

Embodied Vision

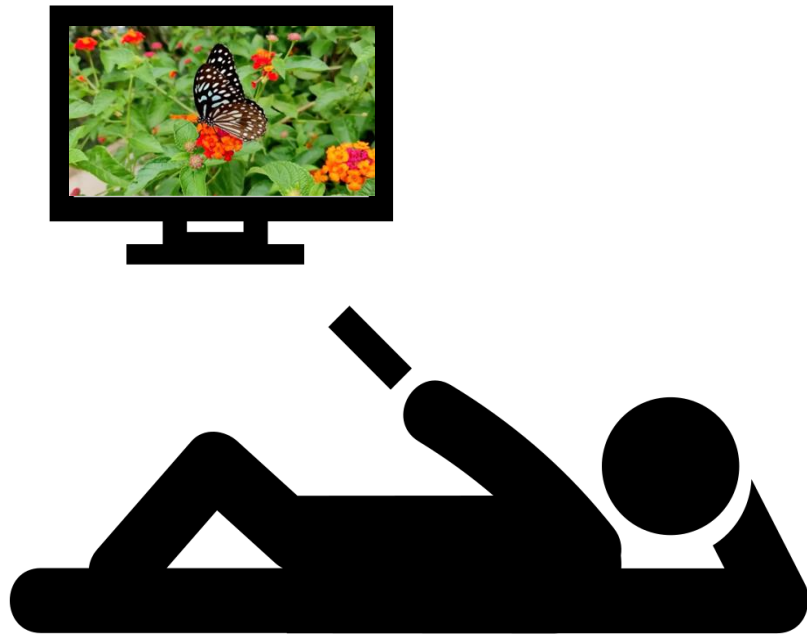
Vision-language Models and Robotics

Tsung-Wei Ke

Spring 2026



Vision is Over-Simplified: Passively Parsing the Information Received from the World

