

Towards Safe and Efficient Robot Foundation Models

Jeongeun Park

2026.08.25

Introduction

Jeongeun Park

Research Interest: Robot Foundation Model, Human Robot Interaction



- Korea University
- 2021.03~2026.02
- Ph.D. in Department of Artificial Intelligence
- Advisor: Sungjoon Choi



- NAVER AI LAB
- 2024.04~2024.10
- Research Intern
- Advisor: Sangdoo Yun

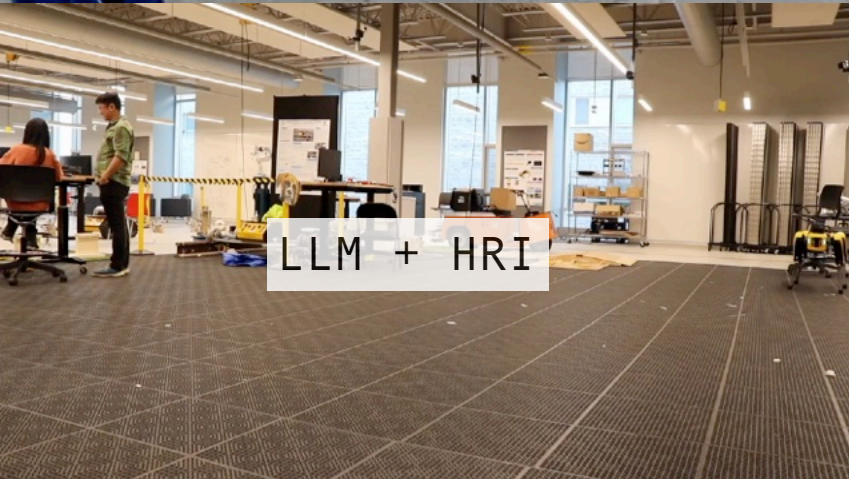
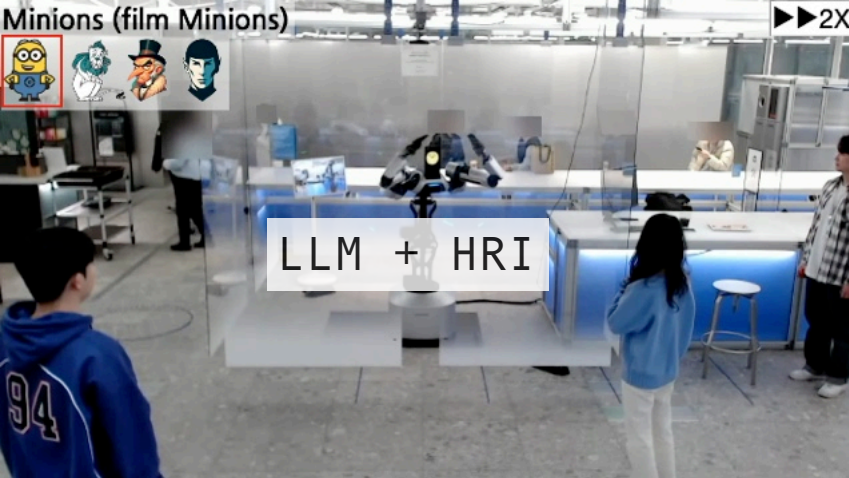


- NAVER AI LAB
- 2026.02~2026.08
- Post-doctoral Researcher



- Georgia Institute of Technology
- 2026.10~ (Expected)
- Post-doctoral Researcher

Minions (film Minions)



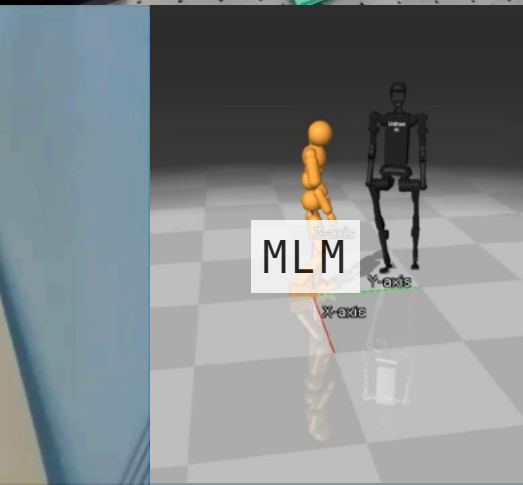
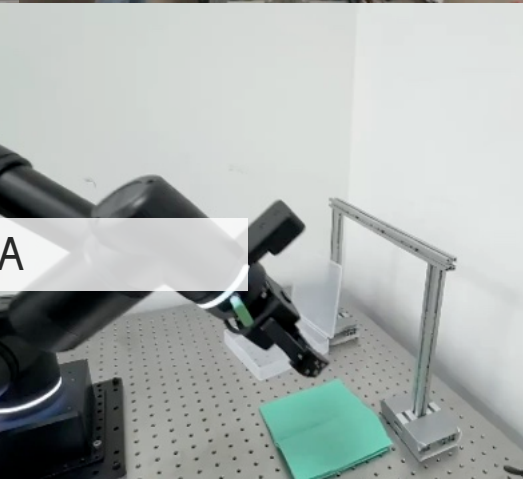
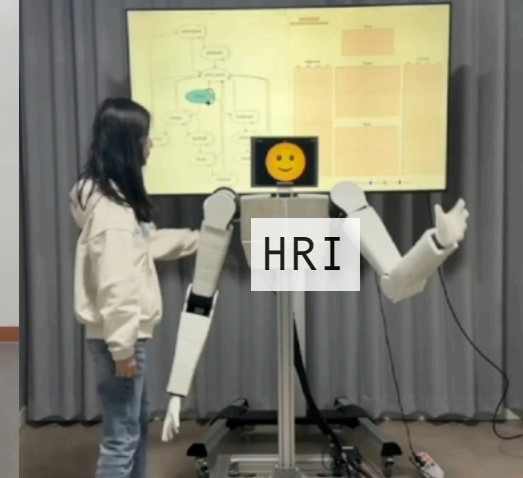
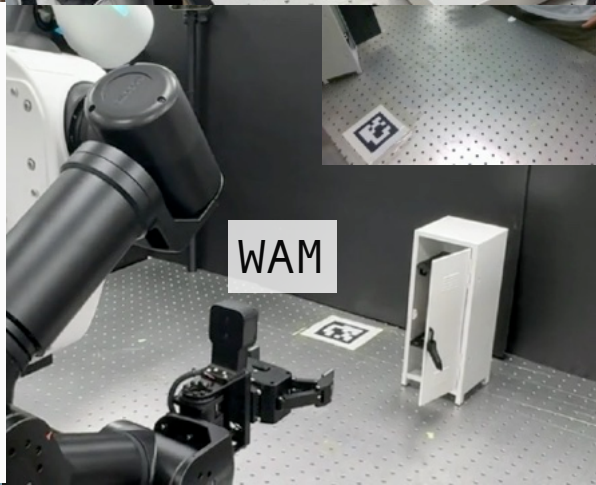
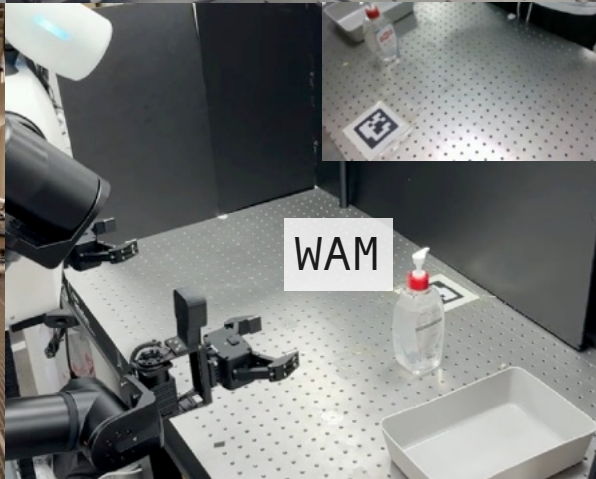
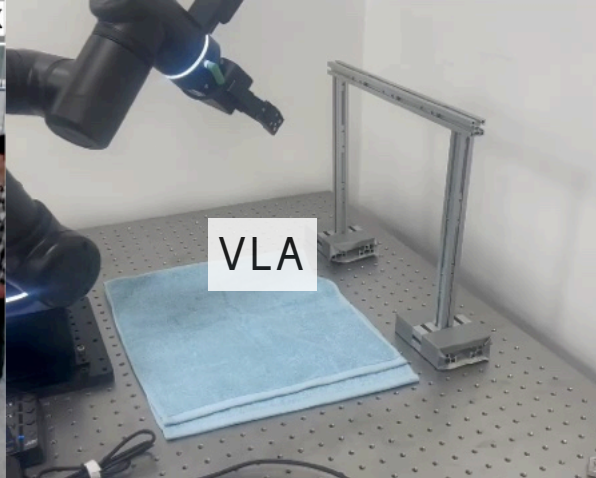
Cellphone



Laptop

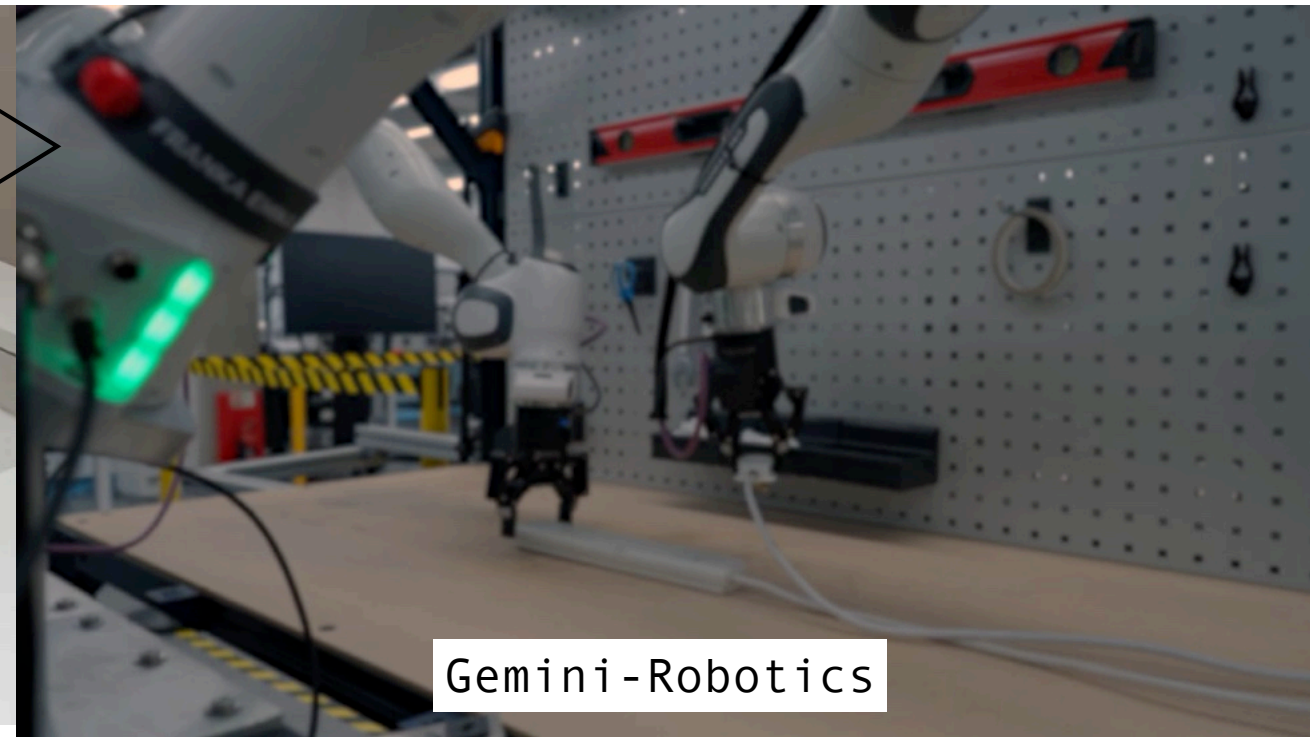


Language Based Navigation



What is “RFM”

- Robot Foundation Model
 - A model that can perform diverse tasks, manipulate every objects, and work in diverse environments



What is “RFM”

- Robot Foundation Model



<https://www.1x.tech/discover/world-model-self-learning>

What is “RFM”

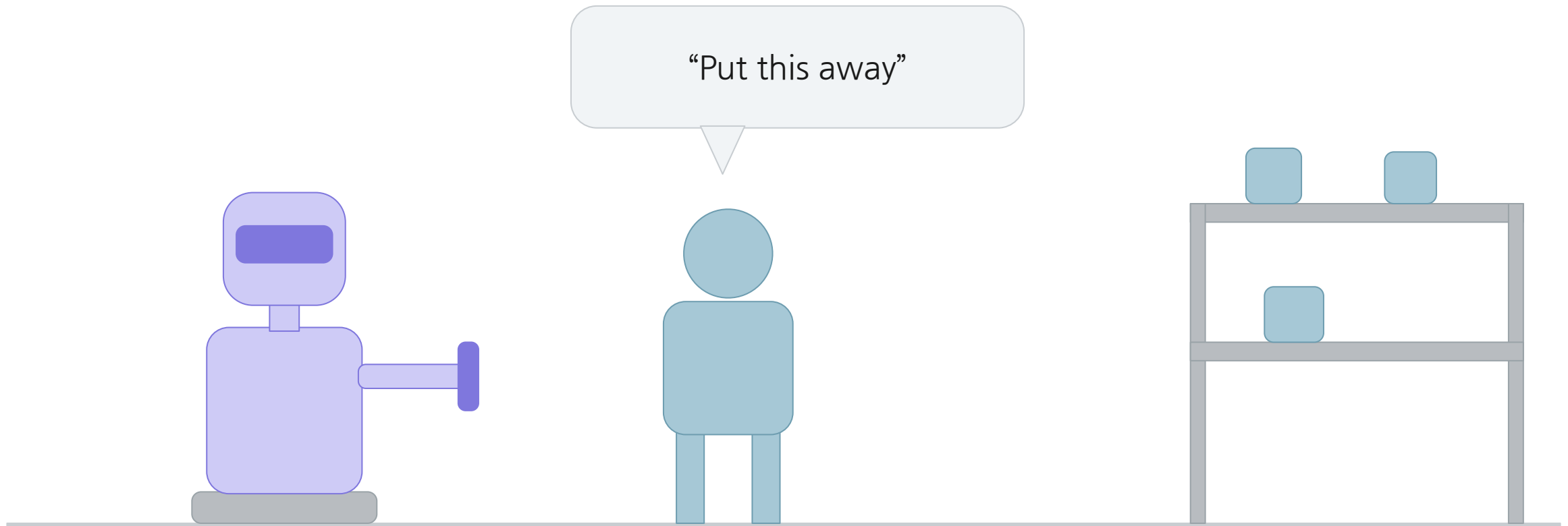
- Robot Foundation Model

One-Shot Policy



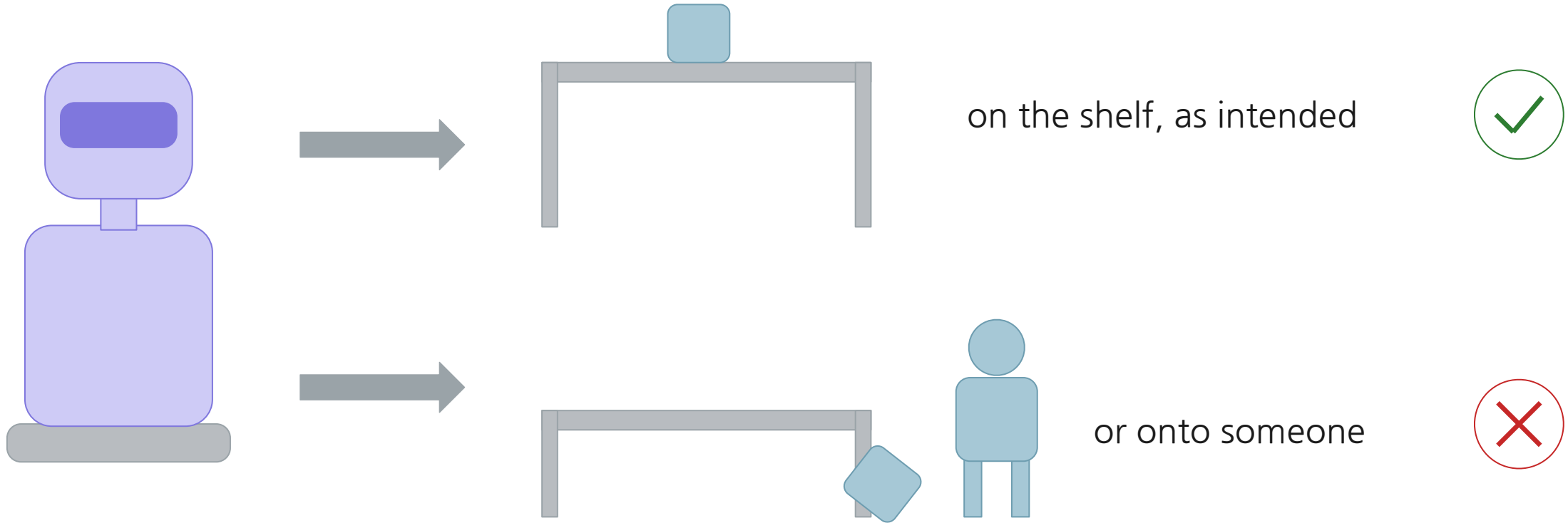
<https://generalistai.com/blog/gen-1.5>

Deployed Next to People



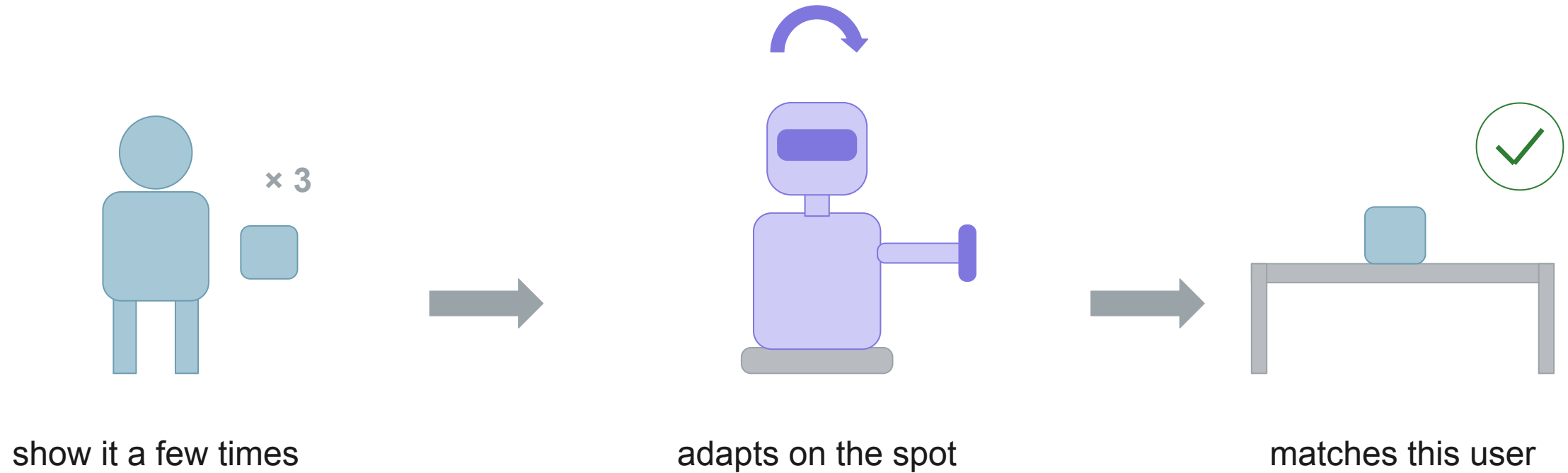
Unfamiliar home. Non-expert user. No engineer in the loop.

Safety



A wrong action here is not merely unhelpful, it can even injure someone.

Efficiency



Minutes, not months — without ever relaxing the safety constraint.

“Safety”

Hardware safety

Collision avoidance

Formal verification

Distribution shift

Human-robot contact

Undesirable actions

This work: avoiding undesirable actions

“Efficiency”

Compute cost

Parameter-efficient tuning

Data collection cost

Sample-efficient RL

Few Shot, In-Context Learning

Representation learning

**This work: learning new tasks without training,
and representations that make policy learning cheap**

Contents

01 Safety

Research Question

How can we avoid undesirable actions?

02 Efficiency

Research Question

How can model learn new tasks without training?

Can we build efficient representation for policy learning?

Contents

01 Safety

Research Question
How can we avoid undesirable actions?

02 Efficiency

Research Question
How can model learn new tasks without training?
Can we build efficient representation for policy learning?

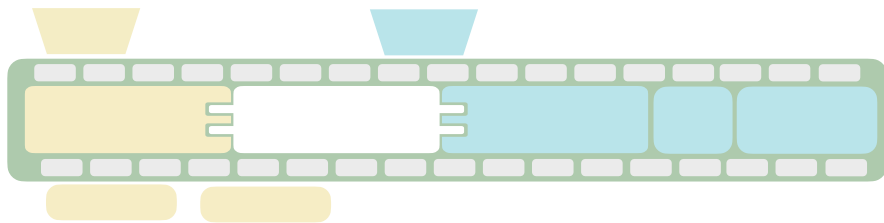
Contents

01 Safety

Research Question
How can we avoid undesirable actions?

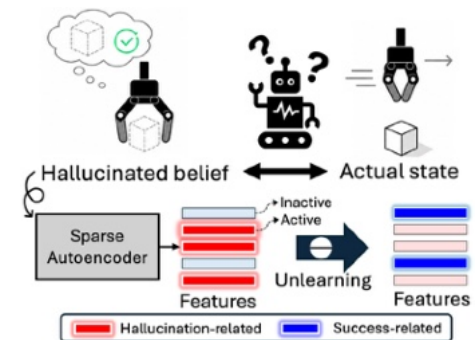
A

Hierarchical VLA

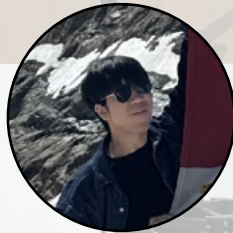
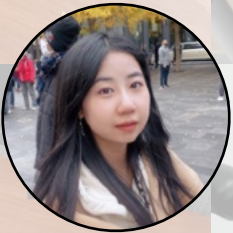


B

VLA Unlearning

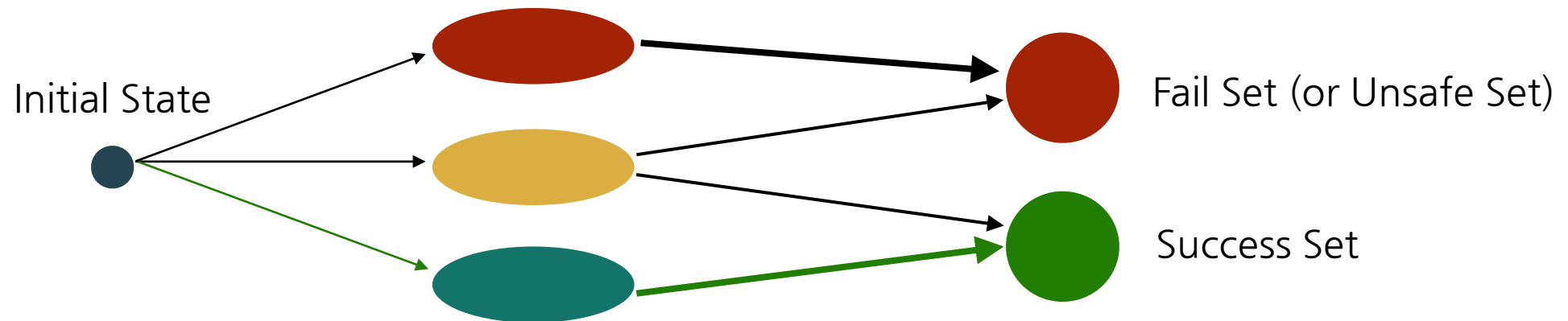
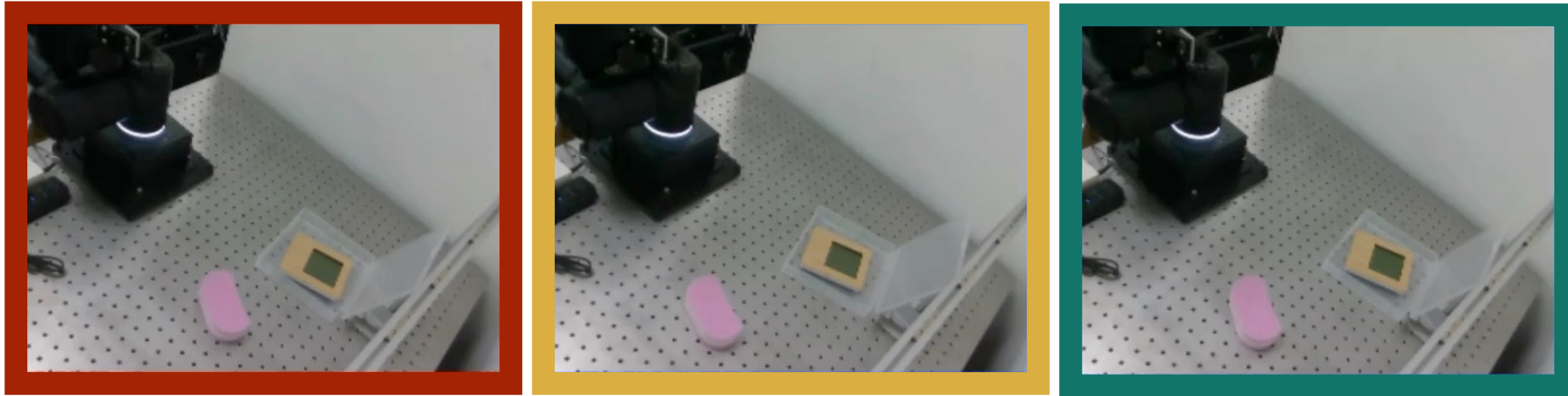


Hierarchical Vision Language Action Model Using Success and Failure Demonstrations



