VLA generalization - Vision





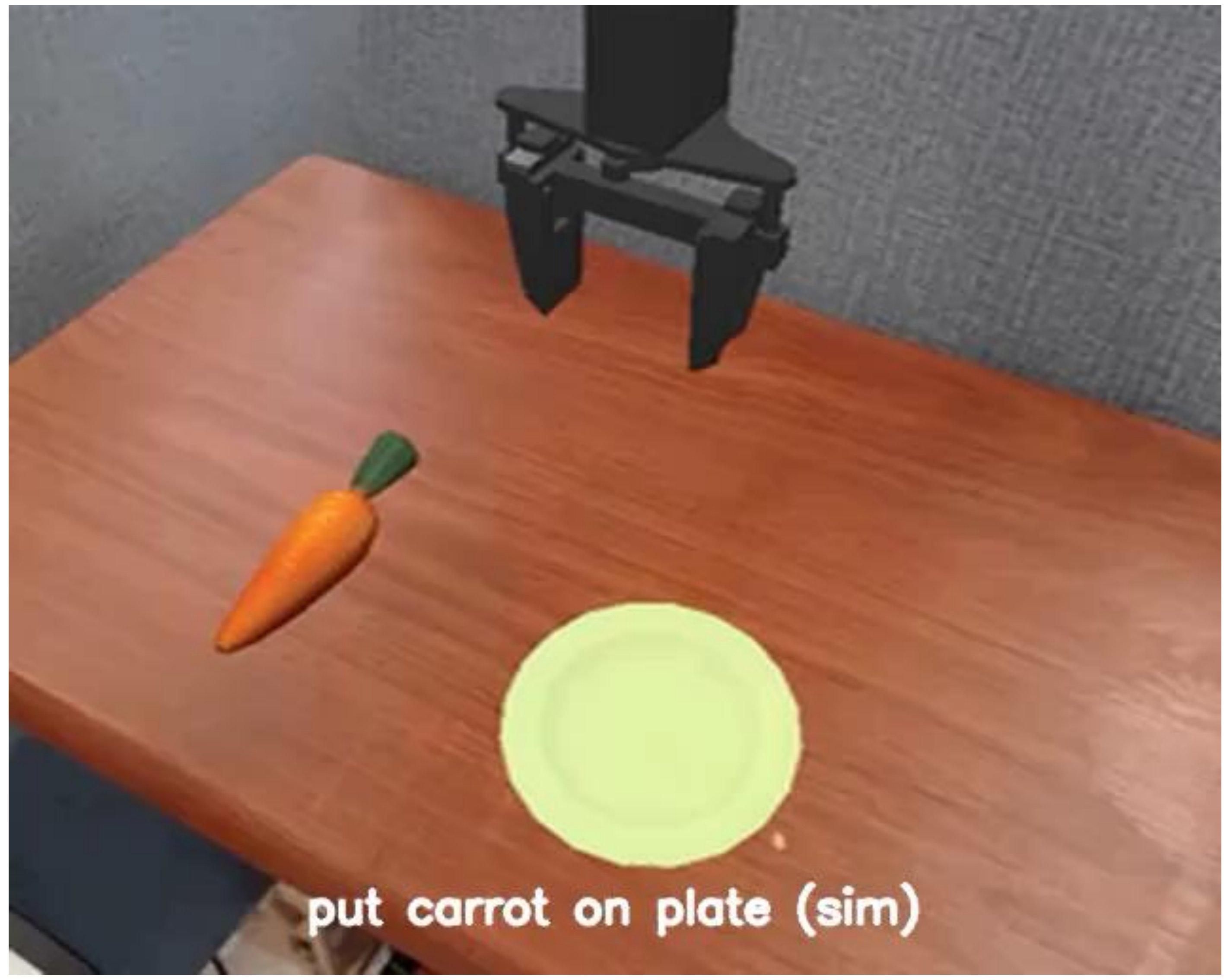
Facets of VLA generalization - Semantics



