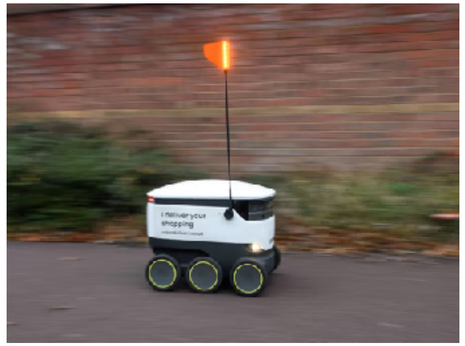
World Models and Planning for Generalizable Robots

Weiyu Liu
National University of Singapore

weiyuliu.ai@gmail.com
weiyuliu.com

Towards General-Purpose Robotics







Past → **Future**

Specialized

Single Environment

Single Task

General-Purpose

Many Environments

Many Tasks

Existing Work observation → neural network policy → action



An **end-to-end neural network policy** learns a new task with **50-100 demonstrations**

Challenges in Generalization

Existing Work

observation → neural network policy → action



Challenges in Generalization

Existing Work

observation → neural network policy → action



new initial state
faucet is on



Challenges in Generalization

Existing Work

observation → neural network policy → action



new initial state
faucet is on



new object
dirty pan



Challenges in Generalization

Existing Work

observation → neural network policy → action



new initial state
faucet is on



new environment
full sink



new object
dirty pan



Challenges in Generalization

Existing Work

observation → neural network policy → action



new initial state
faucet is on



new object
dirty pan



new environment
full sink



new skill
load dishwasher



Challenges in Generalization

Existing Work

observation → neural network policy → action



new initial state
faucet is on



new object
dirty pan



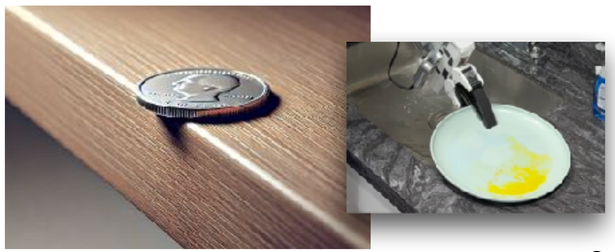
new environment
full sink



new skill
load dishwasher



new task
grasp thin coin



Challenges in Generalization

Existing Work

observation → neural network policy → action



new initial state
faucet is on



new object
dirty pan



new environment
full sink



new skill
load dishwasher



new task
grasp thin coin



Challenges in Generalization

Existing Work

observation → neural network policy → action



Scaling up data?

new initial state
faucet is on



new object
dirty pan



new environment
full sink



new skill
load dishwasher



new task
grasp thin coin



Challenges in Generalization

Existing Work

observation → neural network policy → action



new initial state
faucet is on



new object
dirty pan



new environment
full sink



new skill
load dishwasher



new task
grasp thin coin



Scaling up data?

✕ End-to-end neural policies have **limited generalization**

Challenges in Generalization

Existing Work

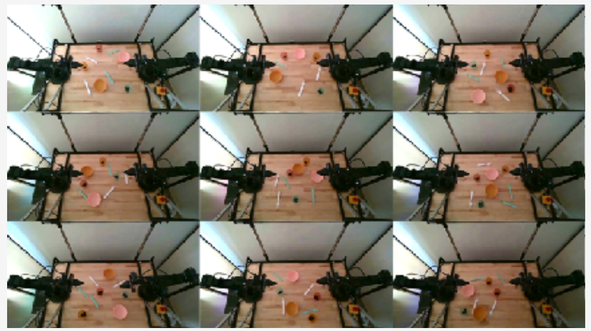
observation → neural network policy → action



table-organizing task, same objects

~**3000 demonstrations, 25% success** with new initial poses





Zhao et al., ALOHA Unleashed, CoRL, 2024.

Scaling up data?

✕ End-to-end neural policies have **limited generalization**

Challenges in Generalization

Existing Work

observation → neural network policy → action



new initial state
faucet is on



new object
dirty pan



new environment
full sink



new skill
load dishwasher



new task
grasp thin coin



different chores



Scaling up data?

✕ End-to-end neural policies have **limited generalization**

✕ **Impossible to enumerate** all situations

Challenges in Generalization

Existing Work

observation → neural network policy → action



new initial state
faucet is on



new object
dirty pan



new environment
full sink



new skill
load dishwasher



new task
grasp thin coin



different chores



changing demands



Scaling up data?

✗ End-to-end neural policies have **limited generalization**

✗ **Impossible to enumerate** all situations

Challenges in Generalization

Existing Work

observation → neural network policy → action



new initial state
faucet is on



new object
dirty pan



new environment
full sink



new skill
load dishwasher



new task
grasp thin coin



different chores



changing demands



dynamic situations



Scaling up data?

✕ End-to-end neural policies have **limited generalization**

✕ **Impossible to enumerate** all situations

Reasoning and Planning

Reasoning

“The process of drawing conclusions from principles and from evidence”

– Wason & Johnson-Laird, Psychology of Reasoning: Structure and Content (1972)

Planning

“The process of choosing and organizing actions to achieve a goal.”

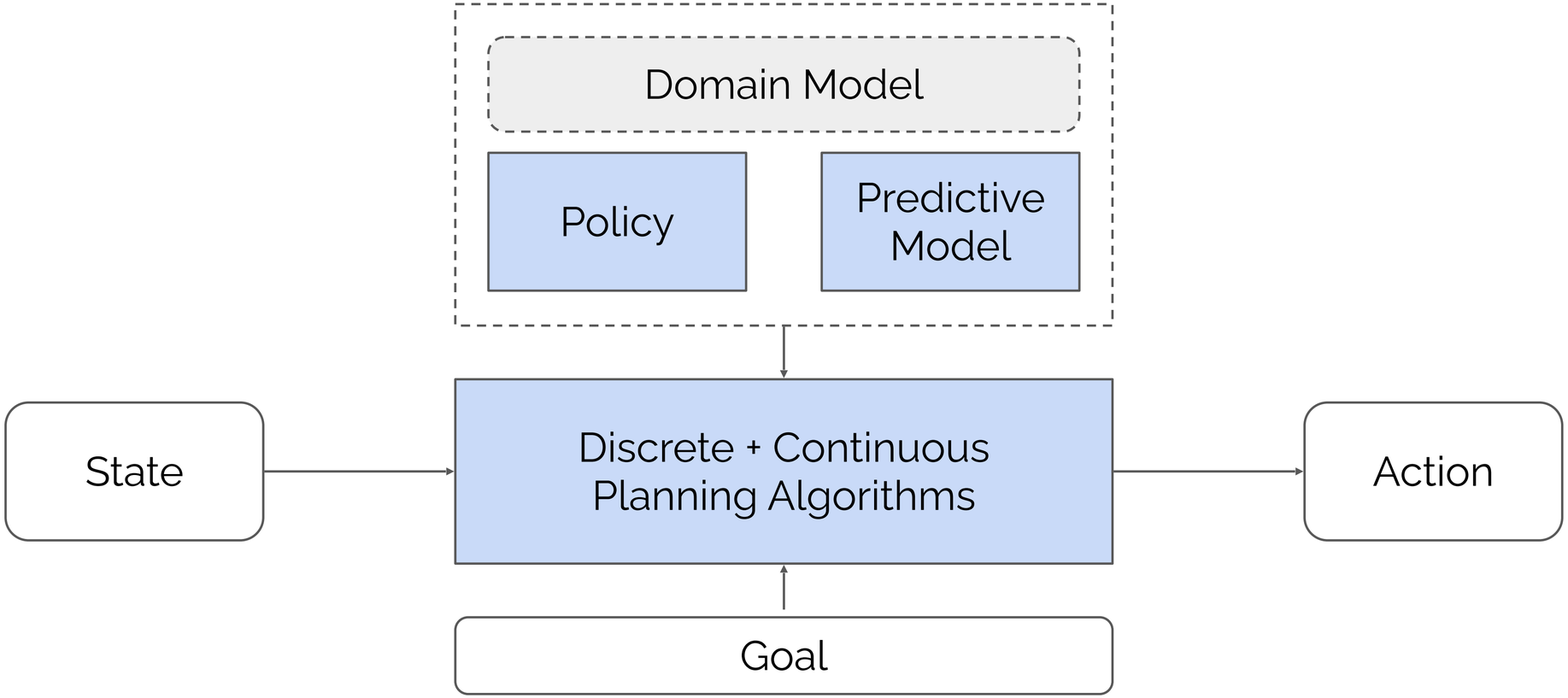
— Ghallab, Nau, & Traverso, Acting, Planning, and Learning (2016)

Why do we need planning?

Can’t we just react? Can’t we memorize everything?

- Plan to solve **complex** problems
- Plan to solve problems **optimally**
- Plan to solve problems **safely**
- Plan to solve **new** problems

Framework for Planning



Challenges

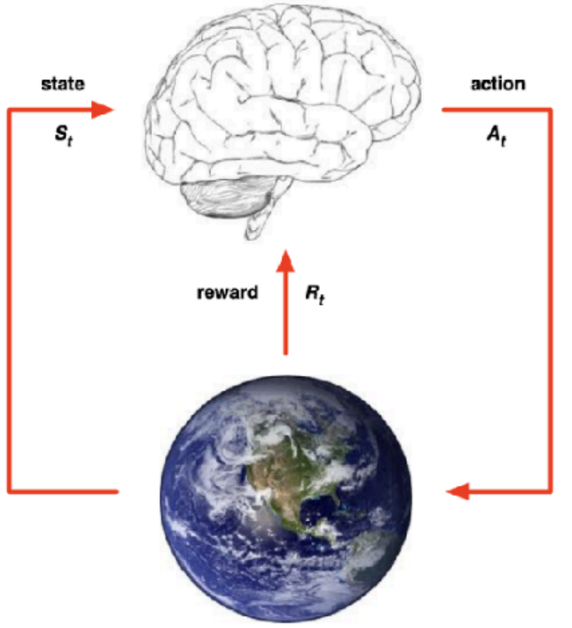
- Build a predictive model
- Search for solutions using the predictive model
- Generalize the policy and predictive model

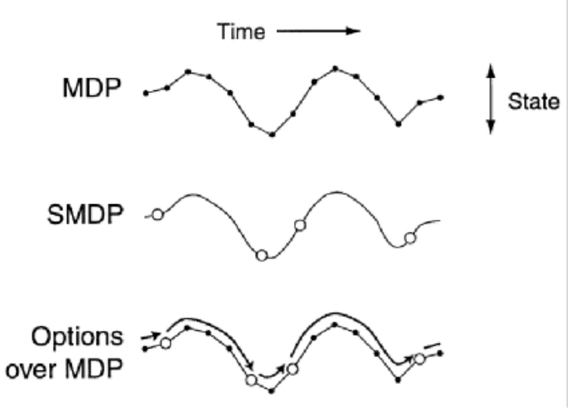
Learning abstractions for planning

1. **Learn state and action abstractions from demos (BLADE)**
2. Discover spatial and temporal structure (STACK)
3. Learn a world model from raw video (Rhoda)
4. Reason at different levels of abstraction (APIVOT)

Action Abstraction

- Behaviors are modular and compositional
- MDP:
  - Tuple: (S, A, R, T, γ)
  - Transition: $T(s'|s, a)$
  - Reward: $R(s, a, s')$
- Option:
  - policy: π_o
  - initiation set: $I_o = \{s|o \in O(s)\}$
  - termination condition: $\beta_o \rightarrow [0, 1]$
- Semi-MDP:
  - Tuple: (S, O, R, P, γ)
  - Transition: $T(s', \tau|o, a)$
  - Reward: $R(s', \tau|s, o)$

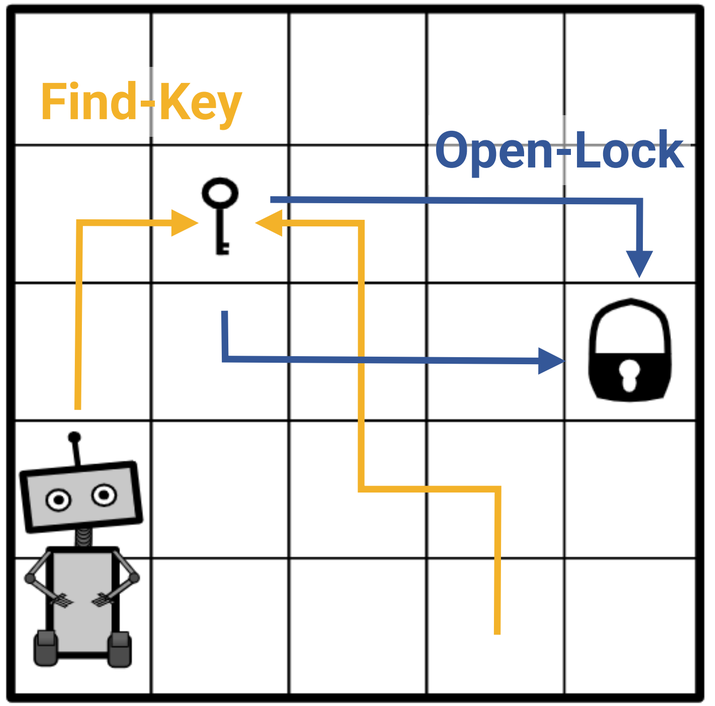




Sutton, Richard S., Doina Precup, and Satinder Singh. “Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning." Artificial intelligence 112, no. 1-2 (1999): 181-211.

Action Abstraction

- Behaviors are modular and compositional
- MDP:
  - Tuple: (S, A, R, T, γ)
  - Transition: $T(s'|s, a)$
  - Reward: $R(s, a, s')$
- Option:
  - policy: π_o
  - initiation set: $I_o = \{s|o \in O(s)\}$
  - termination condition: $\beta_o \rightarrow [0, 1]$
- Semi-MDP:
  - Tuple: (S, O, R, P, γ)
  - Transition: $T(s', \tau|o, a)$
  - Reward: $R(s', \tau|s, o)$



Sutton, Richard S., Doina Precup, and Satinder Singh. “Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning." Artificial intelligence 112, no. 1-2 (1999): 181-211.

State Abstraction

- State representation can be factorized
- Planning domain
  - Proposition: abstract boolean variable p_i
  - Action: $\alpha_i = (\text{precond}_i, \text{effect}_i^+, \text{effect}_i^-)$
  - Problem: initial state s_0, goal states S_g

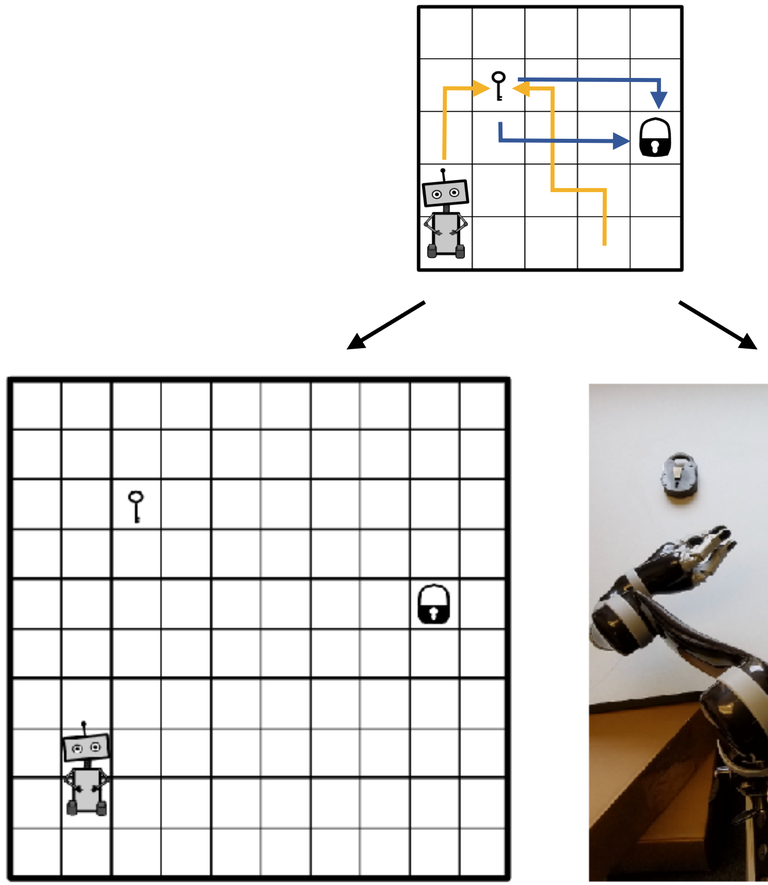
```
(:action Open-Lock
 :precondition LockClosed and KeyFound
 :effect+
 :effect- LockClosed
)
```

Ghallab, Malik, Dana Nau, and Paolo Traverso. Automated Planning: theory and practice. Elsevier, 2004. Harvard

State Abstraction

- State representation can be factorized
- Planning domain
  - Proposition: abstract boolean variable p_i
  - Action: $\alpha_i = (\text{precond}_i, \text{effect}_i^+, \text{effect}_i^-)$
  - Problem: initial state s_0, goal states S_g

```
(:action Open-Lock
 :precondition LockClosed and KeyFound
 :effect+
 :effect- LockClosed
)
```





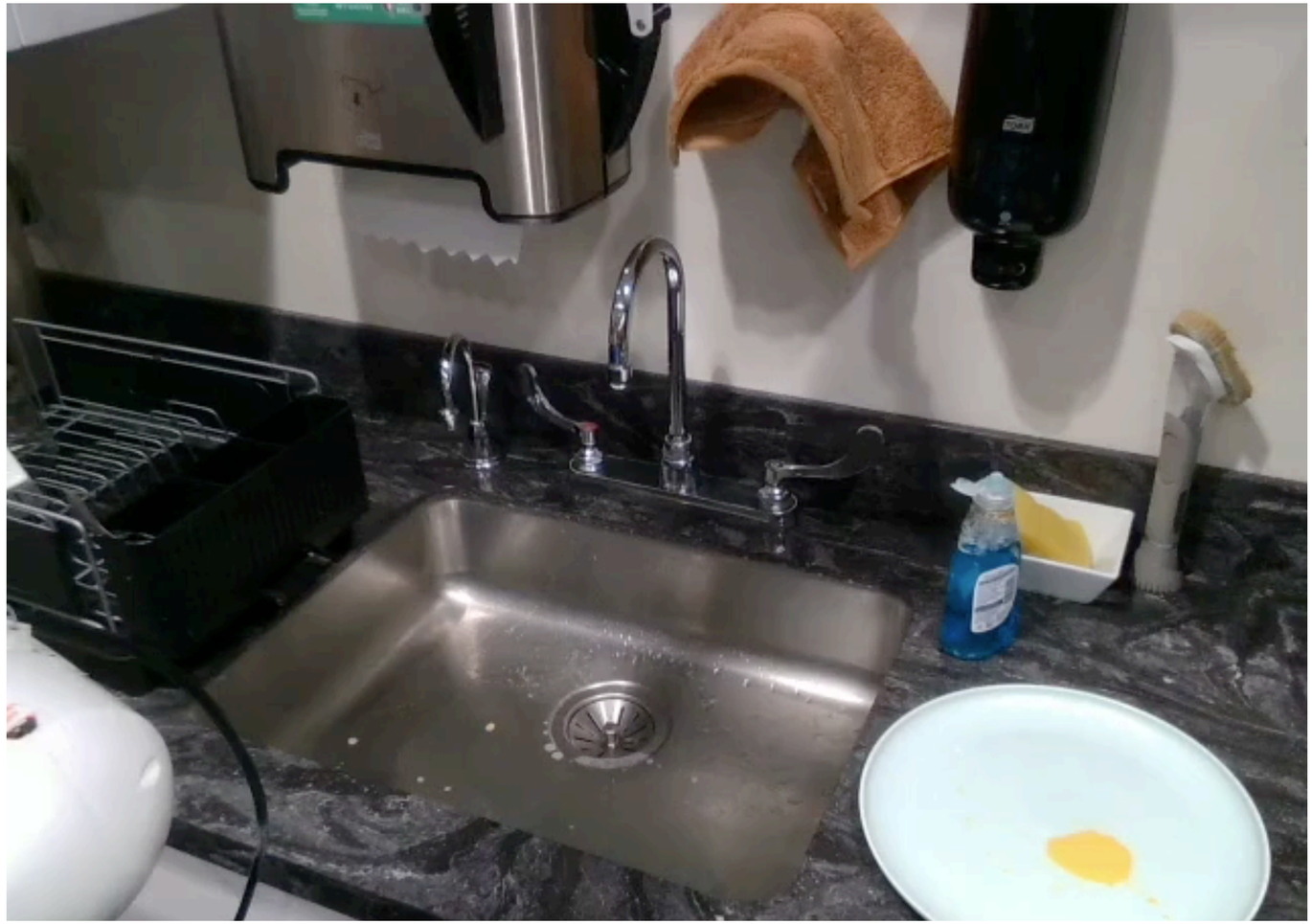
Ghallab, Malik, Dana Nau, and Paolo Traverso. Automated Planning: theory and practice. Elsevier, 2004. Harvard



https://github.com/Learning-and-Intelligent-Systems/open-world-tamp/blob/master/open_world/planning/pddl/domain.pddl