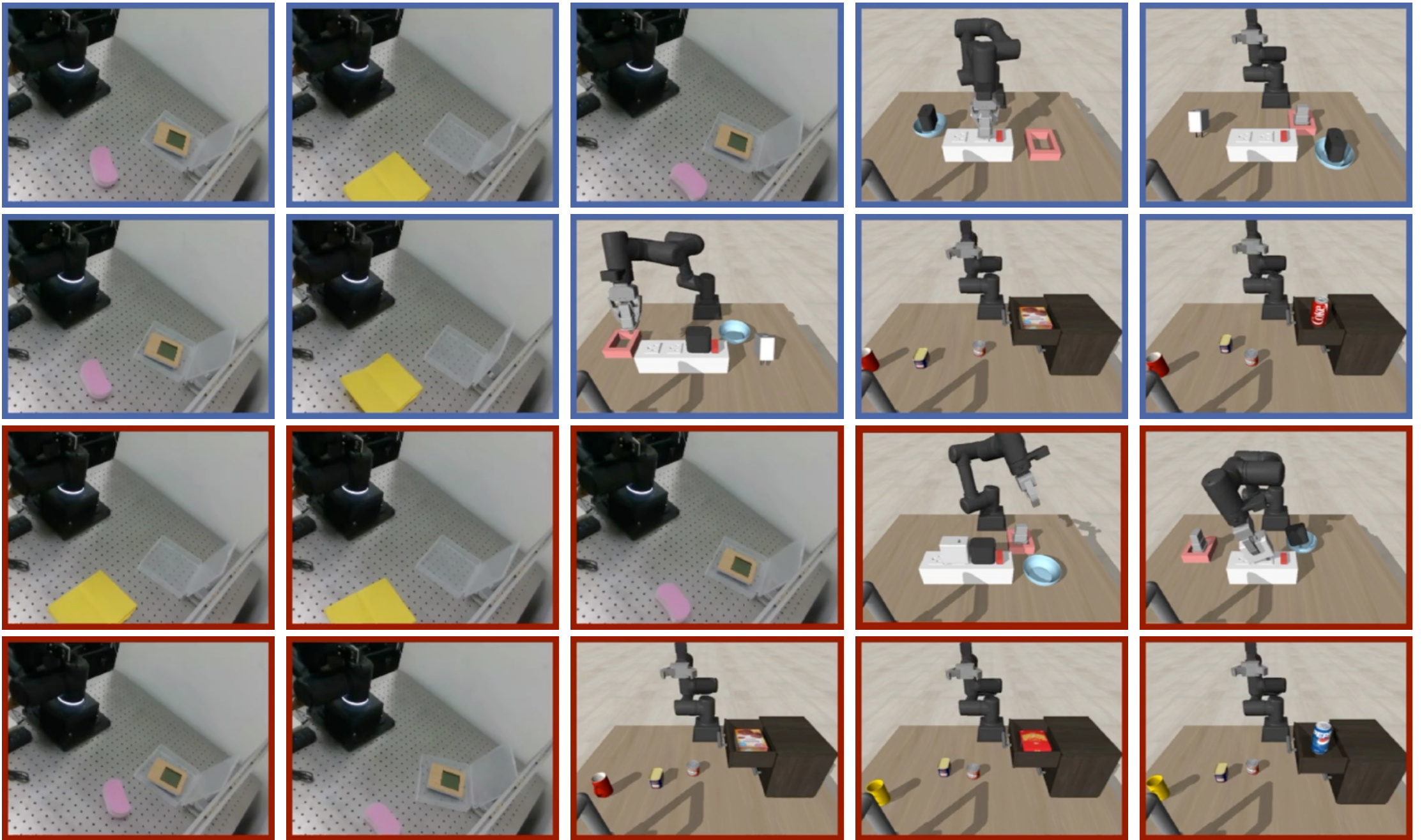
Motivation

Put sponge in cabinet and close it



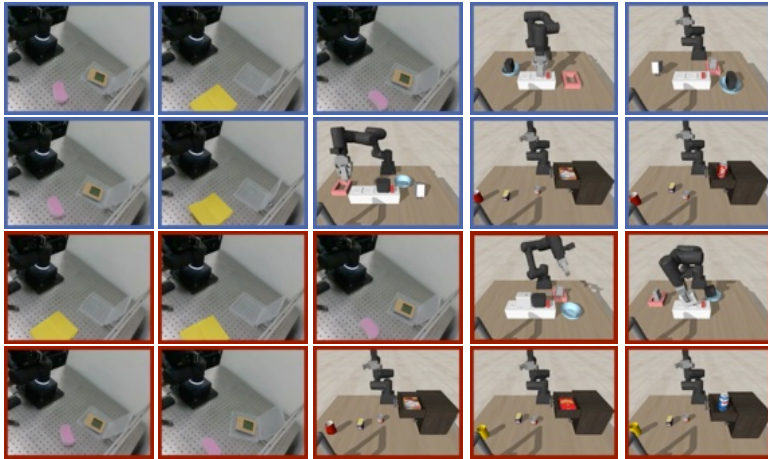
Motivation





Motivation

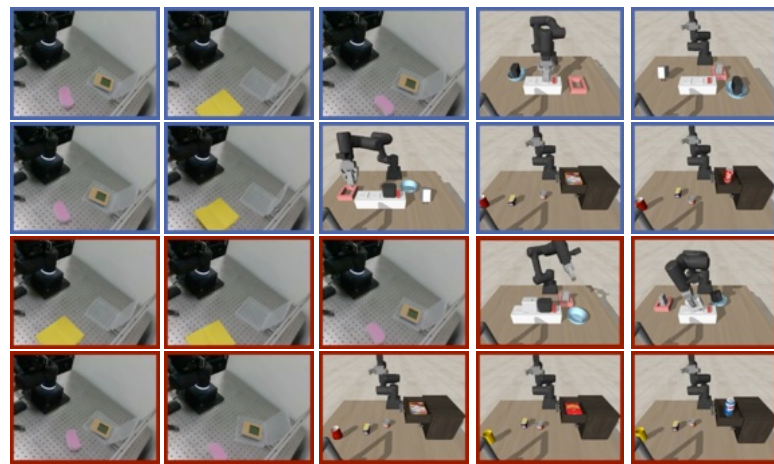
Can we train model to **avoid failures** from mixed quality dataset?



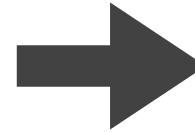
Mixed Quality Dataset

Motivation

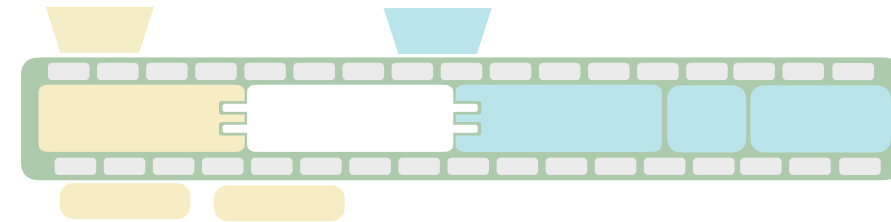
We introduce VINE, Hierarchical Vision Language Action Model that can learn from mixed quality dataset



Mixed Quality Dataset

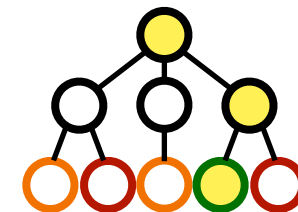


Hierarchical Vision
Language Action Model

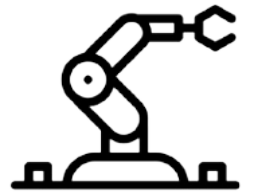


System 2

System 1



Failure Aware
Reasoning



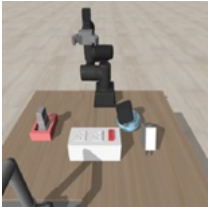
Low-level
Robot Action

Method

System 2 (Planning)

System 1 (Execution)

Initial Image



Vision Tower

Task

“Insert gray and white charger in plug”

Current image



Vision Tower

Task

Edge e_1
Insert gray charger
Node n_2
scene graph



<eoe>

Proprio.
State

Reasoning Adapter

Language Block

Action Adapter

Done Expert

Action Expert

LM Head

Score Head

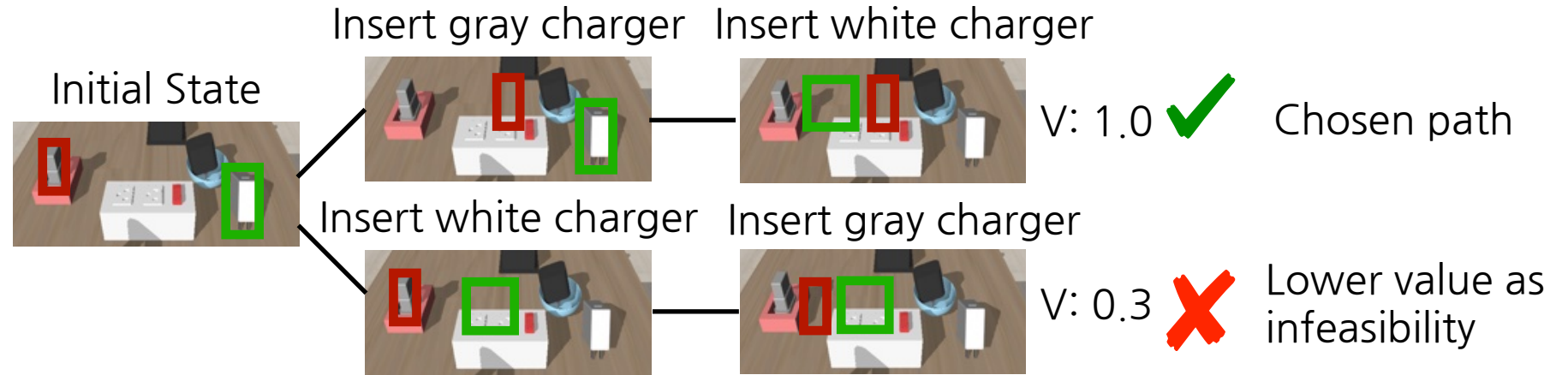
Done signal

Action
(Joint Trajectory)

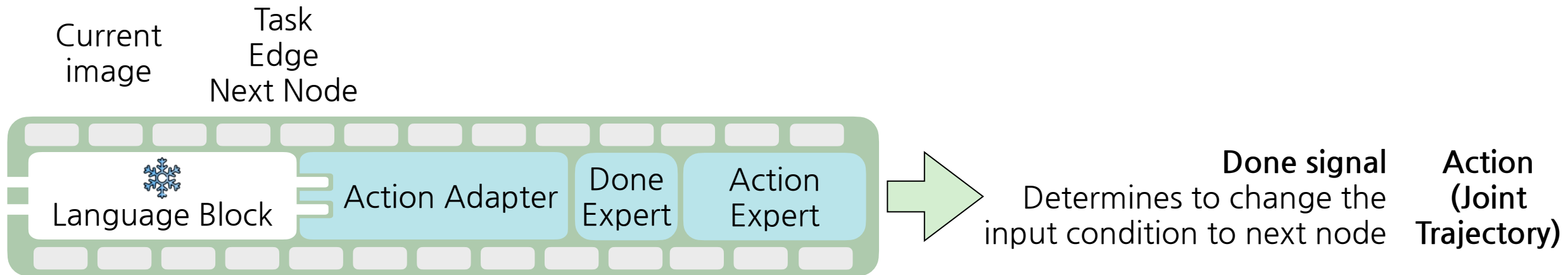
Method

System 2 (Planning)

Imaginary Roll-out
Tree with values



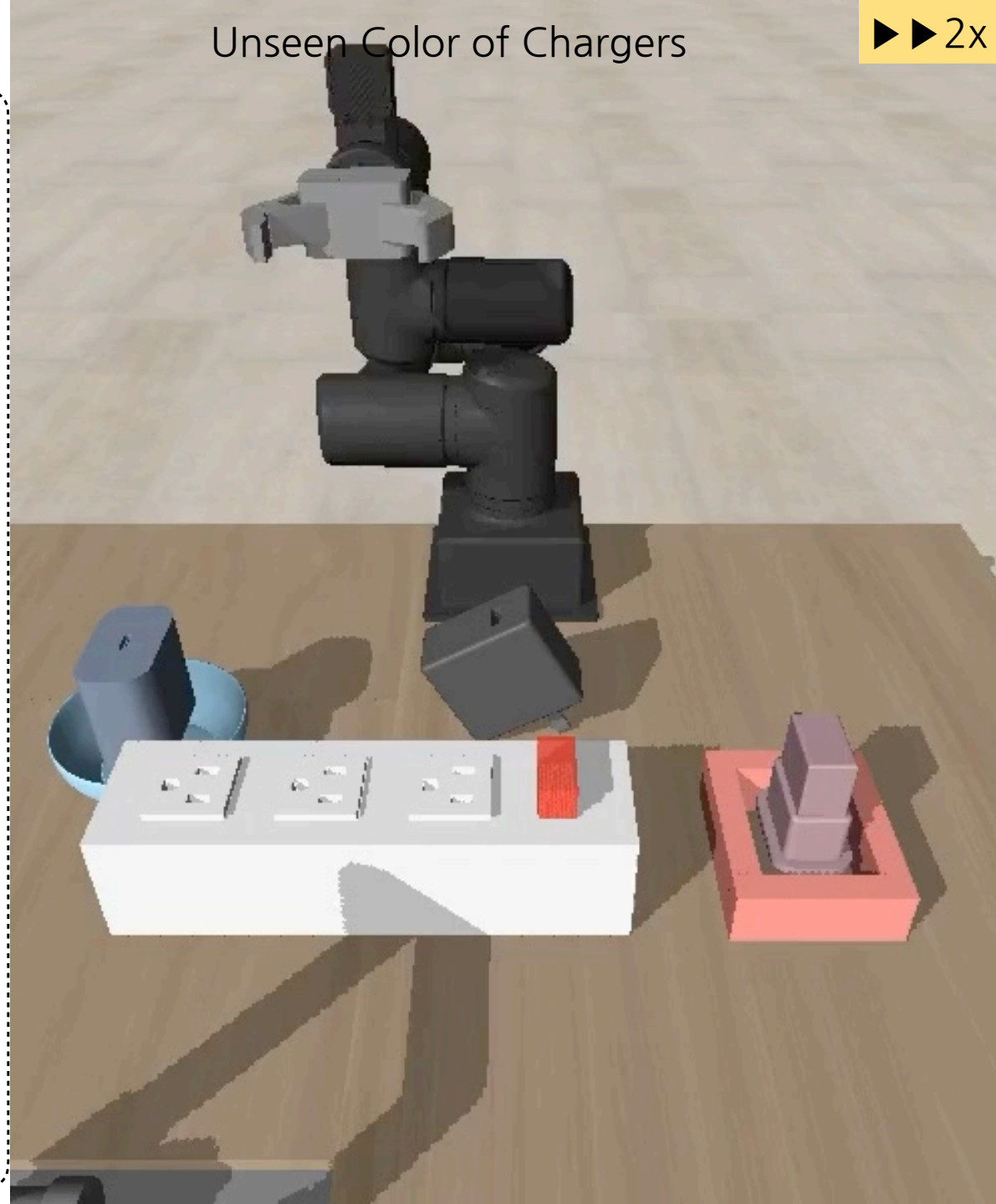
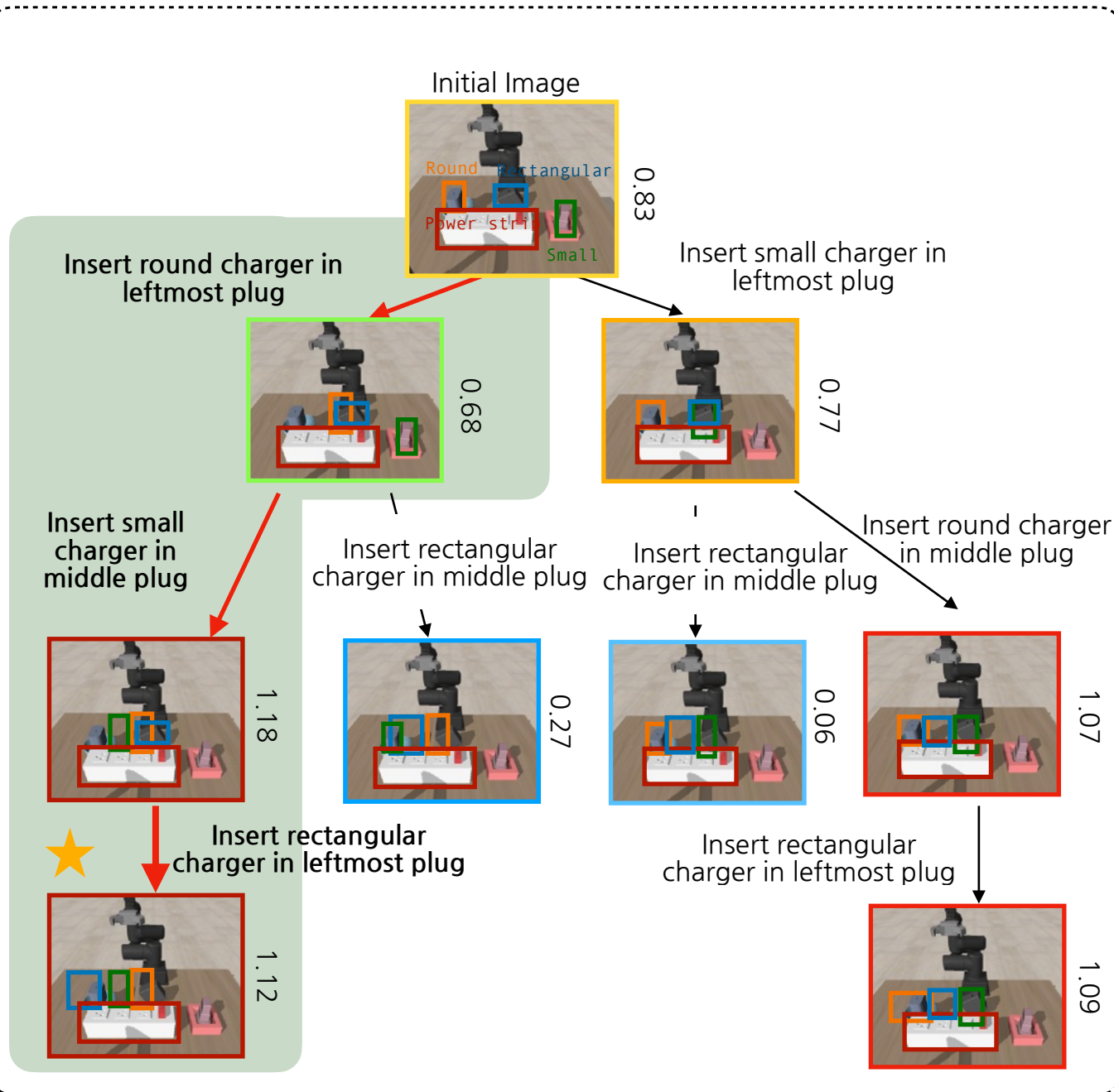
System 1 (Execution)



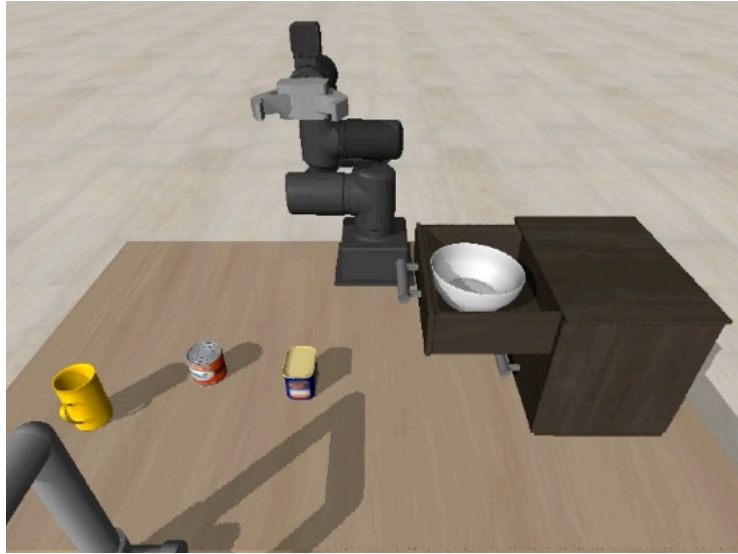
Language Instruction: Insert all chargers in power strip

Unseen Color of Chargers

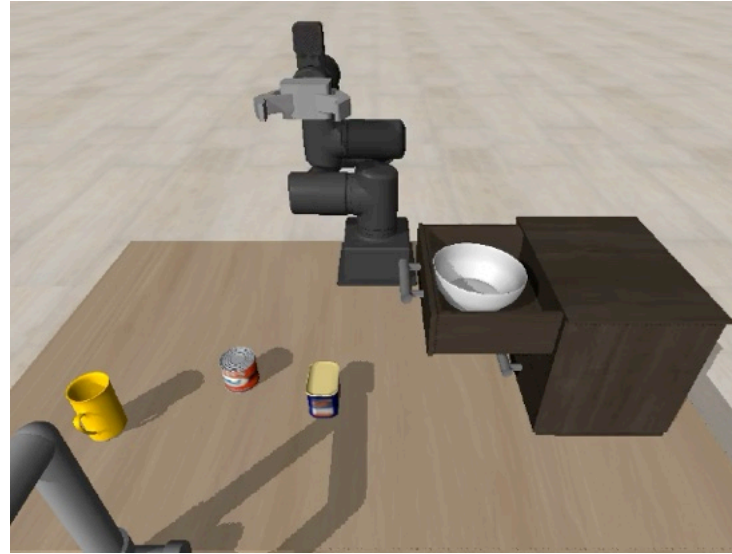
▶▶ 2x



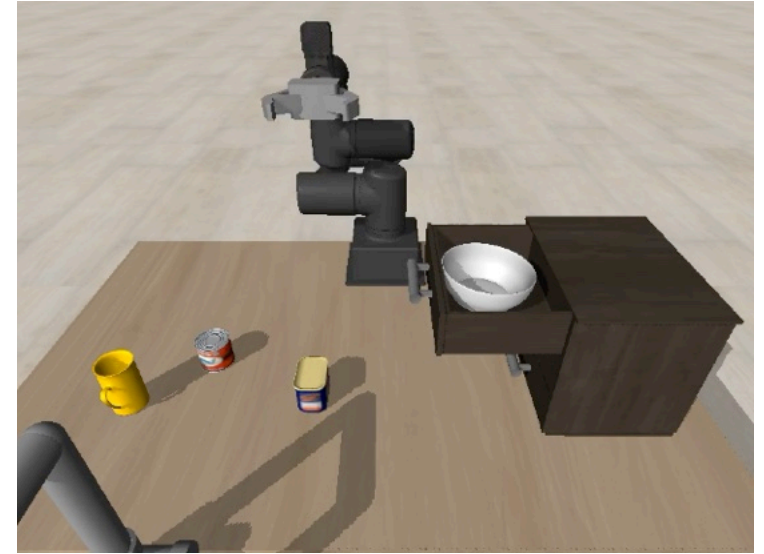
Language Instruction: Put salmon can on top drawer and close it



π_0



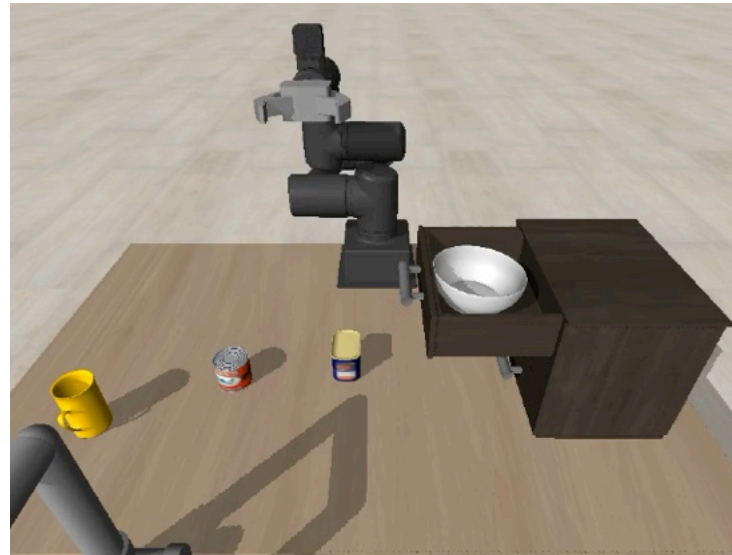
Gemini 2.5 Flash + Our System 1



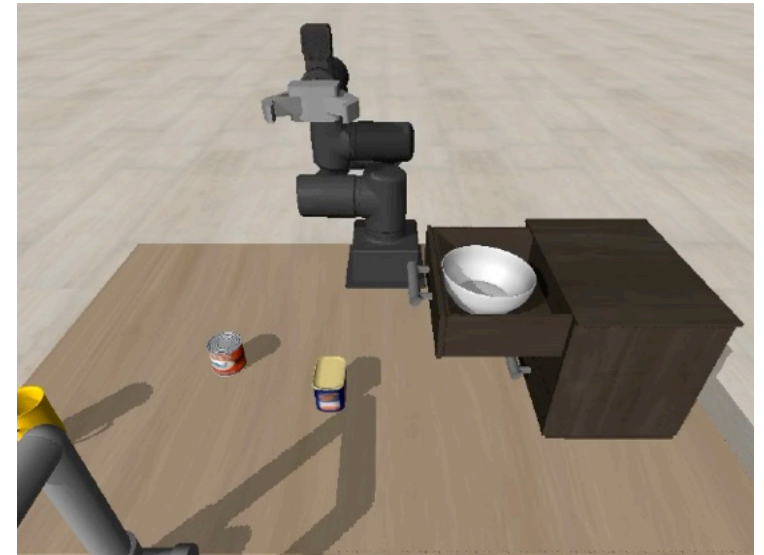
Gemini 2.5 Flash with failure
prompt + Our System 1