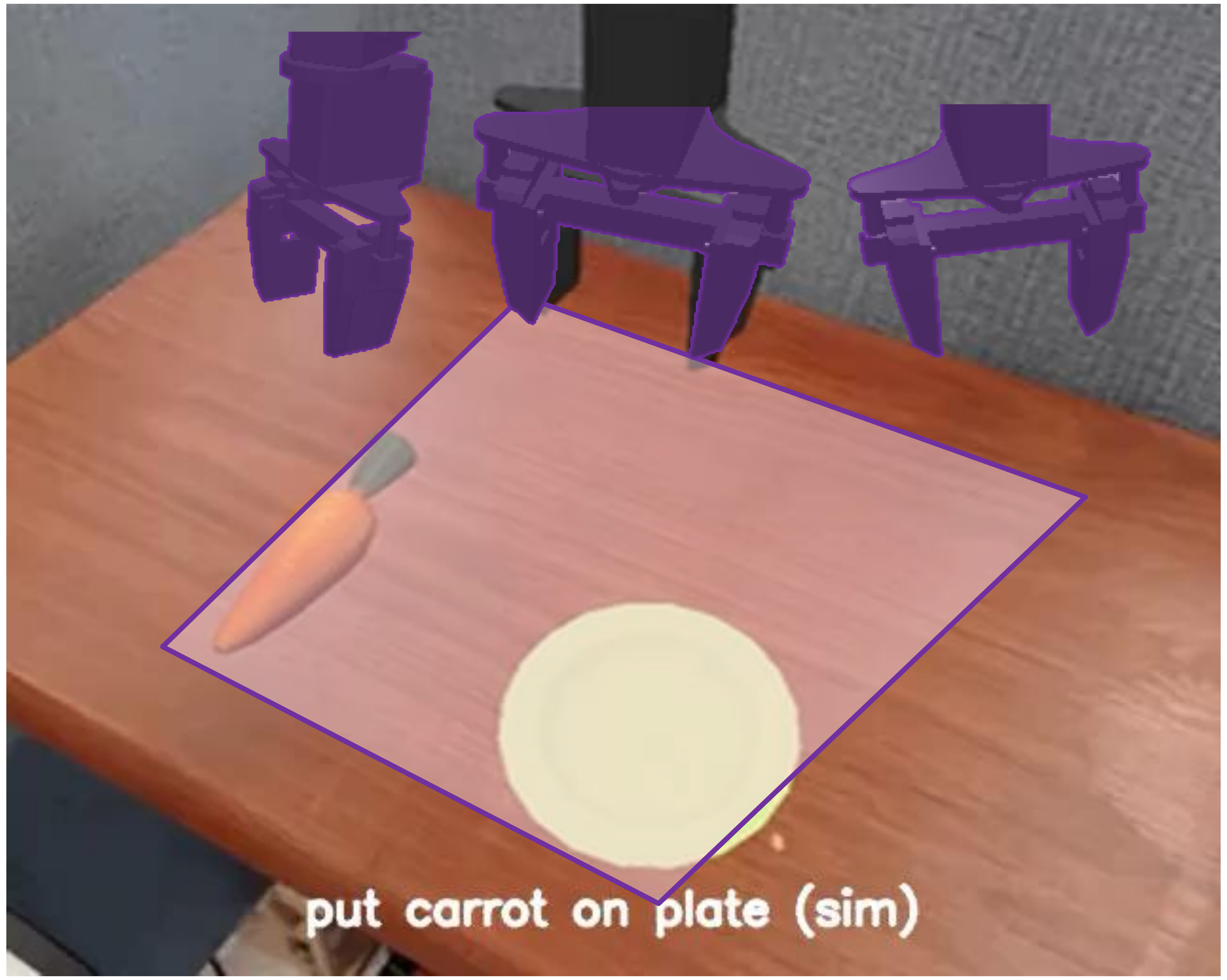
one object, one receptacle

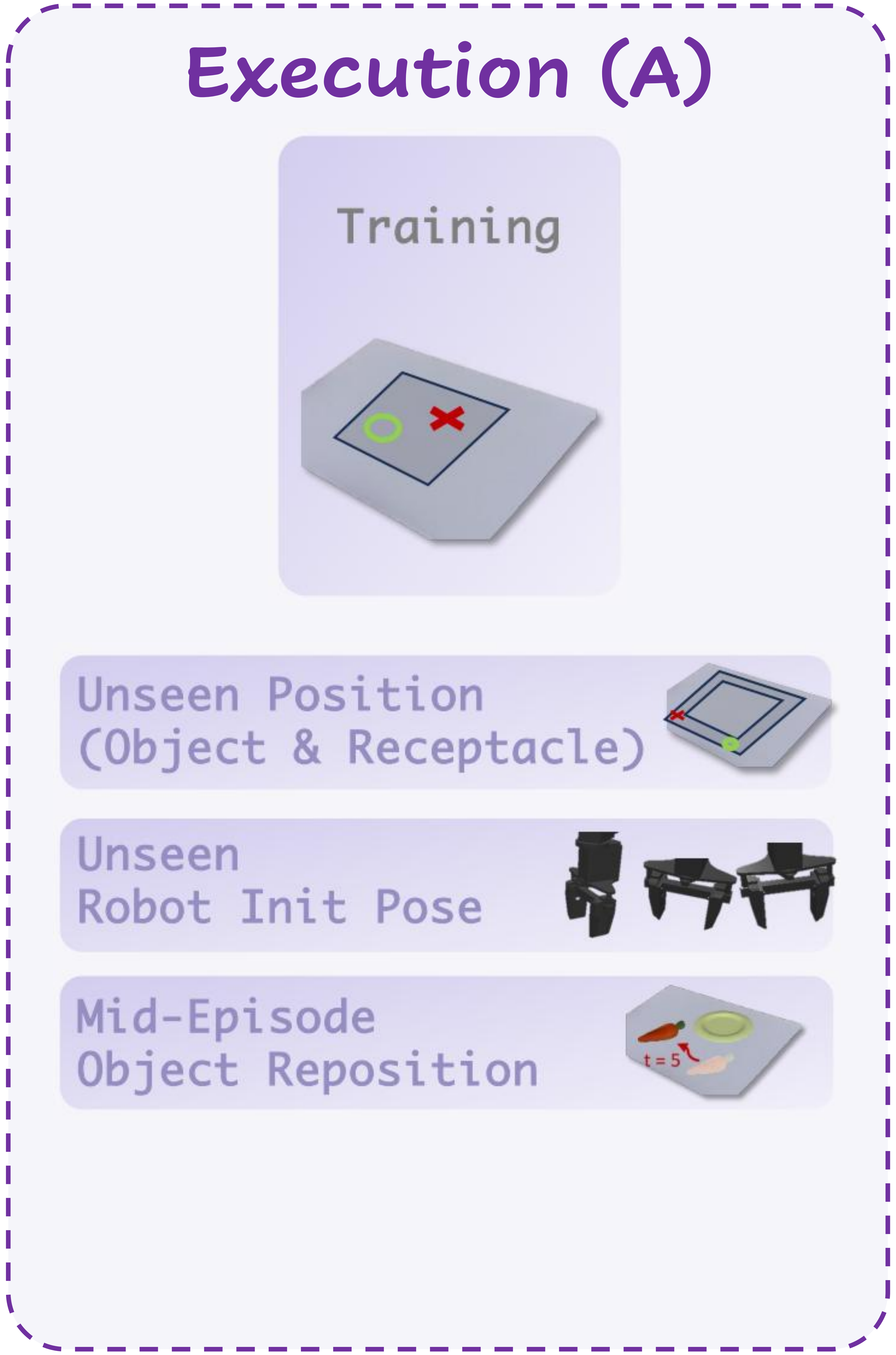
two objects, one receptacle

one object, two receptacles

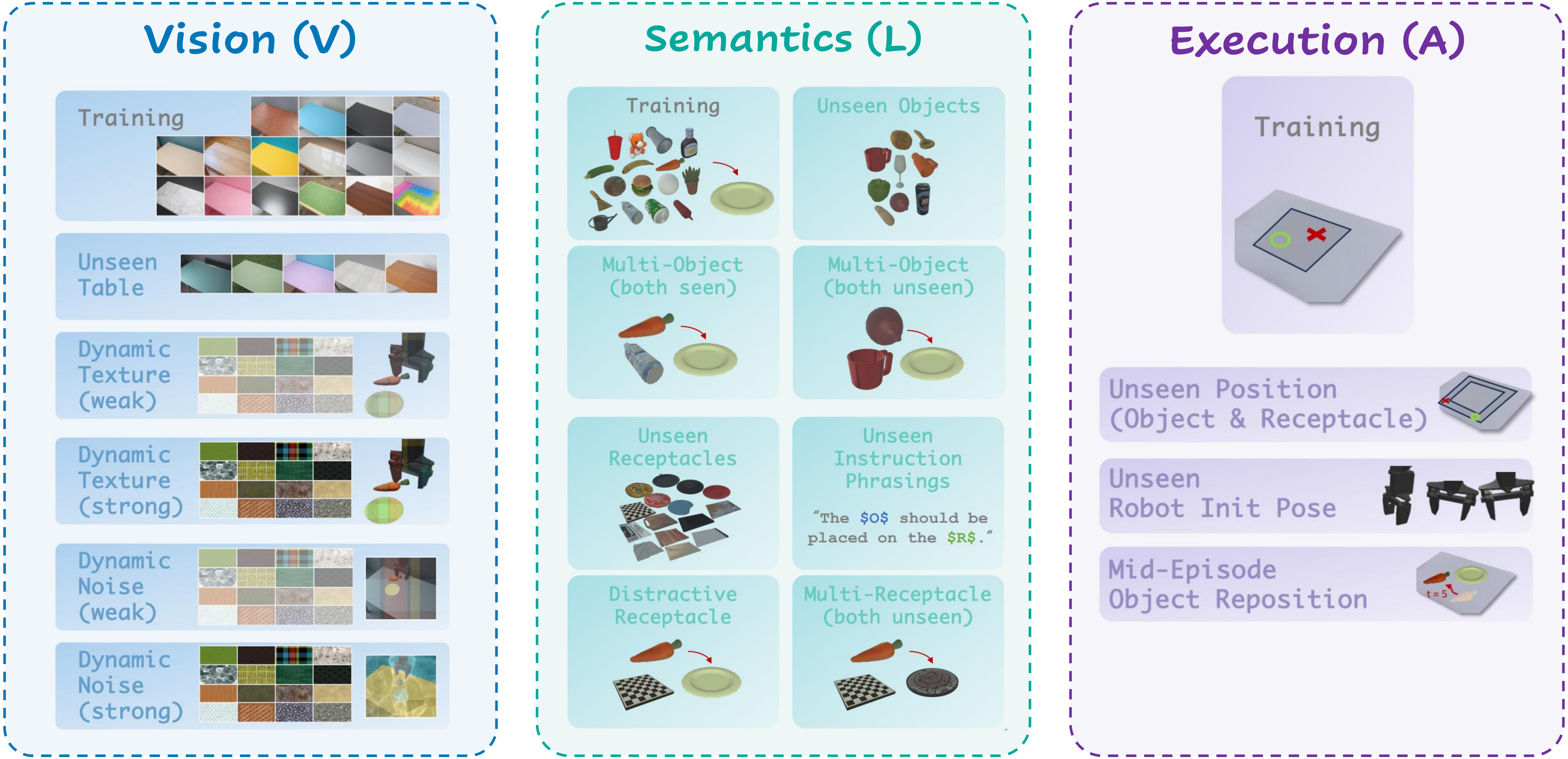


Facets of VLA generalization - Execution

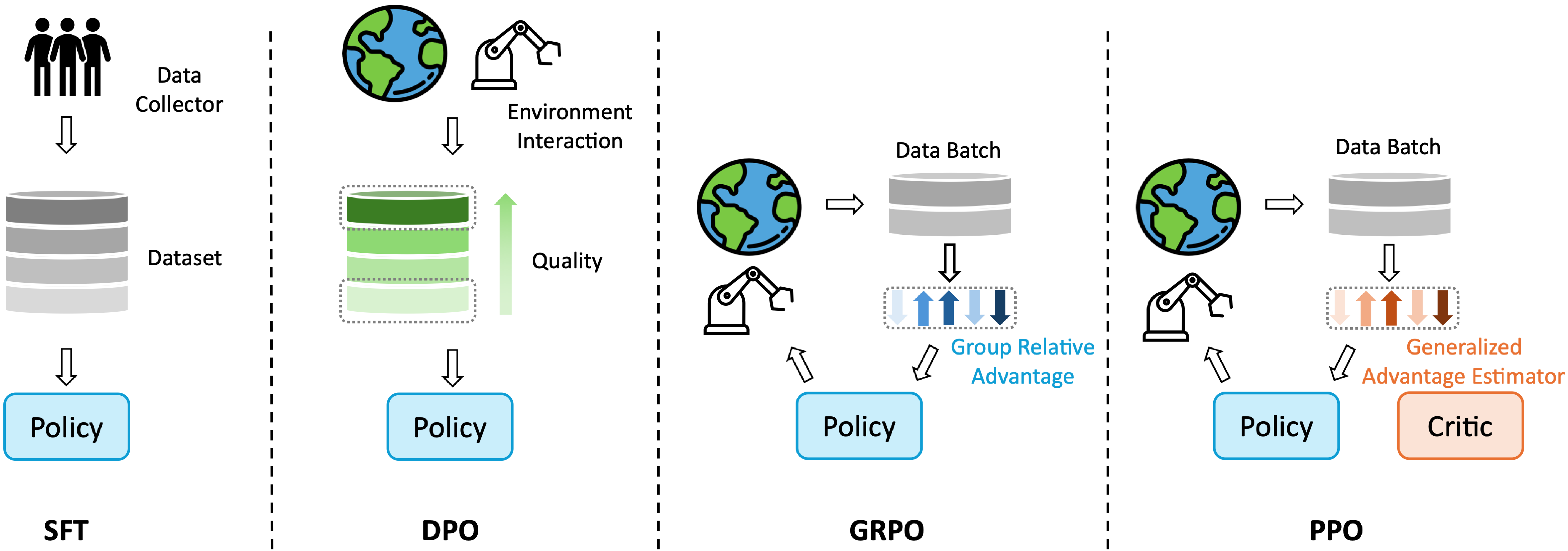




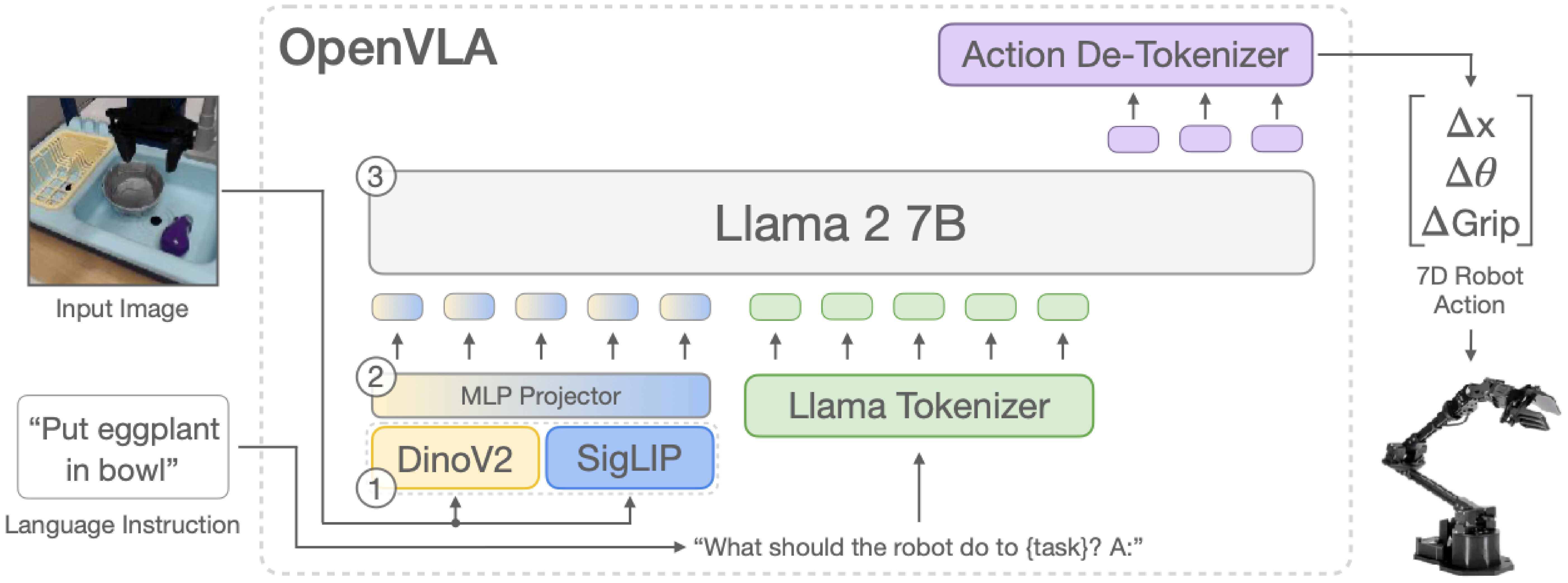