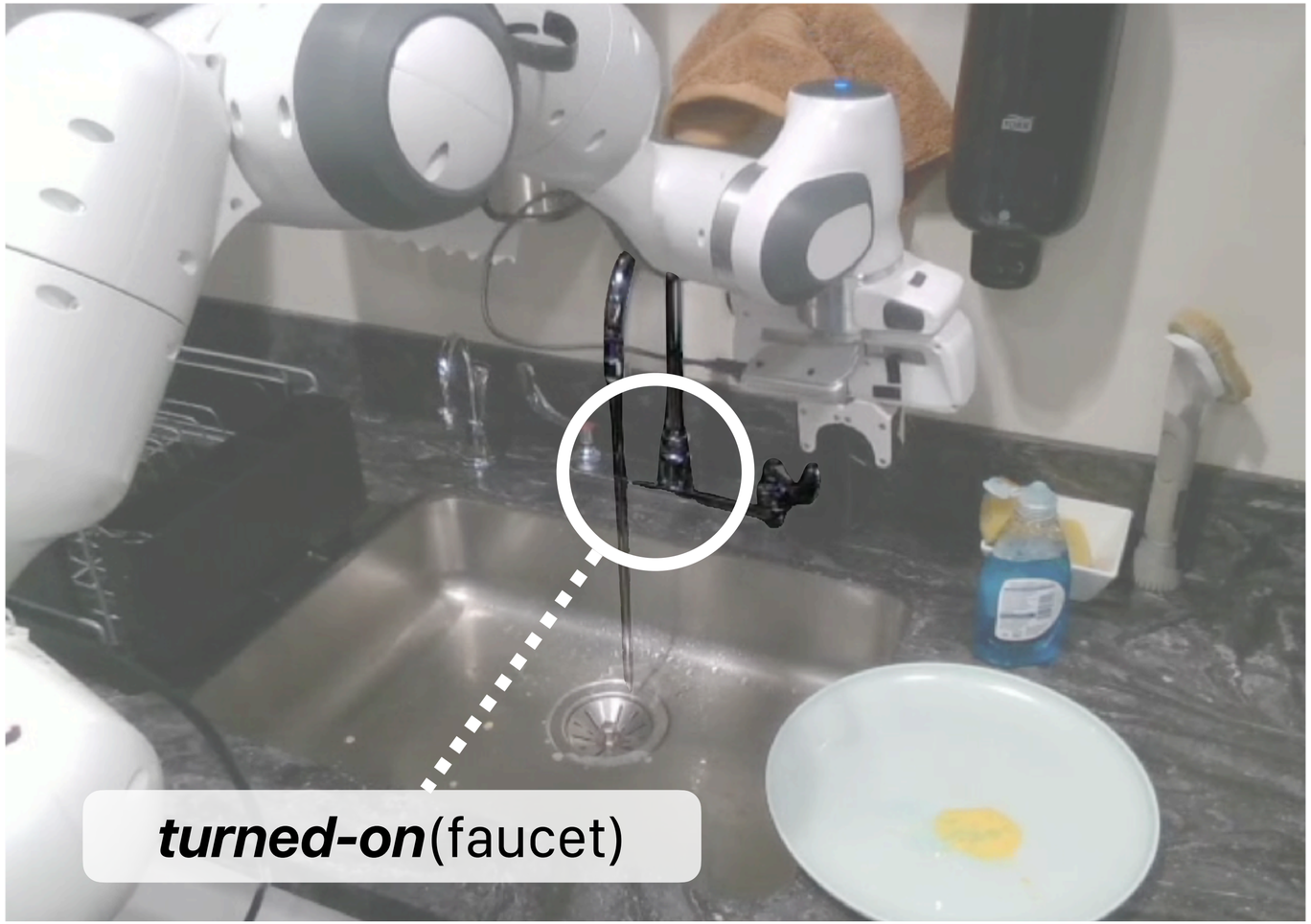
We don’t want to hardcode these abstractions,
so how to build a robot that can discover these abstractions?

Learn from Demonstration



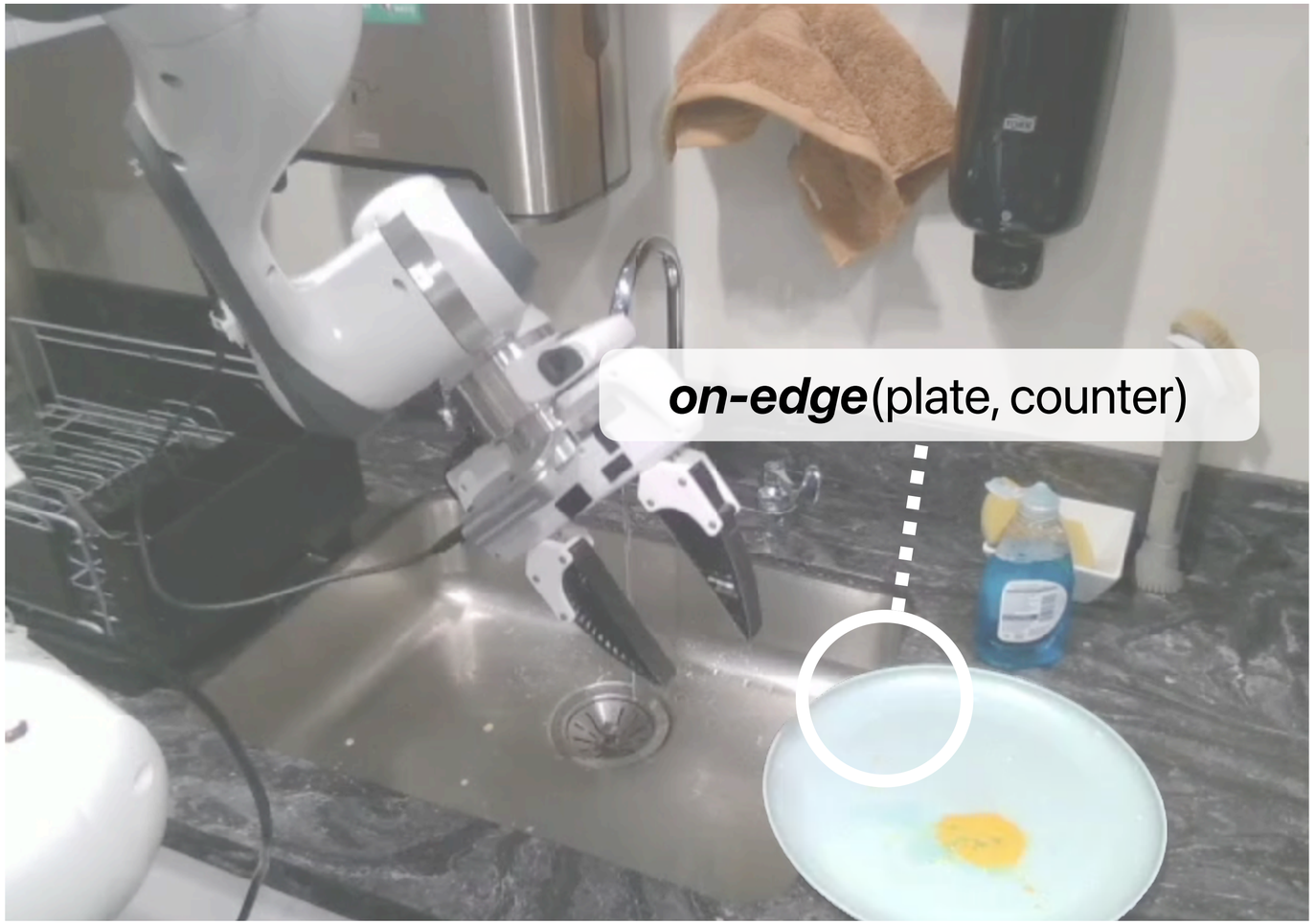


Factorized State





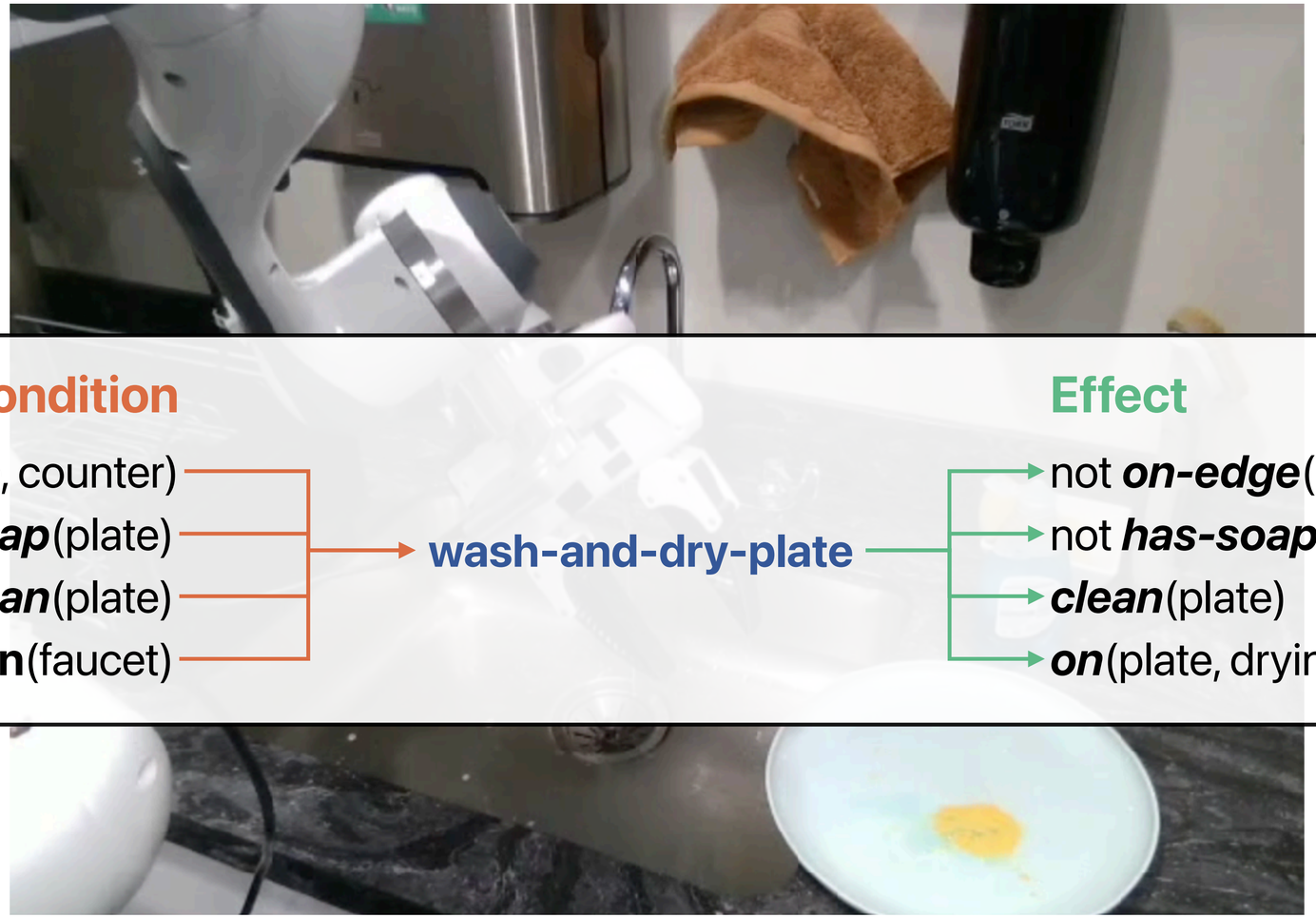
Factorized State



Reusable Skills



Action Model for Encoding Cause and Effect



Precondition

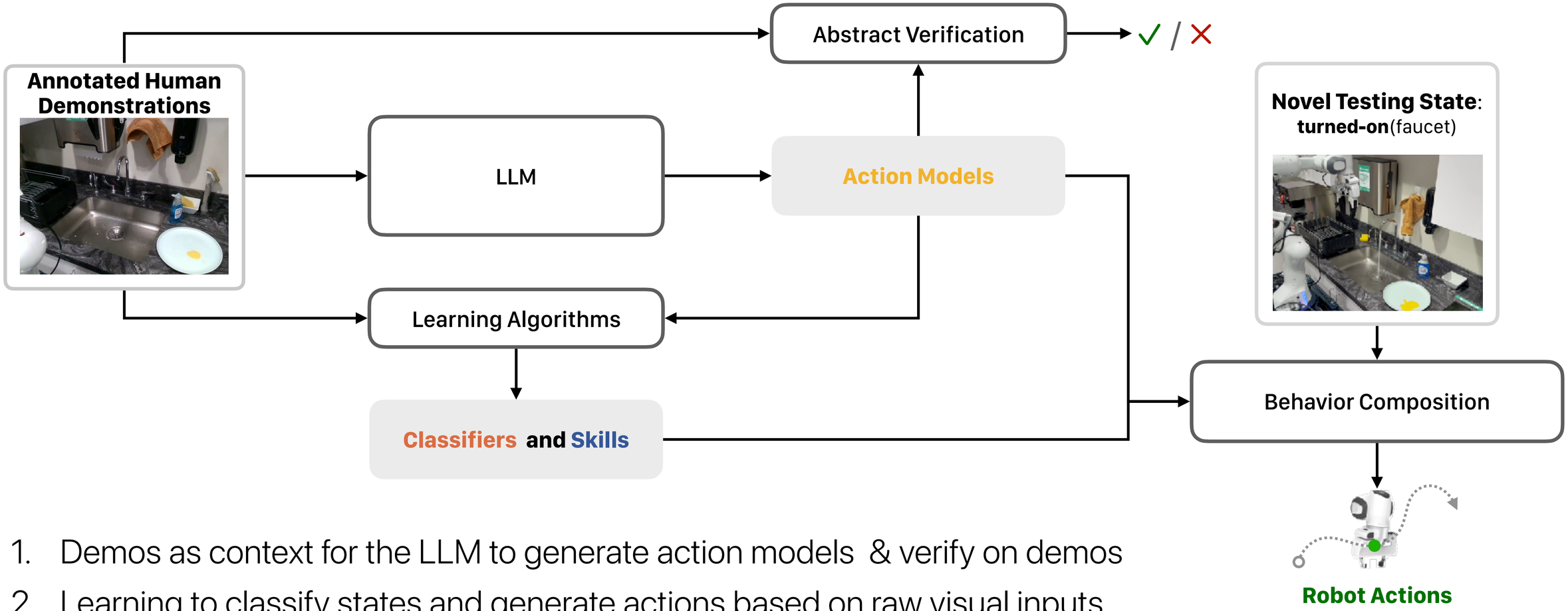
on-edge(plate, counter)
has-soap(plate)
not ***clean***(plate)
turned-on(faucet)

wash-and-dry-plate

Effect

not ***on-edge***(plate, counter)
not ***has-soap***(plate)
clean(plate)
on(plate, drying-rack)

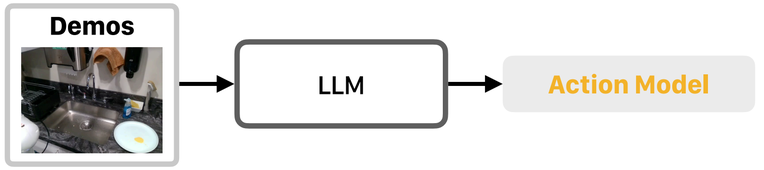
Automatic Construction of Action Models



1. Demos as context for the LLM to generate action models & verify on demos
2. Learning to classify states and generate actions based on raw visual inputs
3. Combine action models and learned modules to complete new tasks

Learning Compositional Behaviors from Demonstration and Language. **Weiyu Liu***, Neil Nie*, Ruohan Zhang, Jiayuan Mao†, Jiajun Wu†. CoRL, 2024

1. Demos as Context for the LLM to Generate Structure



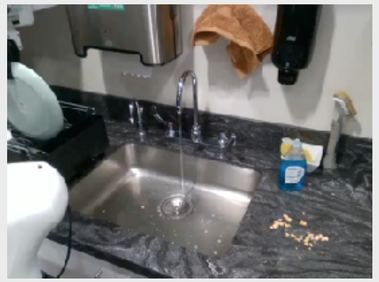
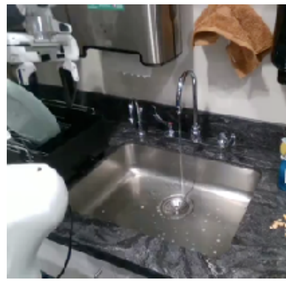
"turn on faucet"

"grasp plate and wash and place on dish rack"

"wipe ..."





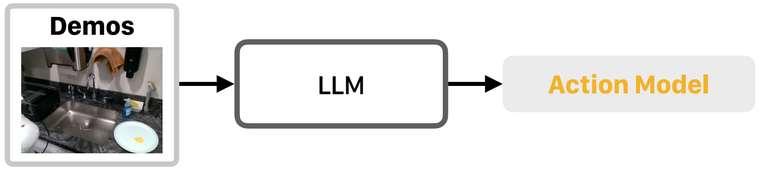




push | open | move-to | grasp | move | place | move-to

LLM

- Detect contact primitives using proprioception:
  - open, close, move-to, grasp, place, move, push
- Prompt the LLM with the language description, contact sequence, and additional context

1. Demos as Context for the LLM to Generate Structure



"turn on faucet"

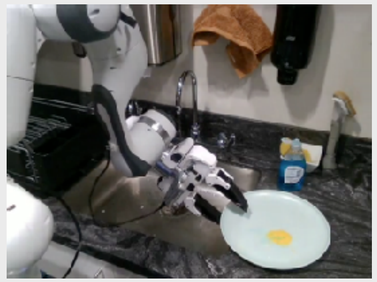


push



open

"grasp plate and wash and place on dish rack"



move-to



grasp

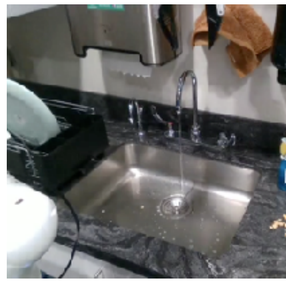


move



place

"wipe ...



move-to

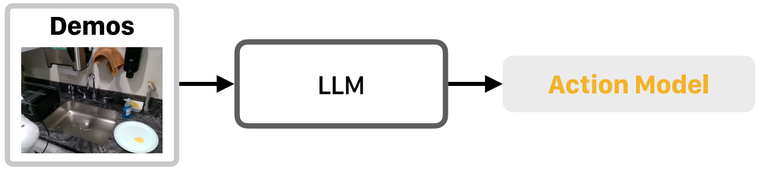
LLM

"The faucet needs to be turned on for washing dish."

"Pushing the dishes to the edge is for better grasping."

"The plate will be clean after washing."

1. Demos as Context for the LLM to Generate Structure



"turn on faucet"

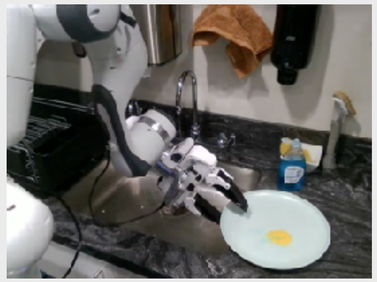


push



open

"grasp plate and wash and place on dish rack"



move-to



grasp

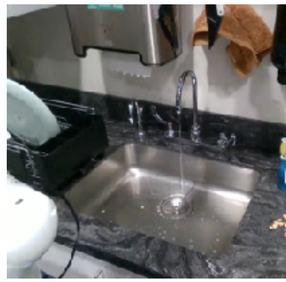


move



place

"wipe ..."