Gr00t 1.5



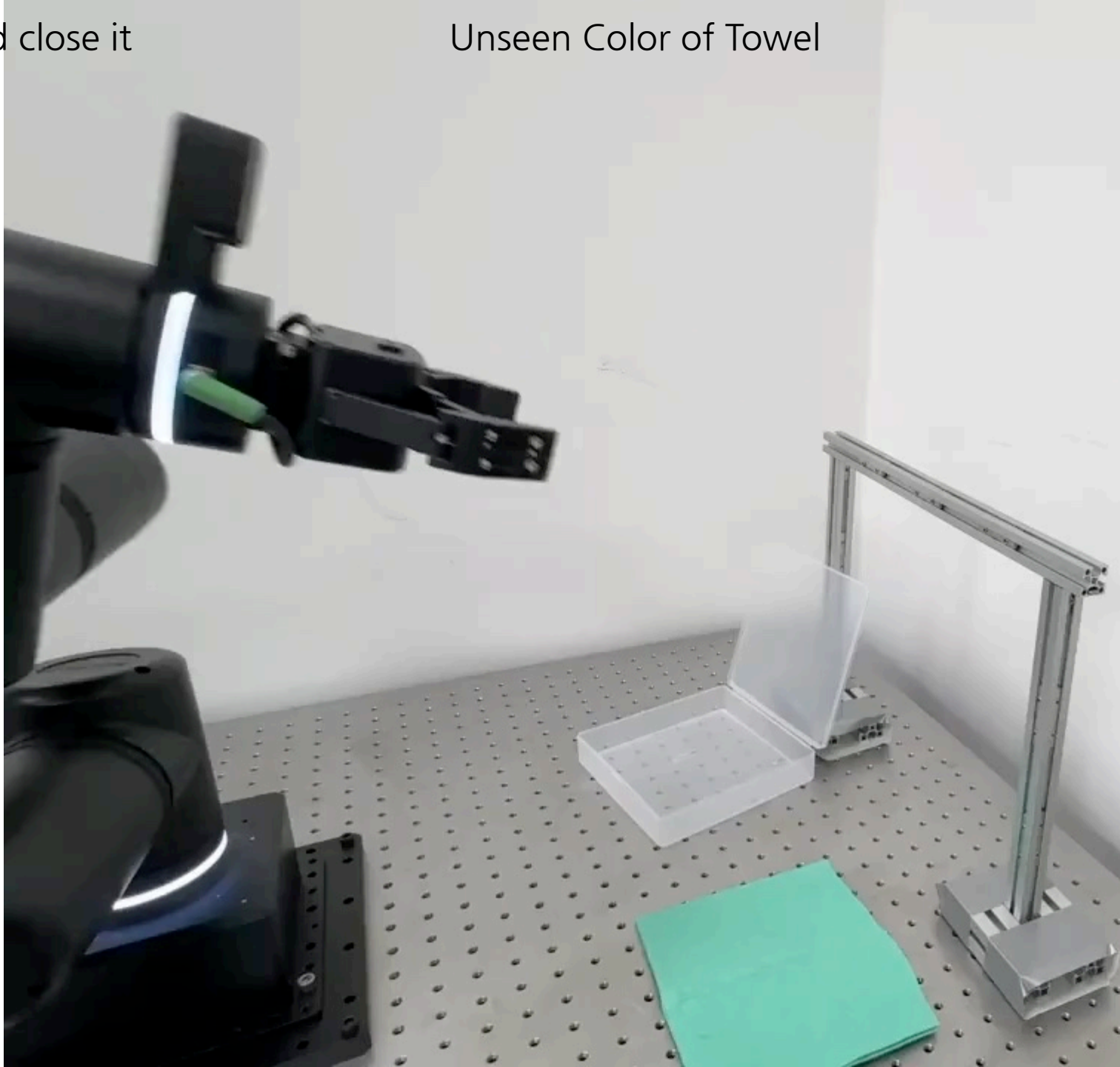
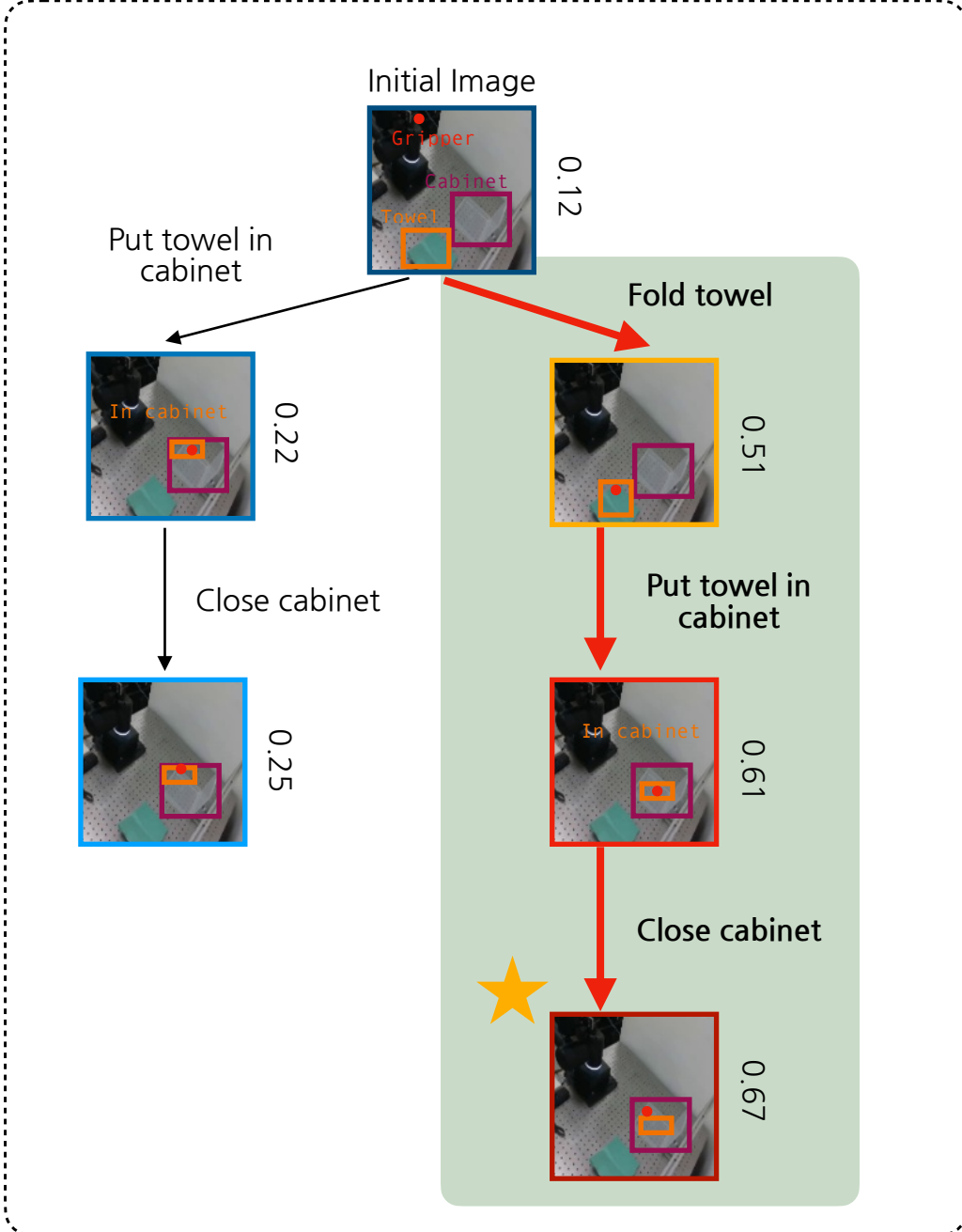
Ours w/o Failure Data



VINE (Ours)

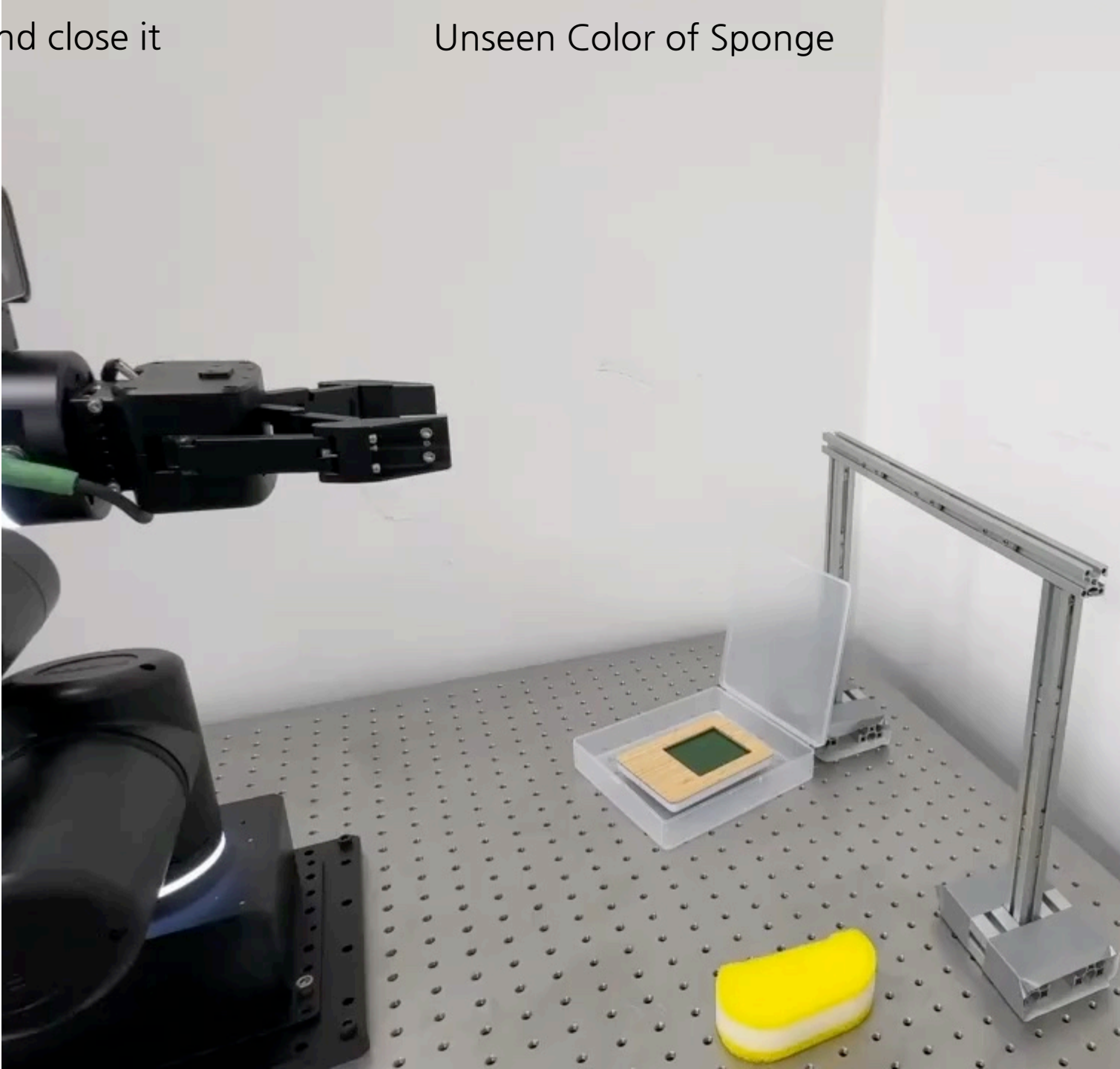
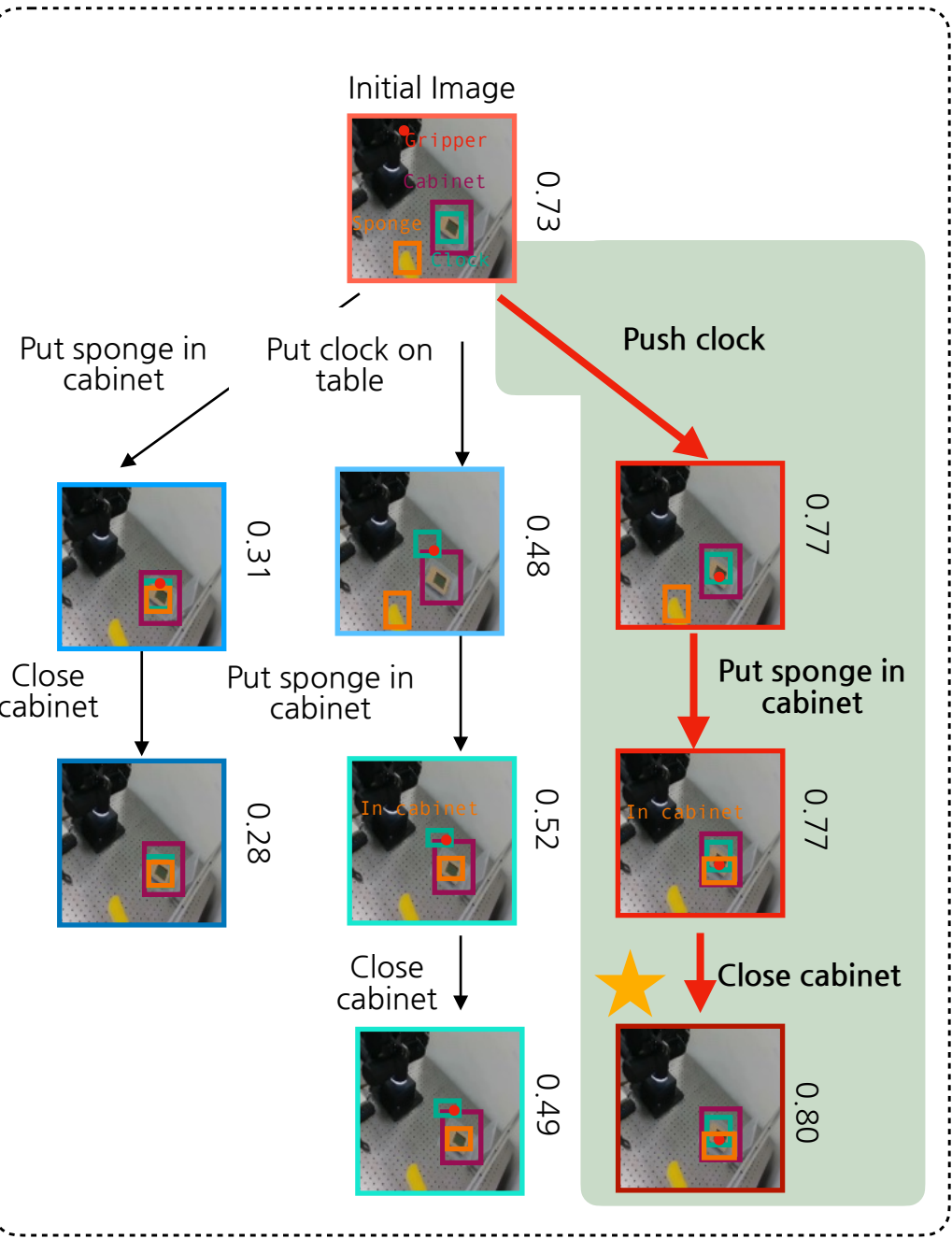
Language Instruction: Put towel in cabinet and close it

Unseen Color of Towel



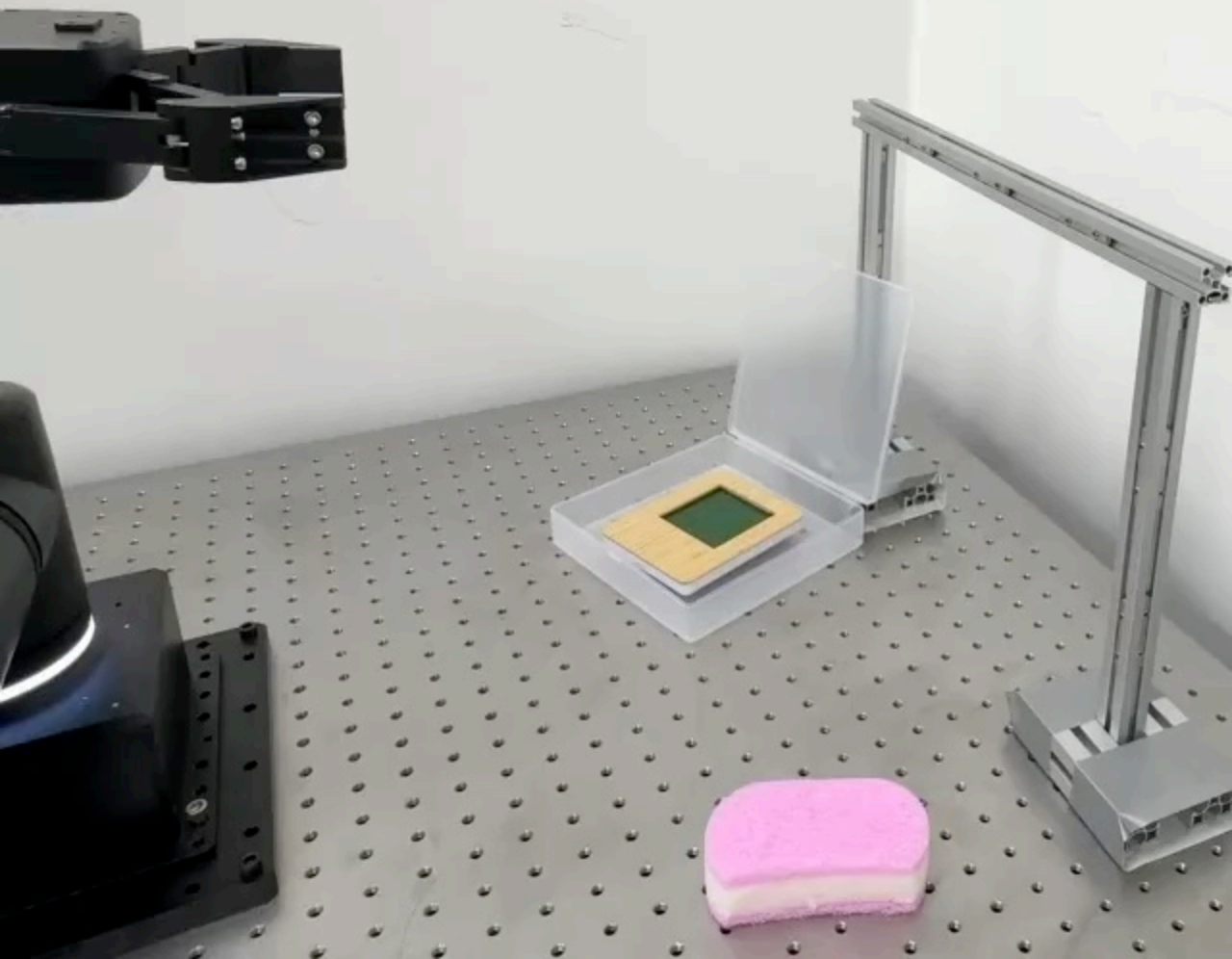
Language Instruction: Put sponge in cabinet and close it

Unseen Color of Sponge

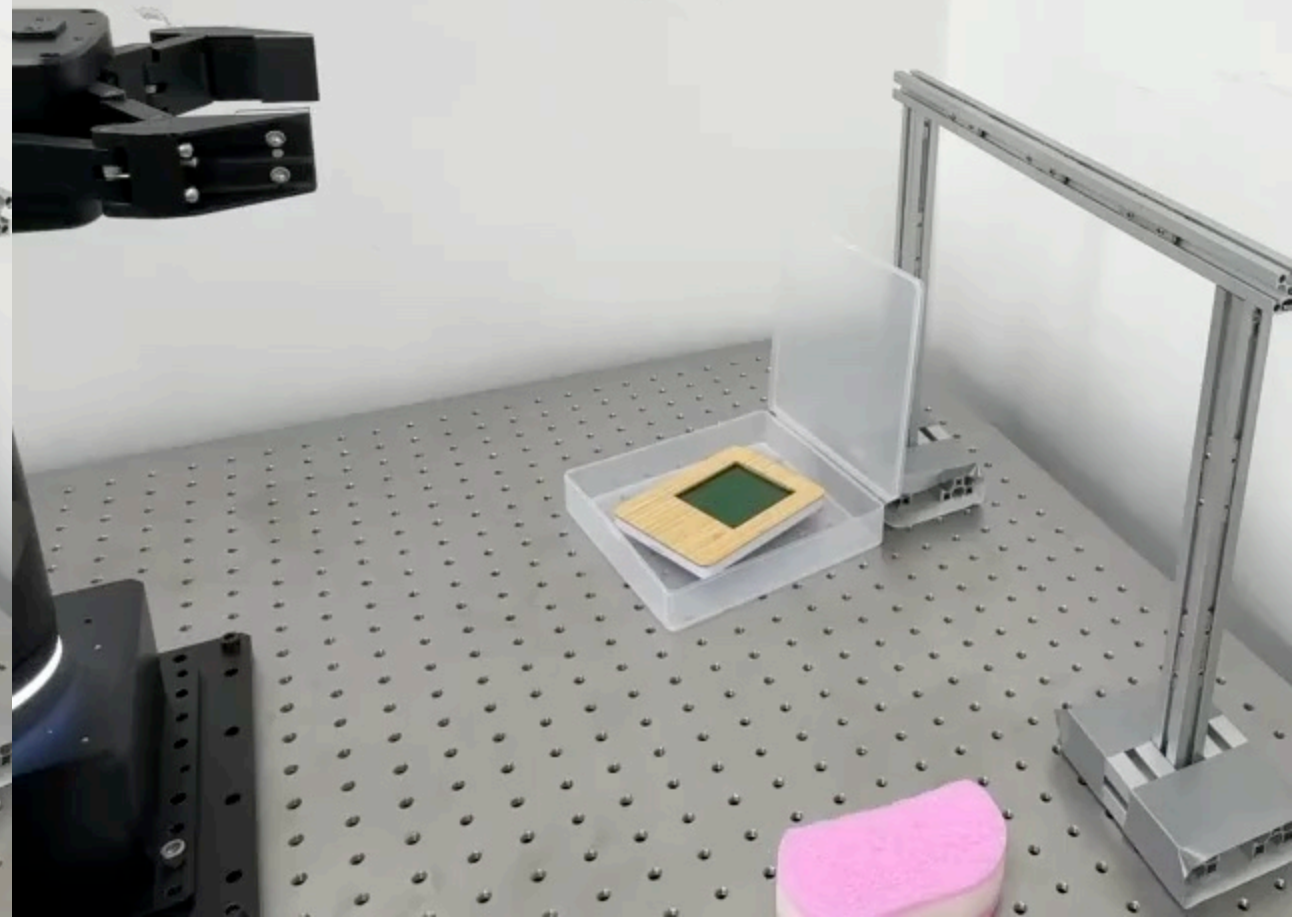


Language Instruction: Put sponge in cabinet and close it

π_0 (Baseline)



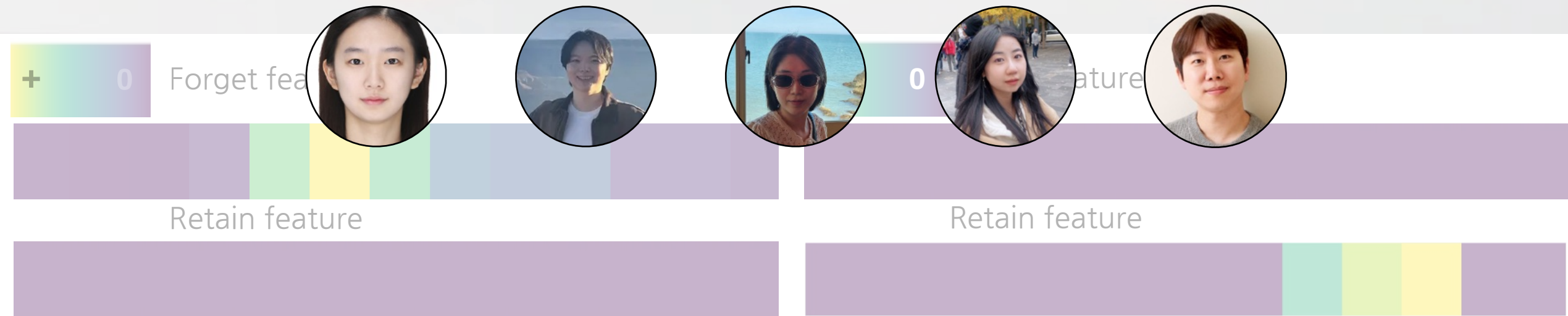
VINE (Ours)



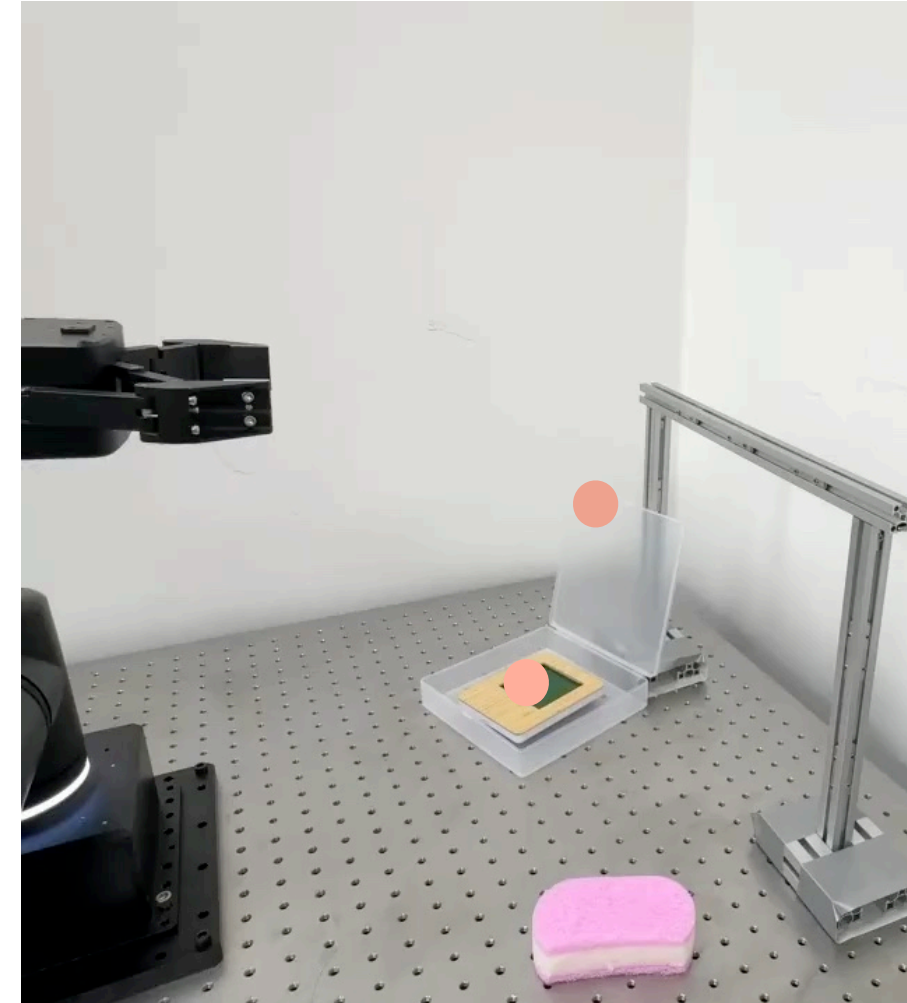
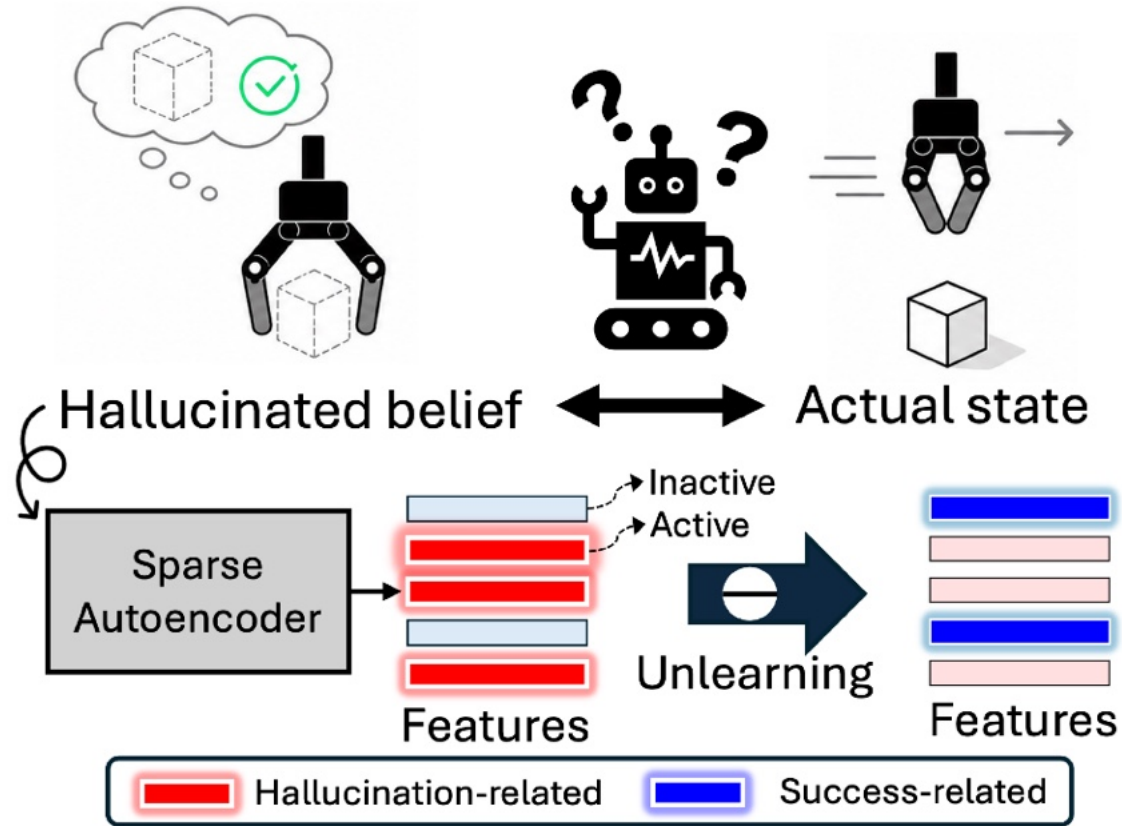
Key Findings

- System 2 was able to find feasible path with system 1 following those.
- Limitations
 - System 2 performs inference only once at the beginning
 - Future improvements could involve re-invoking System 2 at intermediate steps to allow dynamic re-planning or correction.