Overview of our study

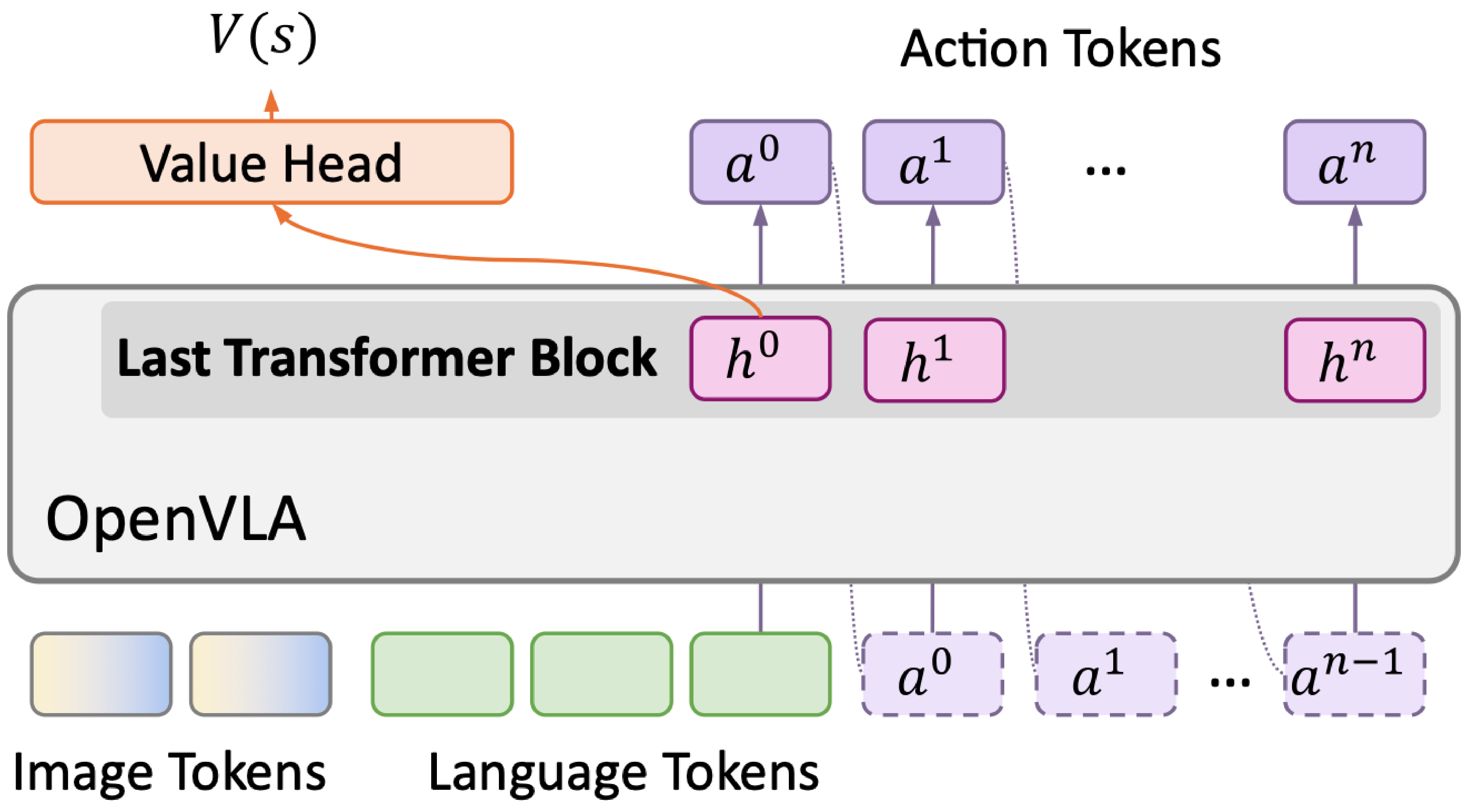


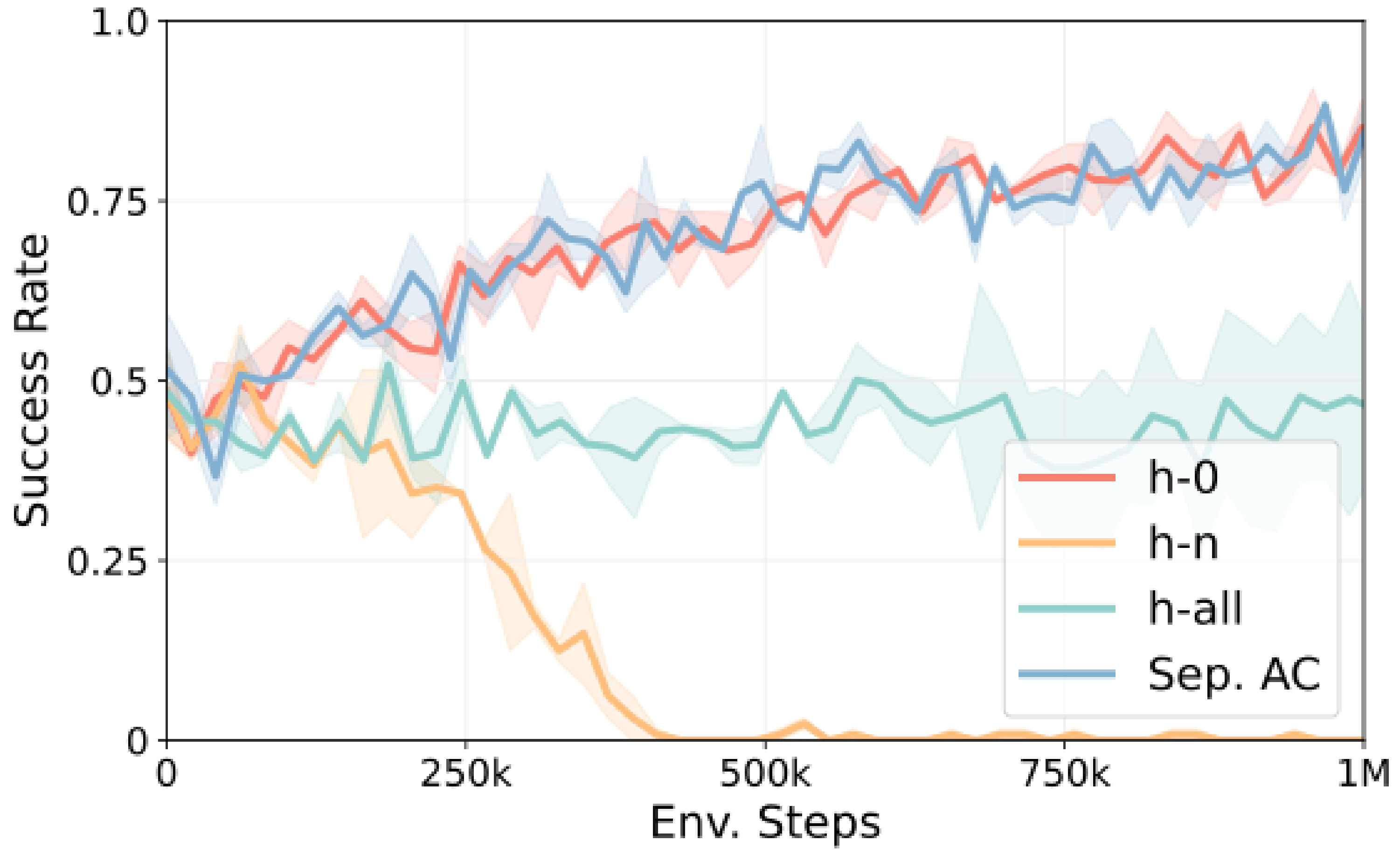
SFT/RL finetuning for VLA



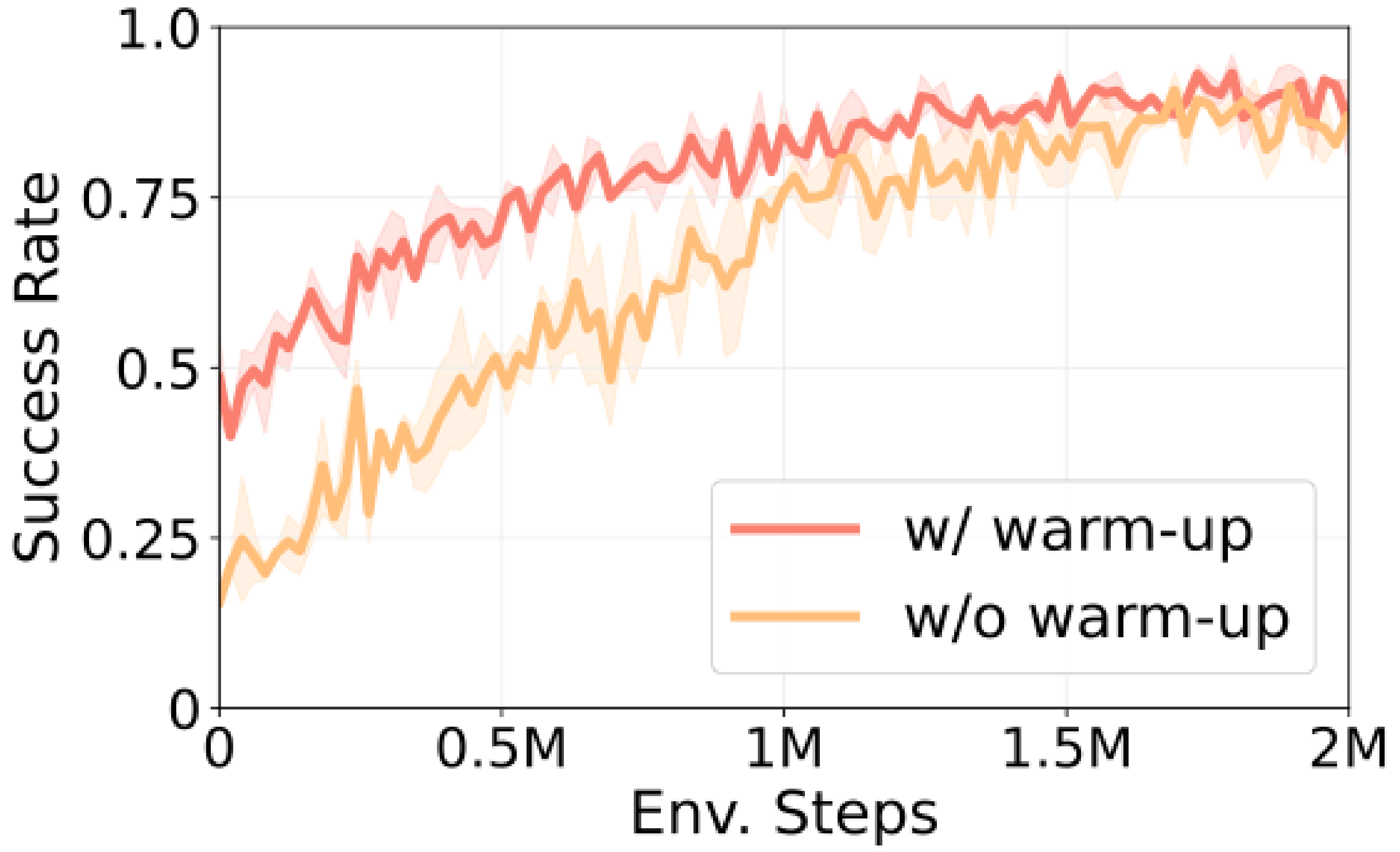
Effective RL fine-tuning



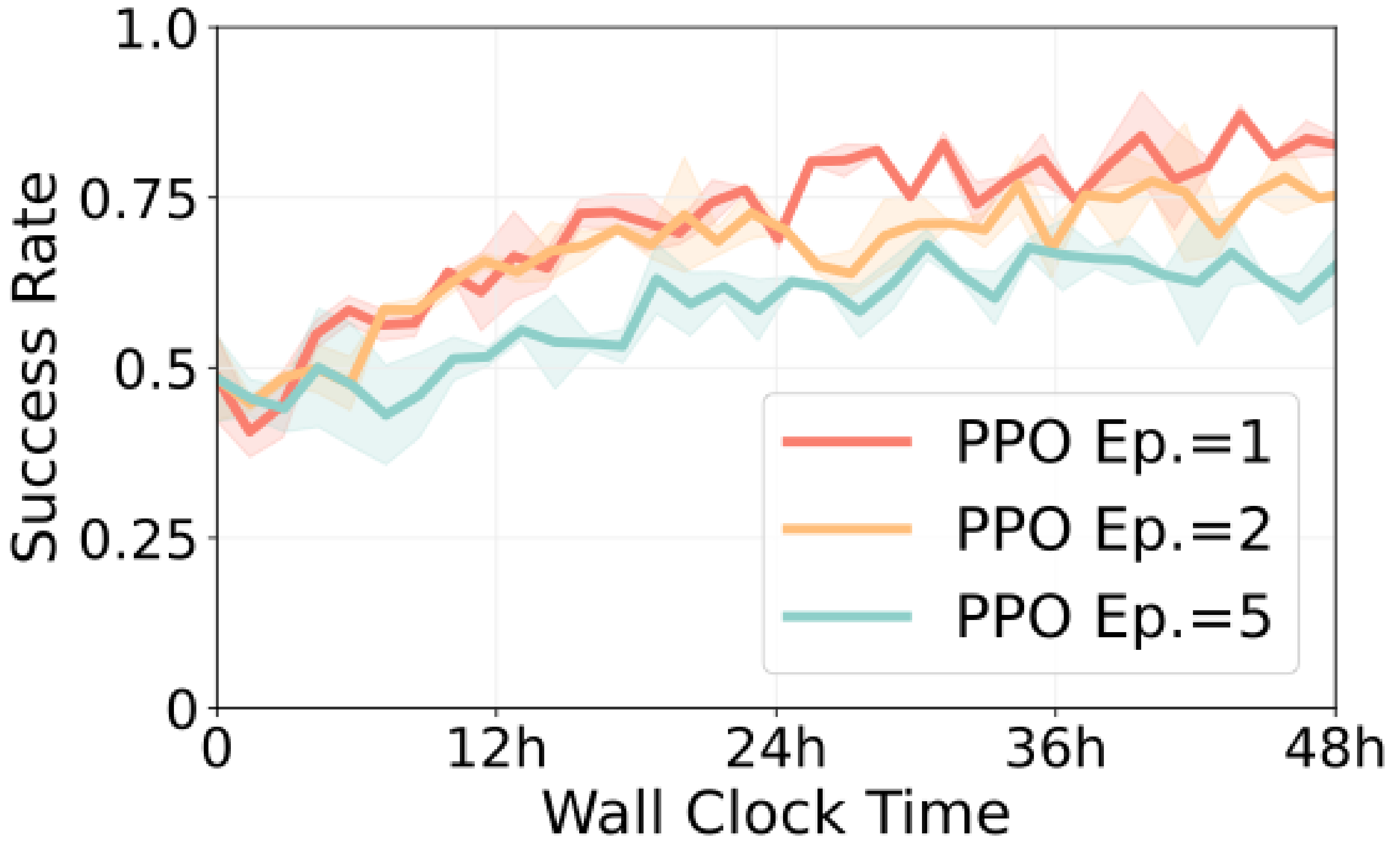




(1) Shared actor-critic backbone

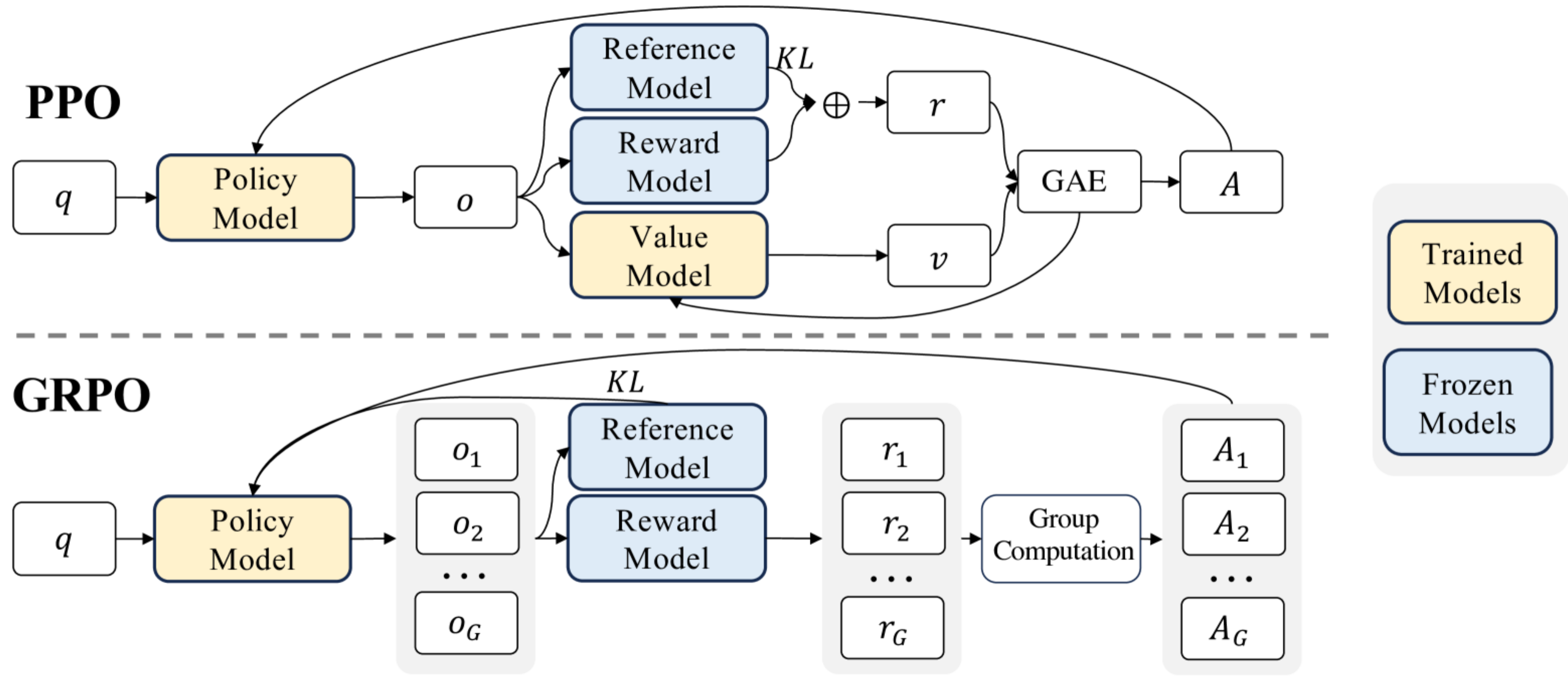