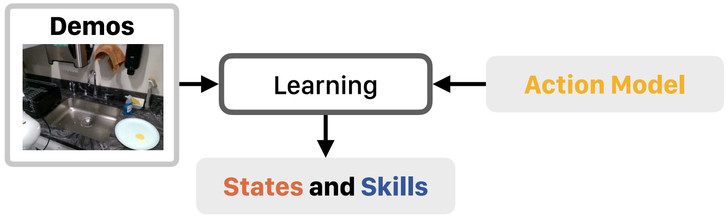


move-to

LLM

```
(:action          wash-and-dry-plate
:precondition     (and (on-edge ?plate ?counter) (has-soap ?plate) (not (clean ?plate)) (turned-on ?faucet))
:effect           (and (not (on-edge ?plate ?counter) (not (has-soap ?plate)) (clean ?plate) (on ?plate ?drying-rack))
:body             (then (grasp ?plate ?counter) (move) (place ?plate ?drying-rack)))
```

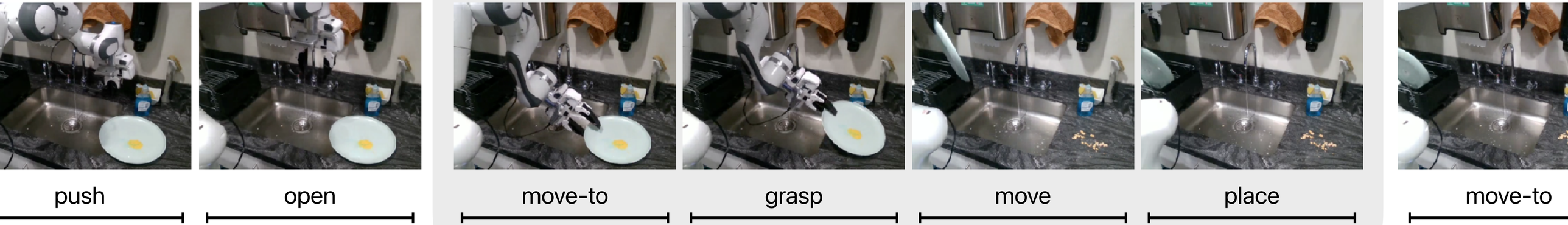
2. Learning to Classify States Based on Raw Visual Inputs



"turn on faucet"

"grasp plate and wash and place on dish rack"

"wipe ..."



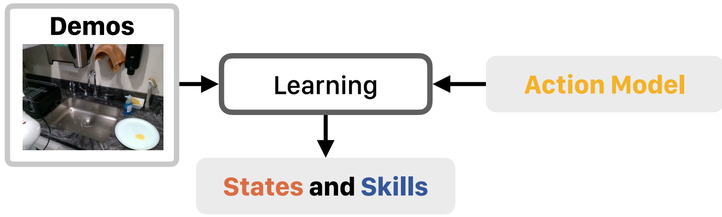
(:action **wash-and-dry-plate**

:precondition (and (***on-edge*** ?plate ?counter) (***has-soap*** ?plate) (not (***clean*** ?plate)) (***turned-on*** ?faucet))

:effect (and (not (***on-edge*** ?plate ?counter) (not (***has-soap*** ?plate)) (***clean*** ?plate) (***on*** ?plate ?drying-rack))

:body (then (**grasp** ?plate ?counter) (**move**) (**place** ?plate ?drying-rack)))

2. Learning to Classify States Based on Raw Visual Inputs



"turn on faucet"

"grasp plate and wash and place on dish rack"

"wipe ..."



push — open — move-to — grasp — move — place — move-to

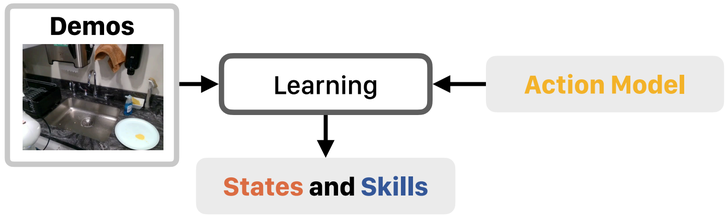
(***on-edge*** ?plate ?counter)

(:action **turn-on-faucet**
:pre (and (not (***turned-on*** ?faucet) ...)
:eff (and (***turned-on*** ?faucet) ...)
:body (then (**push** ?faucet) ...))

(:action **wash-and-dry-plate**
:pre (and (***on-edge*** ?plate ?counter) ...)
:eff (and (not (***on-edge*** ?plate ?counter) ...)
:body (then (**grasp** ?plate ?counter) ...))

(:action **wipe-counter**
:pre (and (not (***clean*** ?counter) ...)
:eff (and (***clean*** ?counter) ...)
:body (then (**grasp** ?towel ?hanger) ...))

2. Learning to Classify States Based on Raw Visual Inputs



"turn on faucet"

"grasp plate and wash and place on dish rack"

"wipe ..."



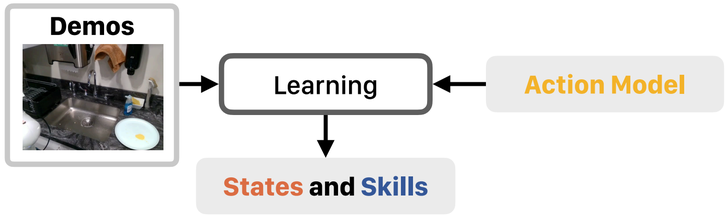
push | open | move-to | grasp | move | place | move-to

(***on-edge*** ?plate ?counter)



ResNet | PointNet | CLIP | VLM

2. Learning Visuomotor Policies for Primitive Skills

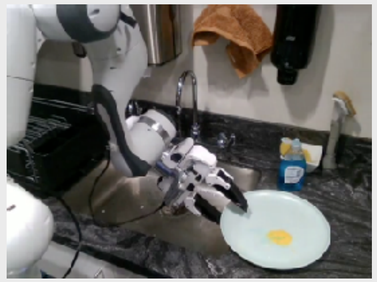


"turn on faucet"

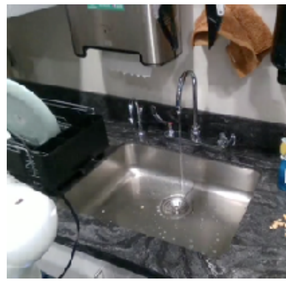
"grasp plate and wash and place on dish rack"

"wipe ...







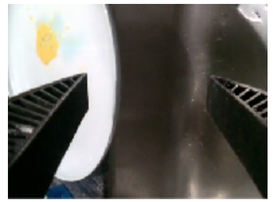


push | open | move-to | grasp | move | place | move-to

(:action **wash-and-dry-plate**
:precondition (...)
:effect (...)
:body (then (**grasp** ?plate ?counter)
(**move**)
(**place** ?plate ?drying-rack)))

$\{o_0, u_1, o_1 ..., o_T\}$

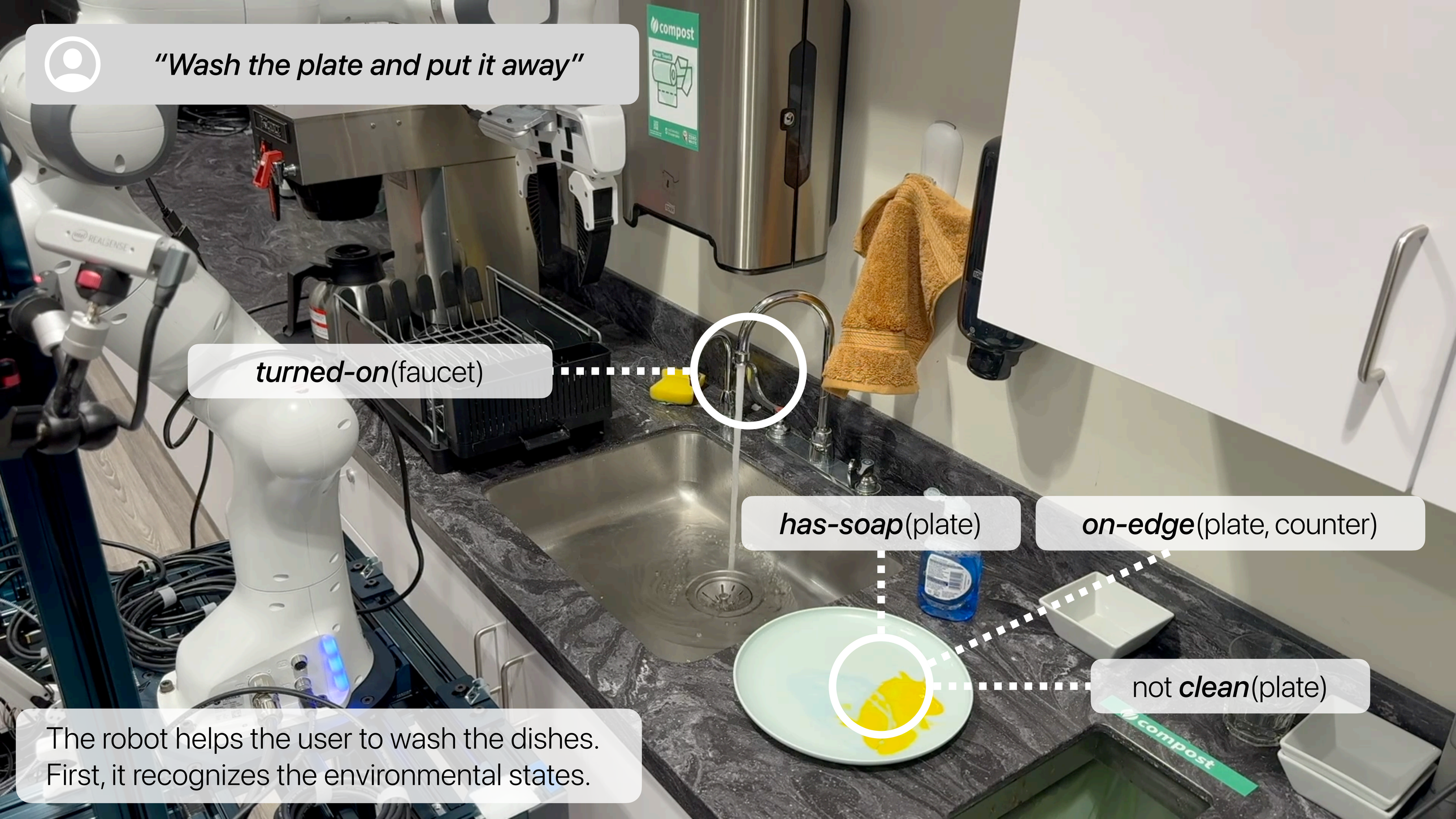


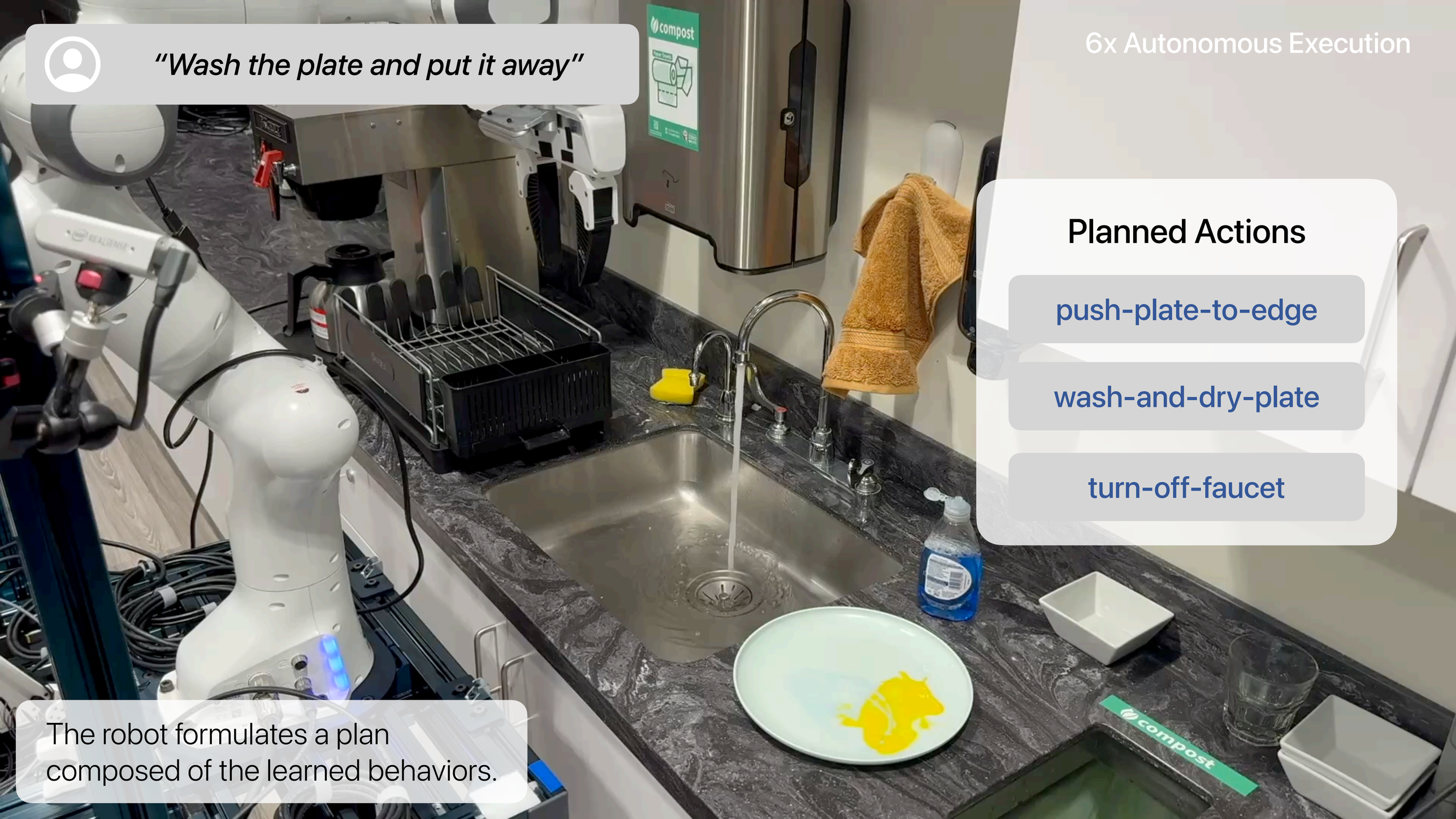


cam 1 | cam 2 | gripper cam

Image-based Diffusion Policy


"Wash the plate and put it away"
compost
compost
The human user pours dish soap on the plate
but has to leave the kitchen to take a phone call.


"Wash the plate and put it away"
turned-on(faucet)
has-soap(plate)
on-edge(plate, counter)
not clean(plate)
The robot helps the user to wash the dishes.
First, it recognizes the environmental states.


"Wash the plate and put it away"
6x Autonomous Execution
Planned Actions
push-plate-to-edge
wash-and-dry-plate
turn-off-faucet
The robot formulates a plan composed of the learned behaviors.

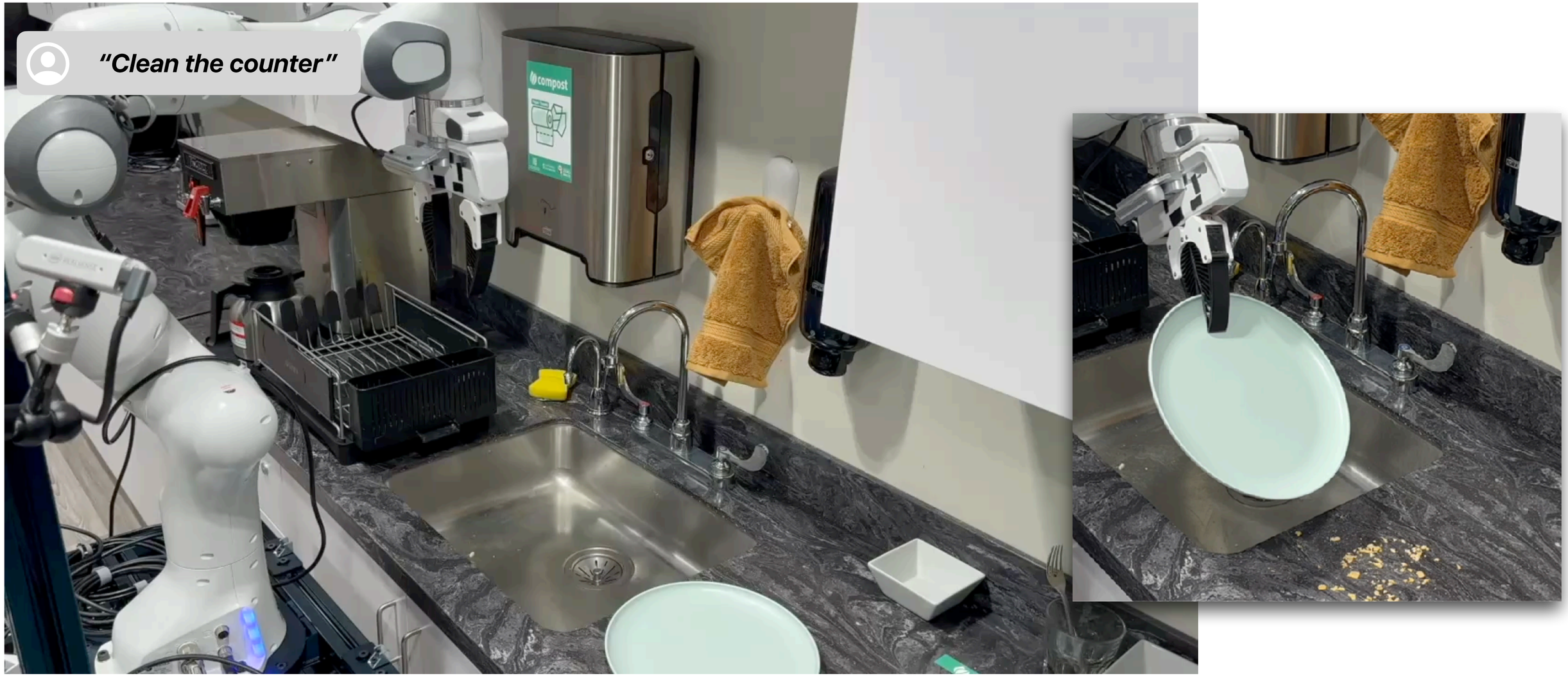
Strong Generalization to Novel Situations

Human Perturbation



Strong Generalization to Novel Situations

Partial Observability



Strong Generalization to Novel Situations

Novel Initial Conditions



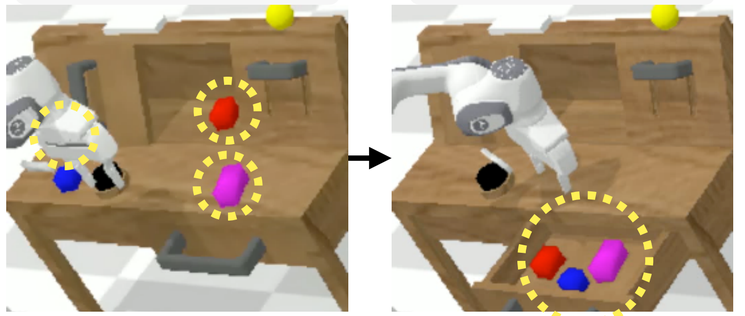






Simulation Experiments

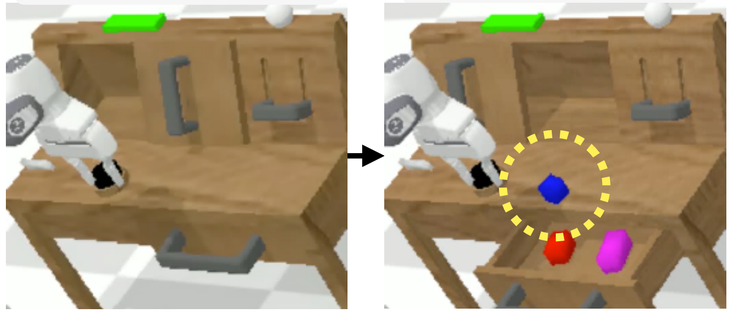
Abstract Goals



Language Goal: "Place All Blocks Inside Drawer"