Sparse Feature Unlearning Mitigates State Hallucination in Vision-Language-Action Models

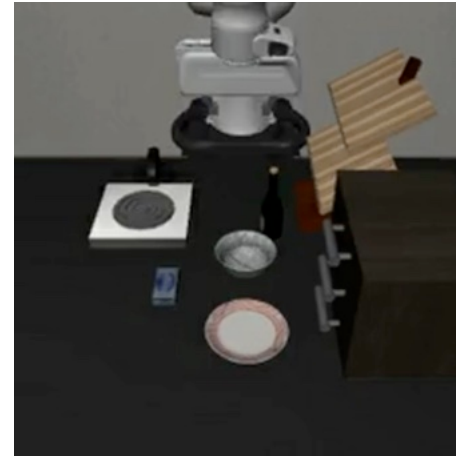
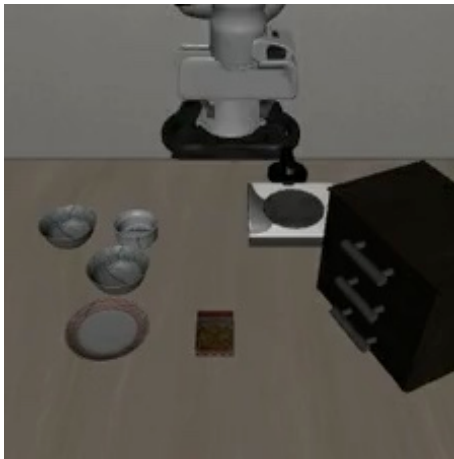


Motivation



Motivation

What triggers **false belief** in **VLA** and how can we mitigate it?



State Hallucinations



Grasp

Transport

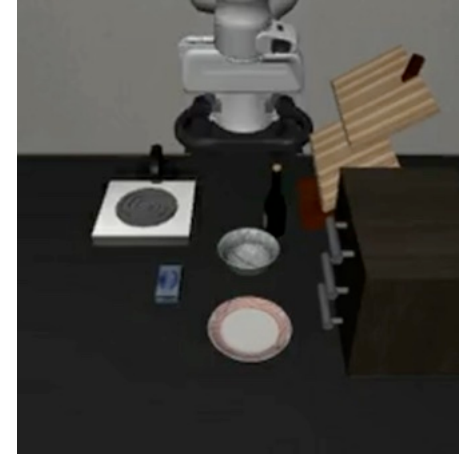
Place



Grasp

Transport

Place



Grasp

Transport

Place

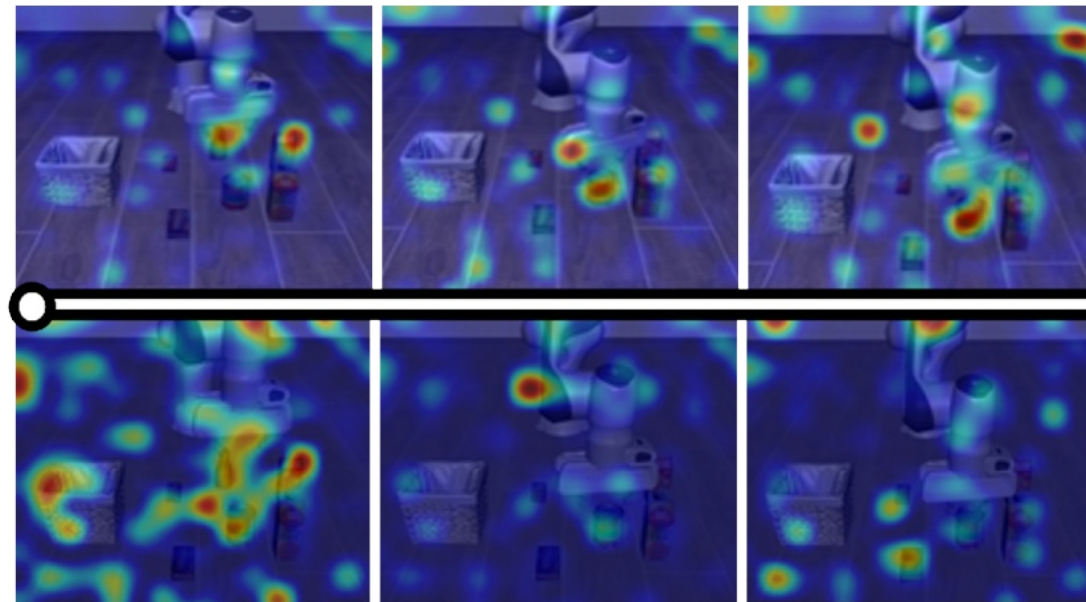
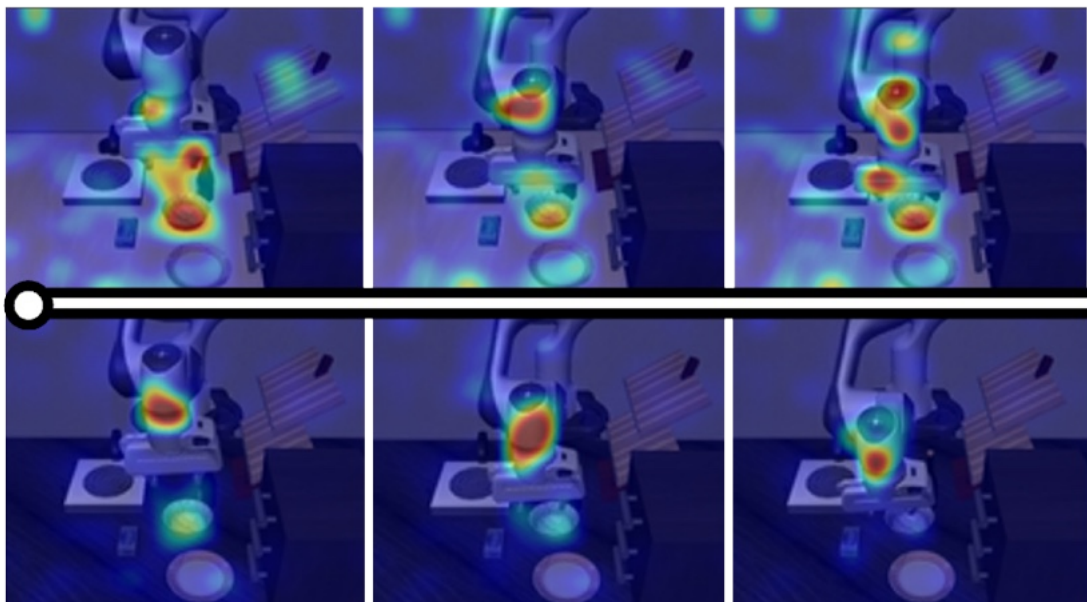
What triggers state hallucinations?

Gripper and target object

Background and target object

Success

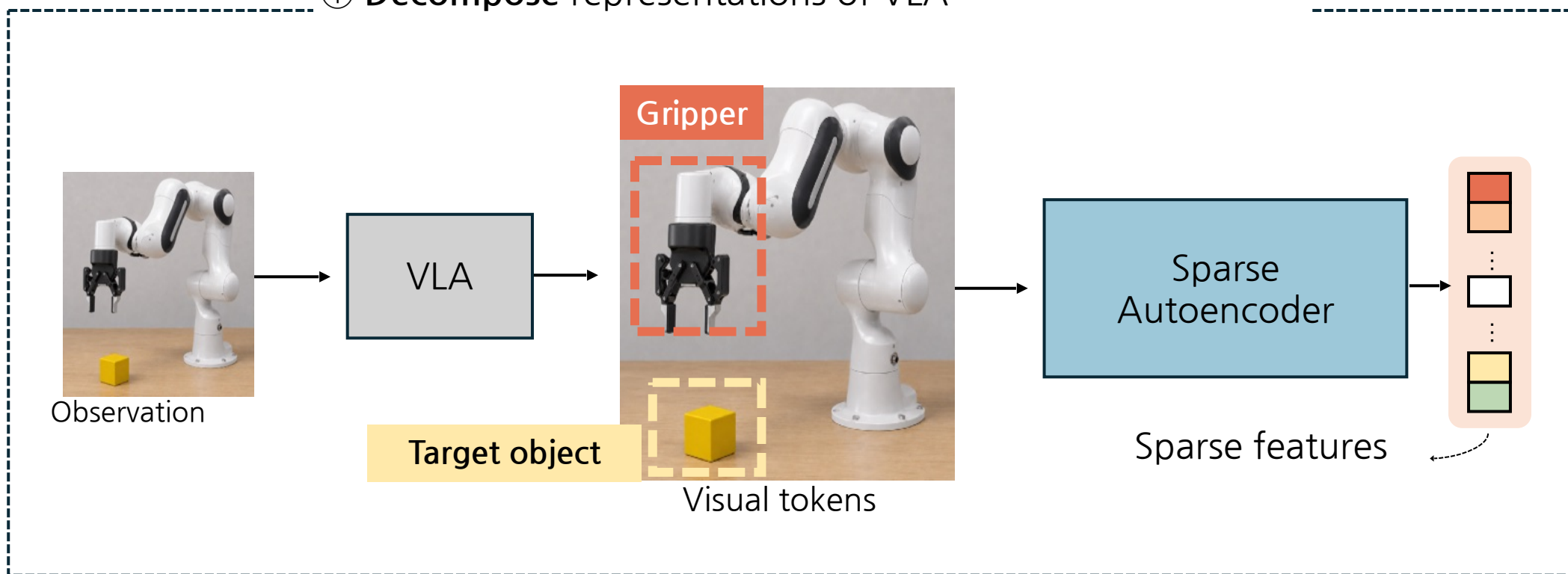
Hallucina
tions



What triggers state hallucination?

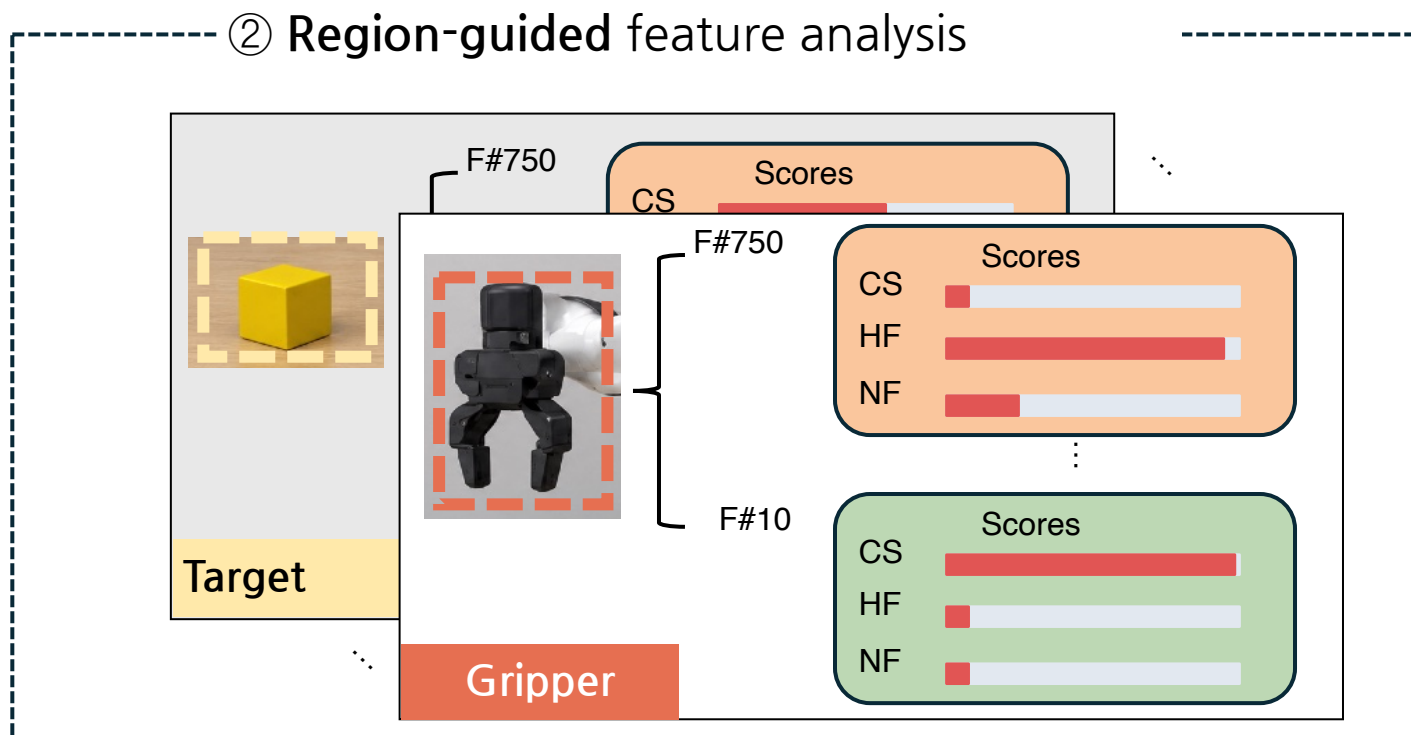
- Sparse AutoEncoder

① **Decompose** representations of VLA

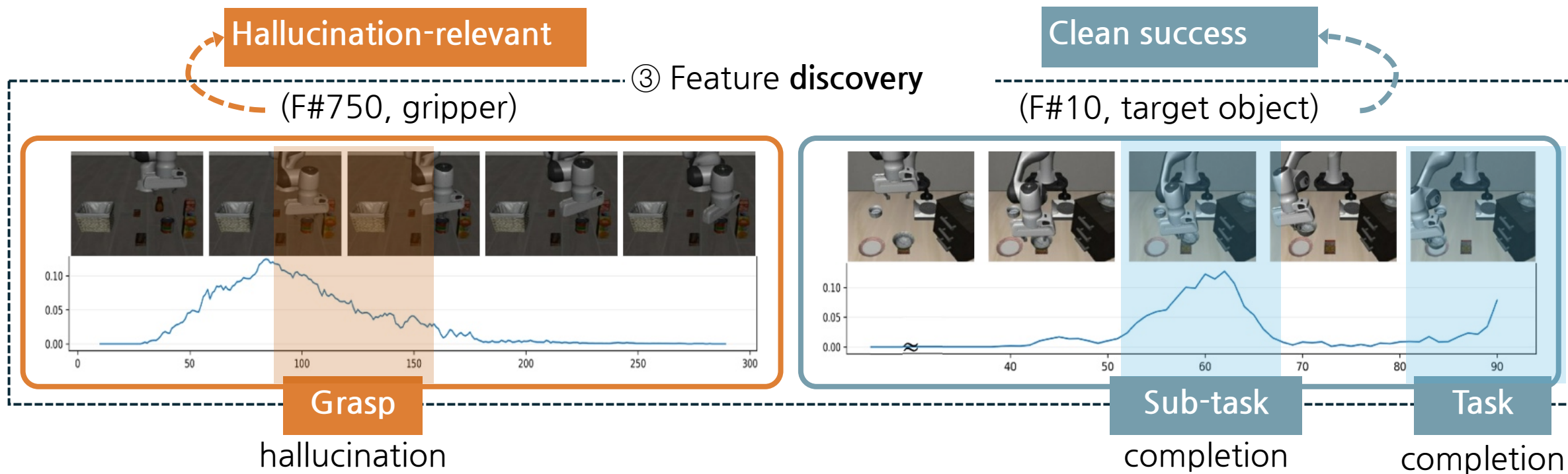


What triggers state hallucination?

- Sparse AutoEncoder



What triggers state hallucination?



Region-Guided Feature Policy Unlearning

④ Selected region-feature pairs



High in
hallucination!

Forget set

(F#2026, target object)



(F#750, gripper)



High in
Clean success!

Retain set

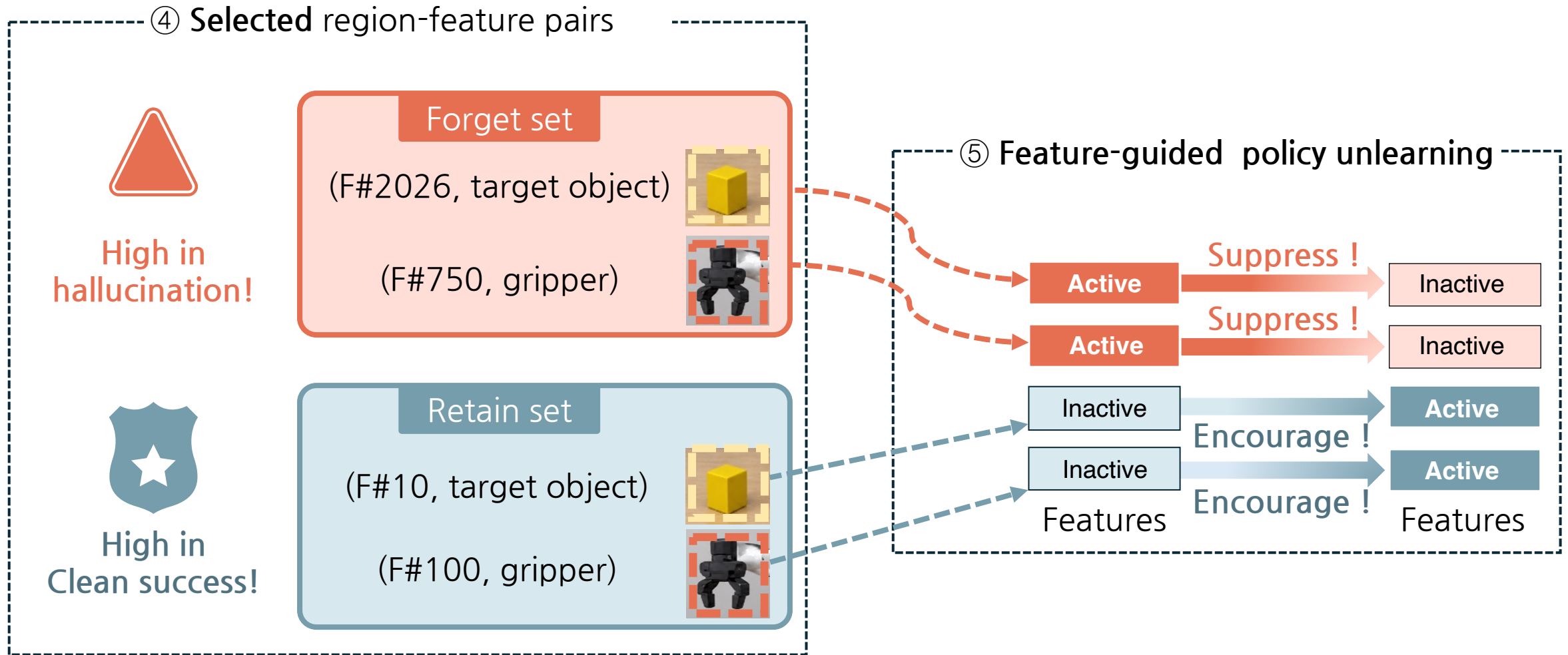
(F#10, target object)



(F#100, gripper)

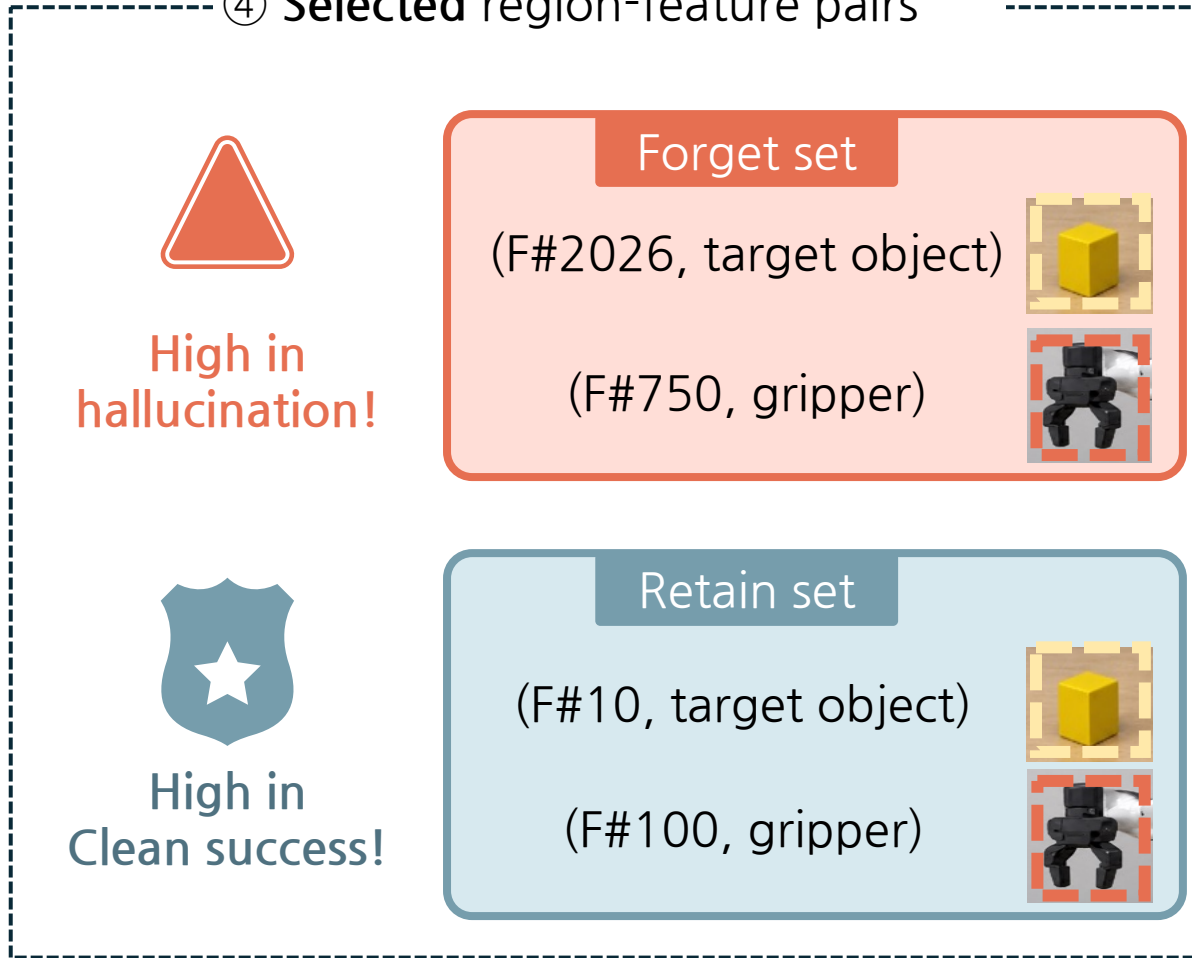


Region-Guided Feature Policy Unlearning

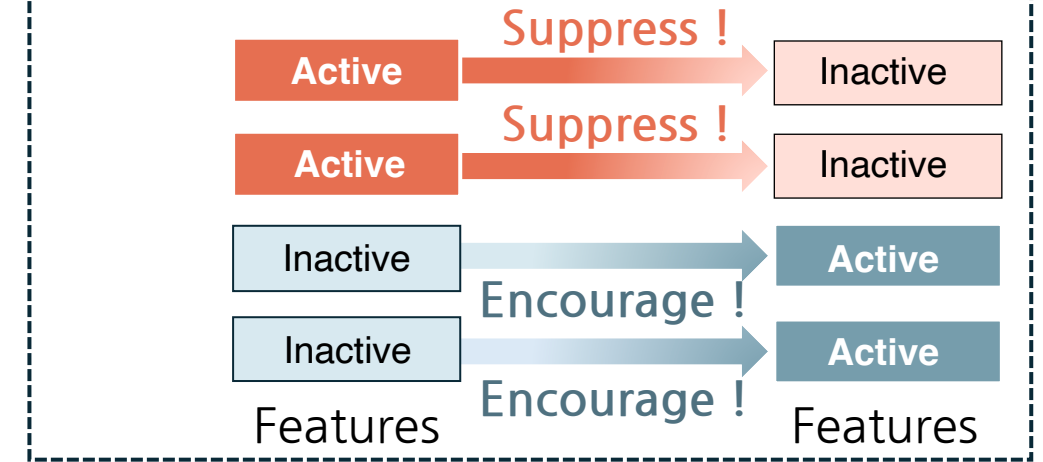


Region-Guided Feature Policy Unlearning

④ Selected region-feature pairs



⑤ Feature-guided policy unlearning



$$\mathcal{L}_{\text{forget}} = \mathbb{E}_{x \in \mathcal{D}_{\text{HF}}} \left[\frac{1}{|\mathcal{F}|} \sum_{(r,j) \in \mathcal{F}} \left(z_{r,j}(x; \theta) \right)^2 \right]$$

$$\mathcal{L}_{\text{retain}} = - \mathbb{E}_{x \in \mathcal{D}_{\text{CS}}} \left[\frac{1}{|\mathcal{G}|} \sum_{(r,j) \in \mathcal{G}} z_{r,j}(x; \theta) \right]$$