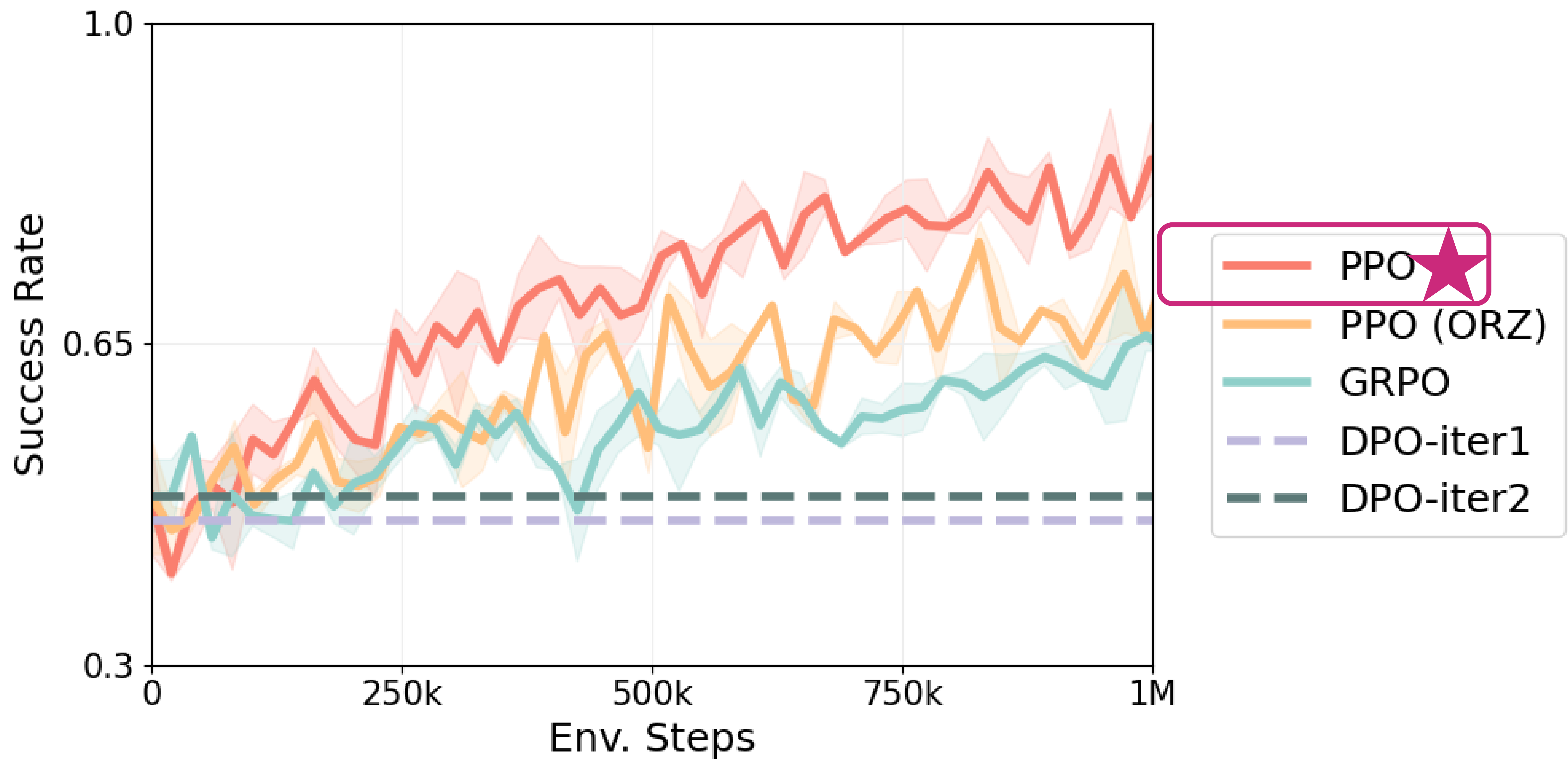
(2) Warm-up with demos



(3) Minimal PPO epoch

PPO works best





PPO (ORZ): disabling GAE ($\gamma = 1, \lambda = 1$)

GRPO:

$$\hat{A}_t^i = \frac{r^i - \text{mean}(\mathbf{r})}{\text{std}(\mathbf{r})}$$

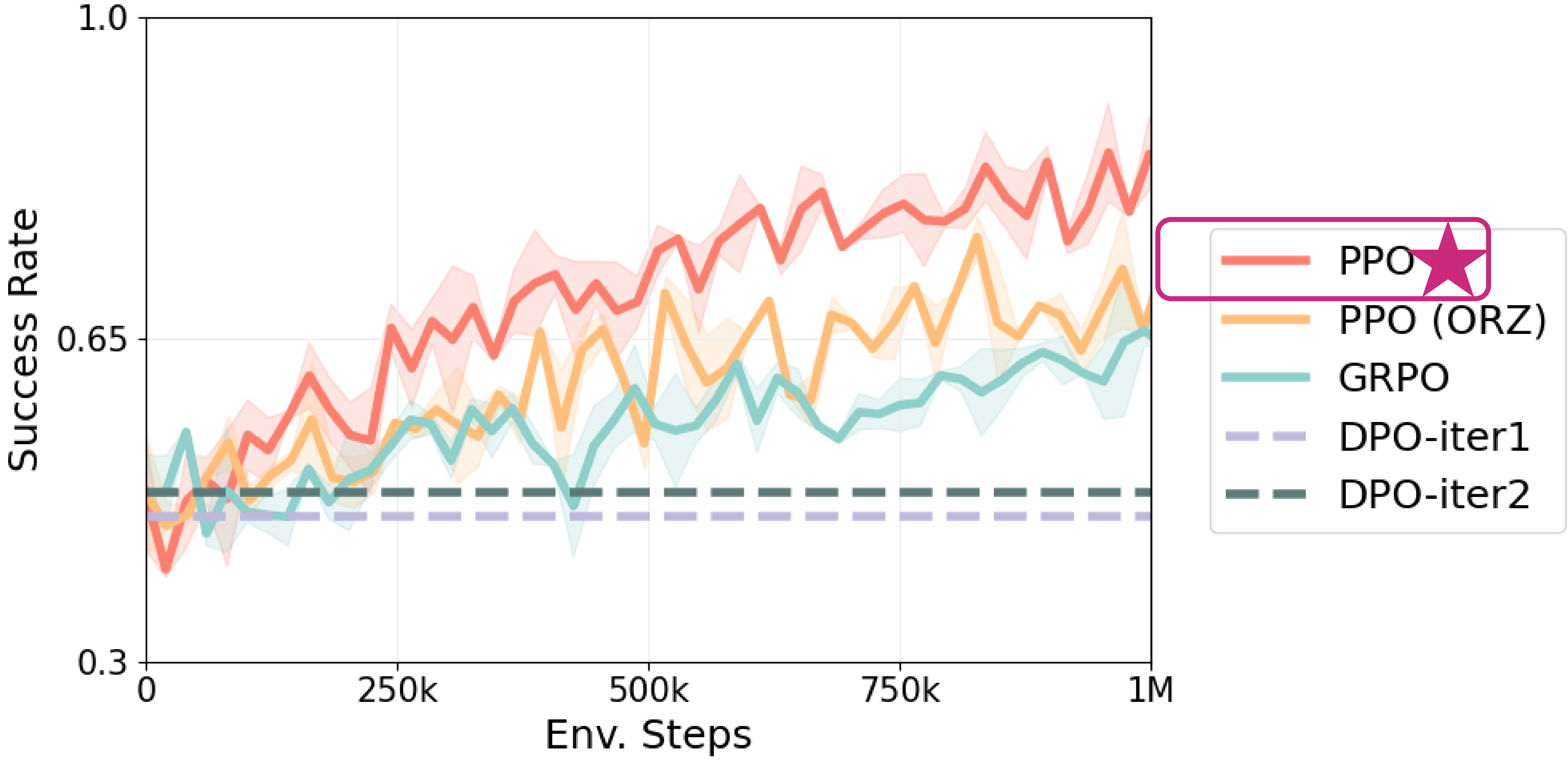
$$\mathbf{r} = \{r^1, r^2, \cdots, r^G\}$$

DPO:

$$\zeta_w, \zeta_l \sim \{s_0^i = \boldsymbol{s_0}\}$$

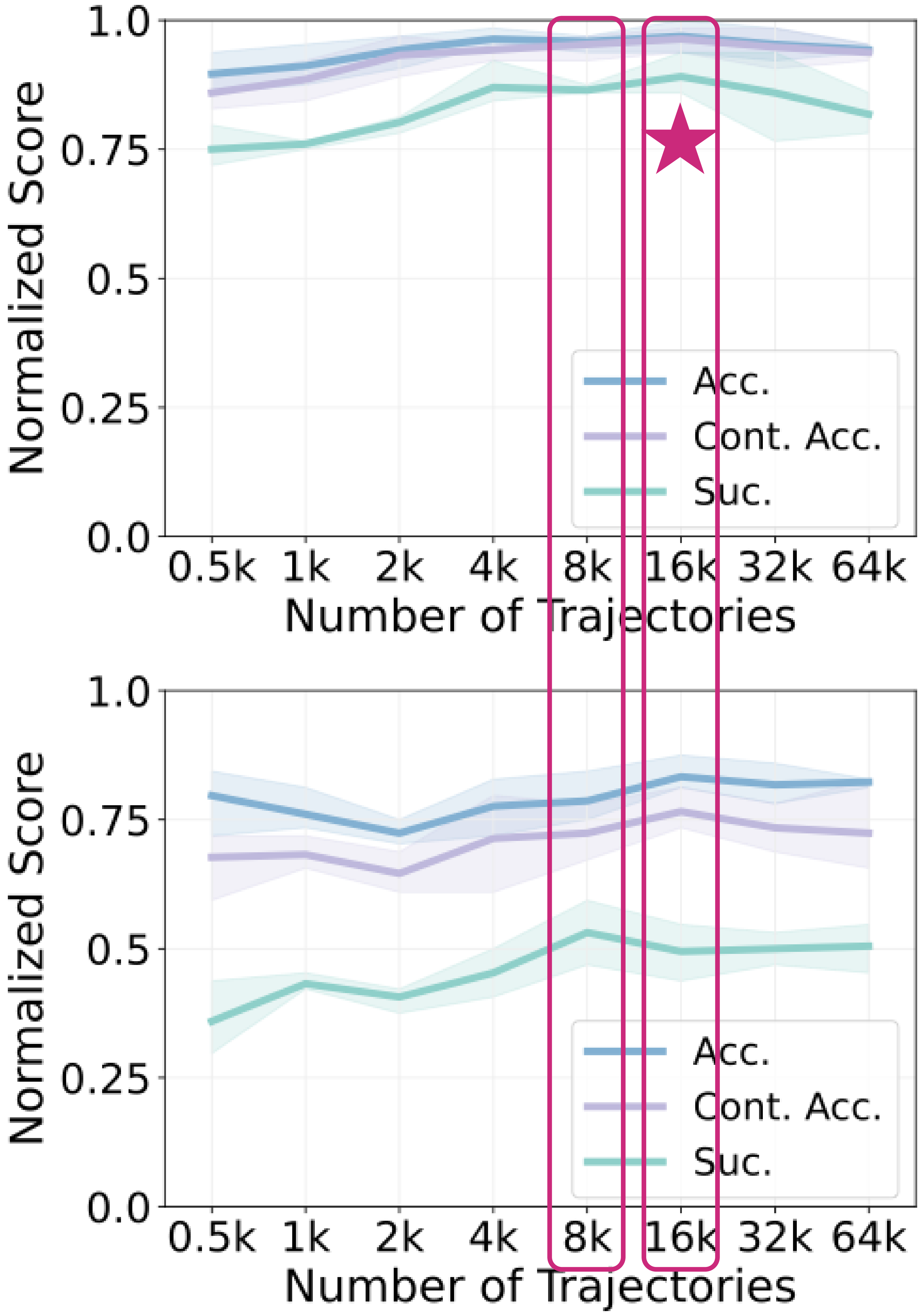
$$\mathcal{L}_{\text{TPO}} = -\mathbb{E}_{(\zeta_w, \zeta_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\left(\log\frac{\pi_\theta(\zeta_w)}{\pi_{\text{ref}}(\zeta_w)} - \log\frac{\pi_\theta(\zeta_l)}{\pi_{\text{ref}}(\zeta_l)}\right)\right)\right]$$