$\forall x$. **is-block**$(x) \Rightarrow$ **in**$(x,$ **drawer**$)$

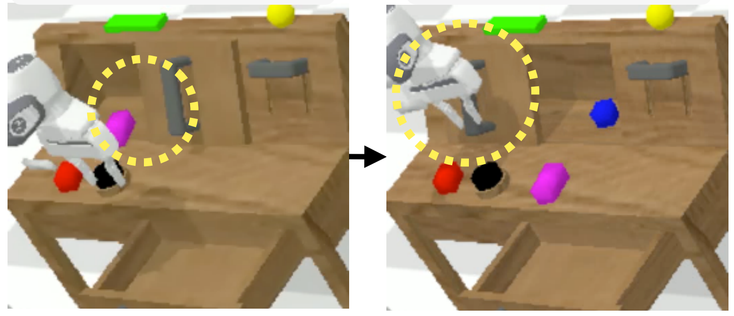
Partial Observability



Language Goal: "Find Block In Slider"

is-block(x), **is-blue**(x), **is-table**(y), **on**(x, y)

Geometric Constraints

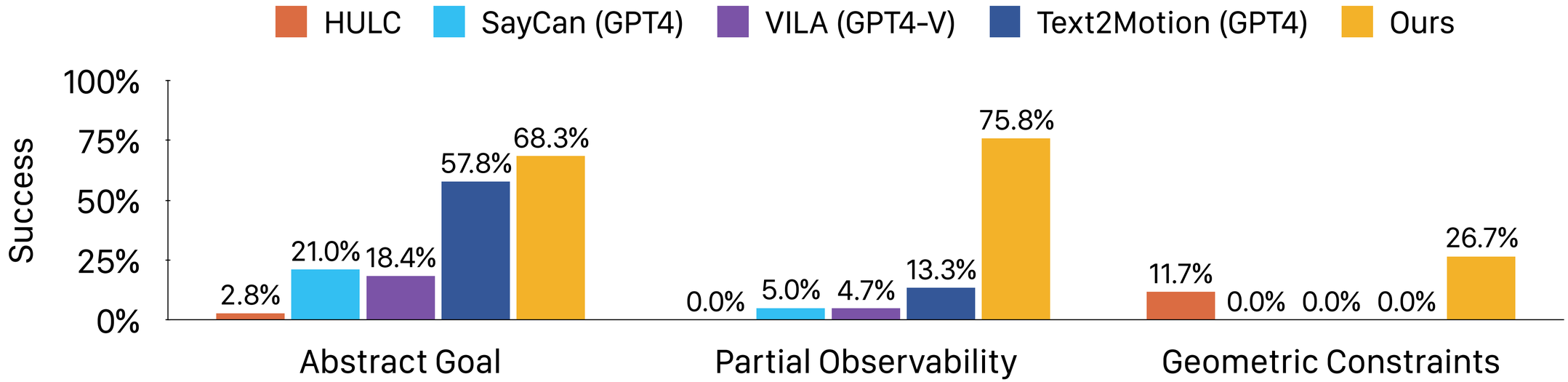


Language Goal: "Move Sliding Door Left"

is-sliding-door(x), **left**(x)

- 6 generalization tasks, 20 randomized task configurations, 3 random seeds
- Baselines:
  - A hierarchical policy with planning in latent spaces: **HULC**
  - LLM/VLM-based methods for robotic planning: **SayCan**, **Text2Motion**, **VILA**

Comparison with Existing Planners



- **HULC** is not able to chain multiple high-level actions due to the strategy used for hindsight relabeling.
- **SayCan** and **Text2Motion** fail due to challenges in learning accurate transition and feasibility models.
- **VILA** struggles to generate feasible plans based on the environmental states.
- **BLADE** leverages structured transition models and scene representations for long-horizon planning.

Learning abstractions for planning

1. Learn state and action abstractions from demos (BLADE)

2. **Discover spatial and temporal structure (STACK)**

3. Learn a world model from raw video (Rhoda)

4. Reason at different levels of abstraction (APIVOT)

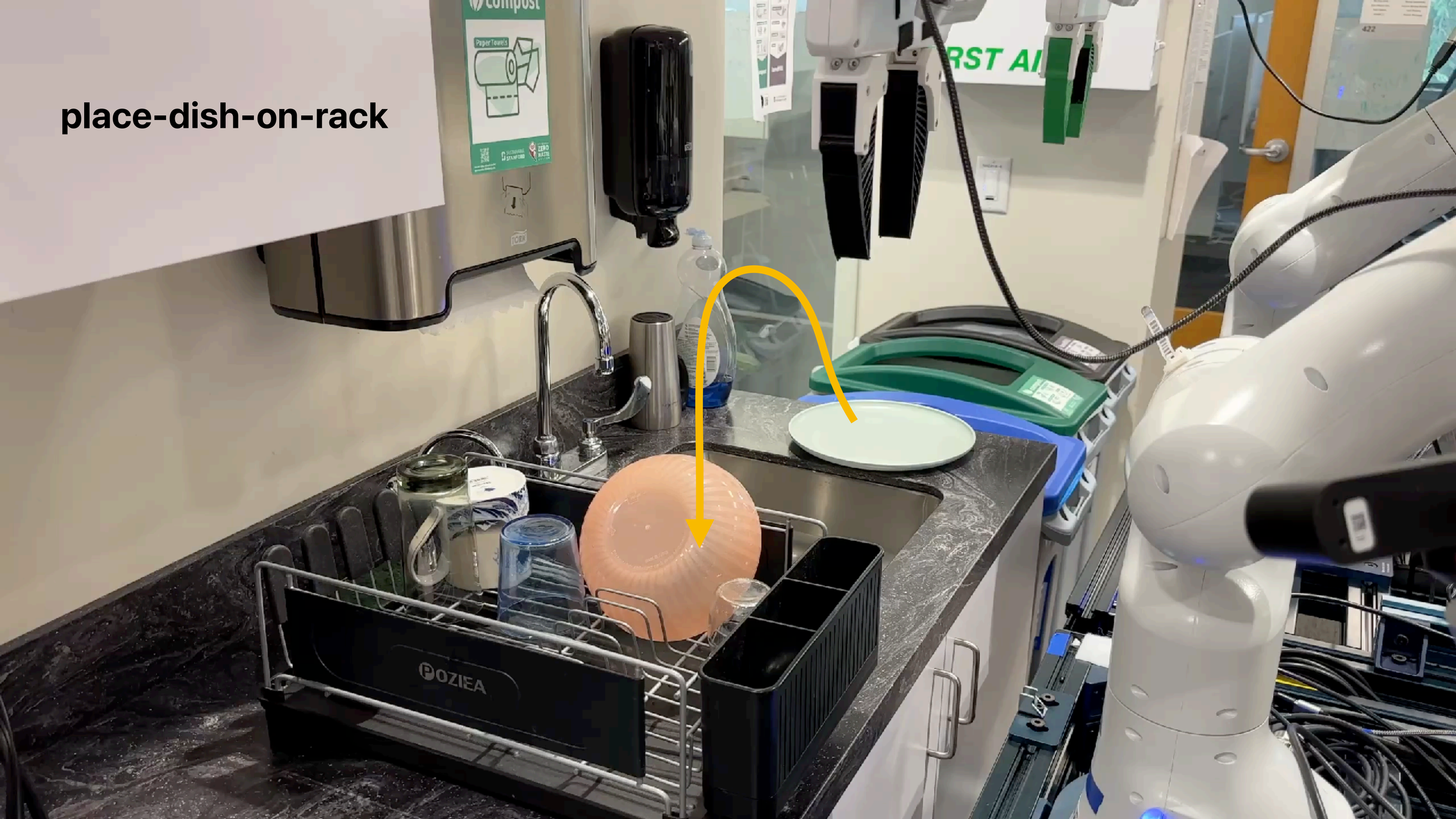
When language alone is not enough

```
(:action          place-dish-on-rack
 :parameters      (?d - dish ?r - dish-rack)
 :precondition    (and (empty ?r) (holding ?d))
 :effect          (and (on ?d ?r) (not (empty ?r)) (not (holding ?d))))
```

(empty ?r)




place-dish-on-rack
compost
Paper Towels
POZIEA

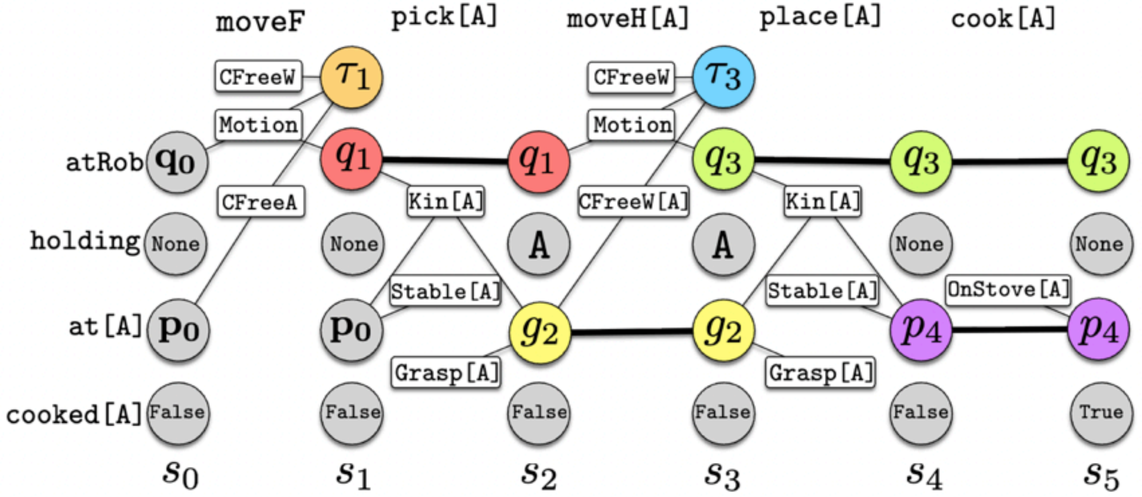

place-dish-on-rack
Compost
Paper Towels
POZIEA

place-dish-on-rack

Integrated Task and Motion Planning

Caelan Reed Garrett [1], Rohan Chitnis [1], Rachel Holladay [1], Beomjoon Kim [1], Tom Silver [1], Leslie Pack Kaelbling [1] and Tomás Lozano-Pérez [1]

[1]CSAIL, MIT, Cambridge, USA, 02139; email: caelan@csail.mit.edu





Step 1: Discovering Skills using Foundation Models

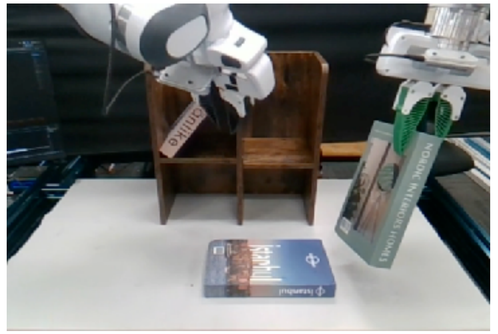
Language Goal:
"Place book on shelf"



Teleoperated Demonstration

Video Language Model







Move | Grasp: Green Book | Adjust: Pink Book | Transit | Place: Green Book, Shelf

Step 1: Discovering Skills using Foundation Models

Language Goal:
"Place book on shelf"



Video Language Model

Temporal Segments







Move | Grasp: Green Book | Adjust: Pink Book | Transit | Place: Green Book, Shelf

Temporal Structure

Step 1: Discovering Skills using Foundation Models

Language Goal:
"Place book on shelf"



Video Language Model

Temporal Segments







Move | Grasp: Green Book | Adjust: Pink Book | Transit | Place: Green Book, Shelf

Step 1: Discovering Skills using Foundation Models

Language Goal:
"Place book on shelf"



Video Language Model

Temporal Segments







Move | Grasp: Green Book | Adjust: Pink Book | Transit | Place: Green Book, Shelf

Detection & Segmentration Model

Bookshelf



Green Book Pink Book

Step 1: Discovering Skills using Foundation Models

Language Goal:
"Place book on shelf"



Video Language Model

Temporal Segments







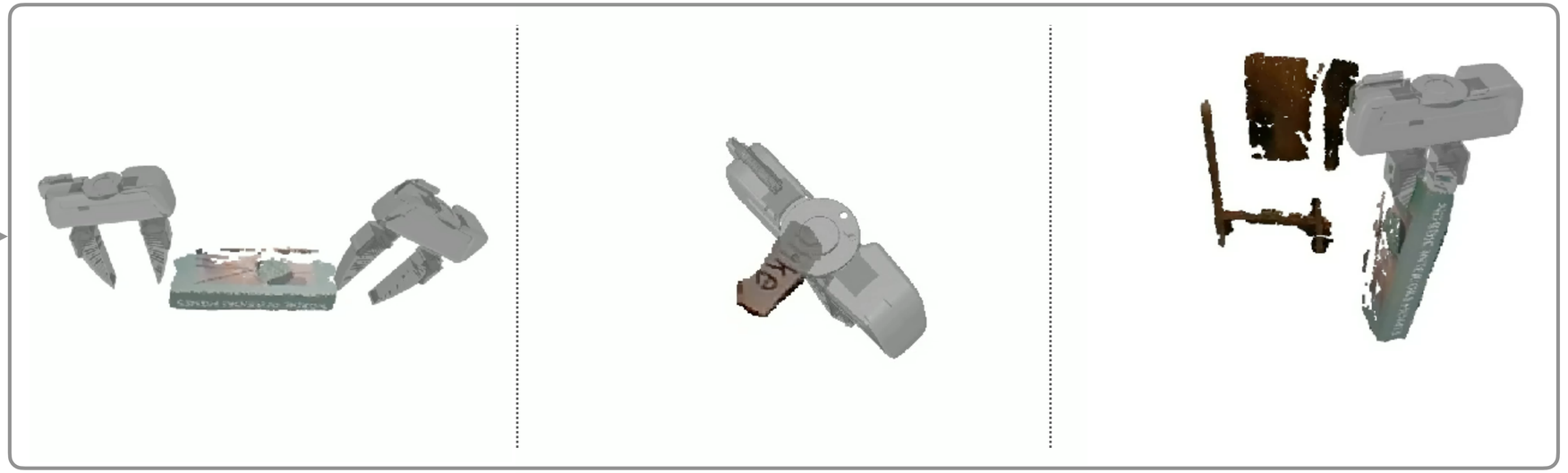
Move | Grasp: Green Book | Adjust: Pink Book | Transit | Place: Green Book, Shelf

Detection & Segmentration Model

Bookshelf



Green Book Pink Book



Relevant Entity Point Cloud & Demonstrated Actions

Step 1: Discovering Skills using Foundation Models

Language Goal:
"Place book on shelf"

Video Language Model

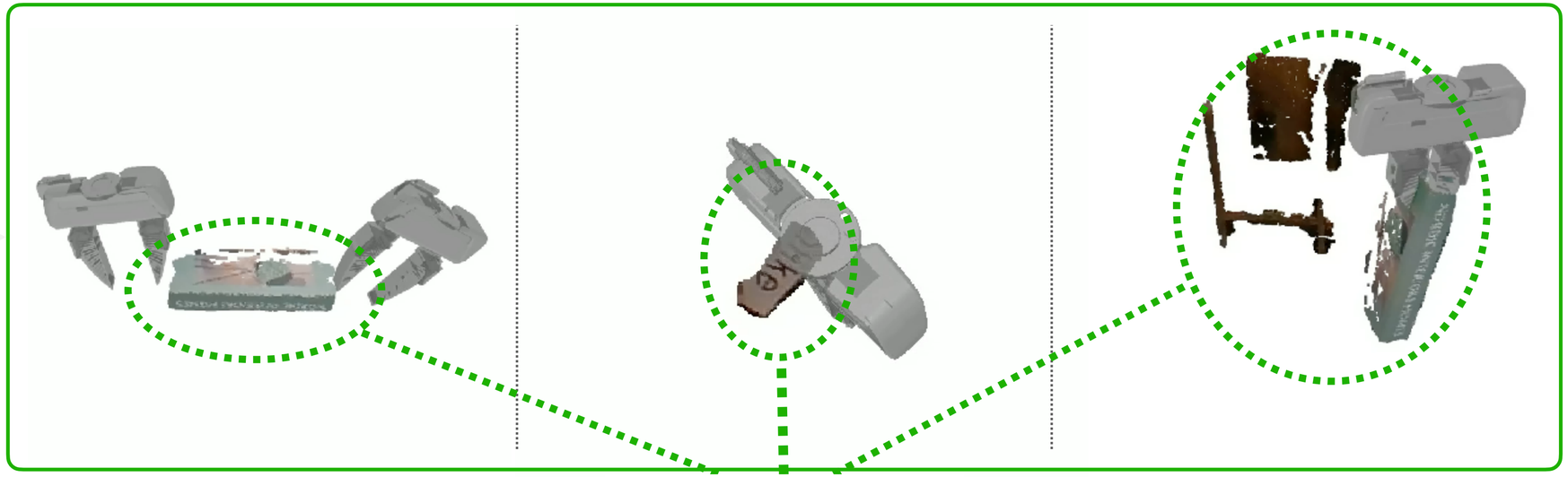
Temporal Segments

Move | Grasp: Green Book | Adjust: Pink Book | Transit | Place: Green Book, Shelf

Detection & Segmentration Model

Bookshelf

Green Book Pink Book



Spatial Structure

Step 1: Discovering Skills using Foundation Models

Language Goal:
"Place book on shelf"



Video Language Model

Temporal Segments



Grasp: Green Book



Adjust: Pink Book



Place: Green Book, Shelf

Detection & Segmentration Model

Bookshelf



Green Book Pink Book

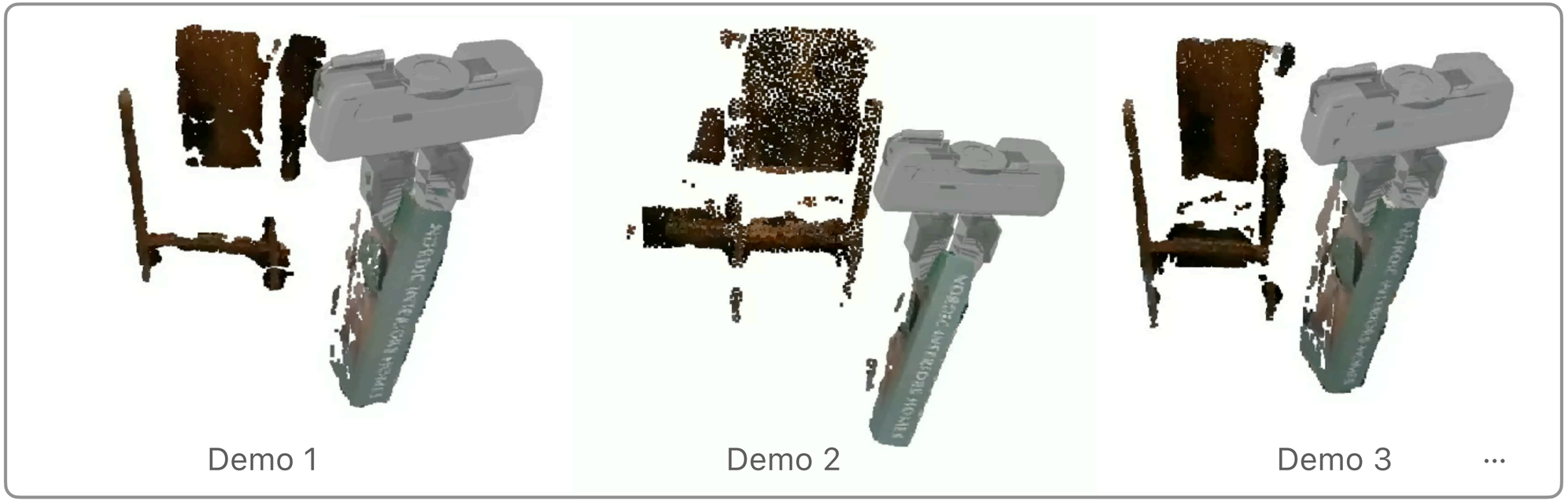






Relevant Entity Point Cloud & Demonstrated Actions

Step 1: Discovering Skills using Foundation Models



Demonstrations for Skills

... ...