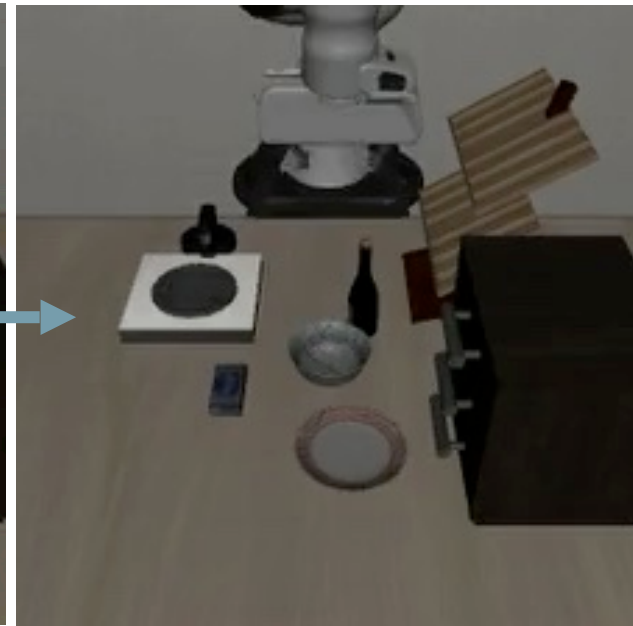
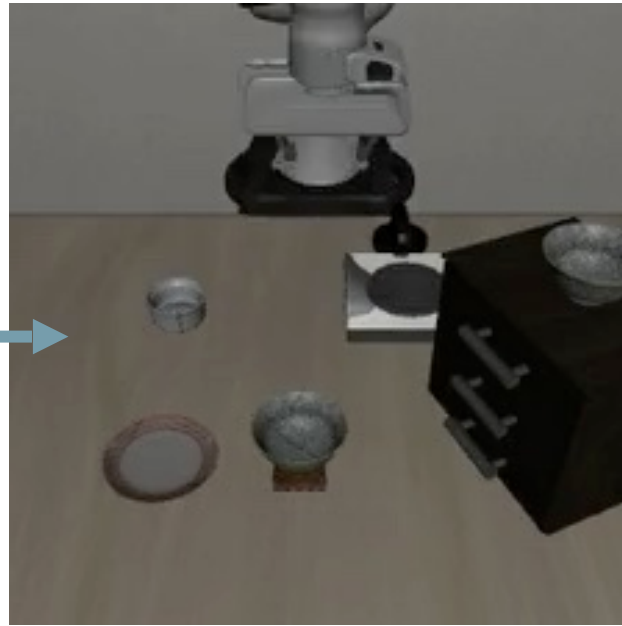
Experiments

Before unlearning

After unlearning

Before unlearning

After unlearning



Forget feature

Forget feature

Forget feature

Forget feature



Retain feature

Retain feature

Retain feature

Retain feature



Experiments

Before unlearning



After unlearning



Forget feature



Retain feature



Forget feature



Retain feature



Before unlearning



After unlearning



Forget feature



Retain feature



Forget feature



Retain feature



Key Findings

- VLA failures often come from a wrong internal belief about the robot-object state, and that belief is legible in the model's attention and sparse features.
- Because the failure has an internal signature, it can be edited out selectively, removing the behavior without erasing the manipulation skill.
- Limitations
 - Evaluated on pick-and-place only; long-horizon and bimanual settings remain open.
 - Requires semantic region partitioning and outcome labels that separate hallucination failures from ordinary ones.

Contents

01 Safety

Research Question
How can we avoid undesirable actions?

02 Efficiency

Research Question
How can model learn new tasks without training?
Can we build efficient representation for policy learning?

Contents

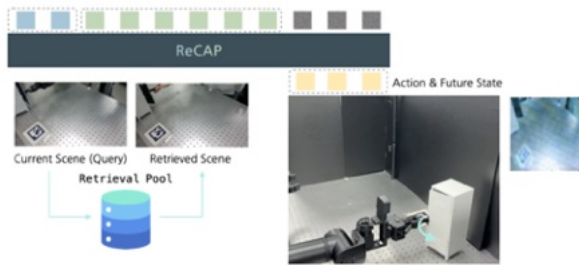
02 Efficiency

Research Question

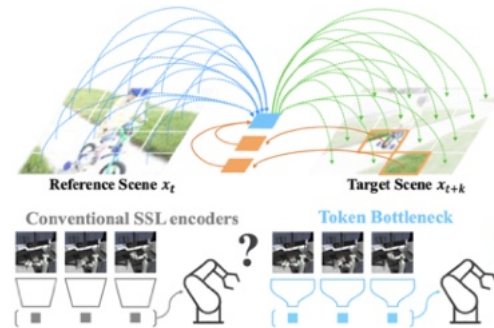
How can model learn new tasks without training?

Can we build efficient representation for policy learning?

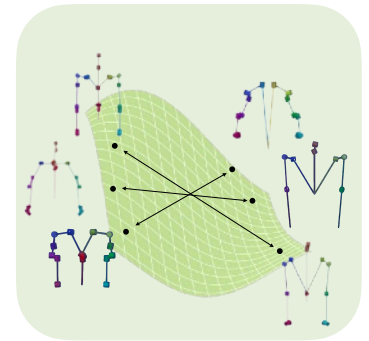
A Test Time Retrieval



B Visual Representation



C Structure Representation



Retrieve, Don't Retrain: Extending Vision-Language-Action Models to New Tasks at Test Time

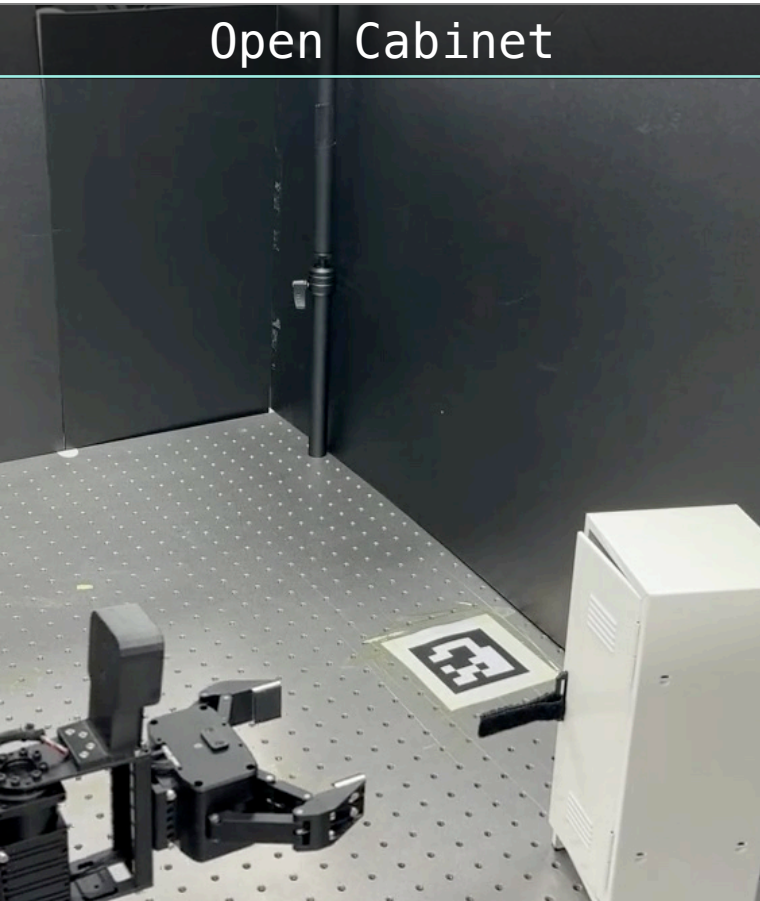
Retrieval Pool



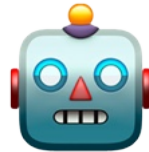
Retrieval Scene



Motivation

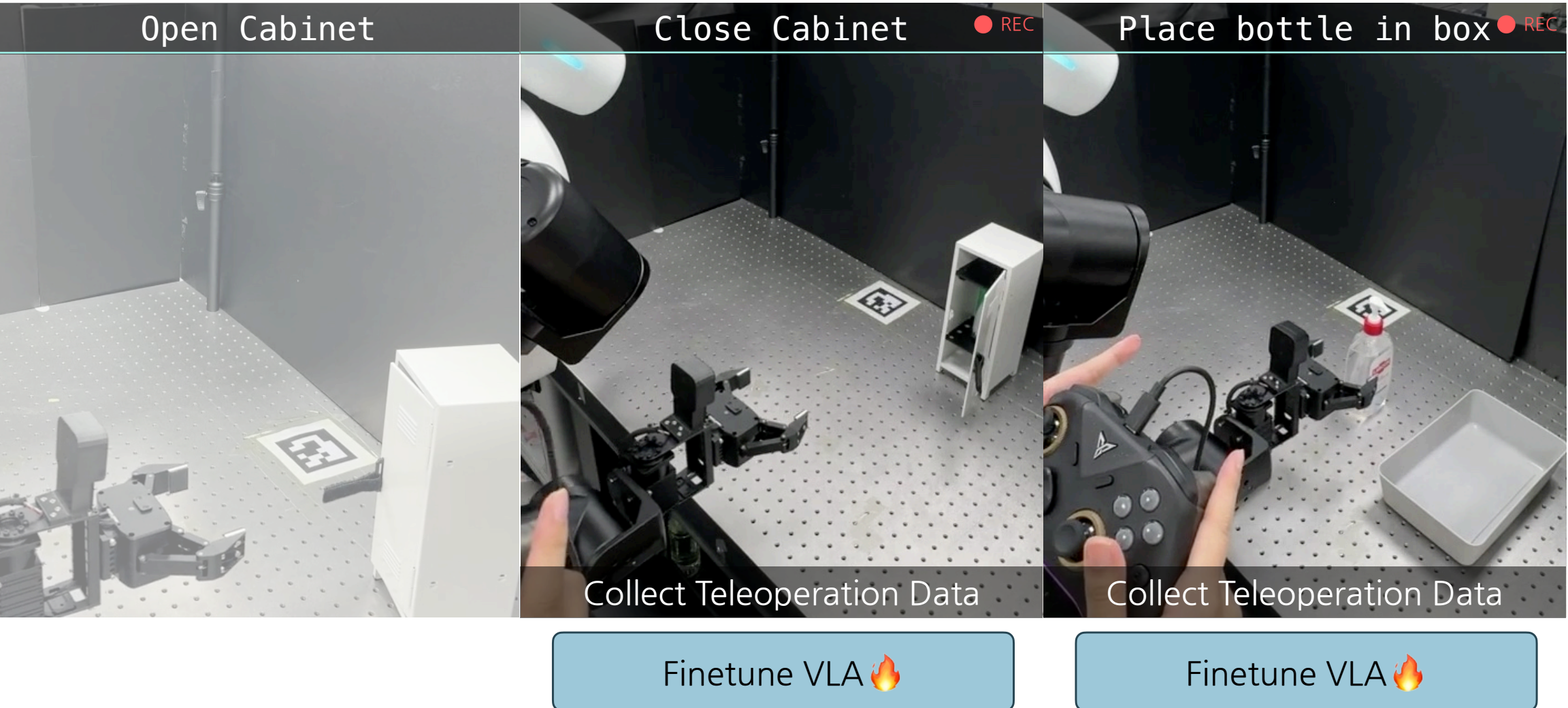


Okay, Let's make robot do "close cabinet"



Collect new teleoperation data and fine-tune policy.

Motivation



Motivation



Training: ~24
gpu-hours/task

Finetune VLA 🔥

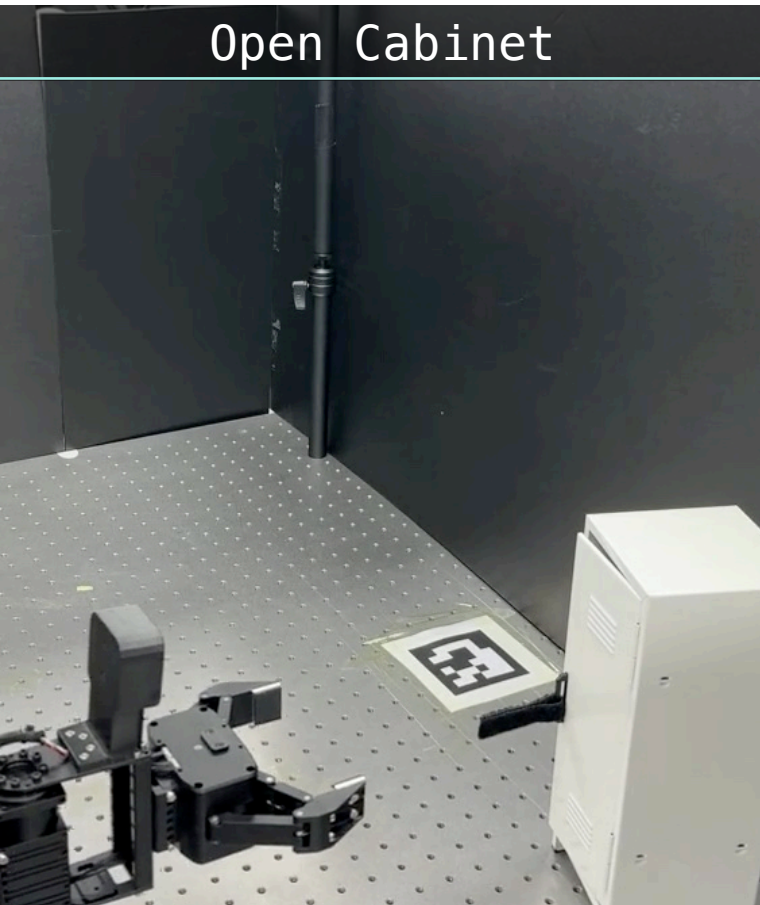


No fine-tuning,
Preprocessing: ~10 mins

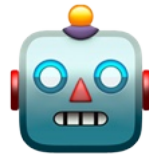
Condition Context

Motivation

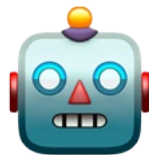
Can we build VLA that can perform new tasks in test time by just **conditioning the demo?**



Okay, Let's make robot do "close cabinet"

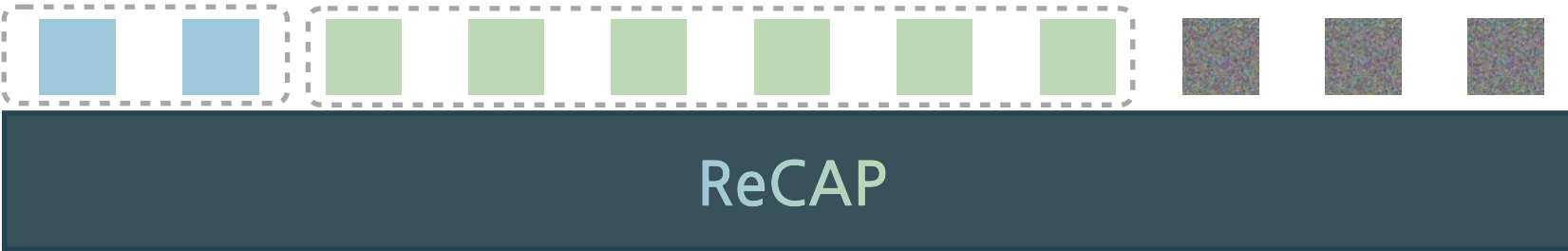


Collect **new teleoperation data and fine-tune** policy.



Freeze policy, and add **cheap embodiment data to pool**, then retrieve

Proposed Method



Action & Future State

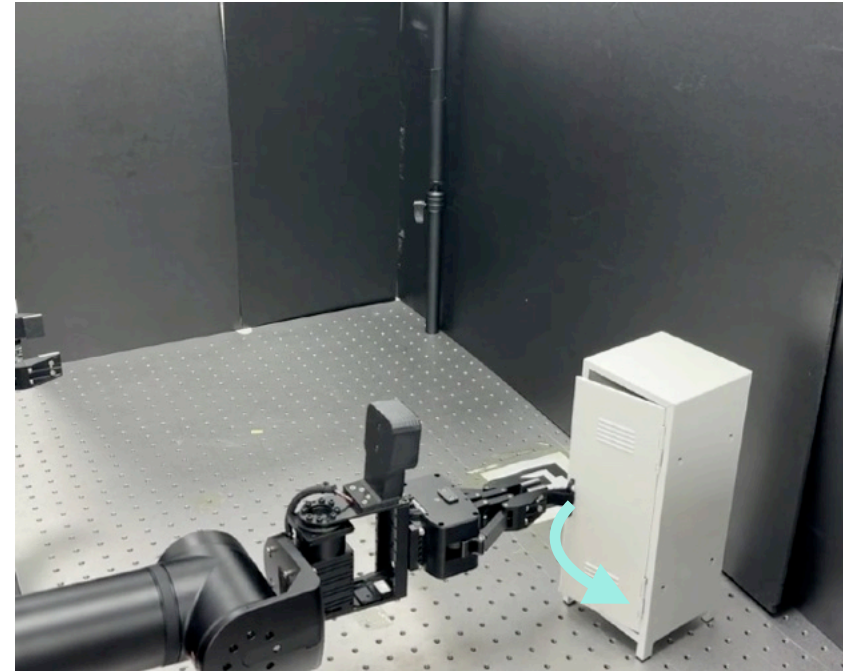
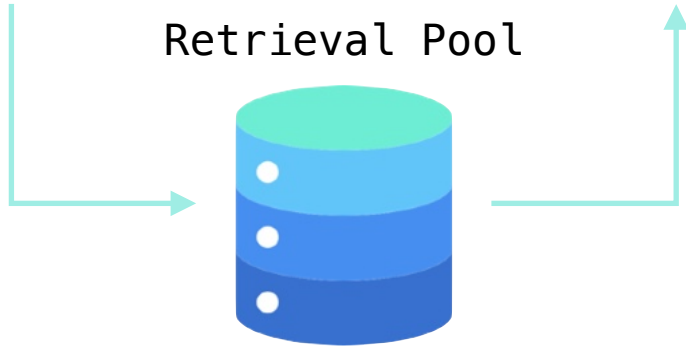


Current Scene (Query)



Retrieved Scene

Retrieval Pool



Proposed Method



Train Policy 🔥

Proposed Method

Open Cabinet

Close Cabinet

REC

Append to pool

Retrieval Pool

ReCAP



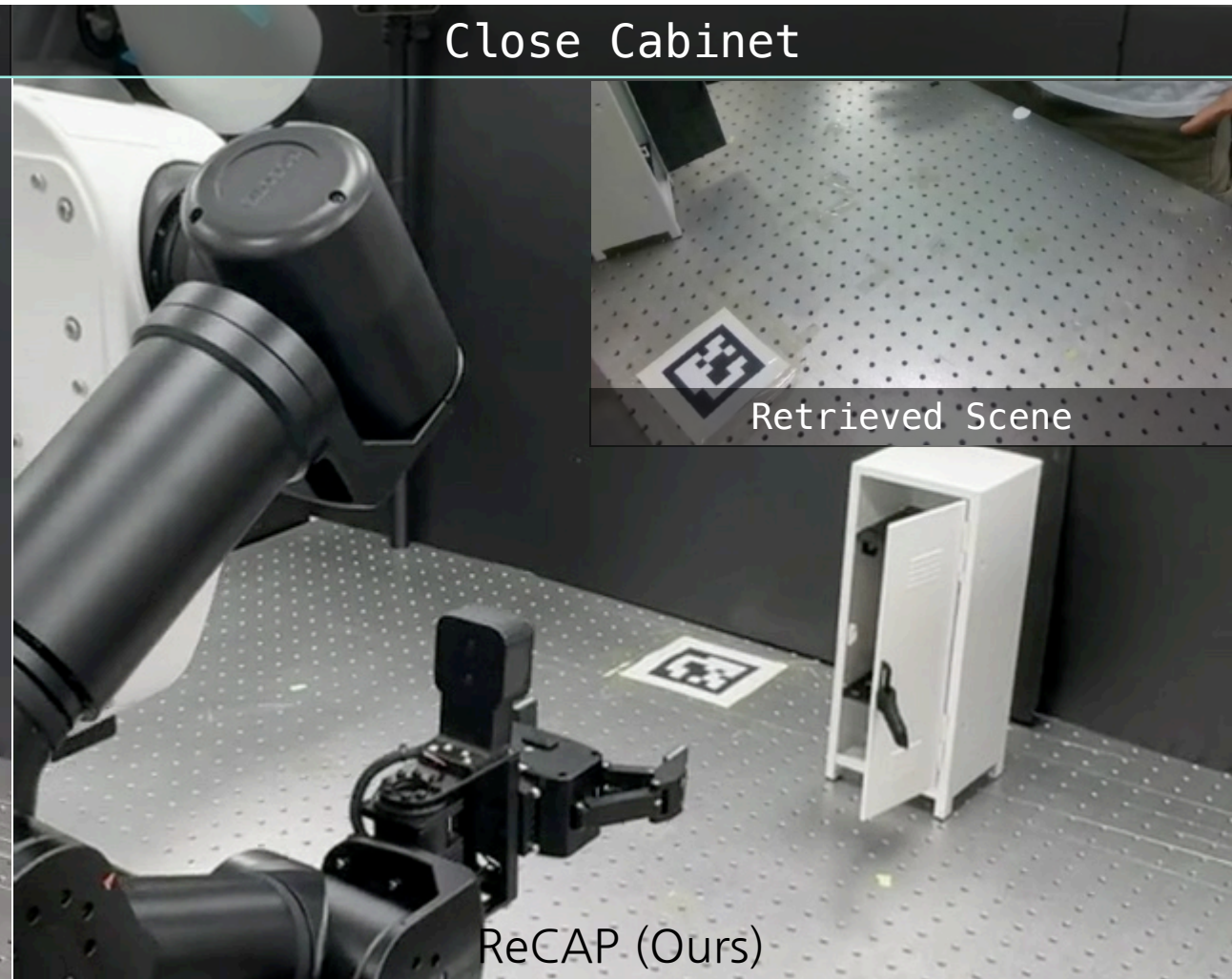
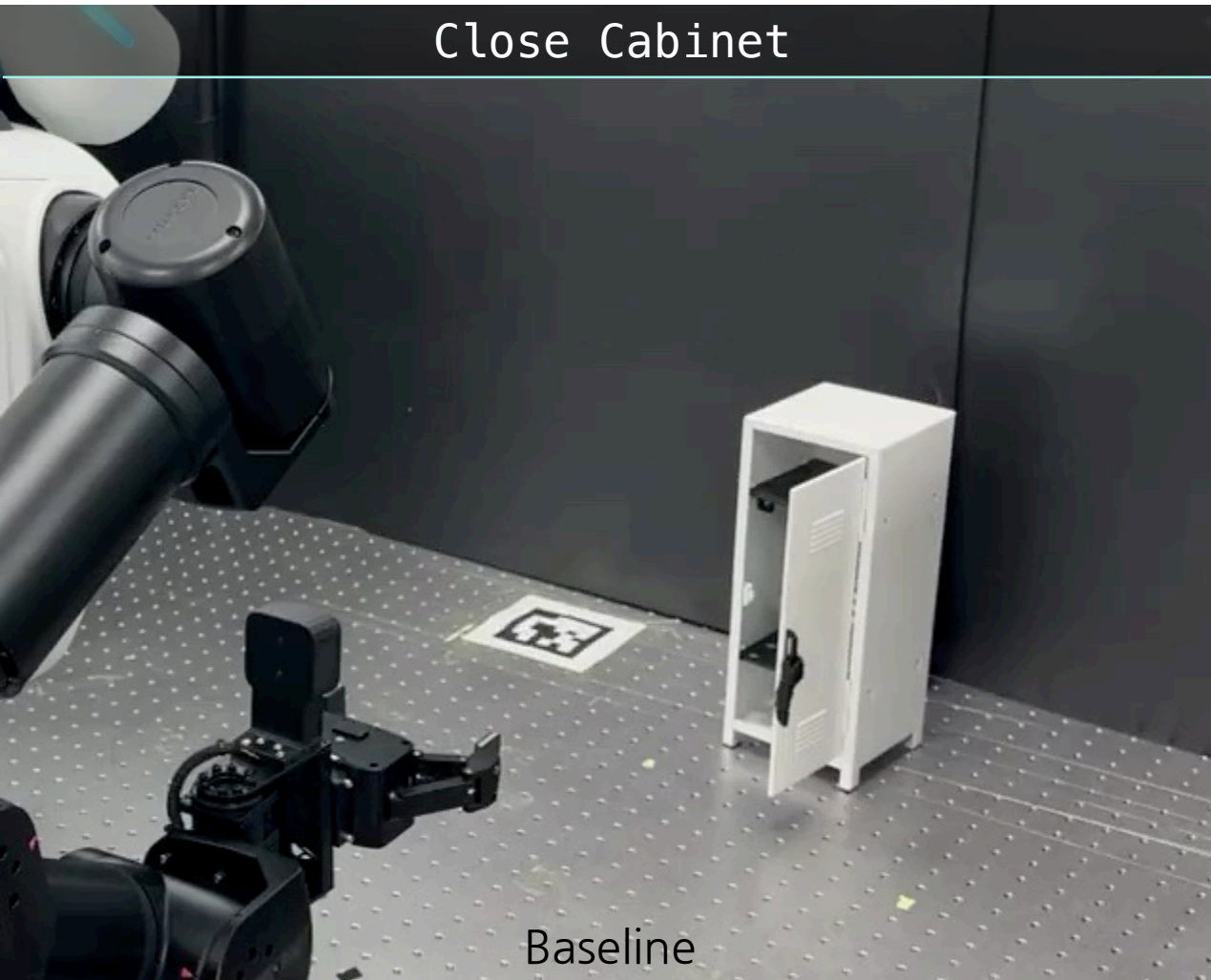
ReCAP