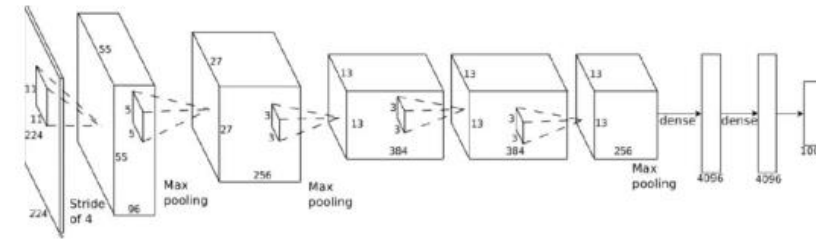


Icon source: <https://www.pngegg.com/en/png-zhyar>

Passive labeling

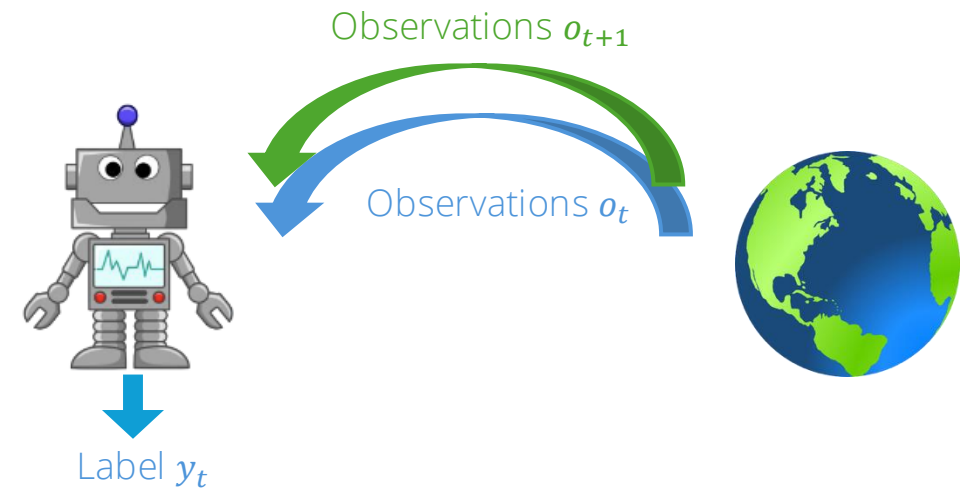


o_t

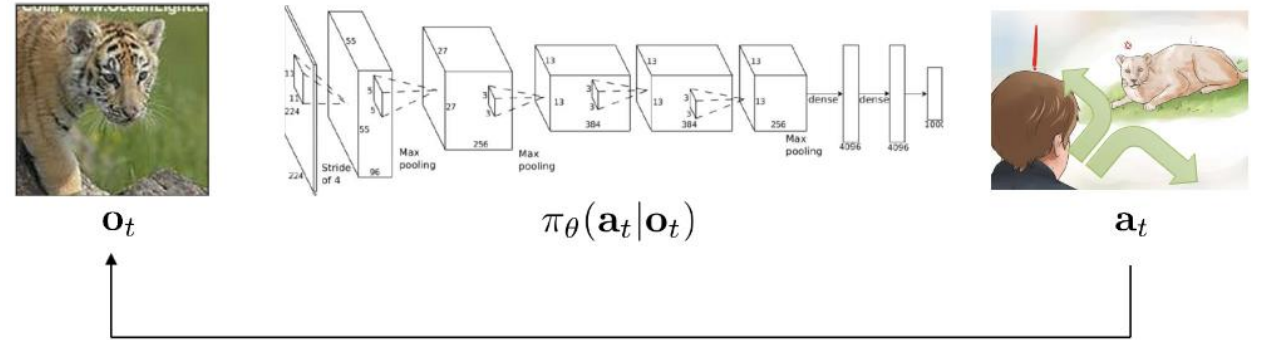
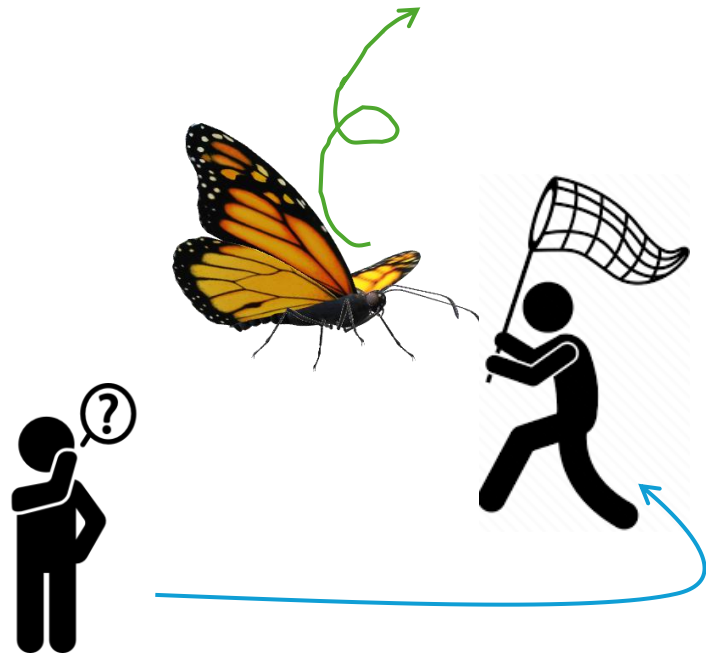