Place: Green Book, Shelf

5-10 collected trajectories

>7000 synthesized trajectories

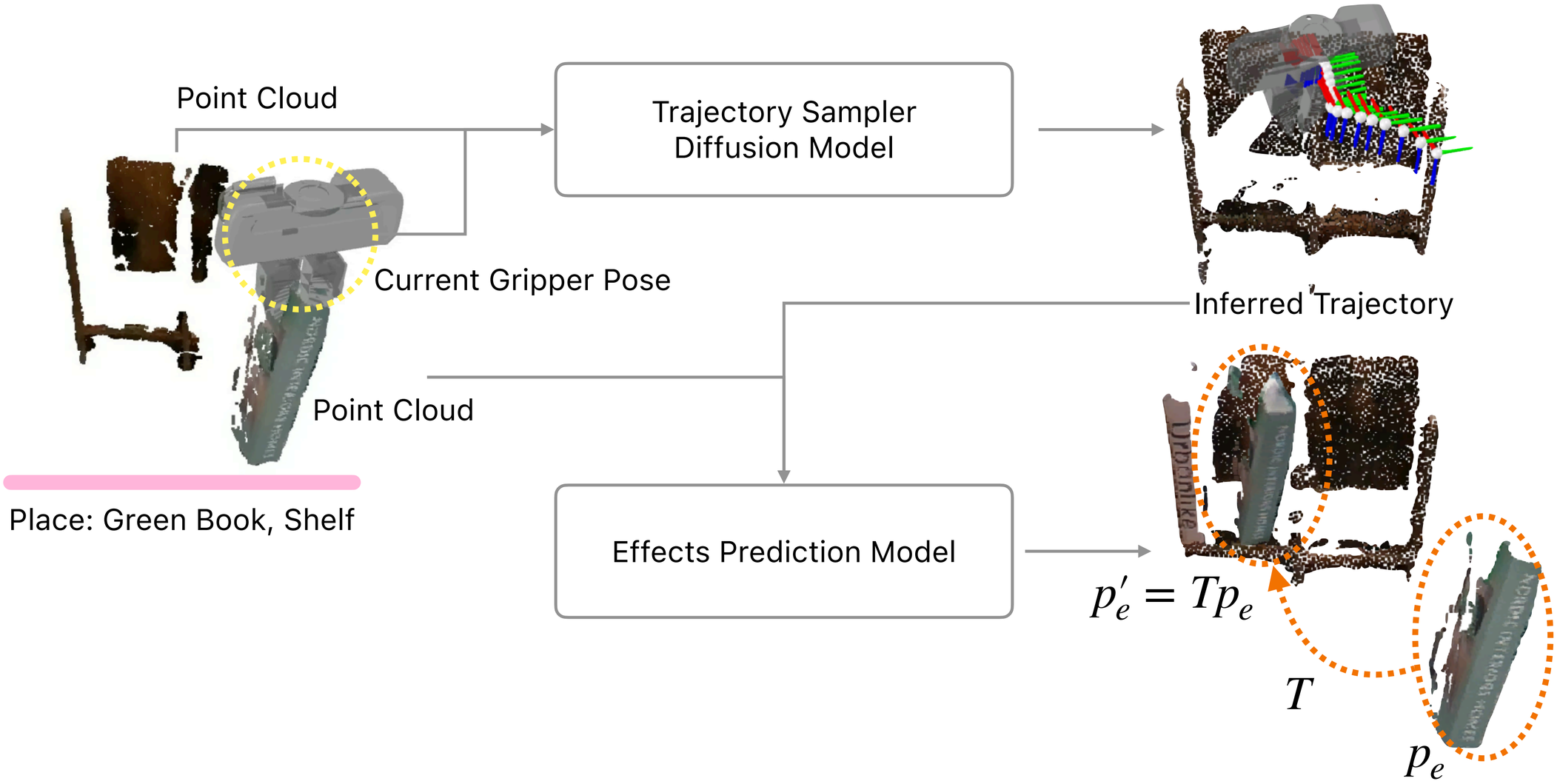
Step 2: Learning Composable Skills



Place: Green Book, Shelf

Step 2: Learning Composable Skills

Train two models for each composable skill



Step 3: Planning with Learned Skills

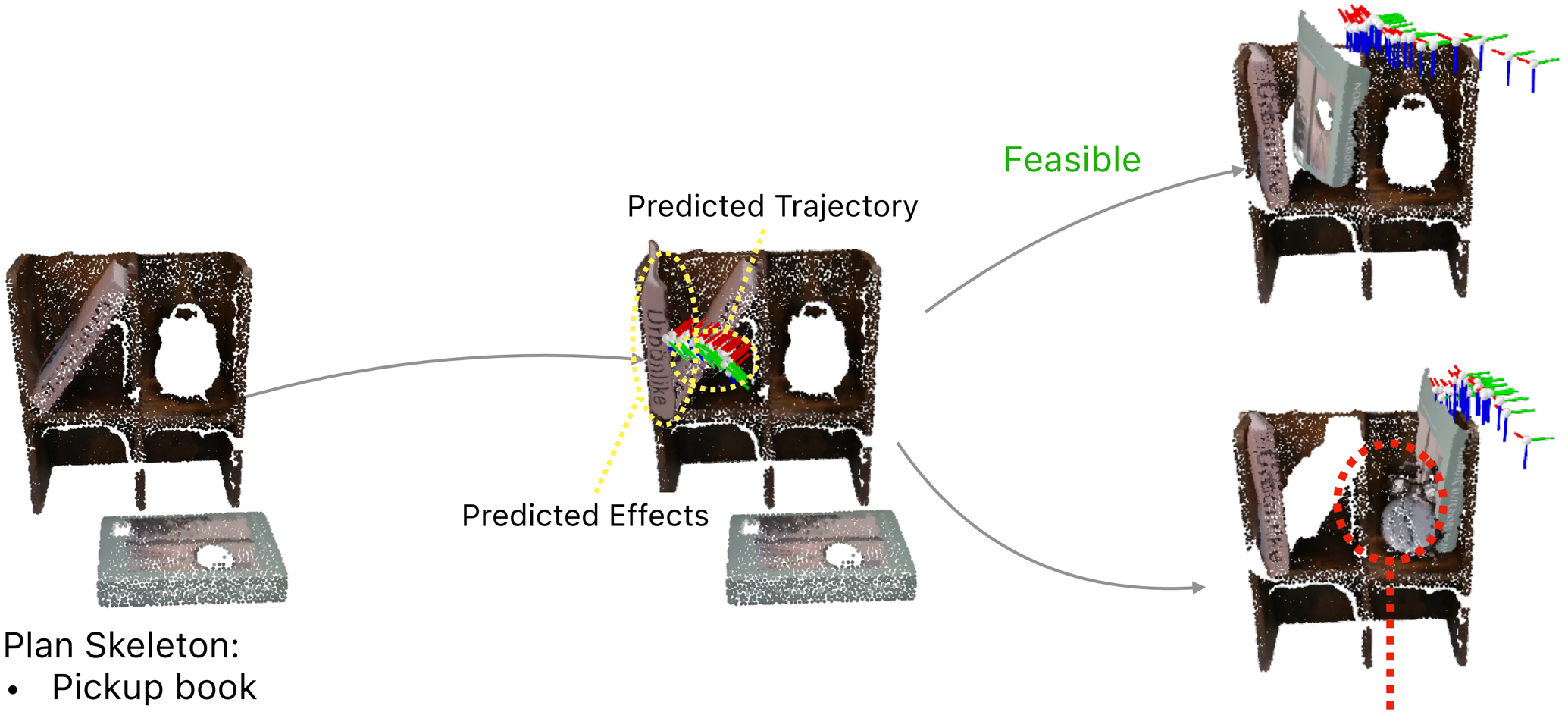


Language Goal: *“Place book on shelf”*

Step 3: Planning with Learned Skills



Step 3: Planning with Learned Skills



Plan Skeleton:

- Pickup book
- Adjust book left
- Place book

Step 3: Planning with Learned Skills

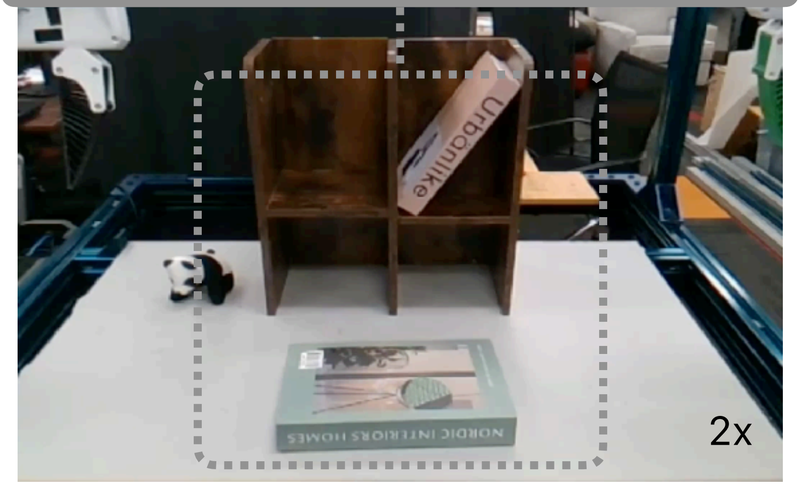


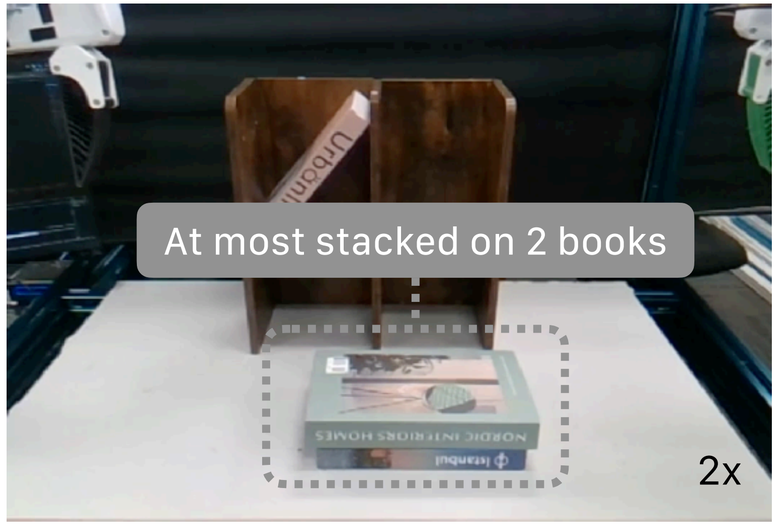
“Place book on shelf”

8x

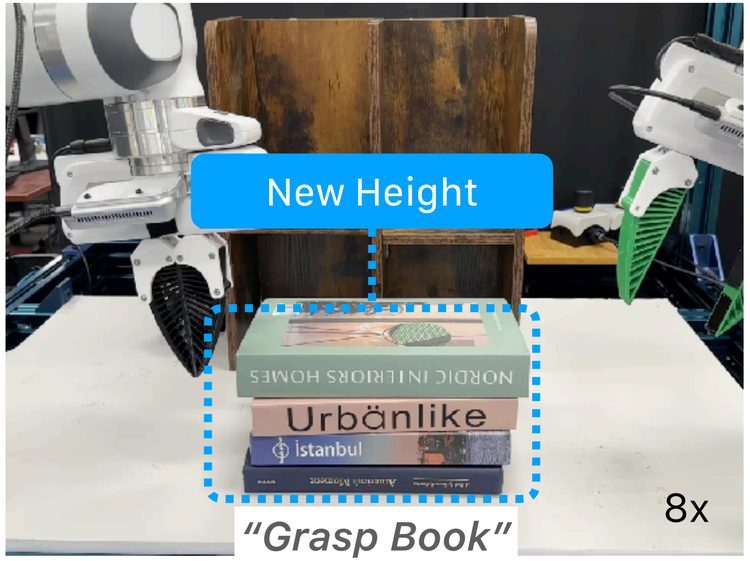
New Scene Configuration

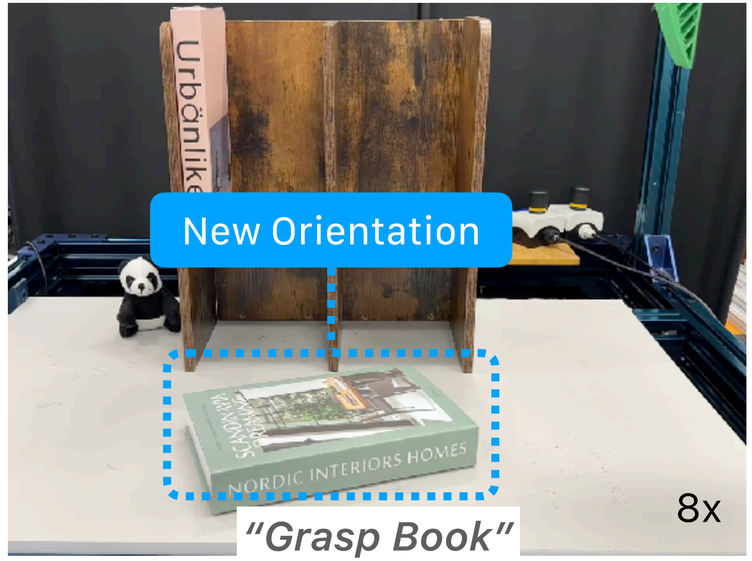
Objects are axis-aligned, little variation

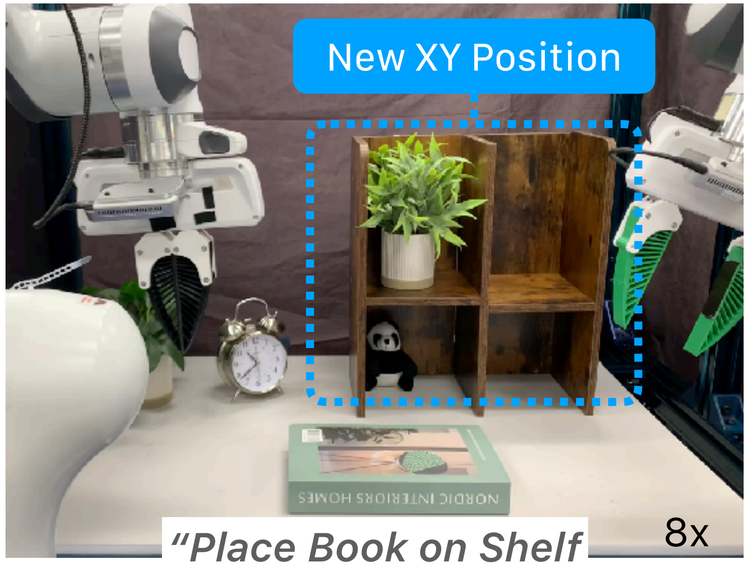


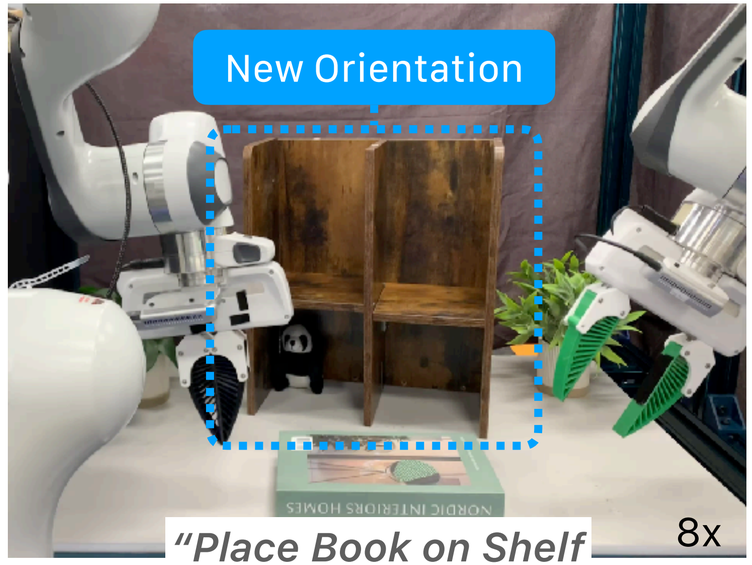


Training Demonstrations









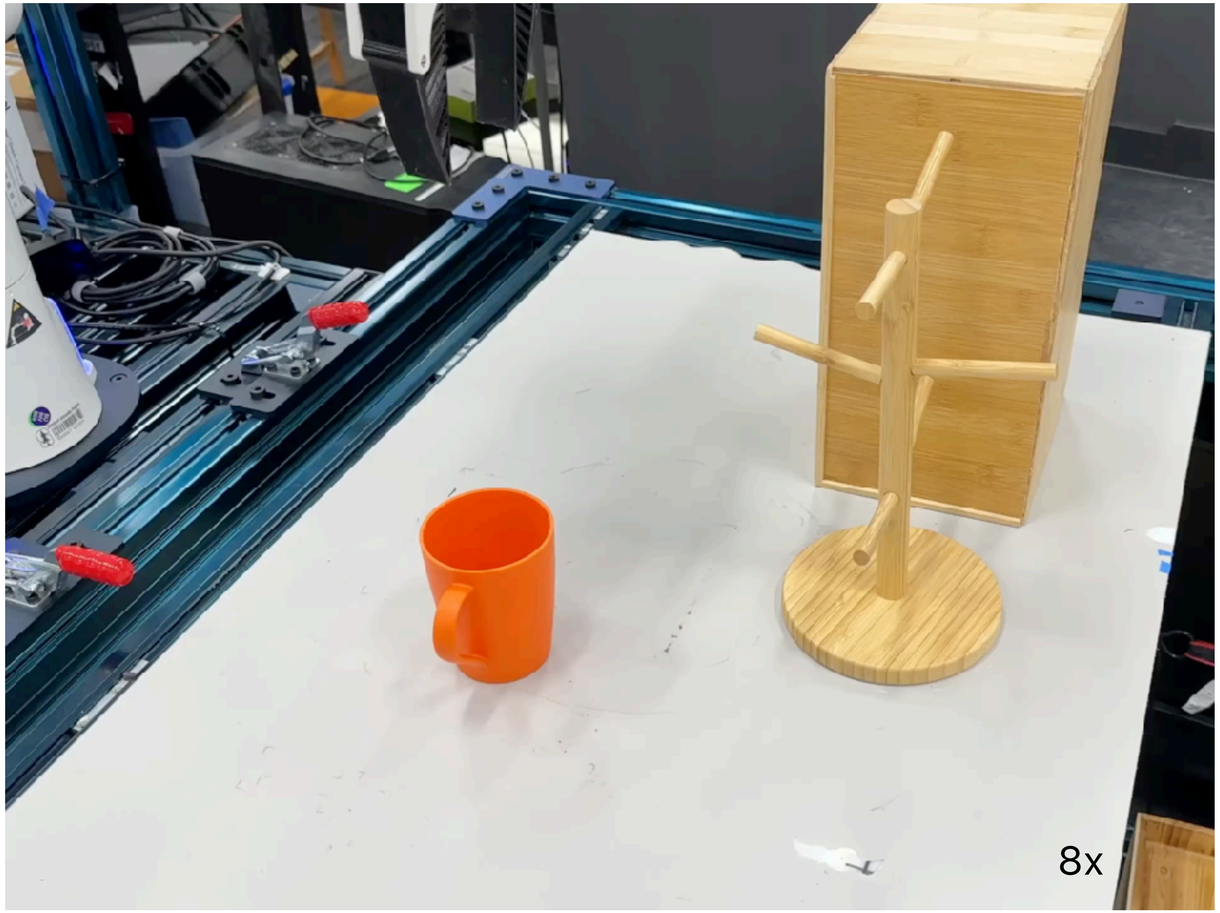
New Scene Configurations

New Geometric Constraints

Training Data: A Common Set of **10x** Demos Per Peg, **2** Steps: Grasp Mug → Hang Mug



“Hang the mug on mug tree”