Collect Human Data

Experiments



Experiments

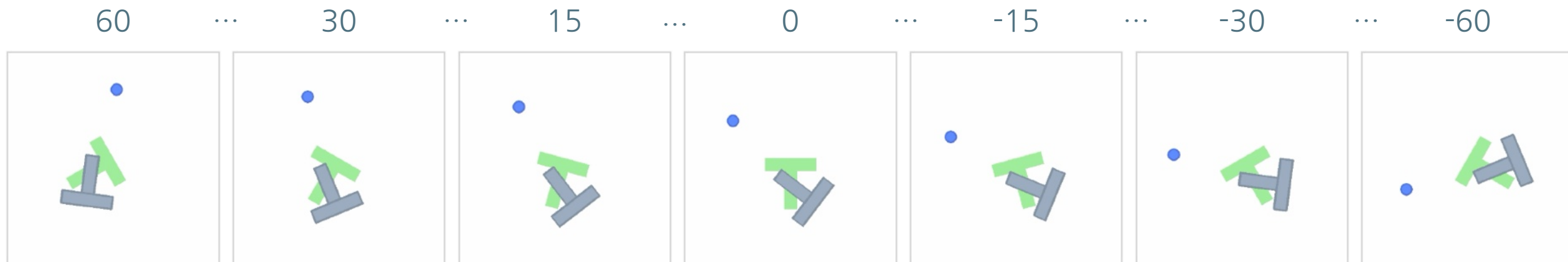


Experiments

Training time database



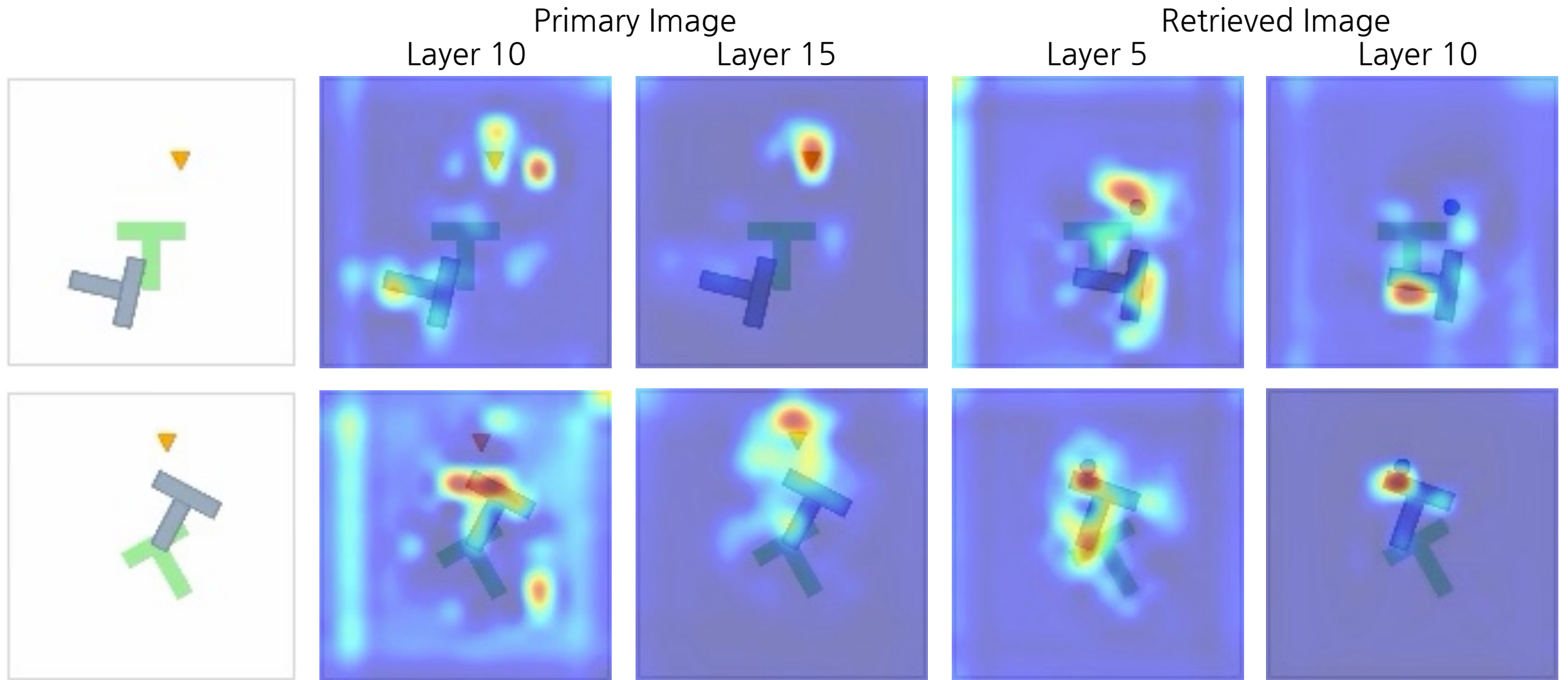
Test time database



Experiments

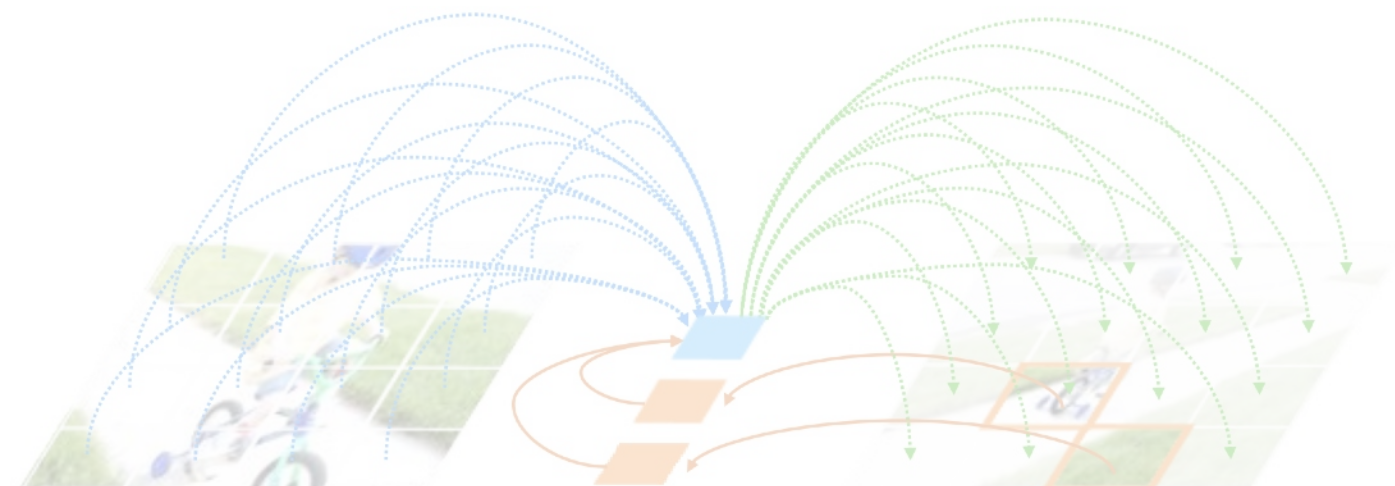
Observation	Generated	Retrieved	Observation	Generated	Retrieved
					
Goal Angle: 0			Goal Angle: -15		
					
Goal Angle: 30			Goal Angle: -60		

Experiments



Key Findings

- New tasks are added by indexing data, not updating parameters. The policy stays frozen.
- Retrieval works because of the world-action backbone. Future-image prediction keeps the retrieved trajectory consistent with the scene.
- Limitations
 - Coverage is bounded by the pool.
 - Training still needs paired query-pool demonstrations.
 -

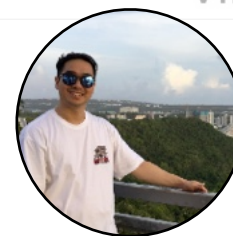
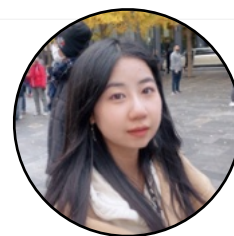
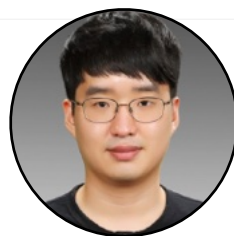
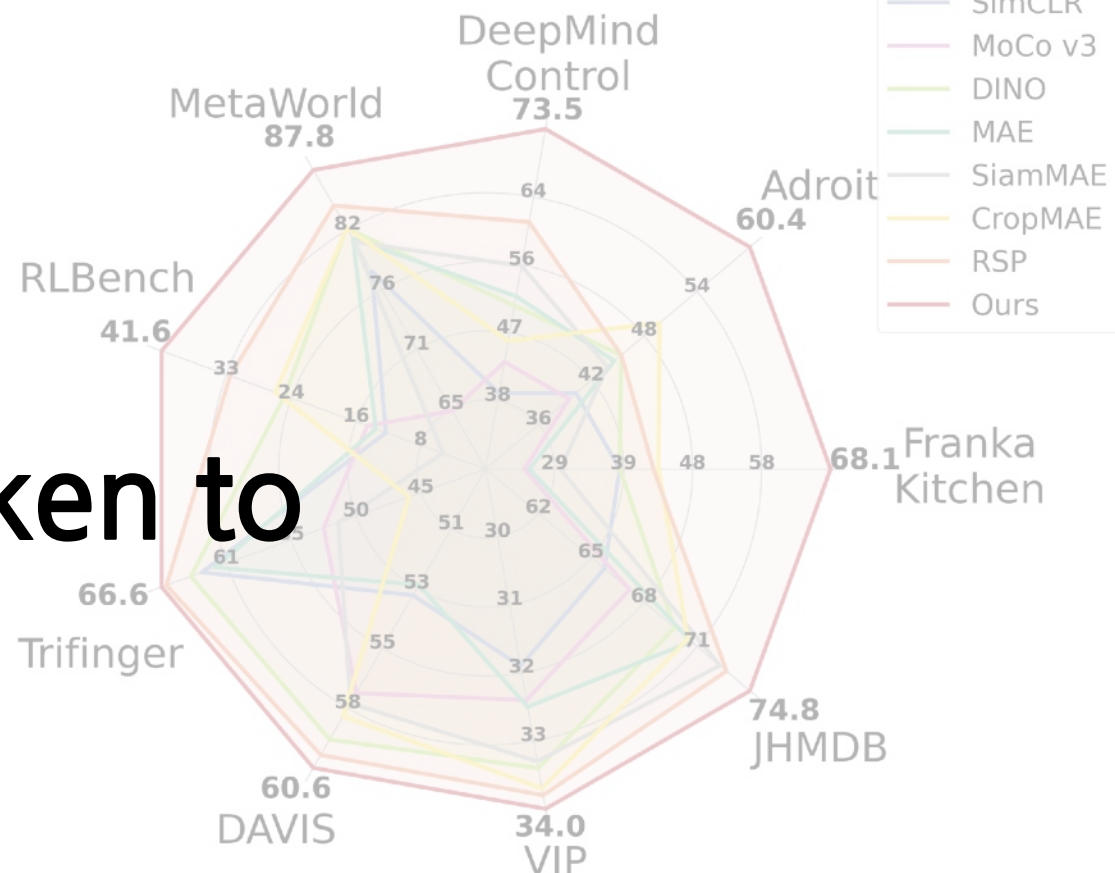


Token Bottleneck: One Token to Remember Dynamics

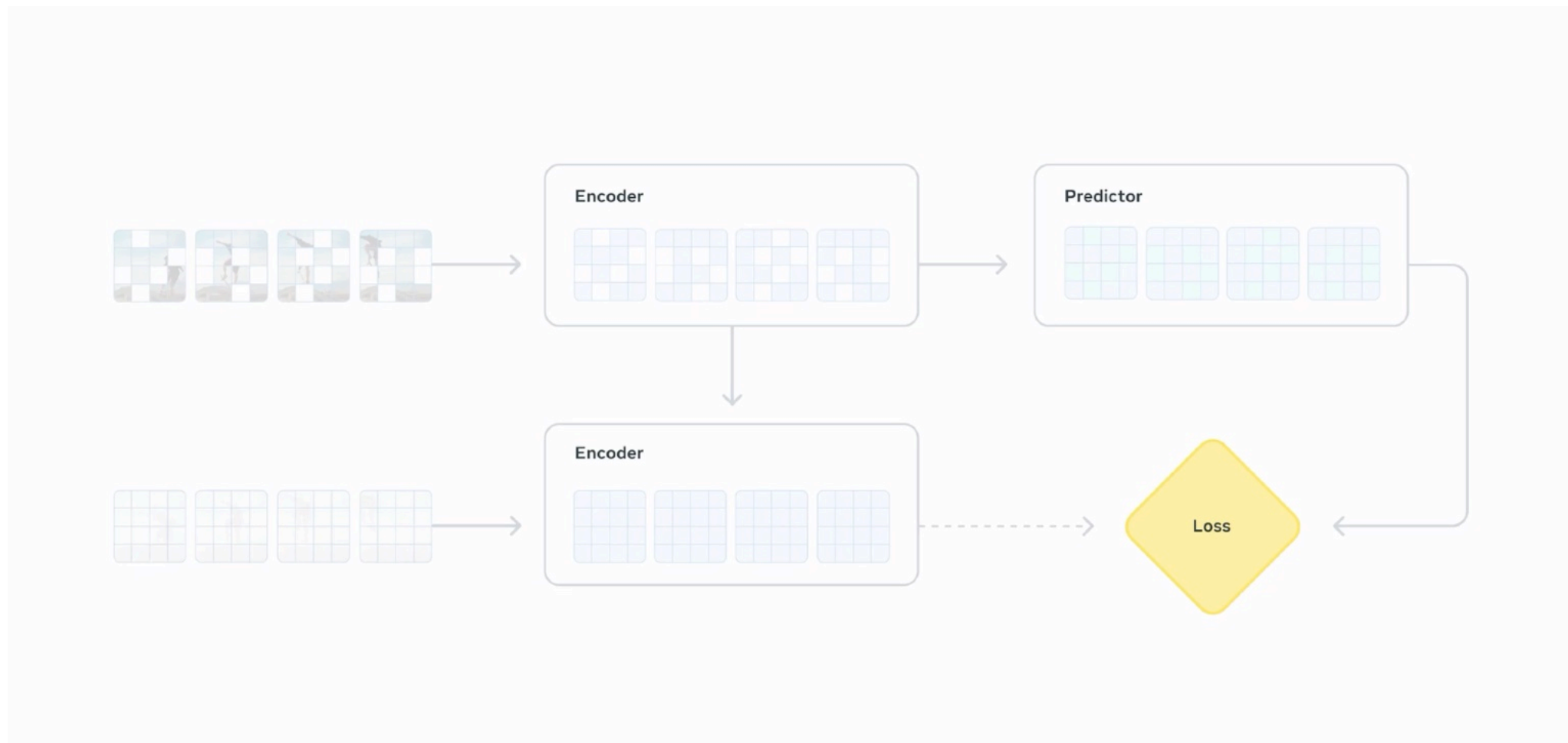
Conventional SSL encoders



Token Bottleneck

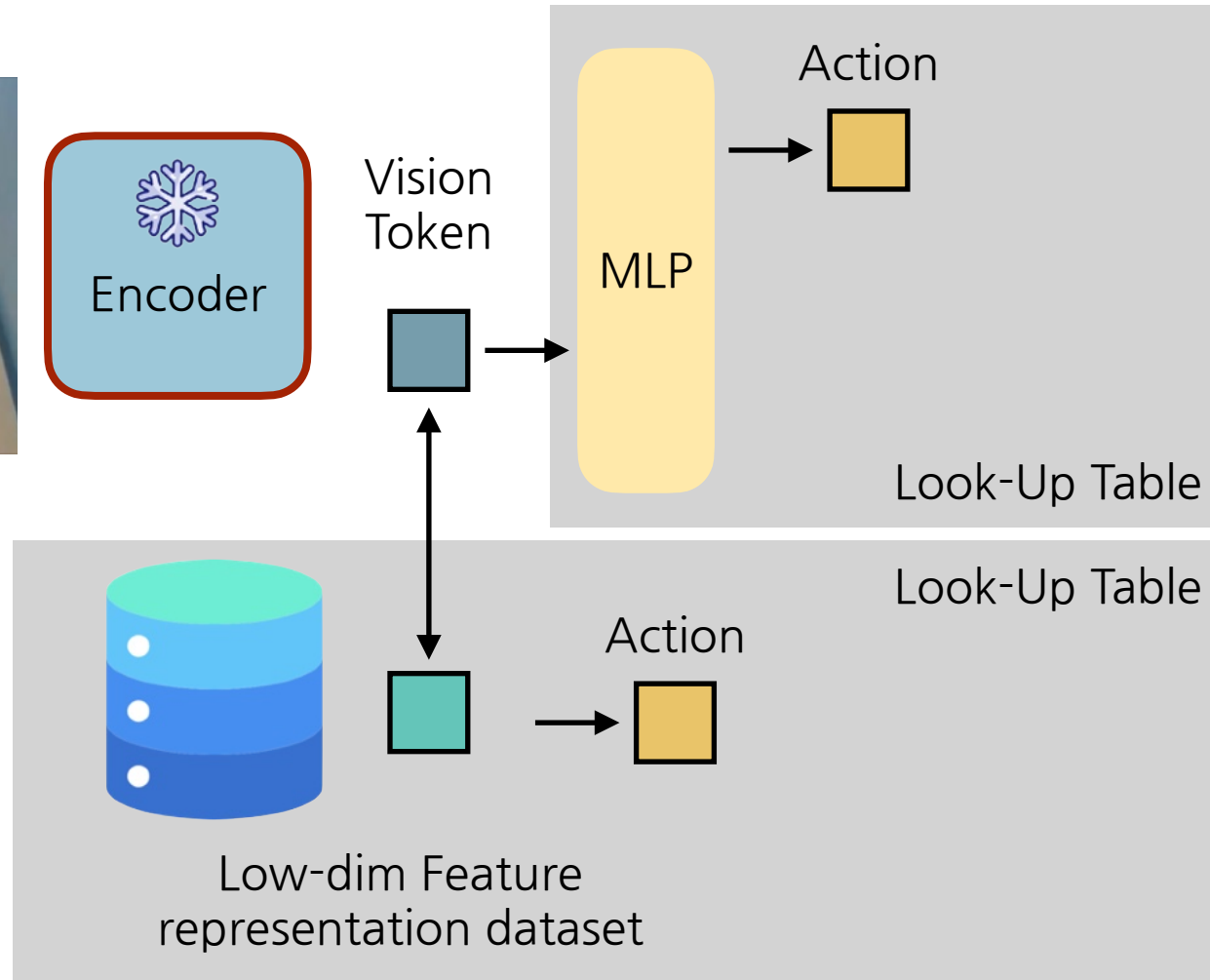


Visual Representation Learning



<https://ai.meta.com/research/vjepa/>

Motivation



Motivation

Is visual representation trained via “world modeling” better than training in static scenes?



Encoder

Vision
Token

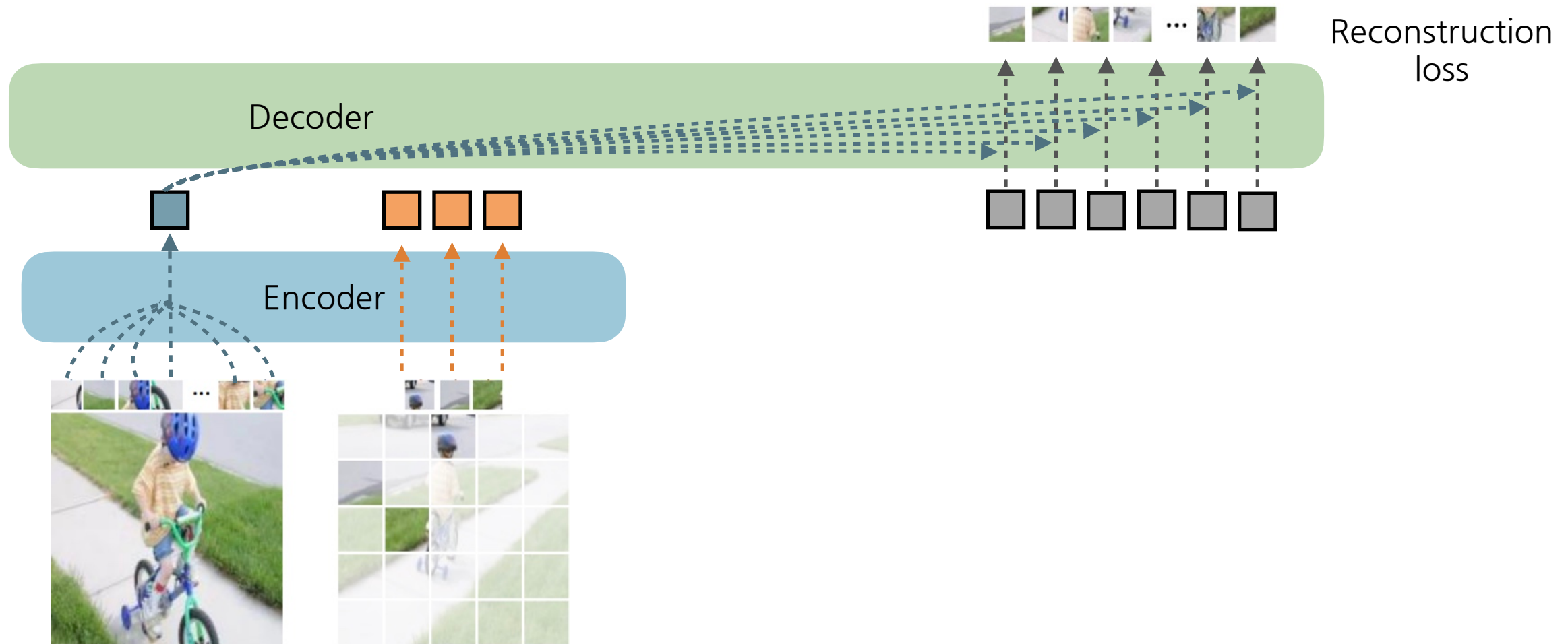


Condition
Token

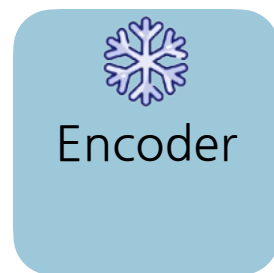
Decoder



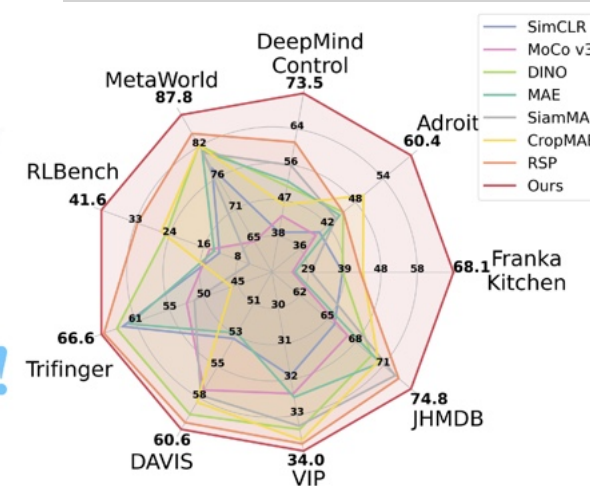
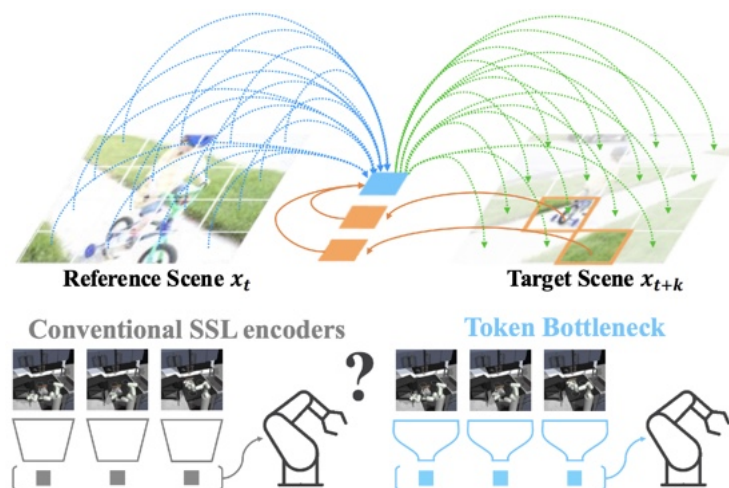
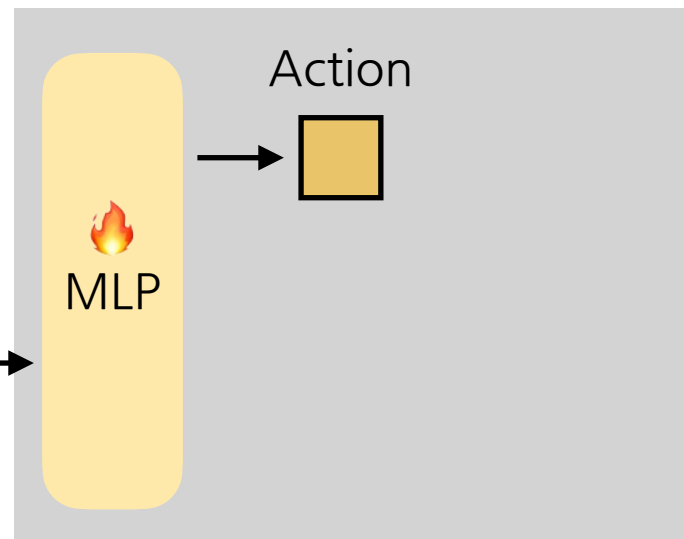
Proposed Method



Downstream Tasks



Vision
Token



Experiments

Table 1: **Experimental results on vision-based robot policy learning on Franka Kitchen.** We report the performance of imitation learning agents on Franka Kitchen [15], which are trained upon representations from the ViT-S/16 model pre-trained on Kinetics-400 [26] dataset. The success rates (%) are reported for all the tasks. We underline the second-best performance. We report the gains of our method over the second-best baseline.