Tiger

y_t



Humans are Embodied Agents: We Act, We Perceive and We Predict



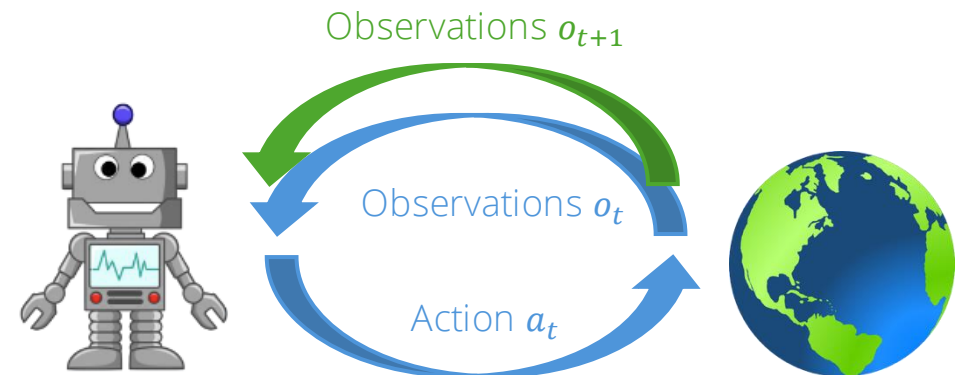
\mathbf{s}_t – state

\mathbf{o}_t – observation

\mathbf{a}_t – action

$\pi_{\theta}(\mathbf{a}_t | \mathbf{o}_t)$ – policy

$\pi_{\theta}(\mathbf{a}_t | \mathbf{s}_t)$ – policy (fully observed)

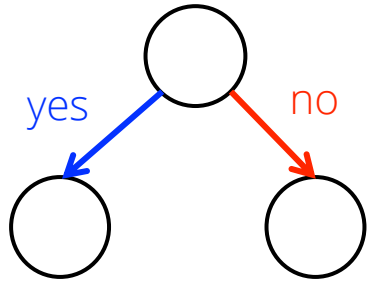


What does it mean acting?

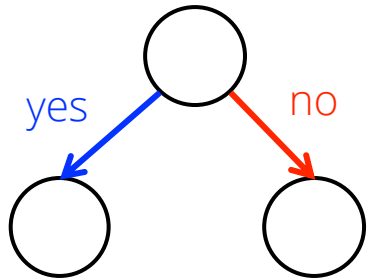
Decision Making: Decide What Actions to Take

I.I.D decision making

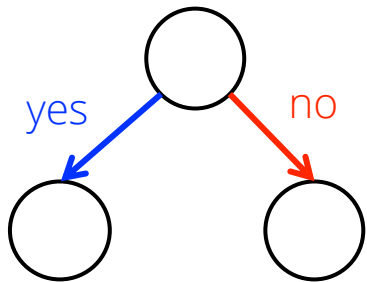
Have a
breakfast



Have a
lunch



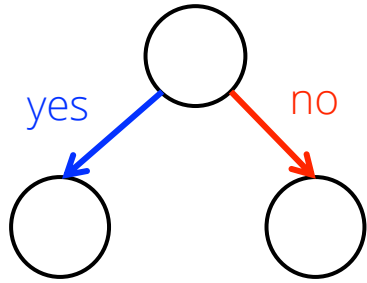
Have a
dinner



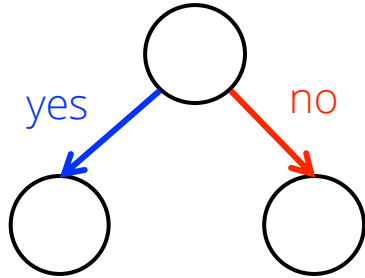
Decision Making Problems Are Often Sequential