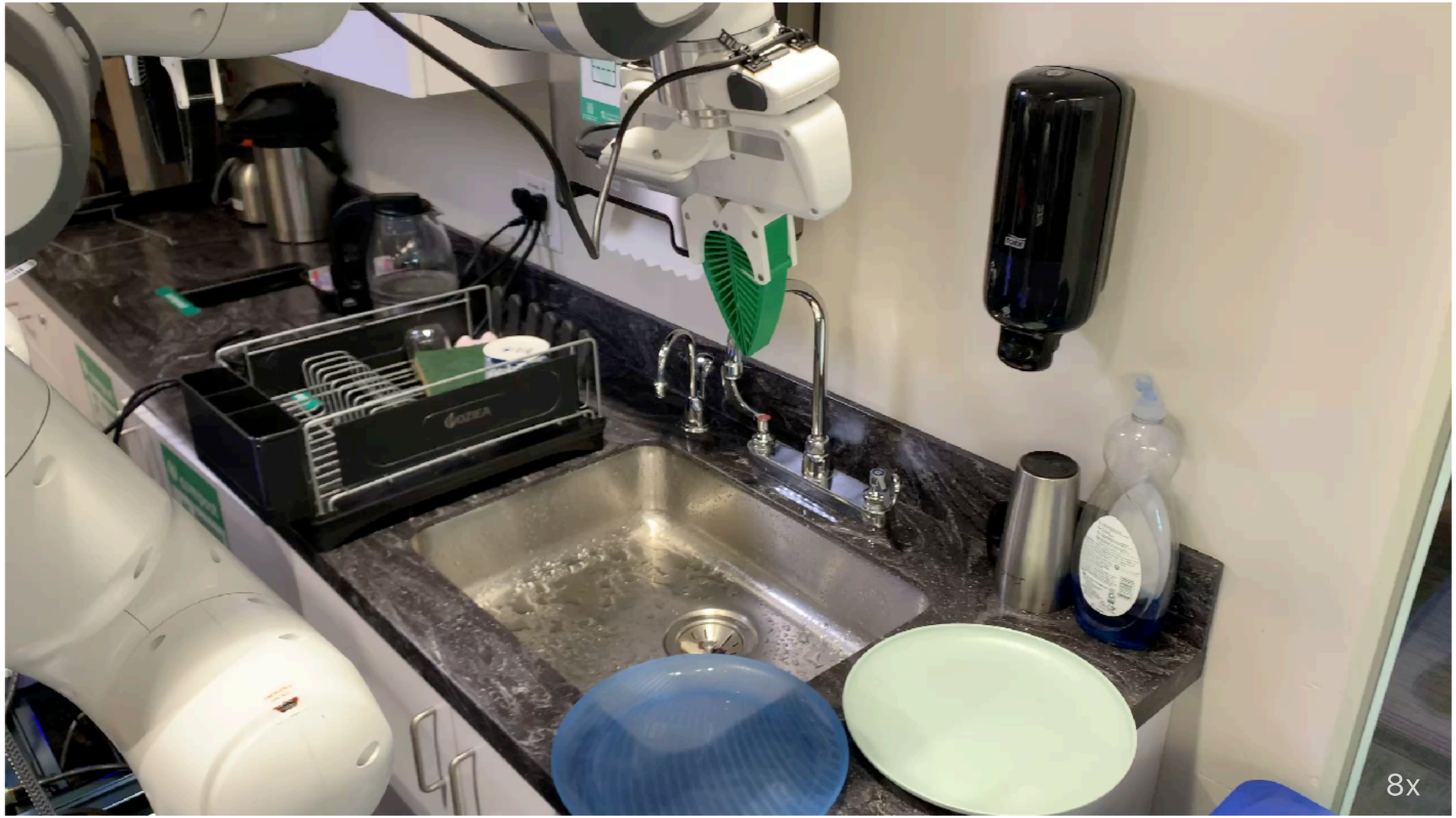


“Hang the mug on mug tree”

Kitchen

Longer Task Horizon

Training Data: A Common Set of **10x** Demos, **5** Steps: Turn on Faucet → Grasp Plate → Wash Plate → Place Plate → Turn off Faucet



8-Step Task: *“Wash, and put away two plates”*

Learning abstractions for planning

1. Learn state and action abstractions from demos (BLADE)

2. Discover spatial and temporal structure (STACK)

3. **Learn a world model from raw video** (Rhoda)

4. Reason at different levels of abstraction (APIVOT)

What video planners can uniquely address?

Build a predictive model

→ The video model can simulate diverse physical effects

Search for solutions using the predictive model

→ The video model might have learned “heuristics” useful for planning*

Generalize the policy and predictive model

→ The video model can achieve both at the same time

* This is related to works that distill planners into neural networks

From small learned models to web-scale video

Then (2017–2020)

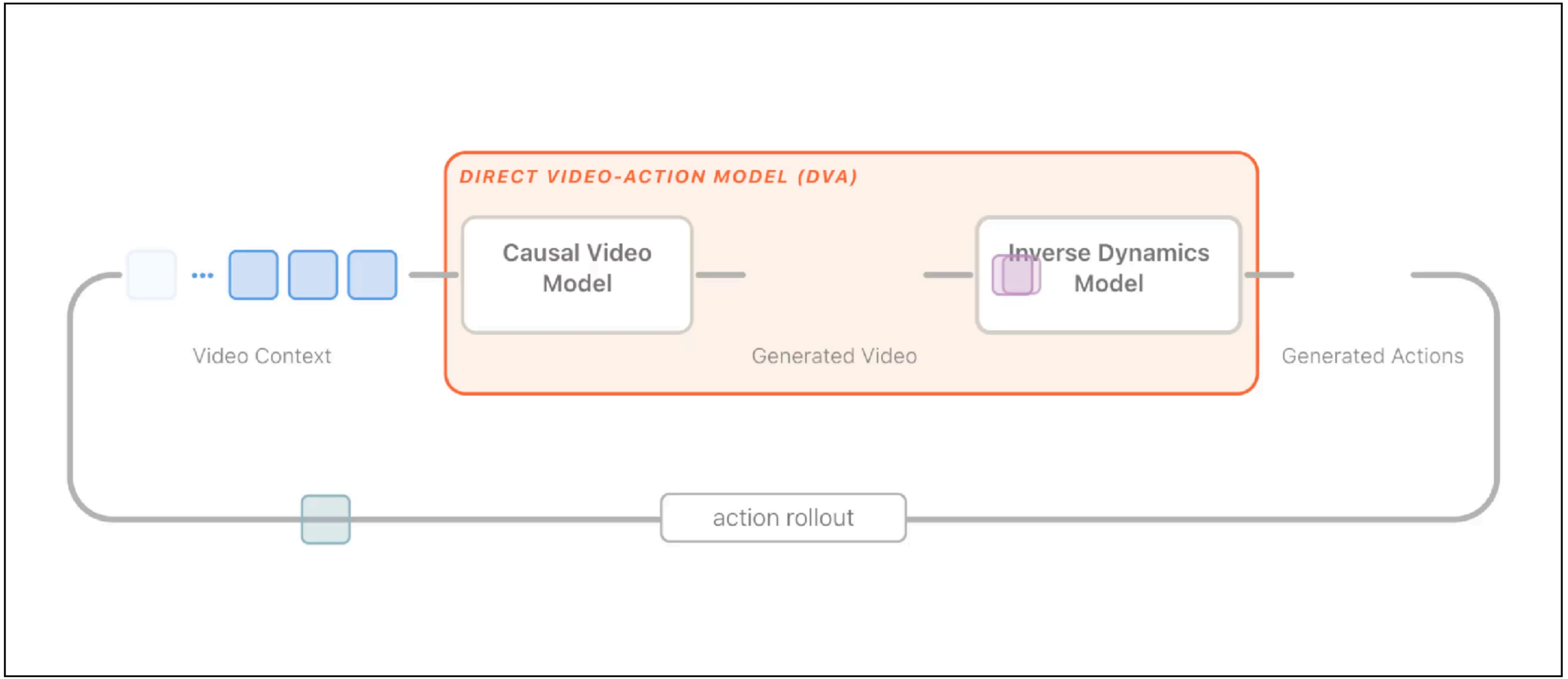
- Visual Foresight
- World Models
- Dreamer
- Small, one environment each

Now

- Web video: the largest record of how the world moves
- Video diffusion works at scale
- Pixels as a universal state

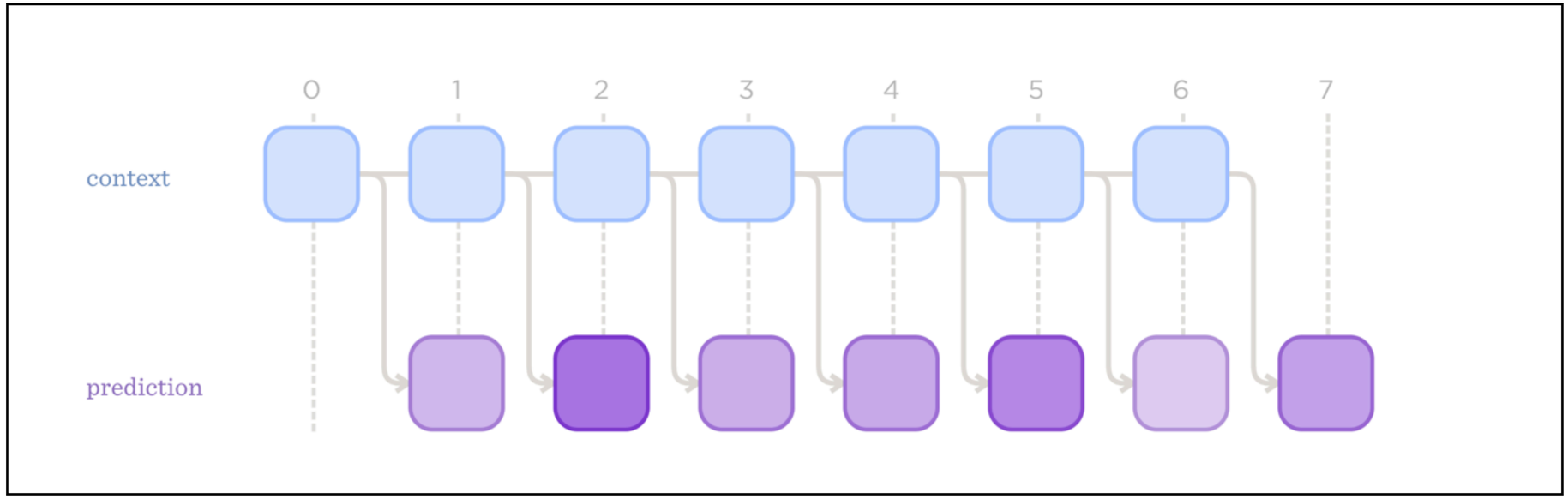
Finn & Levine 2017 · Ha & Schmidhuber 2018 · Hafner et al. 2020

DVA: the video model is the policy



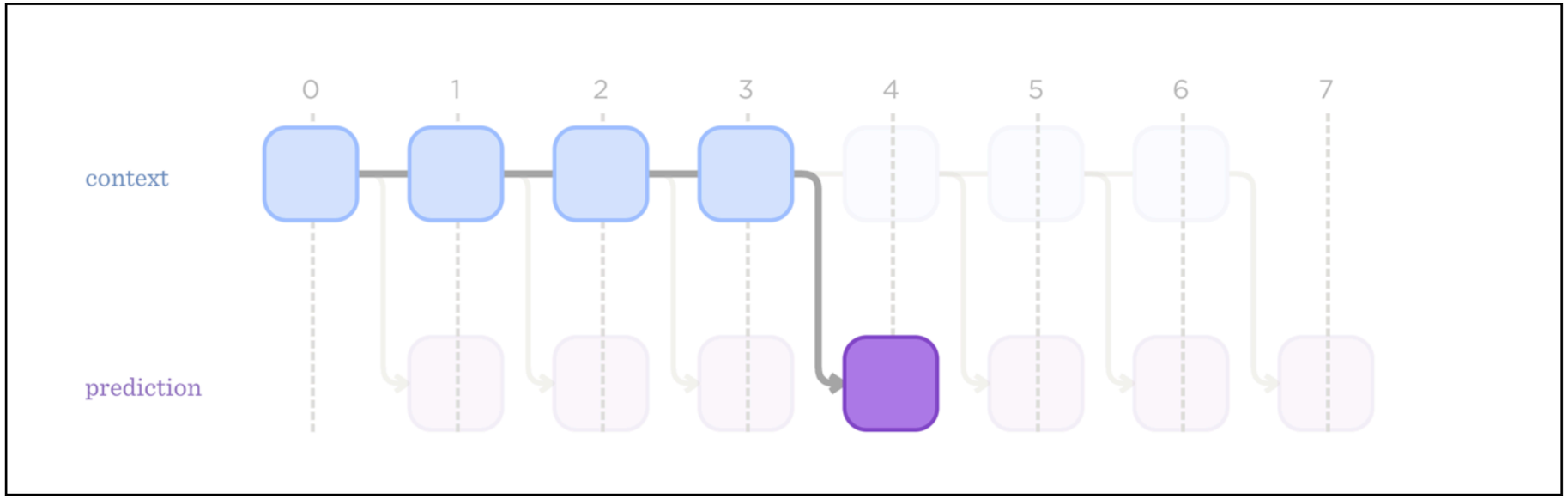
A native causal video model

- Pretrained from scratch on web video: given the past, predict the future
- Training: predict at every position of a long context
- Inference: reuse the encoded context (KV cache)



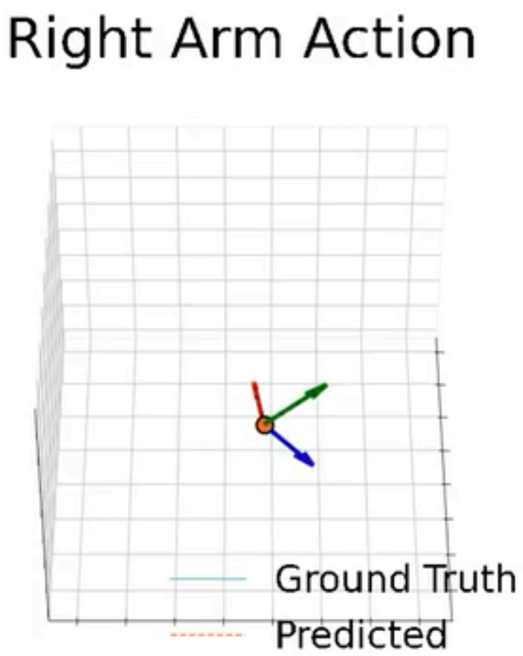
A native causal video model

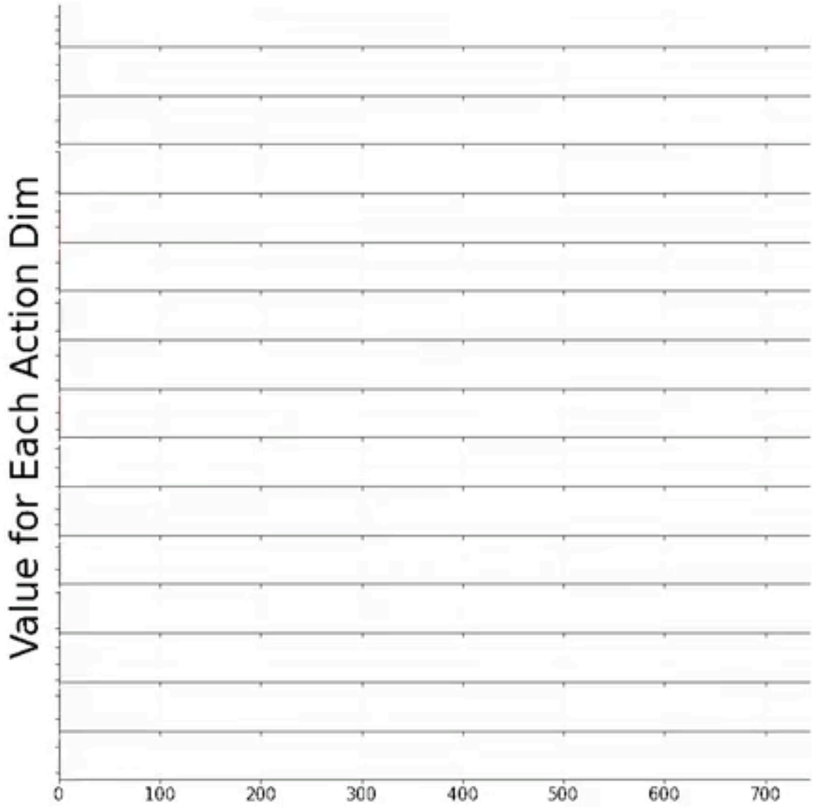
- Pretrained from scratch on web video: given the past, predict the future
- Training: predict at every position of a long context
- Inference: reuse the encoded context (KV cache)



From video to actions: inverse dynamics

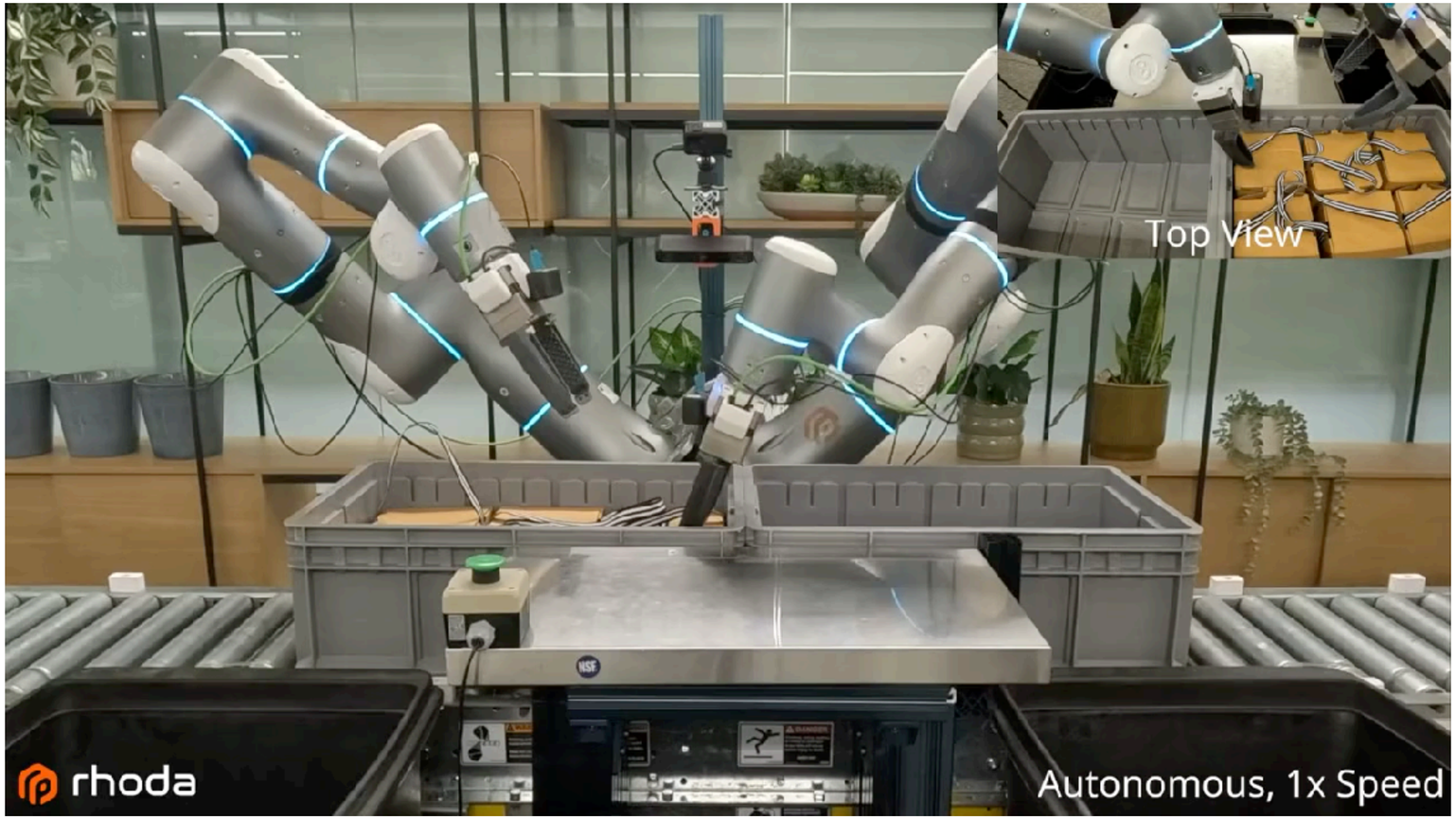






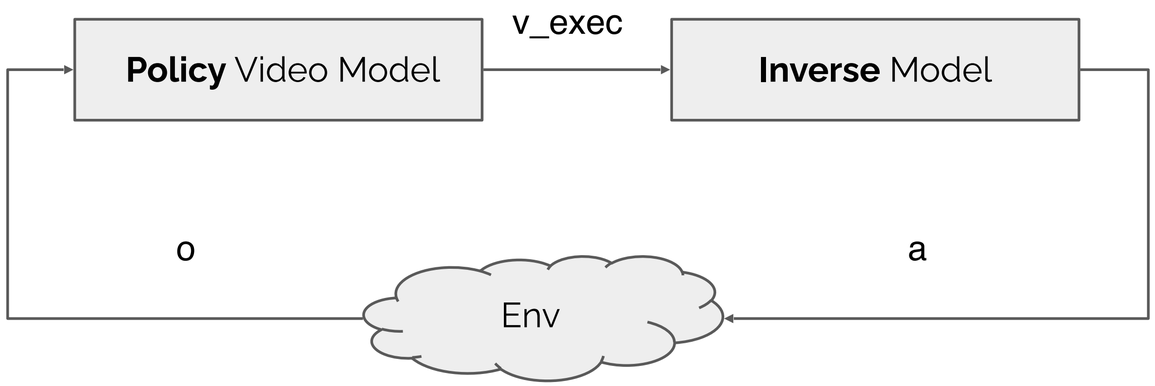
- About 10 hours of data per embodiment type; even random motions work
- Shared across robots and tasks