Tasks	SimCLR	MoCo v3	DINO	MAE	SiamMAE	RSP	CropMAE	ToBo
Knob1 on	25.3 $\pm$ 2.1	11.5 $\pm$ 3.9	27.0 $\pm$ 3.2	12.0 $\pm$ 3.3	16.8 $\pm$ 4.4	31.0 $\pm$ 2.4	<u>31.5</u> $\pm$ 5.3	57.0 $\pm$ 2.0
Light on	<u>55.8</u> $\pm$ 6.4	24.3 $\pm$ 5.0	44.3 $\pm$ 6.5	24.3 $\pm$ 4.2	36.5 $\pm$ 7.0	44.5 $\pm$ 5.6	54.0 $\pm$ 11.2	82.0 $\pm$ 1.6
Sdoor open	72.3 $\pm$ 2.8	66.5 $\pm$ 3.2	77.0 $\pm$ 5.0	71.5 $\pm$ 4.3	68.0 $\pm$ 7.9	<u>82.5</u> $\pm$ 2.7	77.0 $\pm$ 8.1	95.0 $\pm$ 7.1
Ldoor open	17.0 $\pm$ 2.9	10.3 $\pm$ 2.1	16.5 $\pm$ 2.5	12.8 $\pm$ 3.9	17.3 $\pm$ 3.7	<u>28.8</u> $\pm$ 4.8	25.5 $\pm$ 5.7	51.0 $\pm$ 1.4
Micro open	23.3 $\pm$ 2.8	14.3 $\pm$ 2.5	28.5 $\pm$ 4.8	10.0 $\pm$ 2.8	13.5 $\pm$ 4.8	30.3 $\pm$ 5.6	<u>32.5</u> $\pm$ 4.1	55.0 $\pm$ 1.4

Experiments

Table 6: **Comparison with robot representation learning models.** We compare the performance of our method with robot representation learning methods across multiple simulated manipulation tasks. We categorize the methods into self-supervised learning, supervised learning through foundation models, and supervised learning with auxiliary language guidance. Despite the unbalanced training and evaluation setup, ToBo surpasses the RRL models on MetaWorld. Moreover, ToBo exceeds self-supervised RRL models despite of a smaller model with a significantly smaller amount of data. These results demonstrate the effectiveness of the representations learned by ToBo in diverse robot manipulation tasks.

Method	#Param	Dataset	#Seen frames	Adroit	MetaWorld	Franka Kitchen
<i>Supervision through Foundation Models</i>						
Theia [†]	52.9M	Theia dataset	14.4B [*]	66.0	86.1	-
<i>Supervision with Auxiliary Language Guidance</i>						
R3M	25.6M	Ego4D	0.8B	65.0	69.2	53.1
MVP	21.7M	MVP dataset	4.8B	-	84.6	-
Voltron [‡]	21.7M	SS-v2	0.3B	-	68.7	70.5
MPI [‡]	21.7M	Ego4D	0.1B	-	85.7	76.5
<i>Self-supervised Learning</i>						
R3M [°]	25.6M	Ego4D	0.8B	45.6	67.0	47.2
data4robotics	86.0M	Kinetics-700	0.5B	-	87.0	55.0
VC-1	86.0M	Ego4D+N	1.0B	50.0	86.4	-
ToBo (ours)	21.7M	Kinetics-400	0.2B	60.4	87.8	68.0

[†] Uses additional compression layers. [‡] Uses multi-head attention pooling layers for integrating spatial tokens.

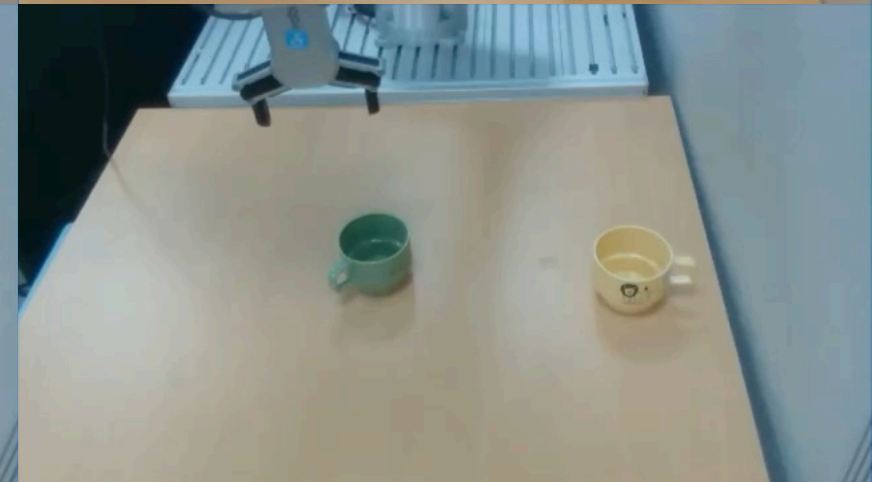
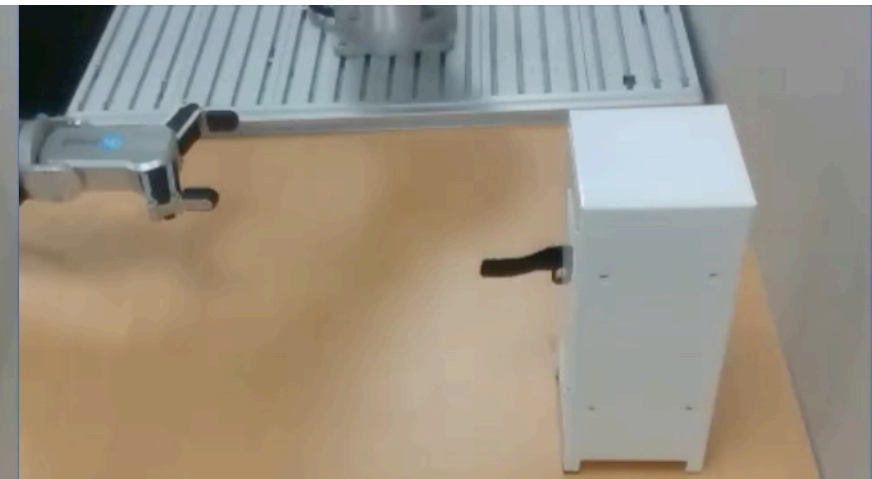
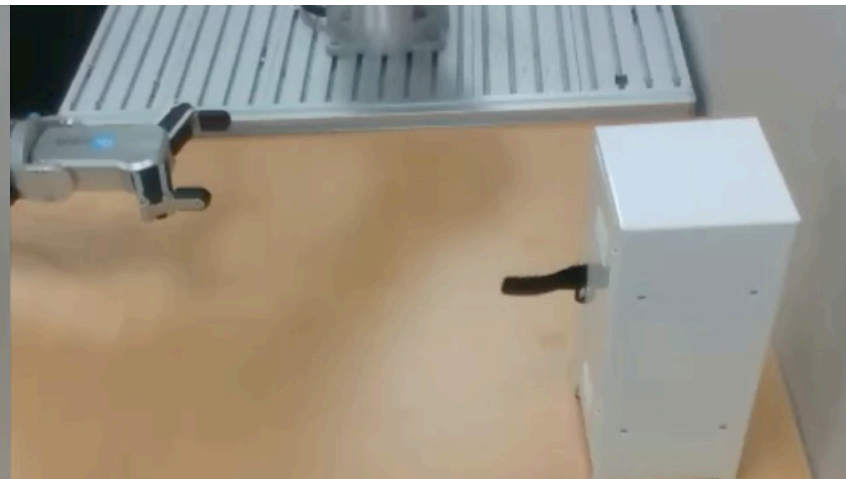
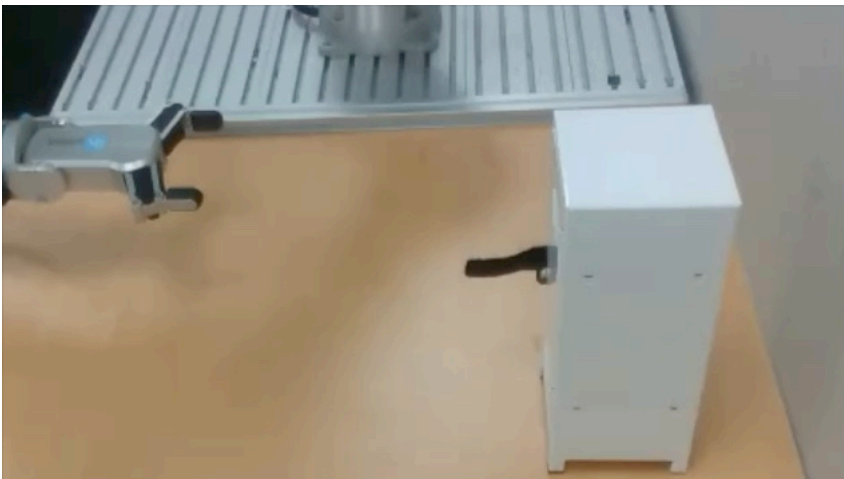
[°] Excludes language guidance from the vanilla recipe. ^{*} Includes data for distillation models.

Experiments

SiamMAE

RSP

Tobo (Ours)



Key Findings

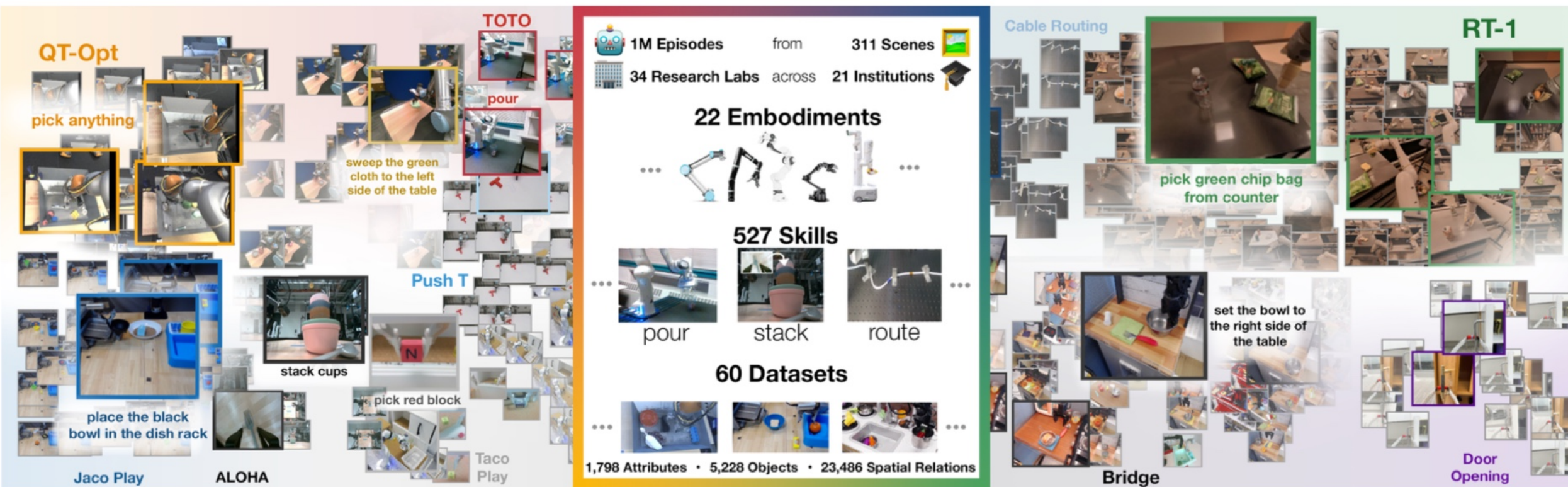
- Predicting what a scene becomes yields better representations than encoding what it is. Temporal structure ends up in the representation itself.
- The bottleneck is what forces this. With one token and a few hint patches, the encoder cannot copy appearance and must keep what carries over.
- Limitations
 - Supervision is frame-to-frame, so the learned structure is short-horizon.
 - Gains are shown on sequential tasks; the benefit where static appearance suffices is untested.



LEGO: Latent-space Exploration for Geometry-aware Optimization of Humanoid Kinematic Design

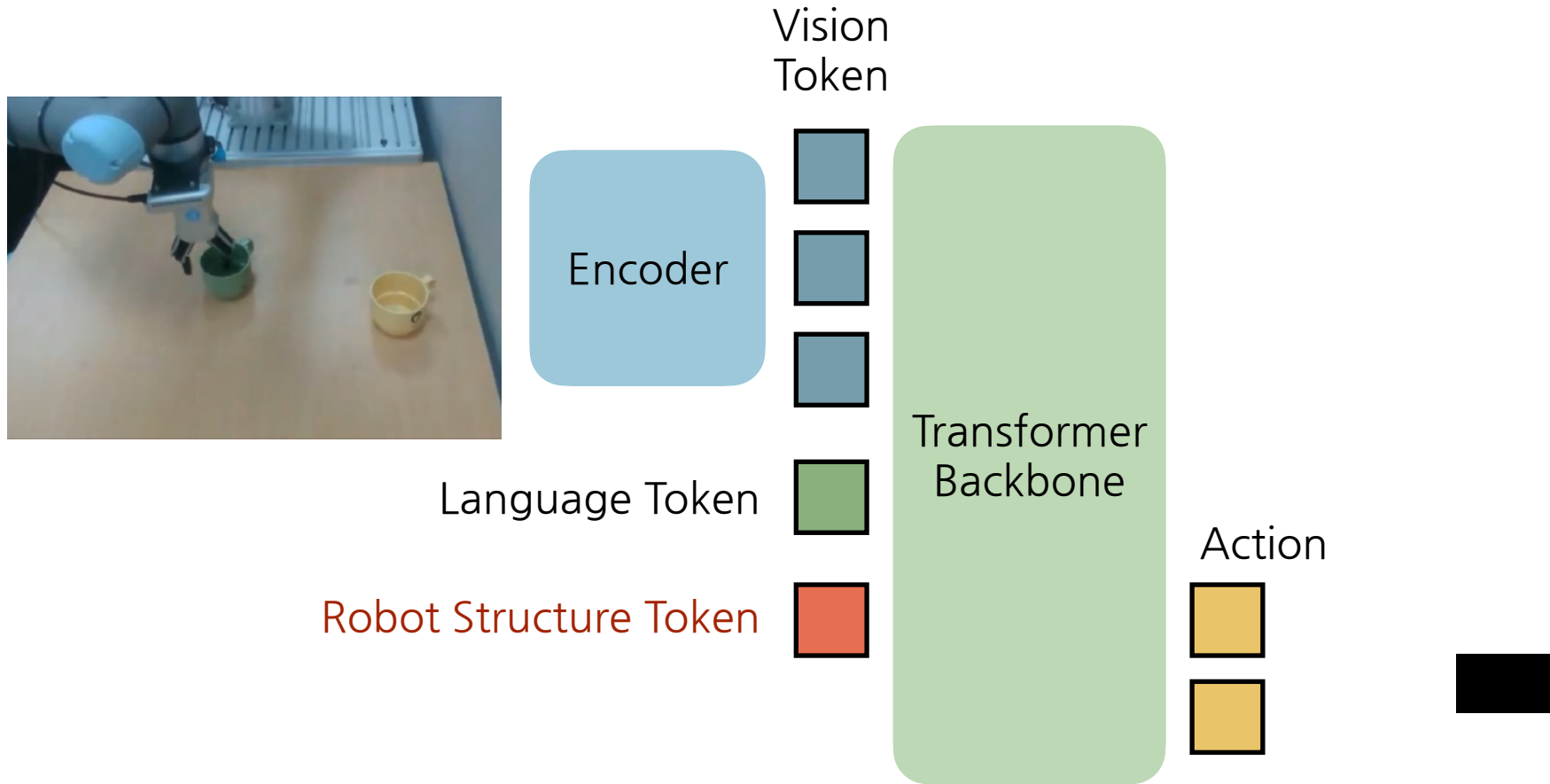


Beyond Visual Representation



<https://robotics-transformer-x.github.io/>

Beyond Visual Representation

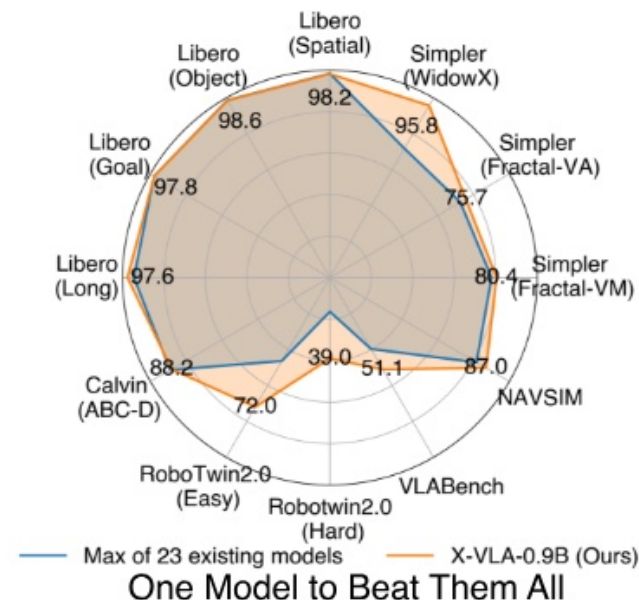
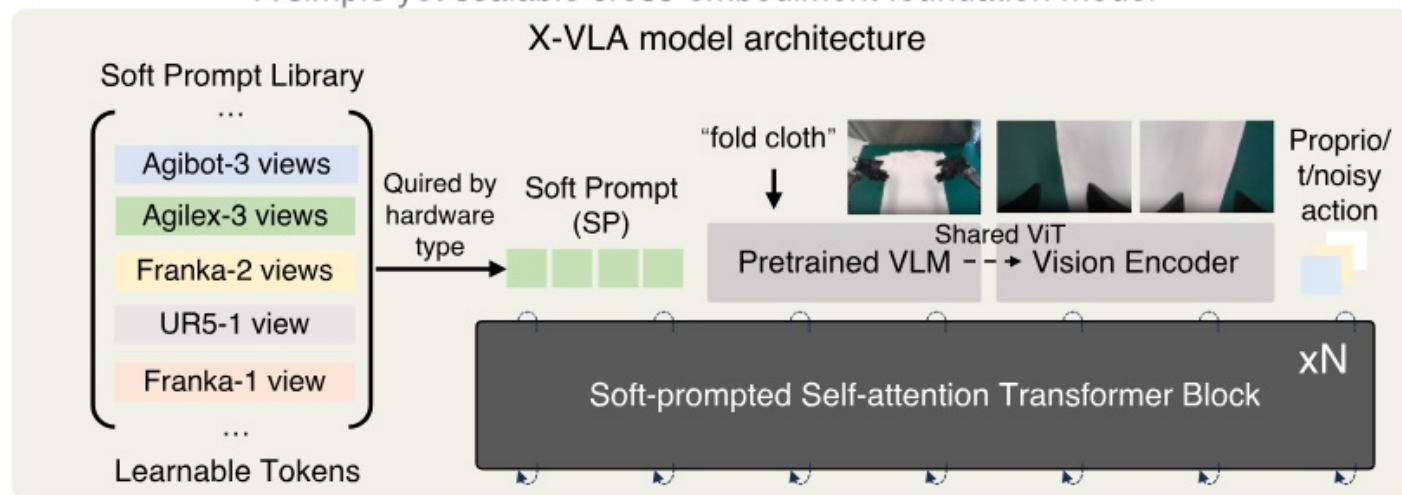


Beyond Visual Representation

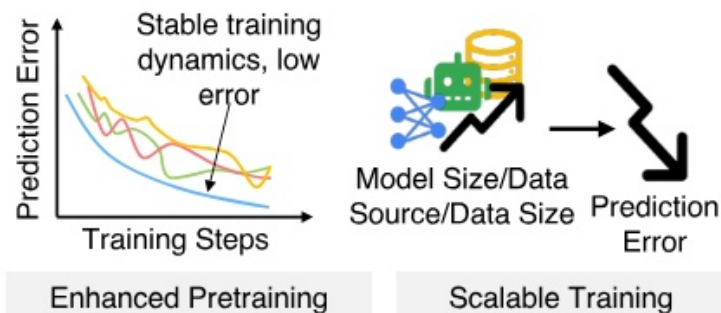
X-VLA

A simple yet scalable cross-embodiment foundation model

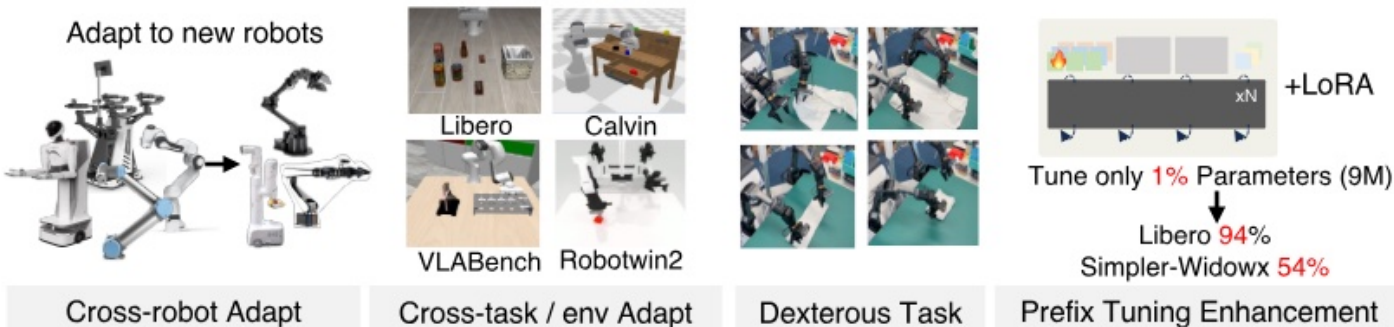
X-VLA model architecture



Scalable Cross-embodiment Pretraining

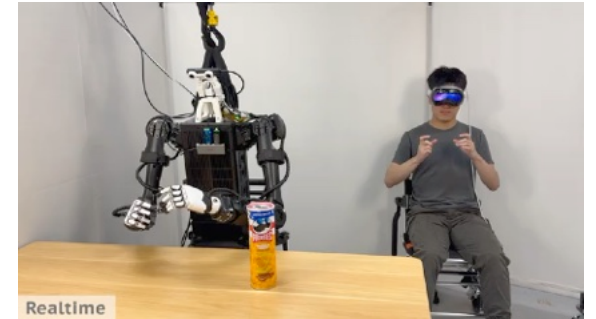


Strong Adaptation Capability

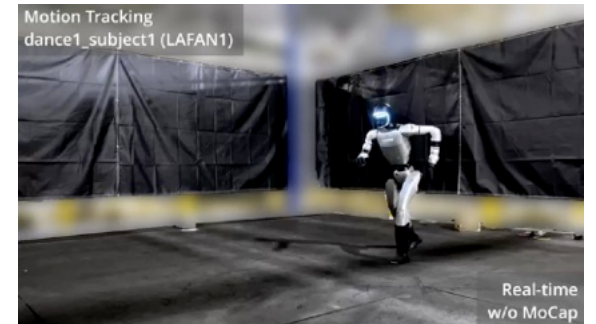


Motivation

Can we **encode robot structure in latent space** and optimize the it to perform (mimic)certain task?