PPO works best, SFT scales with data

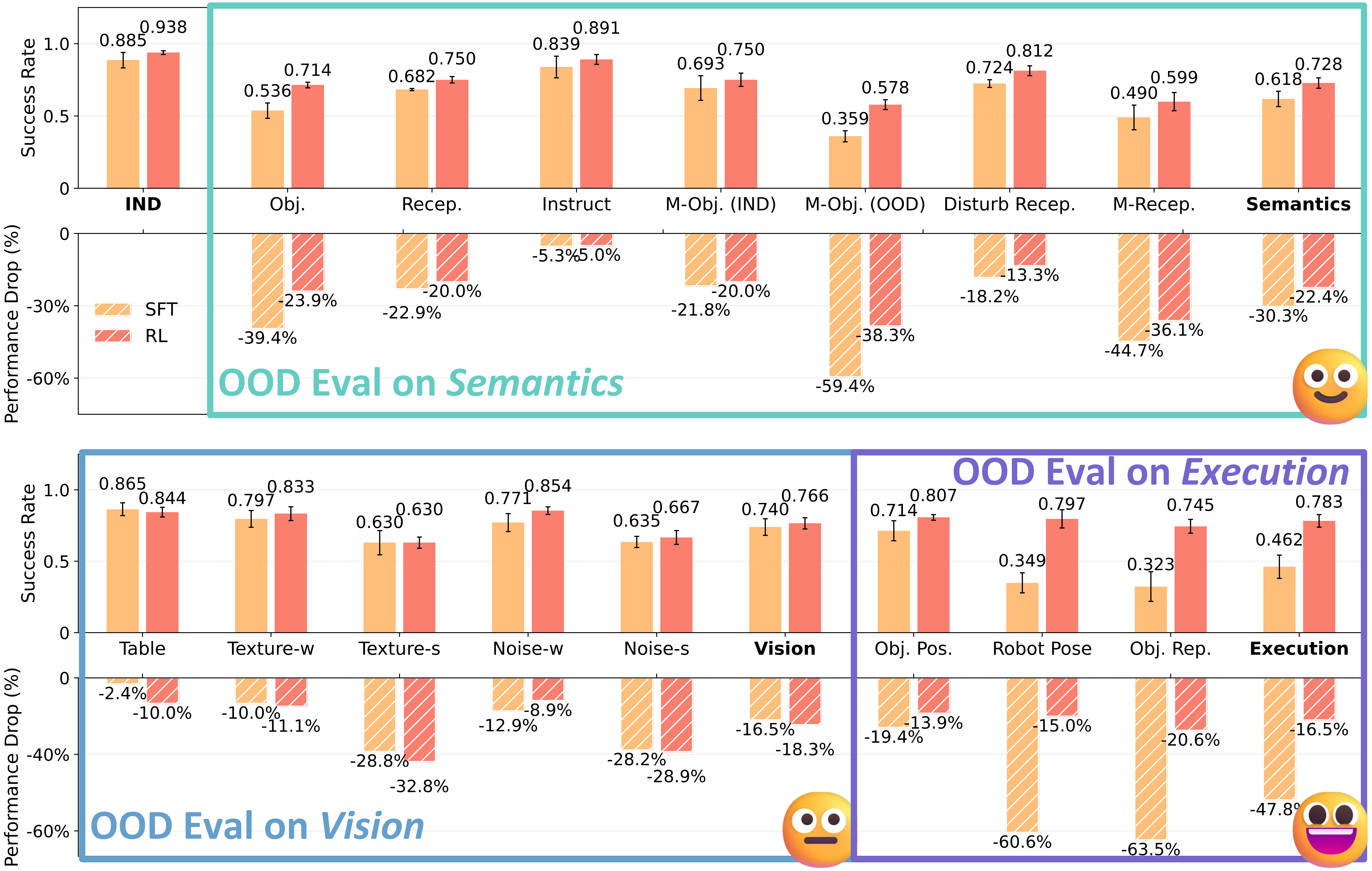


IND

OOD



RL vs SFT: does RL always win?



For VLA **generalization,**

Vision: SFT and RL perform comparably

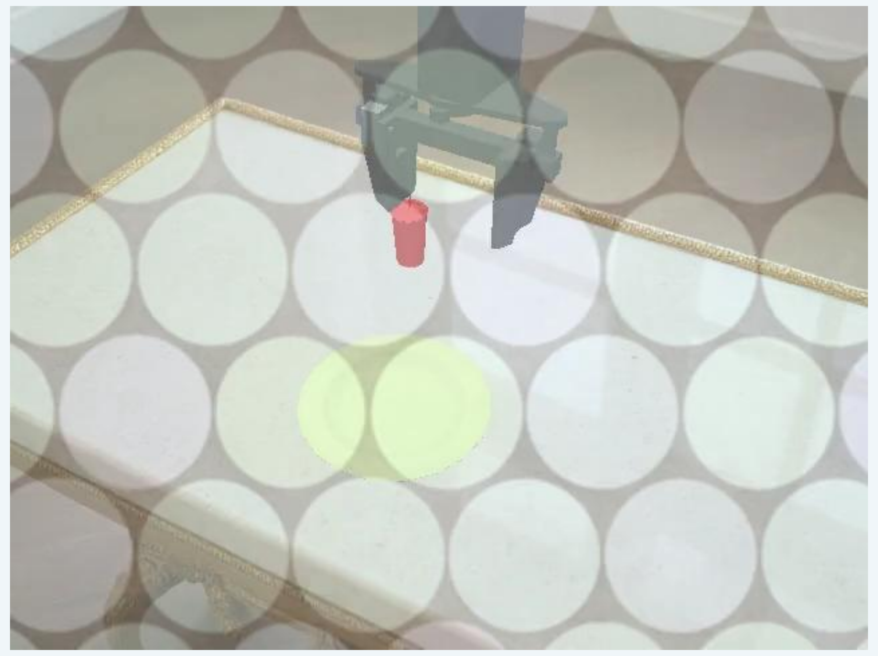
Semantics: RL improves moderately

Execution: RL enhances substantially

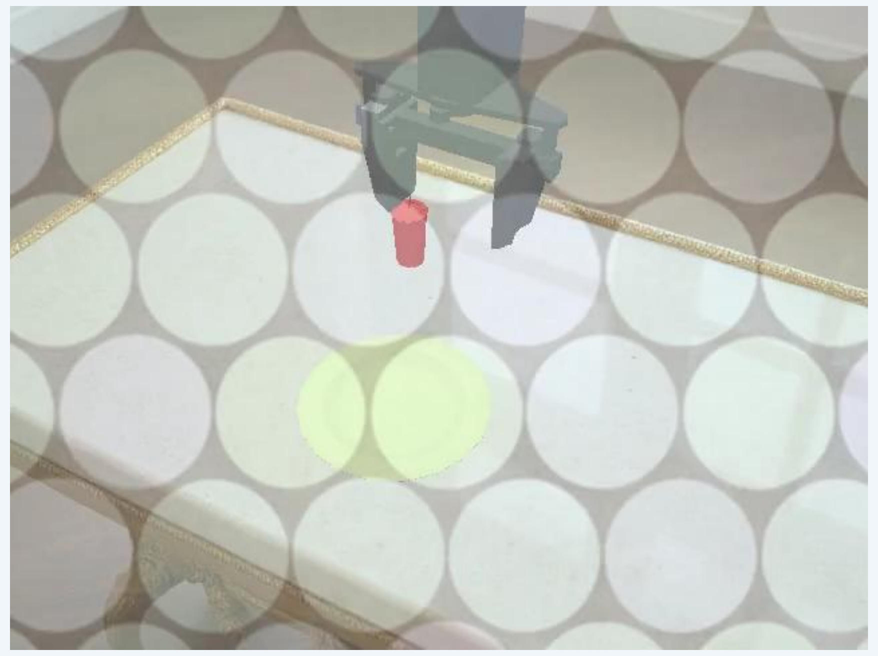
RL vs SFT: does RL always win?

Vision (V)

Noice Strong



SFT Fail



RL Fail

Semantics (L)

New Object

SFT Fail

RL Success

Execution (A)

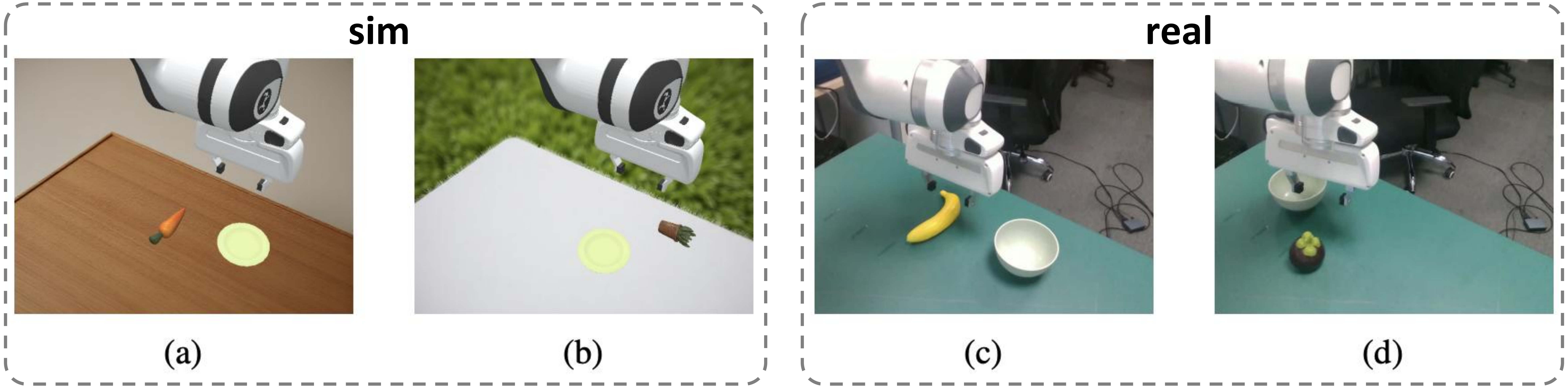
Mid Episode Reposition

SFT Fail

RL Success

RL vs SFT: what about real-world deployment?