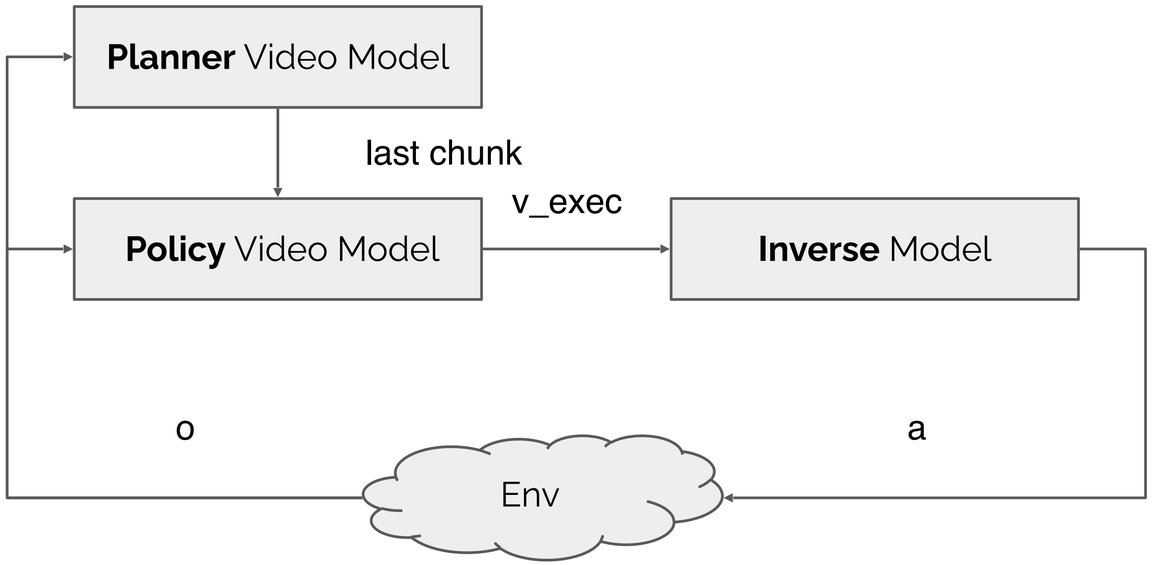
Data-efficient: real deployments



Decanting: 11 hours of robot data, 1.5 hours autonomous

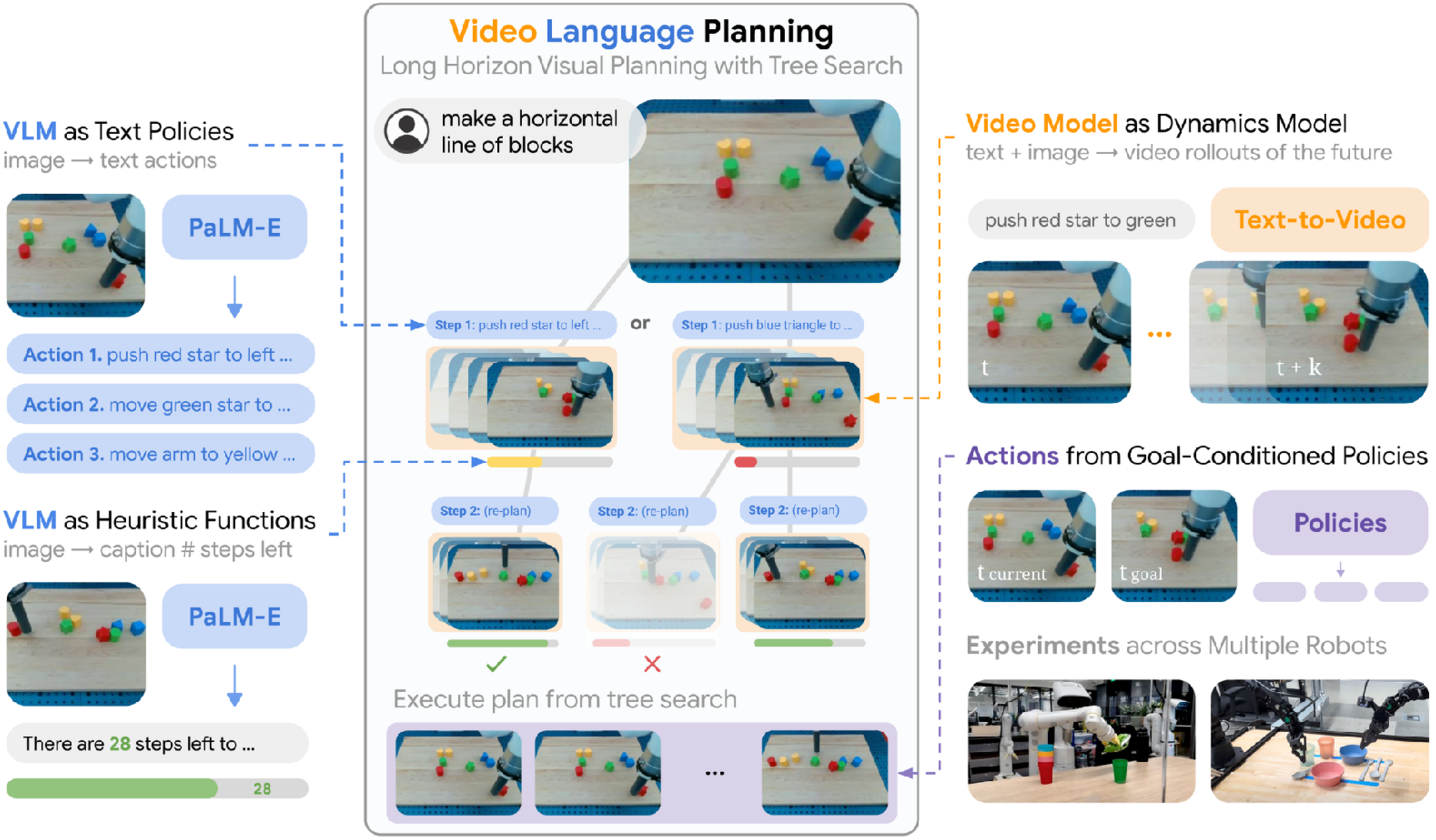
How to do planning with a video model?





Alternatives:

- Language
- Keyframe
- Keypoint/trace



UniPi (Du et al., NeurIPS 2023) · HiP (Ajay et al., NeurIPS 2023) · VLP (Du et al., ICLR 2024, shown)

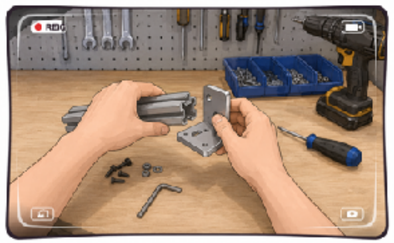
Language Goal

"Move all blocks to the top right corner"

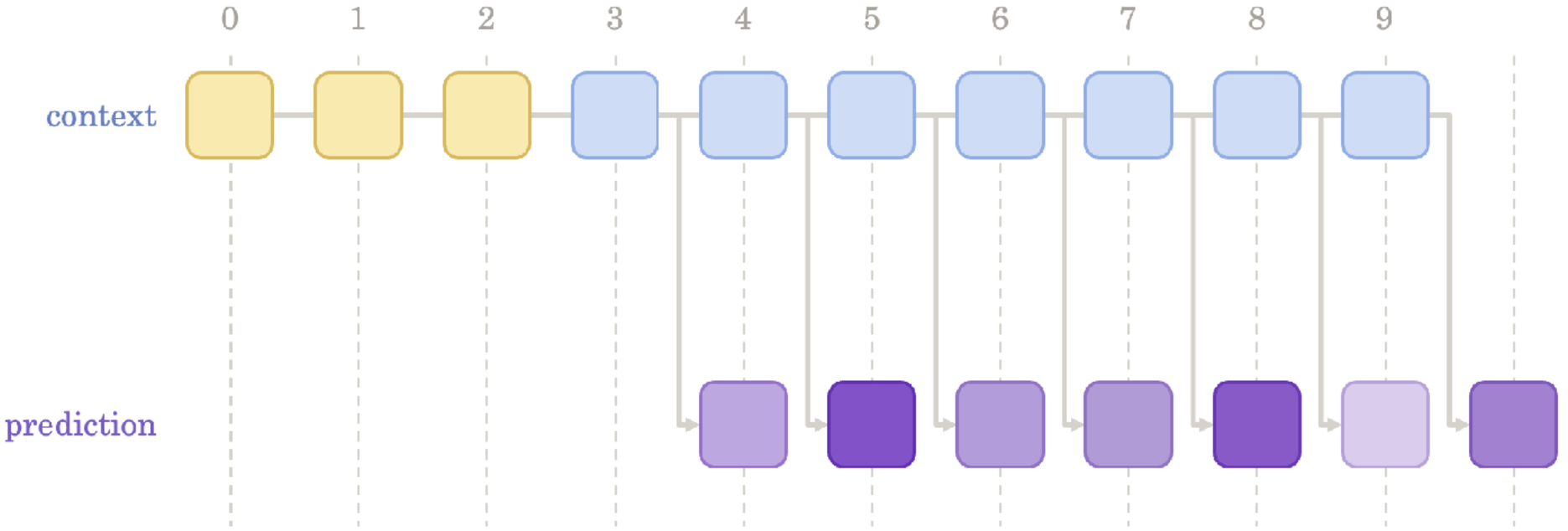
Initial State







previous failed execution / ...



Two challenges

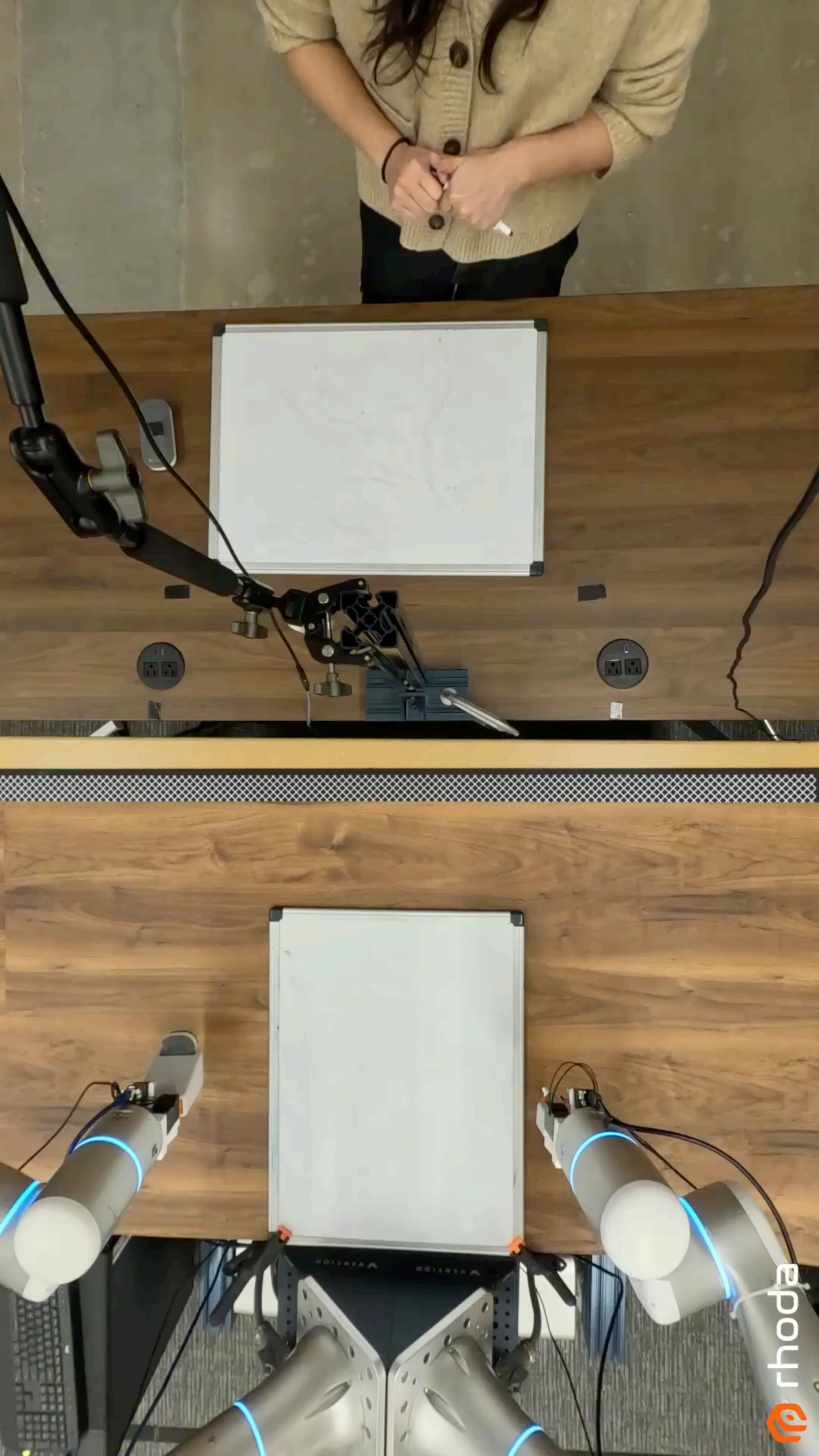
1. Long context

2. Learning to follow


rhoda
Autonomous, 1x Speed

rhoda


rhoda
4x Speed
rhoda
4x Spee



Learning abstractions for planning

1. Learn state abstractions and behaviors from demos (BLADE)

2. Discover spatial and temporal structure (STACK)

3. Learn a world model from raw video (Rhoda)

4. **Reason at different levels of abstraction** (APIVOT)

"Store the leftovers in the fridge"

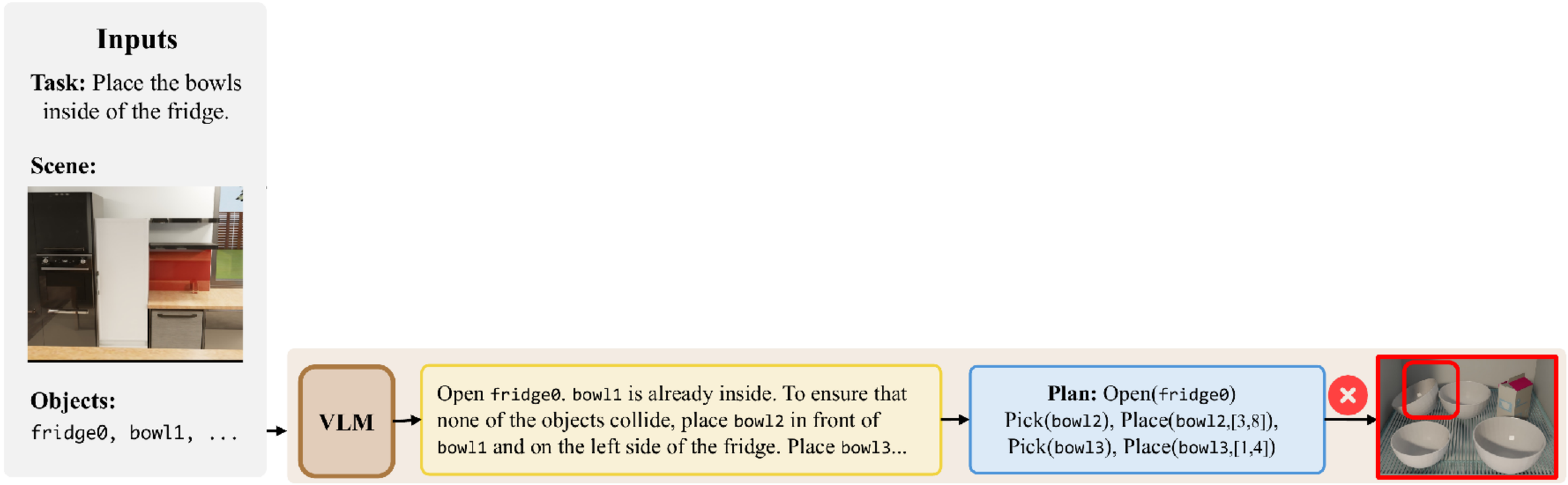
Semantic

- Which items?
- Which containers?
- Open the fridge first

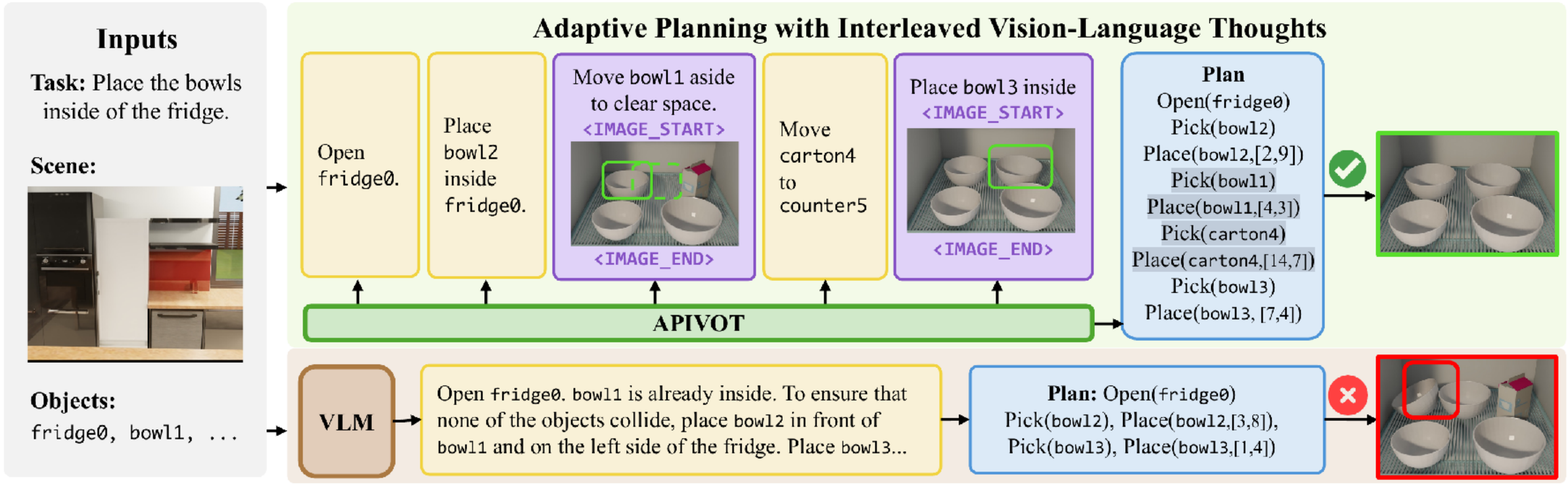
Geometric

- Does it fit?
- What must move?
- Will the next item still fit?