I.I.D decision making

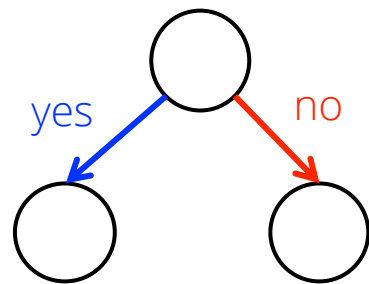
Have a breakfast



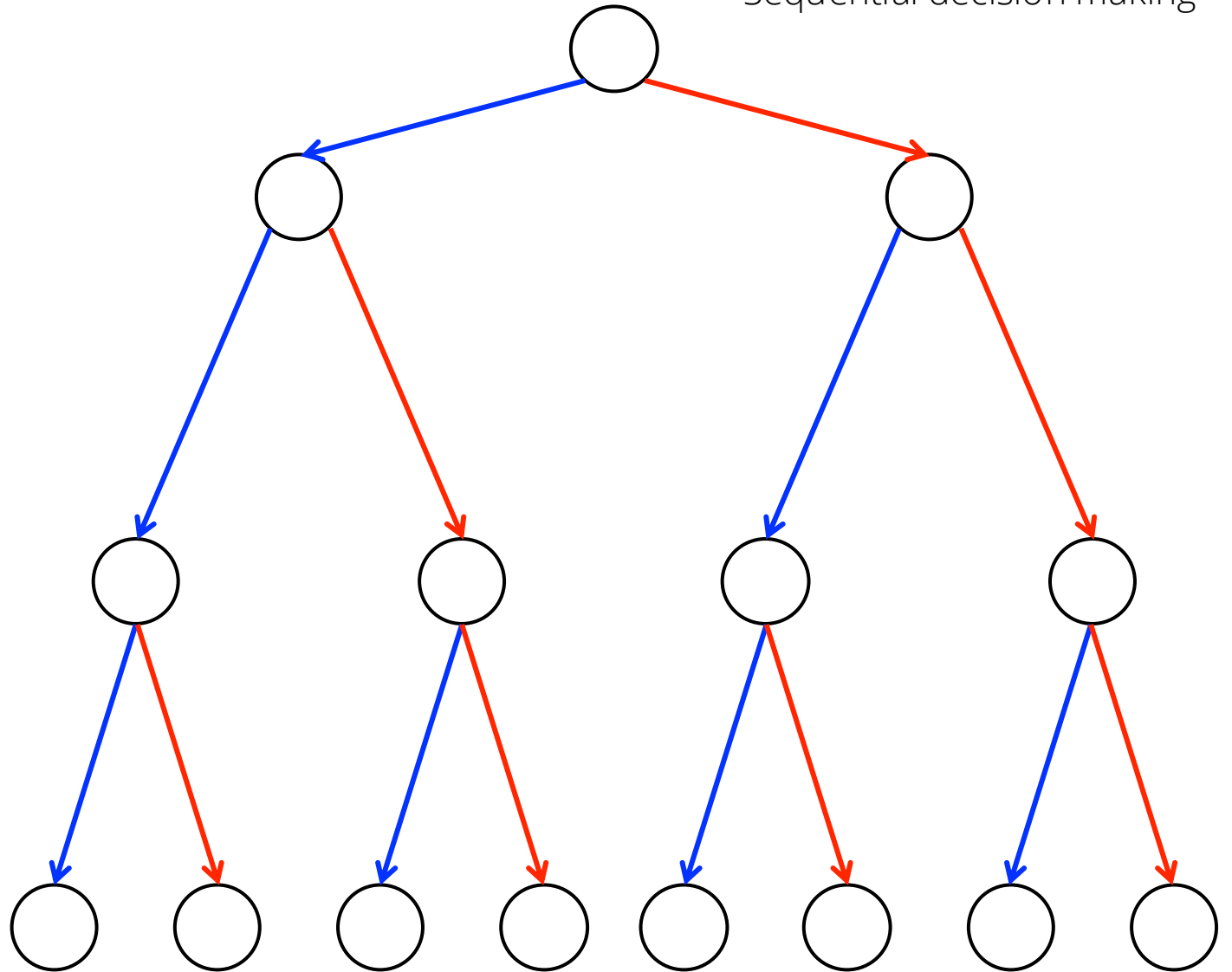
Have a lunch



Have a dinner



Sequential decision making



Decision Making Problems Are Often Sequential



https://youtu.be/a_YGPbWJO5g?si=3ZEr3rc5gjQ9JluX



<https://youtu.be/6Zbhvaac68Y?si=Ivi7bak6xi-ByGzA>



<https://youtu.be/l-Y5FHI4JXc?si=mzqvEBCyu-YnYKun>



<https://www.youtube.com/watch?v=BhMSzC1crr0>



<https://youtu.be/cM6woUhjsCo?si=dKOi0WOUOLfbgkAk>



<https://youtube.com/shorts/GYeSEKnHjDI?si=8xSn3p9EBoW4nwBy>

Each Decision You Make Influence Future Outcomes



League of Legends

What is my first item?