RL can also help **zero-shot sim-to-real generalization**



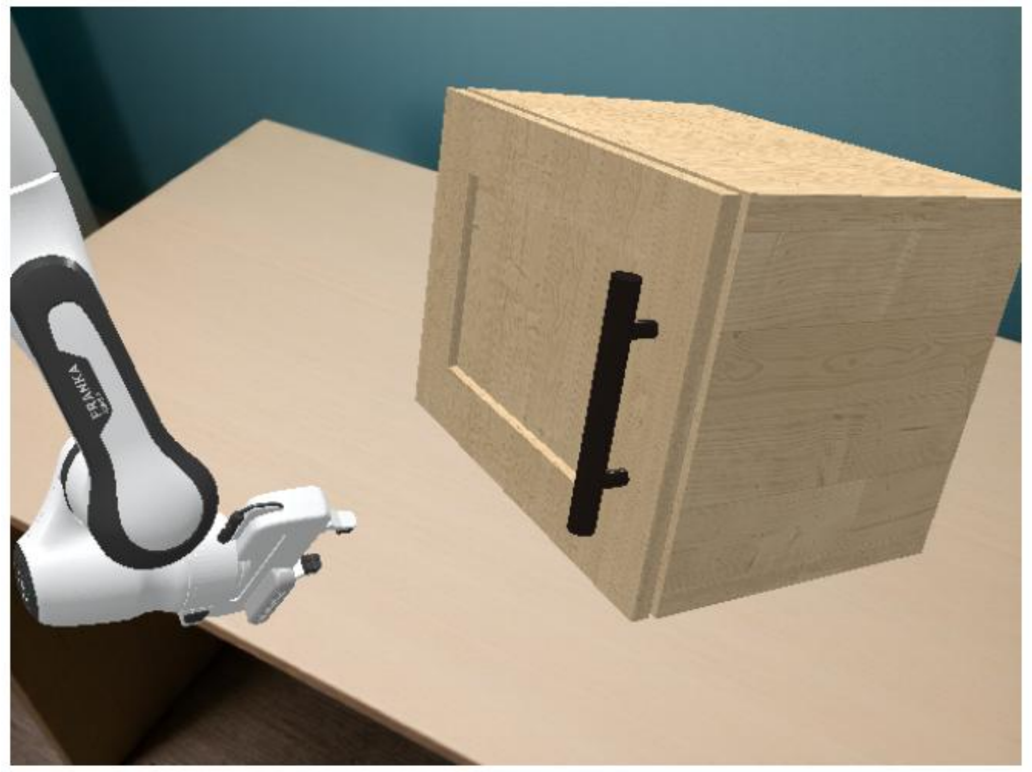
	SFT	RL
Grasp Success Rate	0.10	**0.43**
Pick-and-Place Success Rate	0.00	**0.27**

SFT

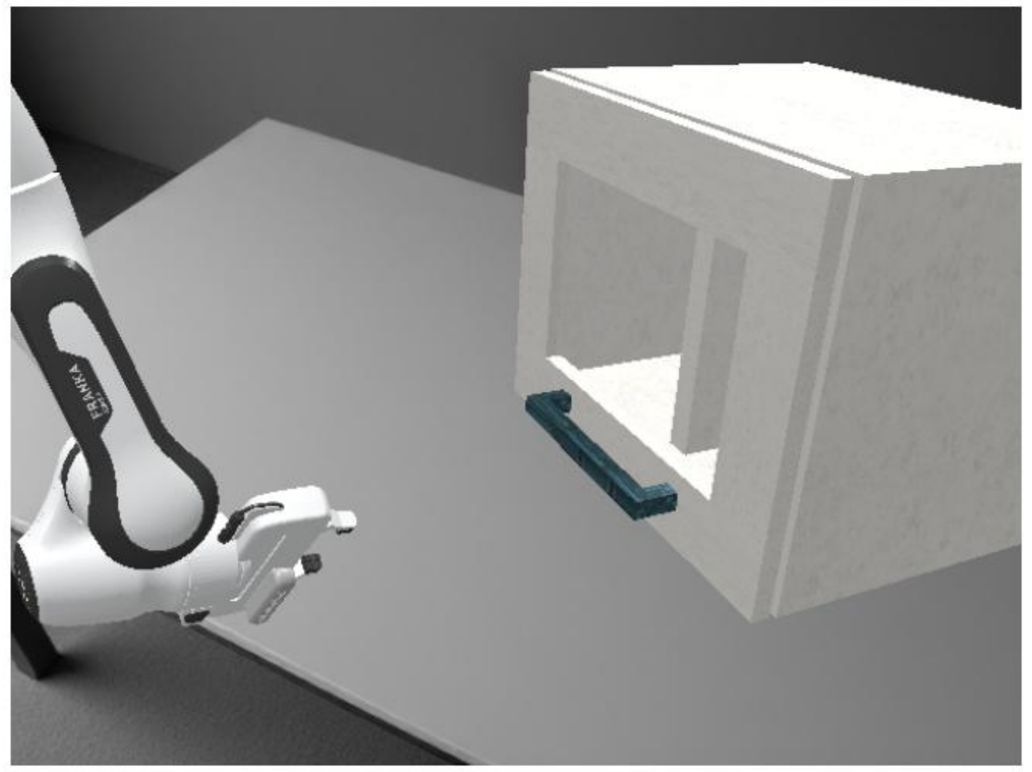
RL

RL vs SFT: what about more challenging tasks?

Open articulated objects



(a)



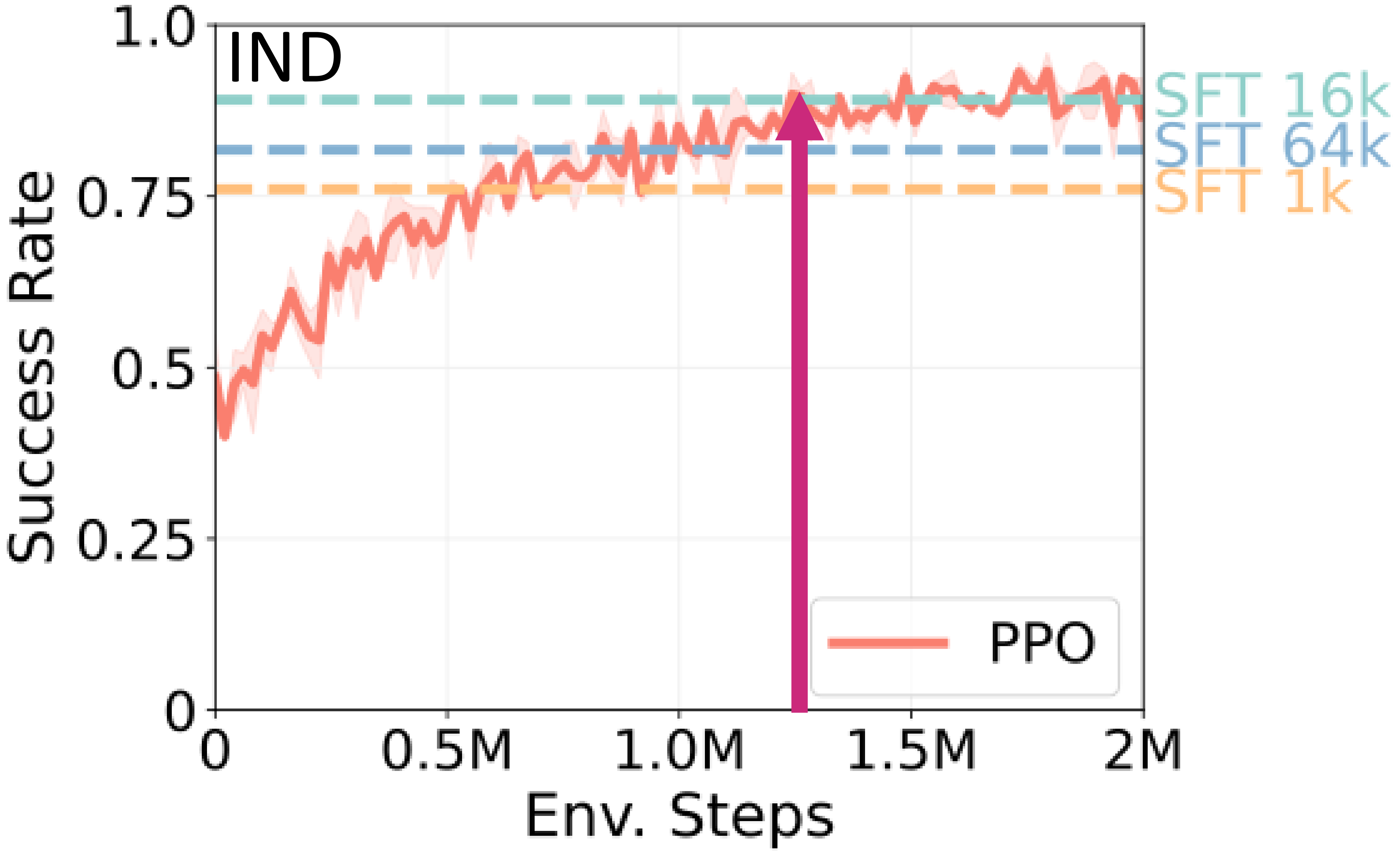
(b)

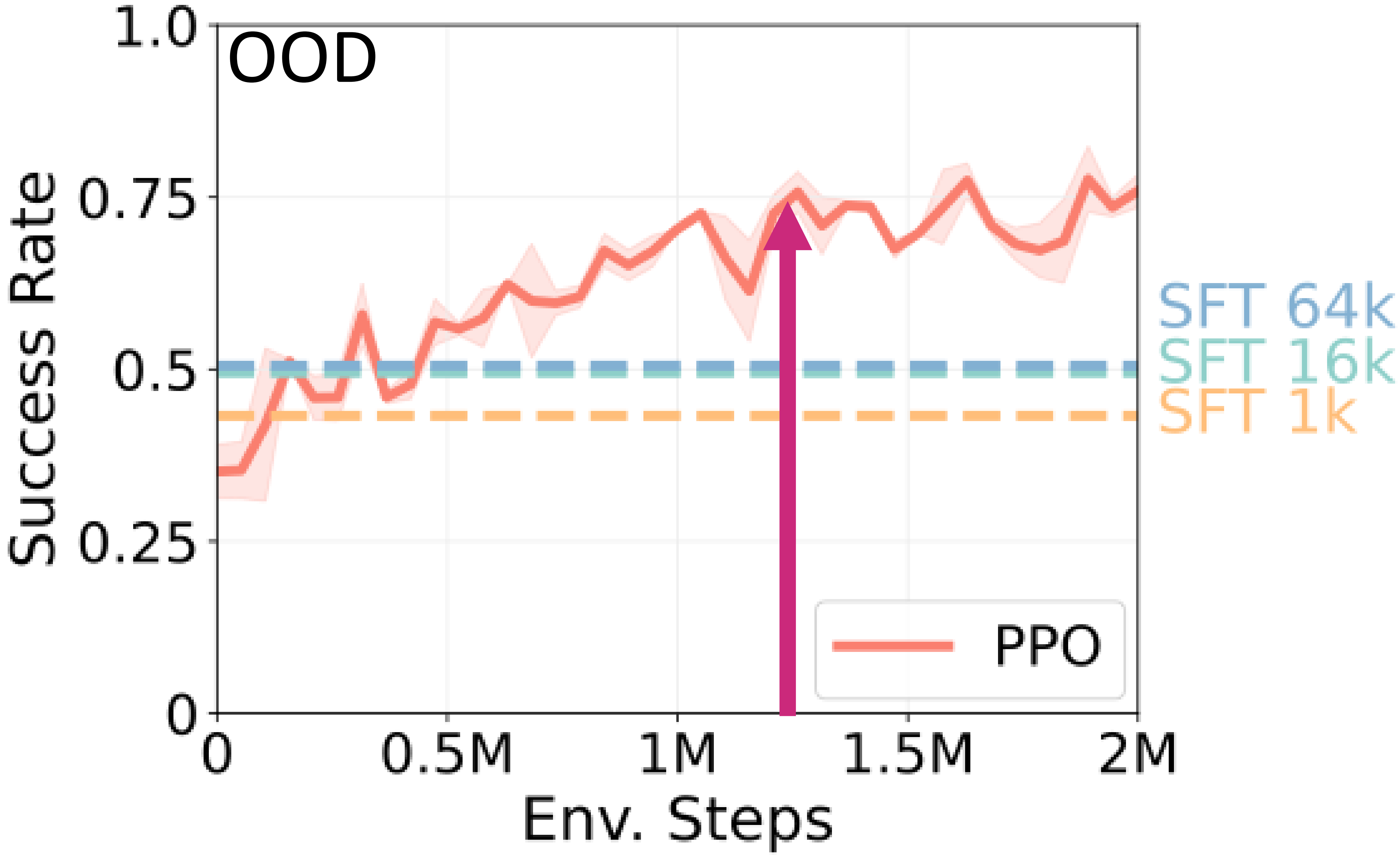
(c)