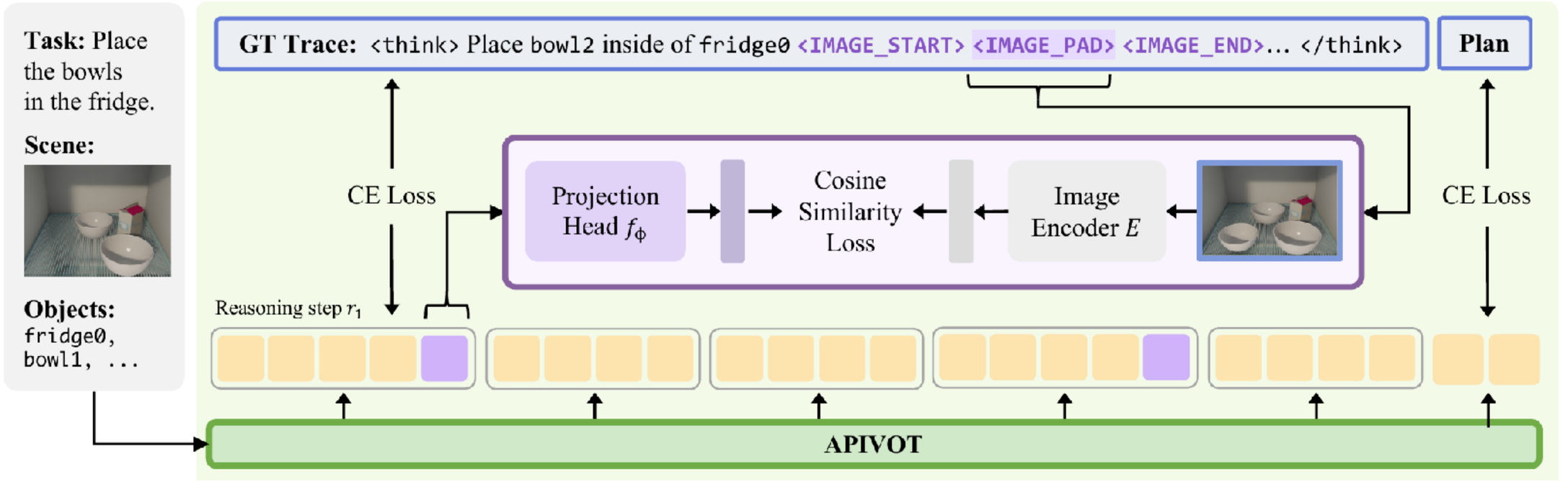
Text-only plans look right but collide



Text-only plans look right but collide



From labels to reasoning traces



- An LLM expands the labeled skeleton into reasoning text
- Visual thoughts aligned to the true future frame (cosine loss)

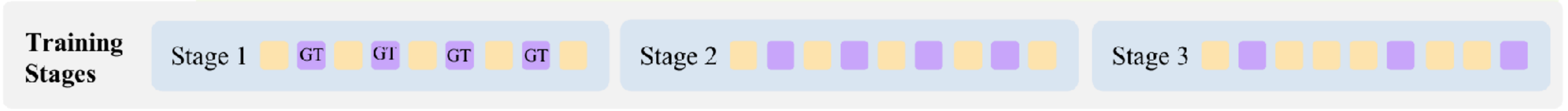
Which steps need imagination?

- **Symbolic precondition**: text only
  - e.g., open the fridge first
- **Current geometric constraint**: imagine if tight
  - e.g., a big bowl into a nearly full fridge
- **Future feasibility**: imagine if tight
  - e.g., place the small bowl so the big one still fits

Tight = small free-space margin, computed in the simulator

Three-stage curriculum

1. Comprehend: use ground-truth visual thoughts
2. Generate: produce visual thoughts itself
3. Adapt: decide when a visual thought is needed



Drop Stage 1 or Stage 2: success falls from 0.419 to 0.301 or 0.345

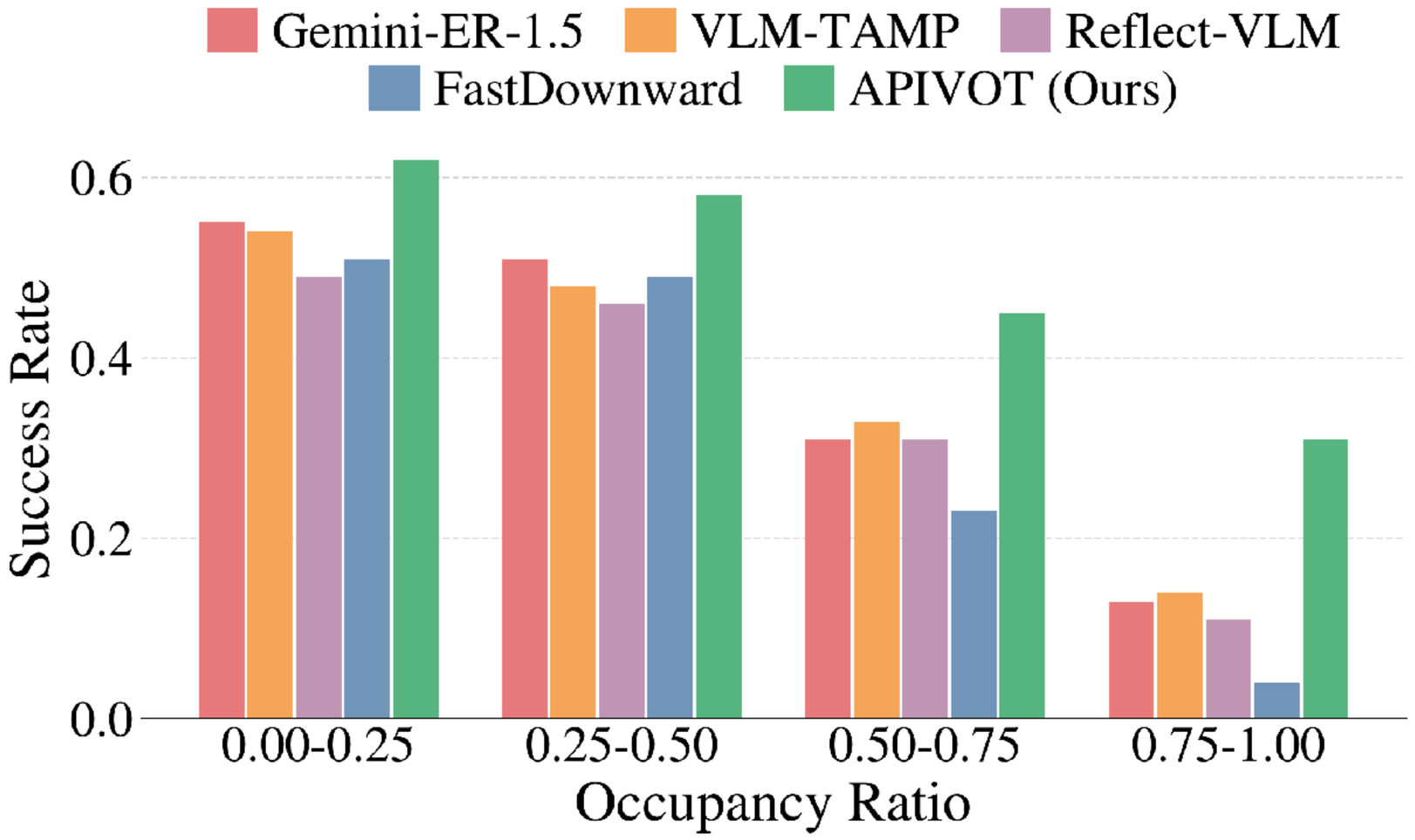
Emily Jin, Joy Hsu, Yiqing Xu, Weiyu Liu†, Nick Haber†, Jiajun Wu†. APIVOT, NeurIPS 2026

APIVOT plans best

Success rate	Avg	Contain	Sort	Store (held out)
Qwen3-VL-8B (base model)	0.188	0.218	0.183	0.162
Gemini-3.1-Pro (frontier VLM)	0.245	0.281	0.241	0.213
Gemini-ER-1.5 (best VLM)	0.338	0.364	0.339	0.311
VLM-TAMP (best planner)	0.329	0.370	0.311	0.307
APIVOT	0.419	0.472	0.421	0.365

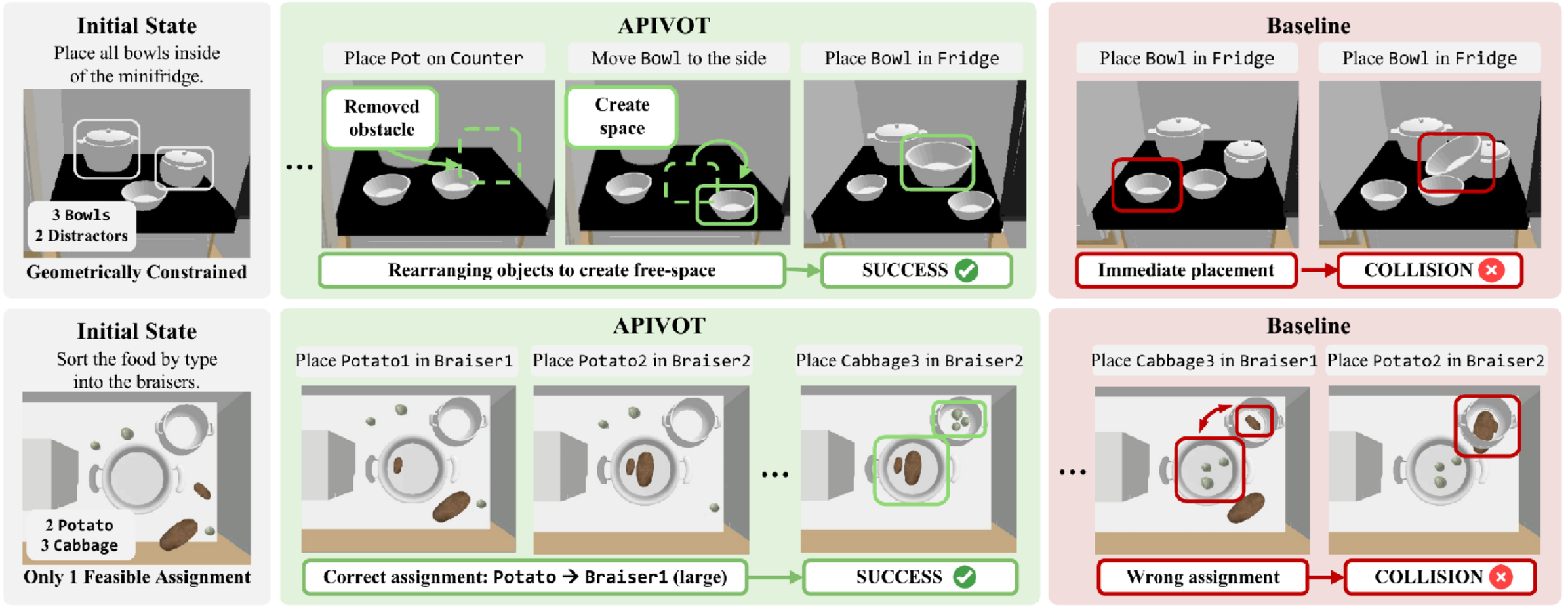
Emily Jin, Joy Hsu, Yiqing Xu, Weiyu Liu†, Nick Haber†, Jiajun Wu†. APIVOT, NeurIPS 2026 · Table 1 (all baselines in the paper)

Gains grow as geometry gets tighter



Gap over the best baseline grows from 7 to 17 points

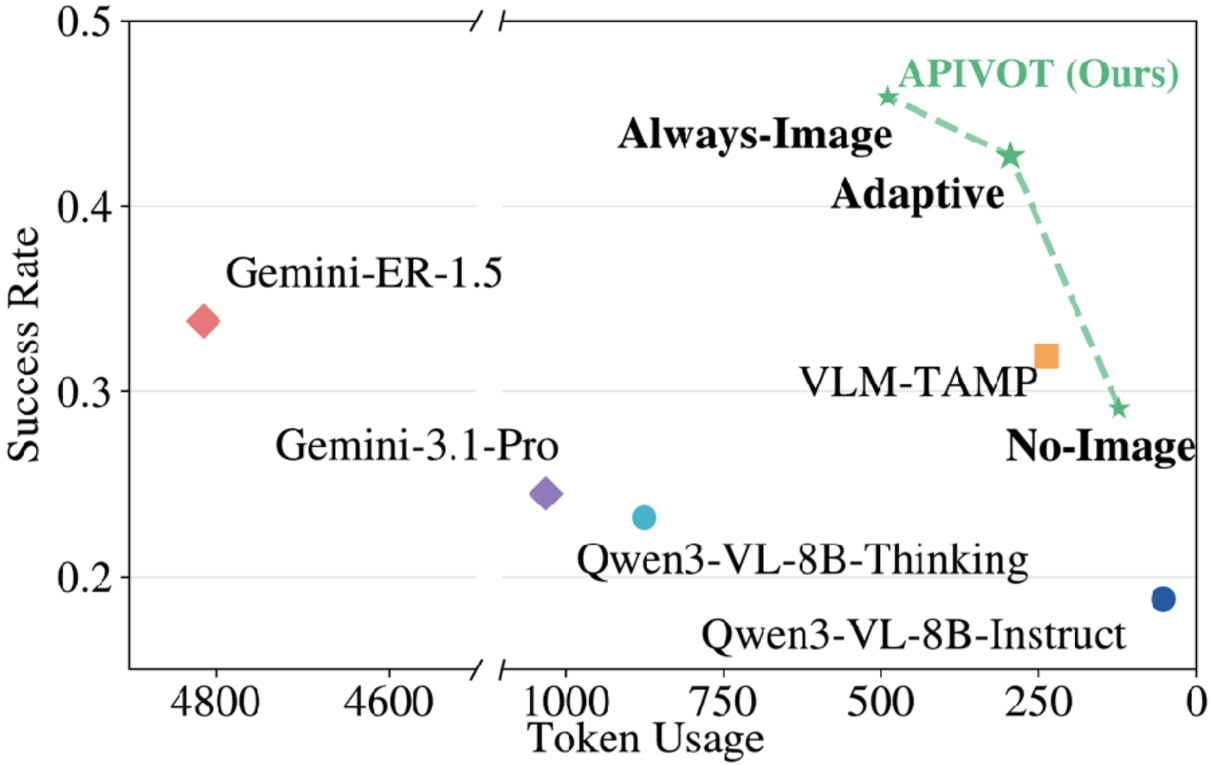
APIVOT clears space first



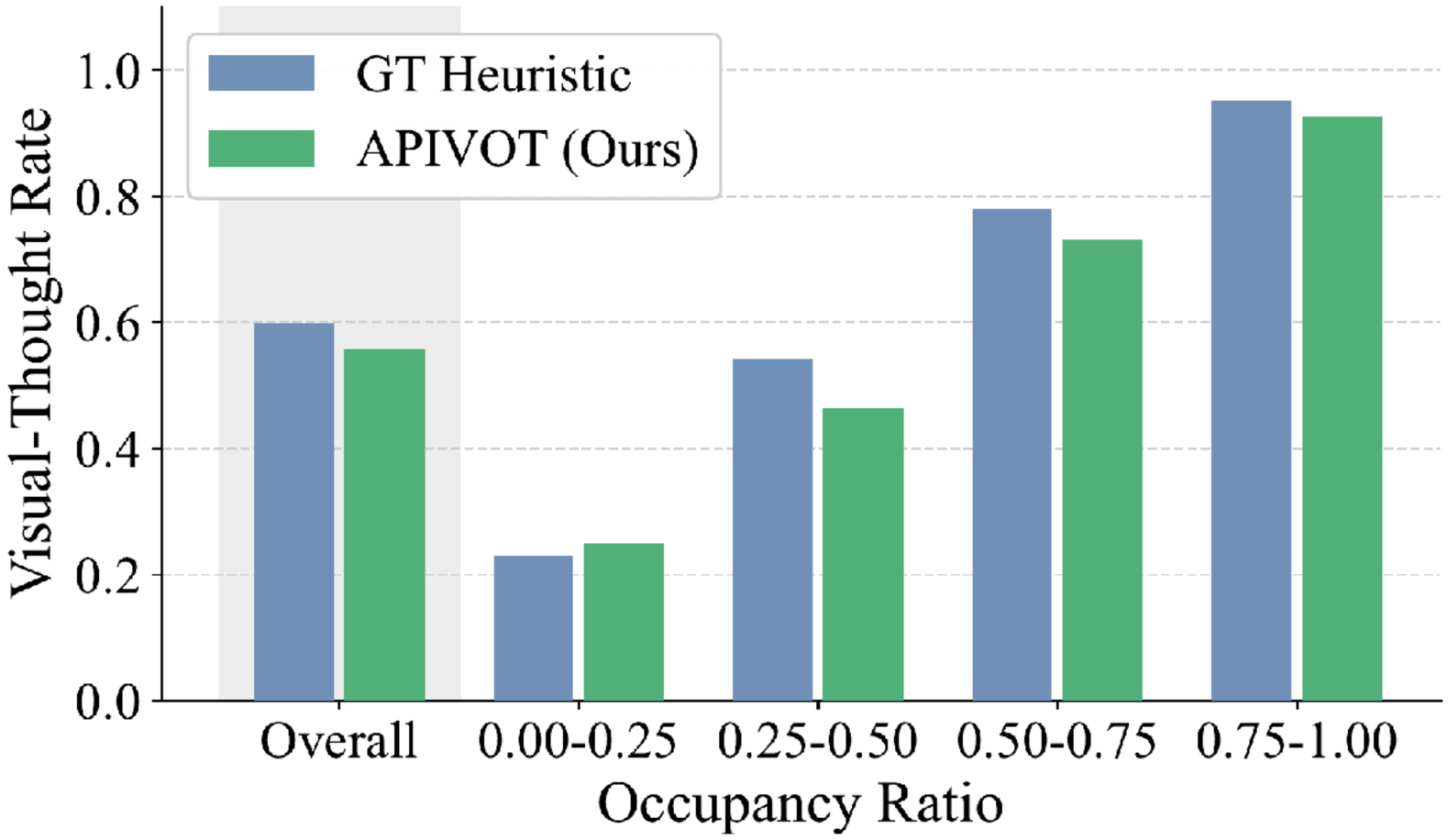
It's the visual thoughts, used efficiently

- No image: **0.282**
- Always imagine: **0.459**
- Adaptive (APIVOT): **0.419**

91% of the success with 39% fewer tokens

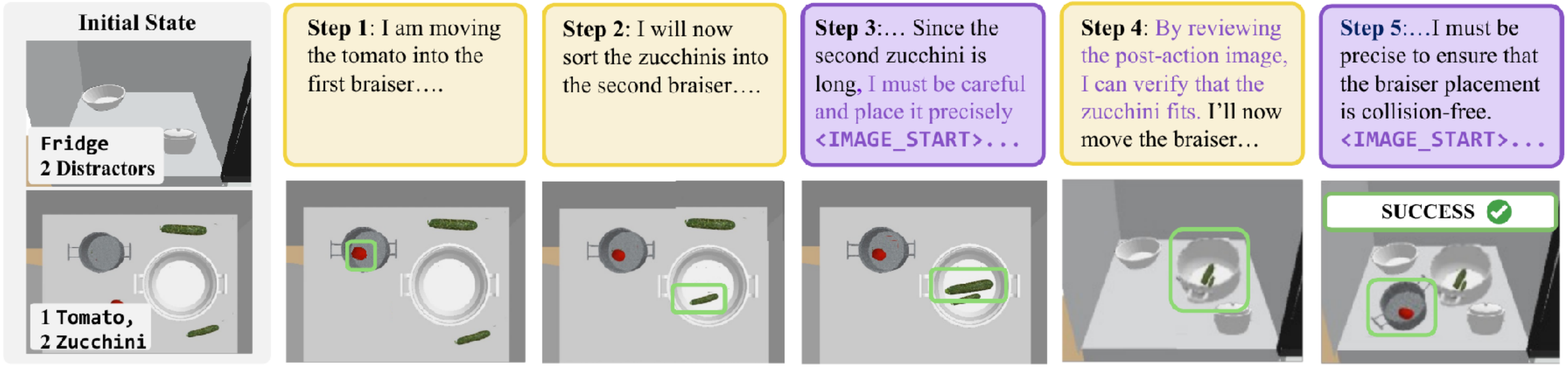


It learned when to imagine



Visual-thought rate tracks how tight the geometry is

Imagine only where it matters



Thank you!

BLADE: blade-bot.github.io

STACK: icra-stack.github.io

Rhoda: rhoda.ai/research

APIVOT: emilyzjin.github.io/projects/apivot.html