HumanPlus: Humanoid Shadowing and Imitation from Humans(CoRL 2024)

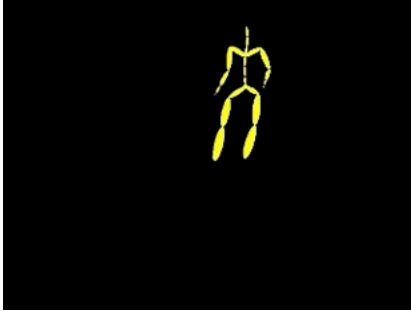


BeyondMimic: From Motion Tracking to Versatile Humanoid Control via Guided Diffusion(Arxiv 2025)

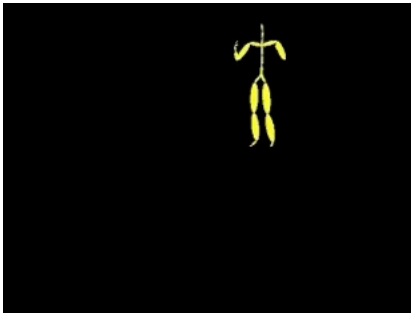
<https://www.merphi.se/download-robotic-poster/#link-humanoids2025>

Motivation

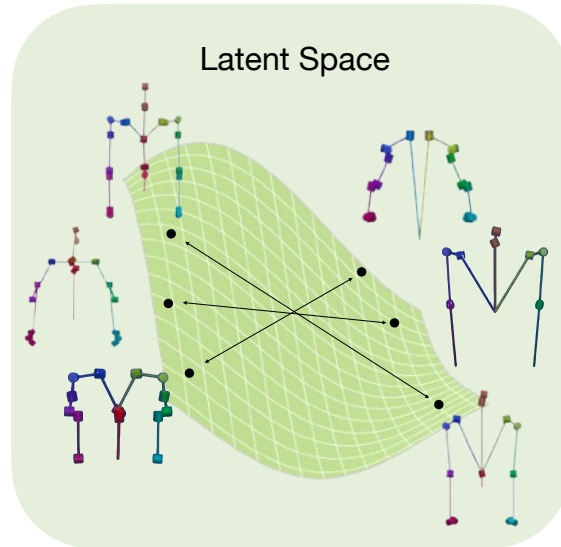
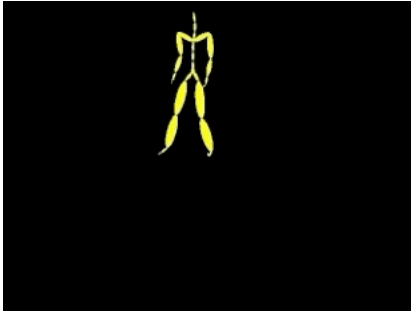
141_16 Waving Hello



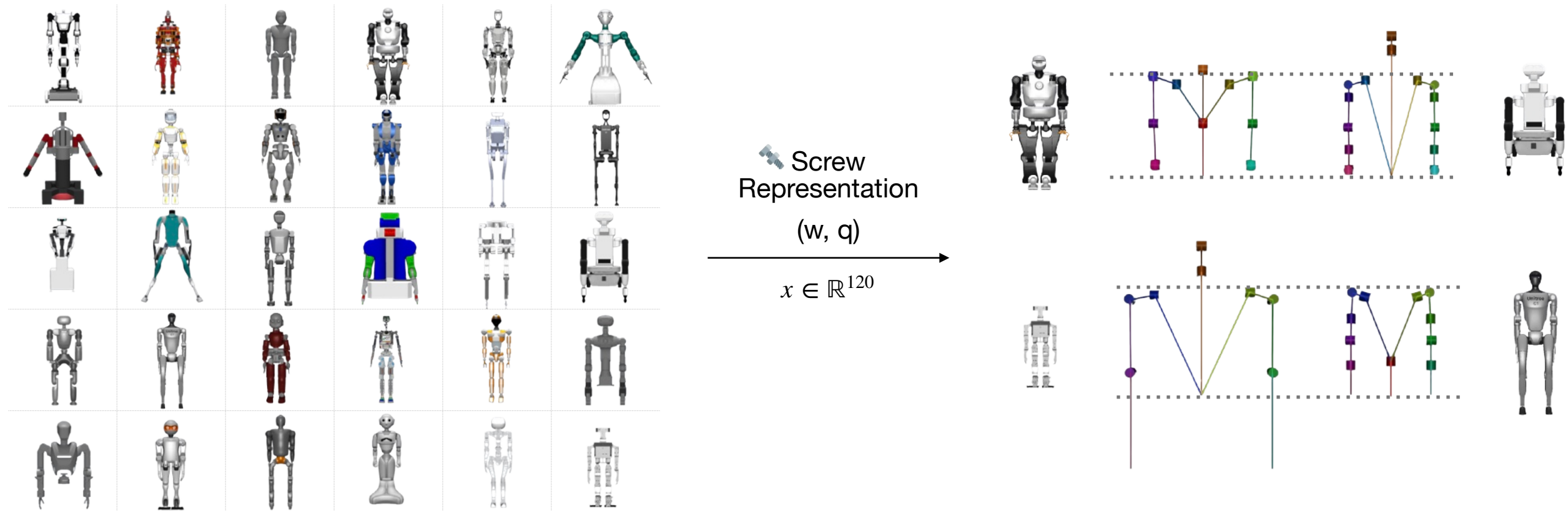
18_15 Chicken Dance



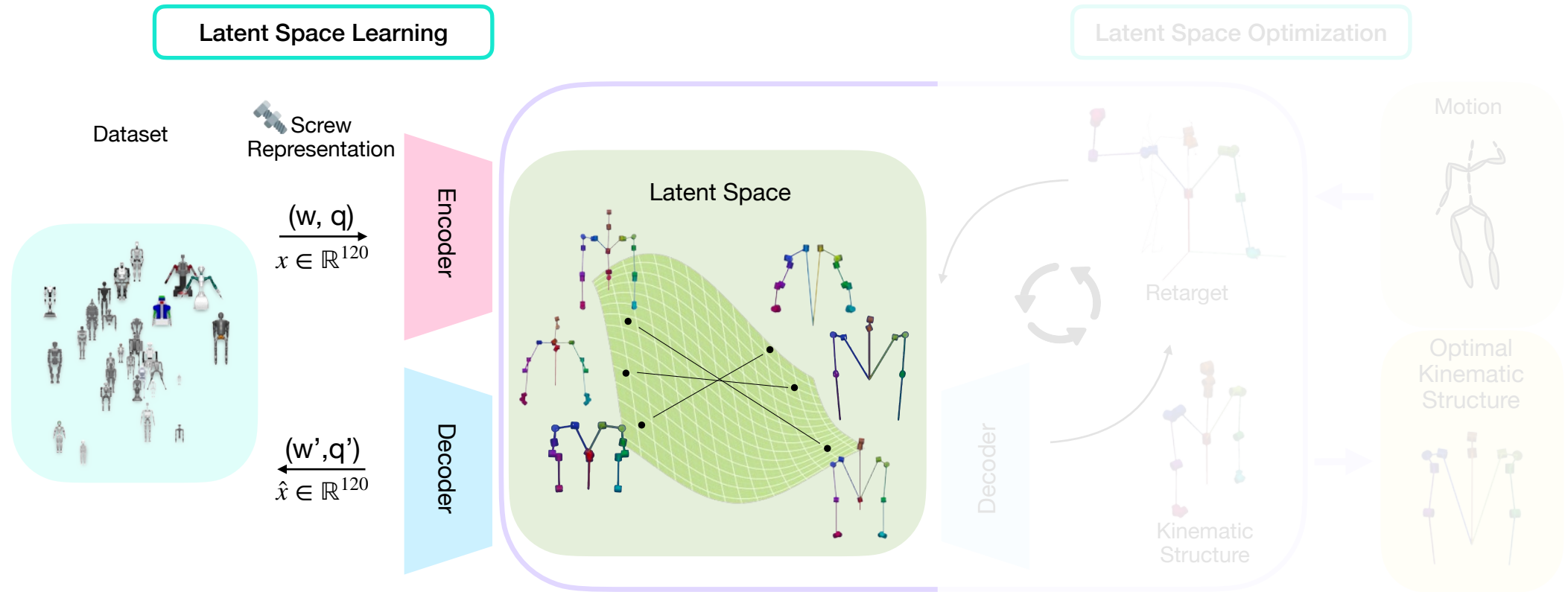
79_02 Swimming



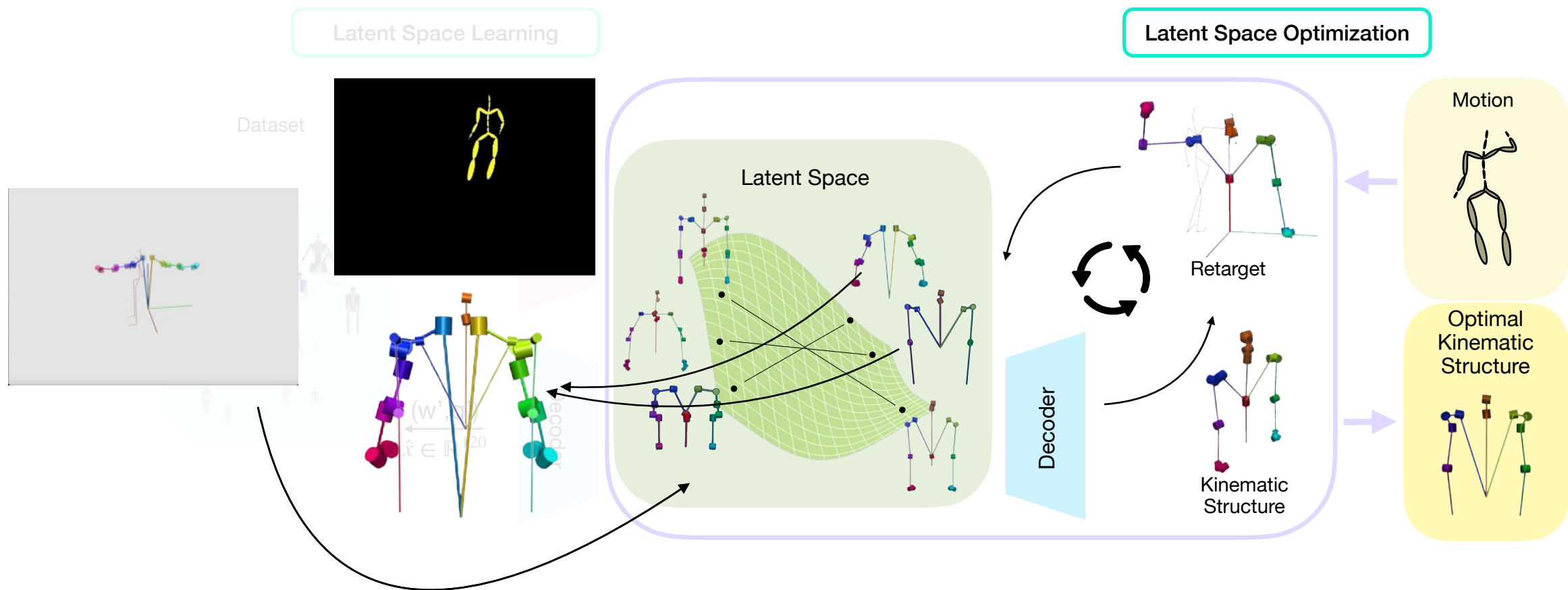
Robot Structure Representation



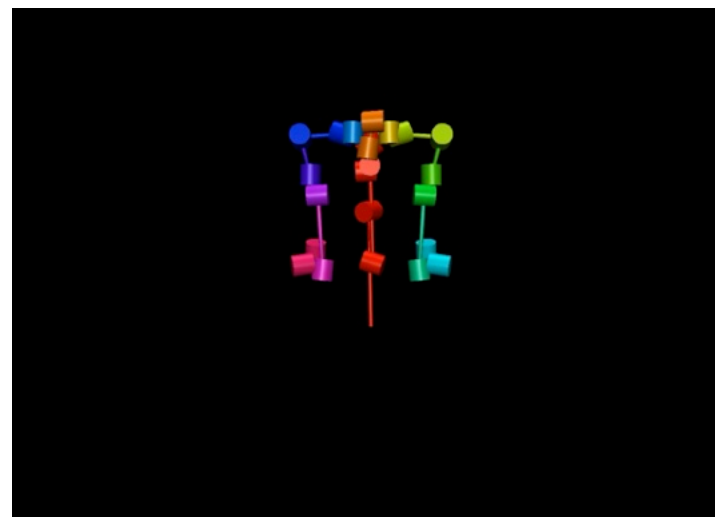
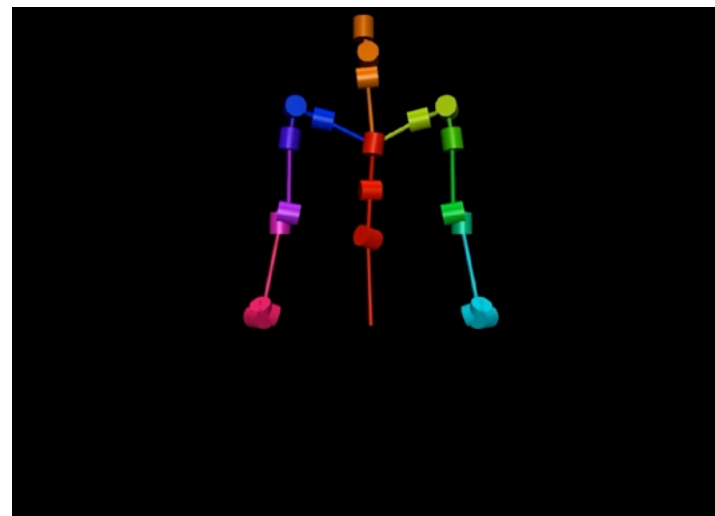
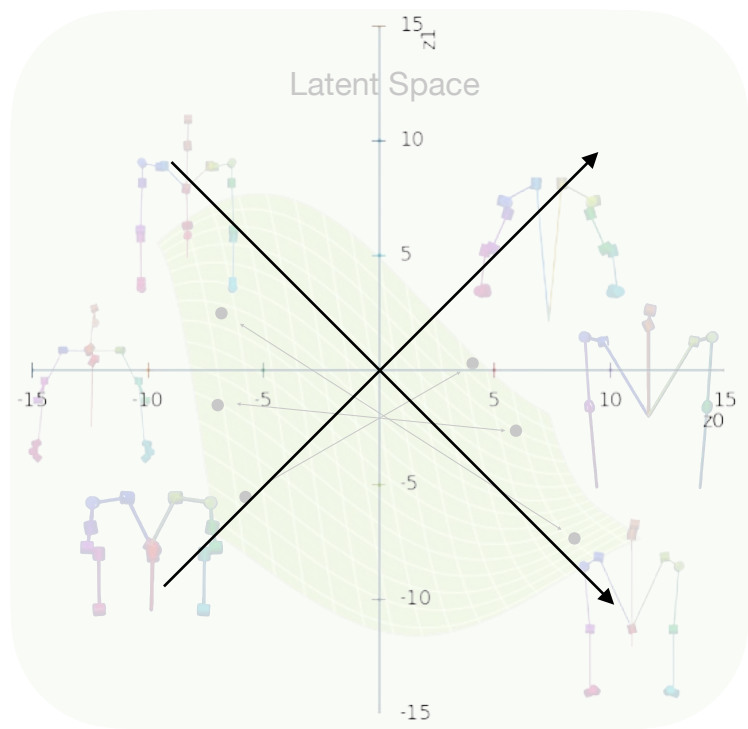
Robot Structure Representation



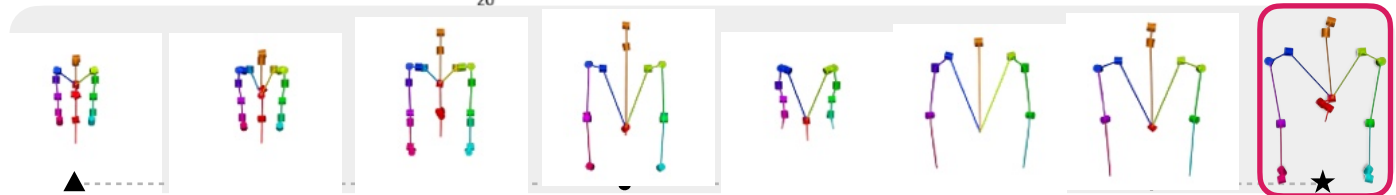
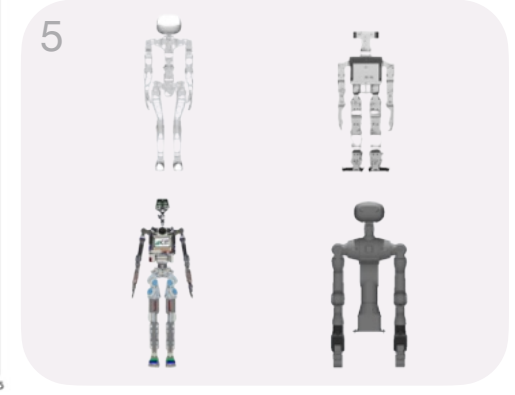
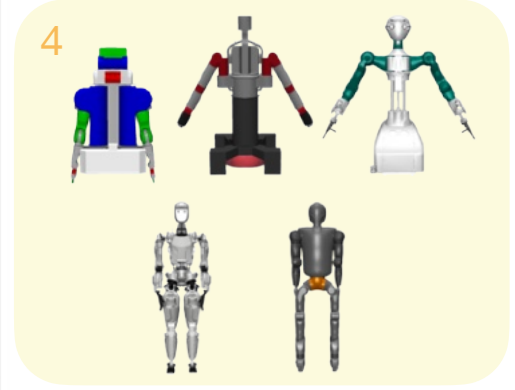
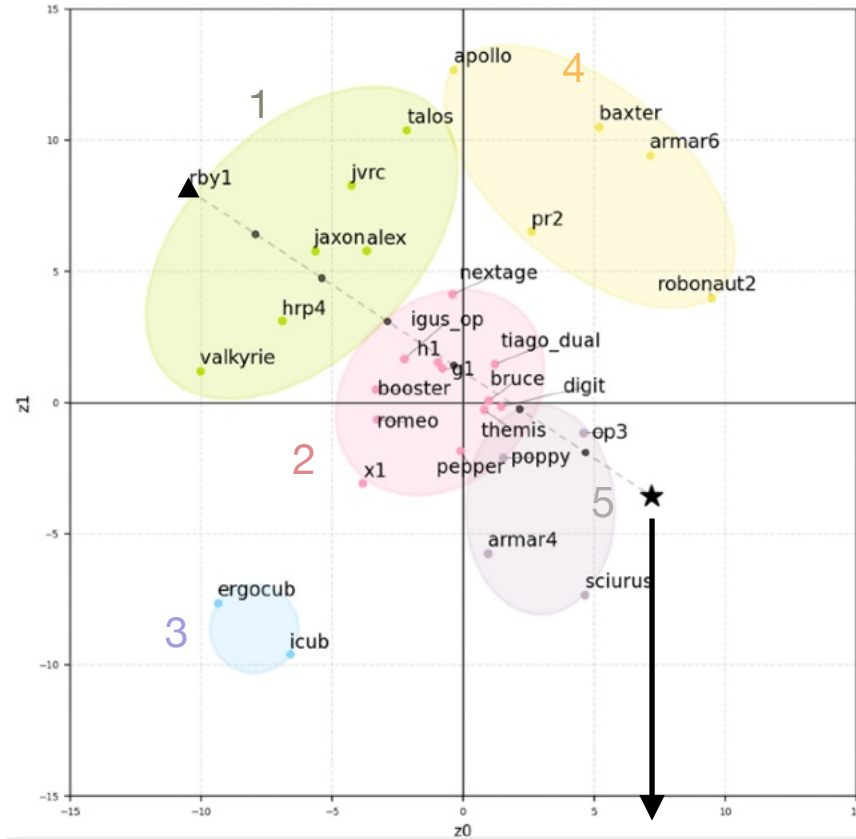
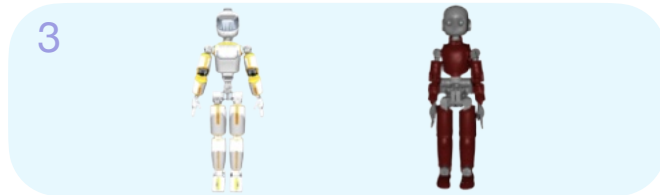
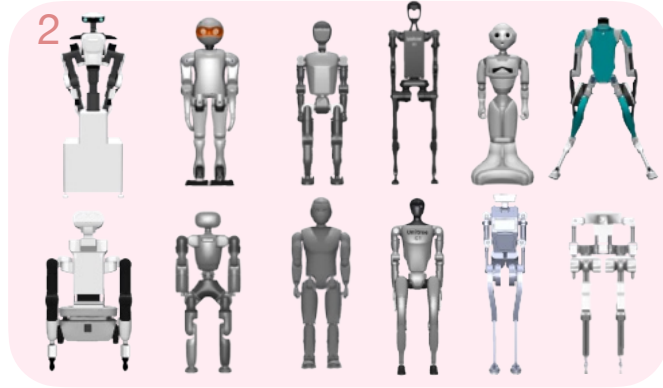
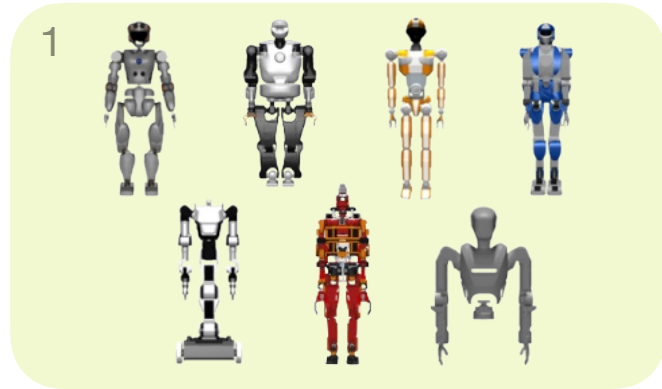
Latent Space Optimization



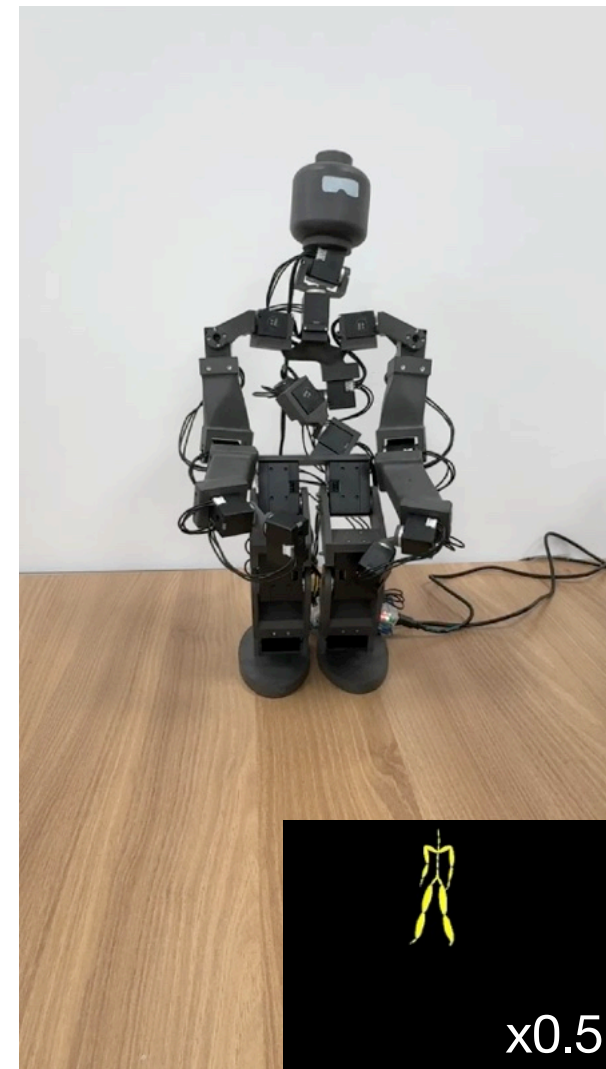
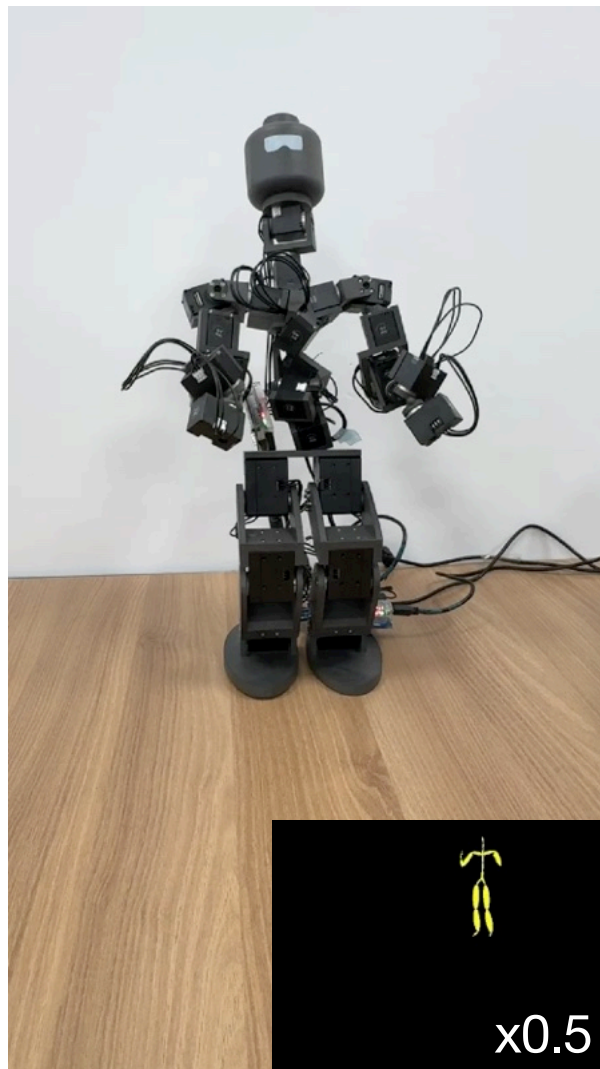
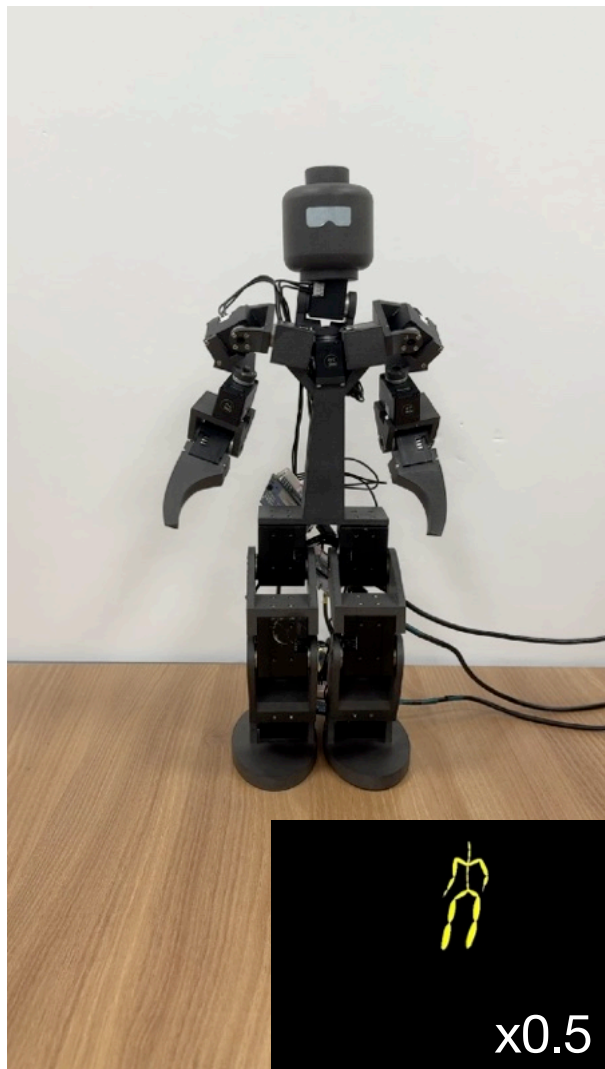
Latent Space Visualization



Latent Space Visualization



HardWare Prototype



Key Findings

- Screw-theoretic joint representation
- Compact latent design space
- Limitations
 - Geometric validity: Self-collision, mesh constraints
 - Dynamic plausibility: Torque/energy budget, contact stability

Concluding Remarks

- Reliability and efficiency are not a trade-off. Both come from making the model's internals legible and editable rather than collecting more data.
- Capability can be added by indexing, and behavior can be removed by editing, even without touching the weights at deployment.
- What we choose to represent, features, retrieved trajectories, dynamics, robot structure, determines what we can control.
- Next: a robot foundation model that can be inspected and adjusted after it is deployed, not only before.
-

Thank you for listening
:)