	SFT		RL	
	Success Rate	Degradation Ratio	Success Rate	Degradation Ratio
IND	**0.745** (.015)	–	0.724 (.045)	–
New Backgrounds	**0.766** (.071)	**+0.028**	0.734 (.058)	+0.014
Dynamic Noise	**0.526** (.015)	**-0.294**	0.448 (.037)	-0.381
Vision Avg.	**0.646** (.043)	**-0.130**	0.591 (.048)	-0.184
Articulated Obj.	0.042 (.015)	-0.944	**0.151** (.019)	**-0.791**
Re-phrased Inst.	**0.703** (.077)	**-0.056**	0.620 (.041)	-0.144
Semantic Avg.	0.373 (.046)	-0.500	**0.386** (.030)	**-0.468**
Obj. Pose	0.448 (.032)	-0.399	**0.484** (.089)	**-0.331**
EE Pose	0.151 (.027)	-0.797	**0.312** (.013)	**-0.568**
Execution Avg.	0.300 (.030)	-0.600	**0.398** (.051)	**-0.450**

Why can RL work better?

Hypothesis 1: Does it benefit from using more data?



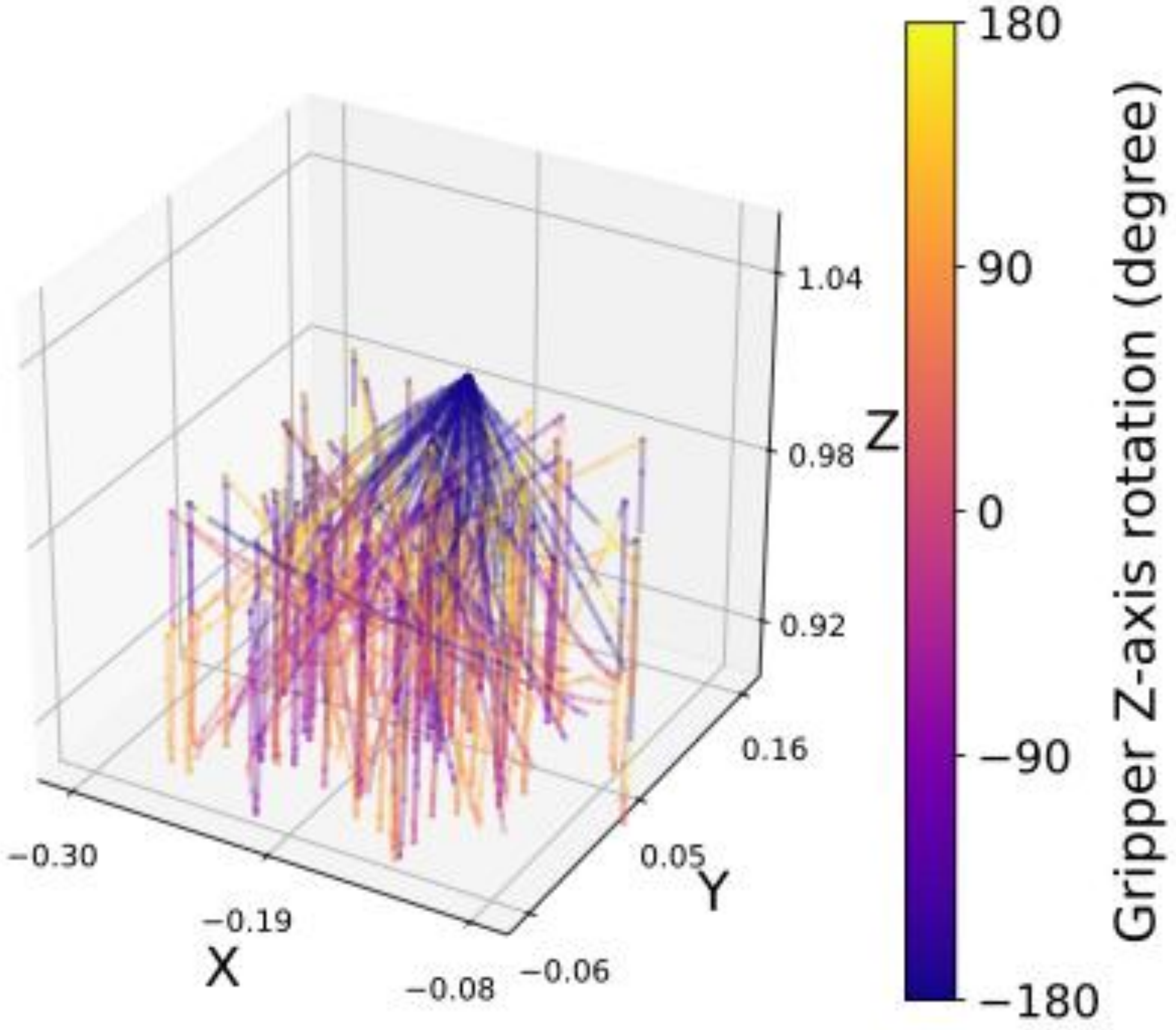


64k trajectory ~ 1.26M transitions

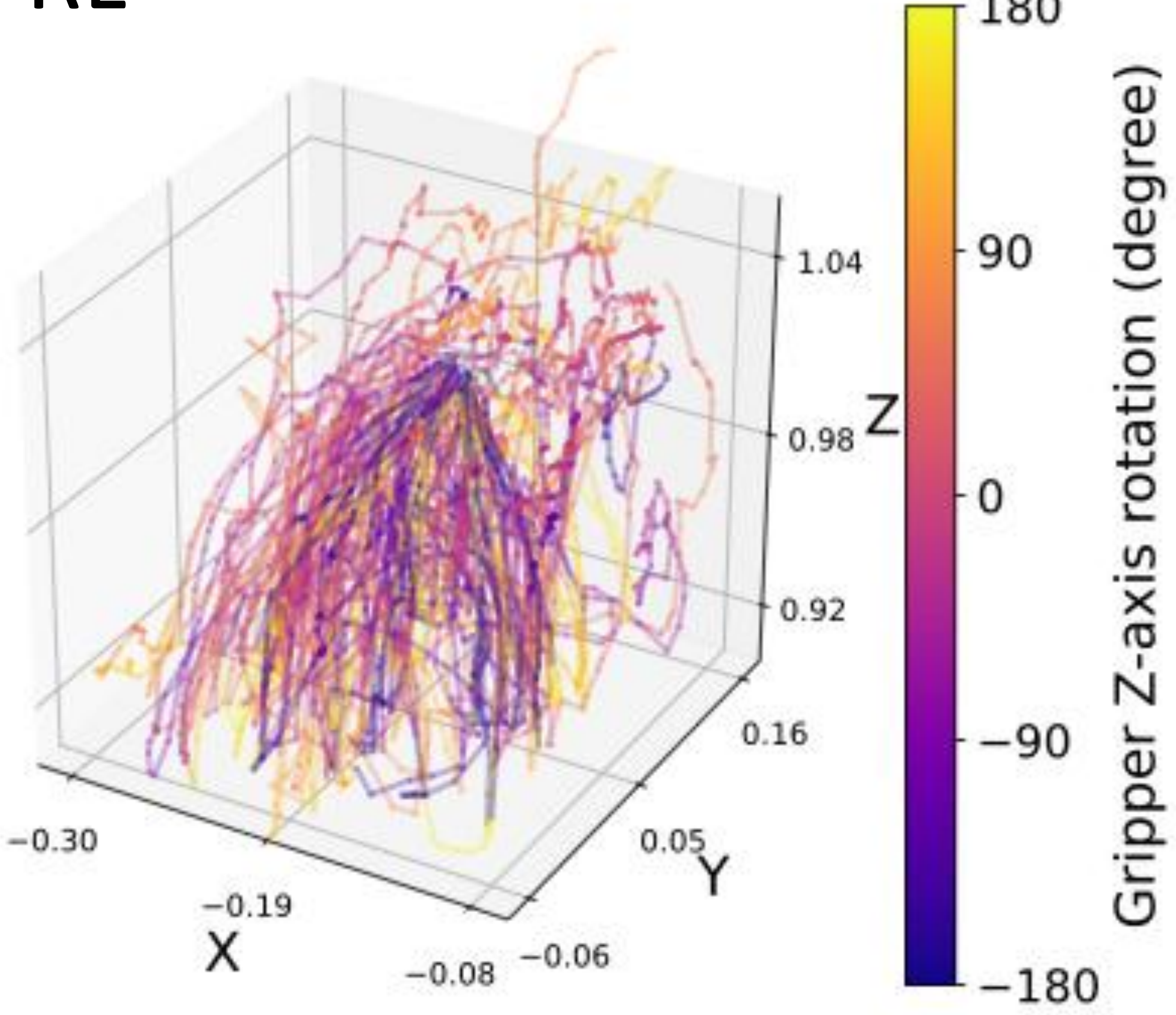
Why can RL work better?

Hypothesis 2: Does it benefit from diverse exploration?

SFT



RL



RL trajectories span a broader workspace and a richer range of end-effector orientations

Thanks for your time!

Project: https://rlvla.github.io

Code: https://github.com/gen-robot/RL4VLA

Project page 👉