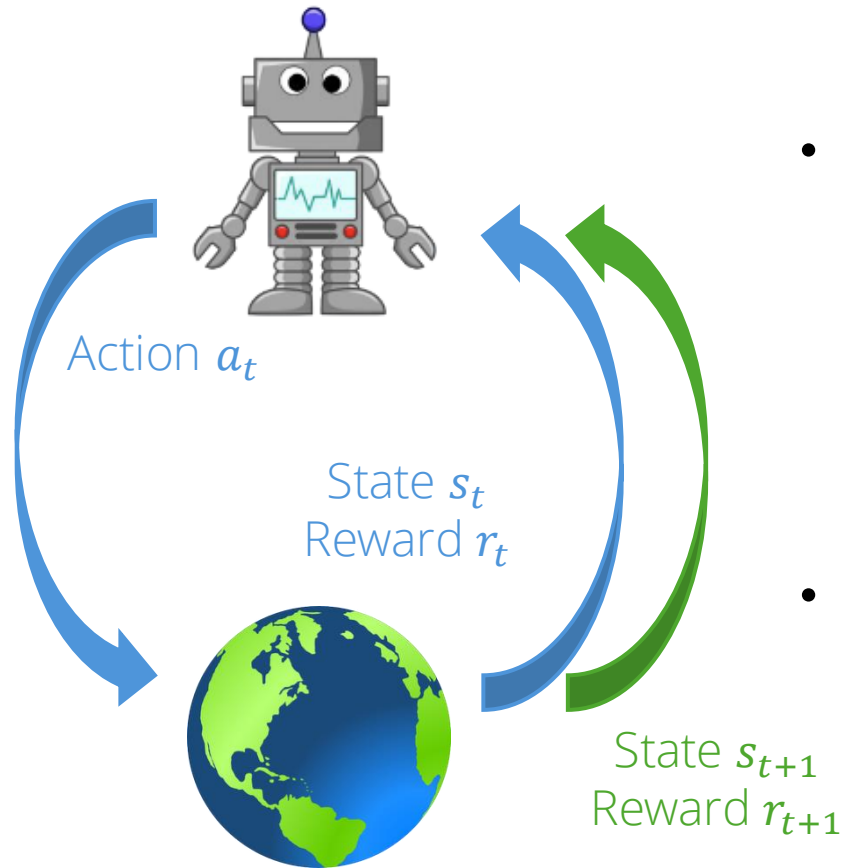
Age of Empire 2

Where should I drop the town center?
How many villagers on wood?
When should my villagers explore the map?

⋮

How to decide which action to take?

Reinforcement Learning: Learns to Maximize the Total Reward of an Episode of Interaction



- A trajectory of interaction in the environment

$$\underbrace{p_{\theta}(\mathbf{s}_1, \mathbf{a}_1, \dots, \mathbf{s}_T, \mathbf{a}_T)}_{p_{\theta}(\tau)} = p(\mathbf{s}_1) \prod_{t=1}^T \pi_{\theta}(\mathbf{a}_t | \mathbf{s}_t) p(\mathbf{s}_{t+1} | \mathbf{s}_t, \mathbf{a}_t)$$

- Maximize the expected value of the cumulative sum of reward

$$\theta^* = \arg \max_{\theta} E_{\tau \sim p_{\theta}(\tau)} \left[\sum_t r(\mathbf{s}_t, \mathbf{a}_t) \right]$$

Reinforcement Learning Enables High-Fidelity Robot Actions

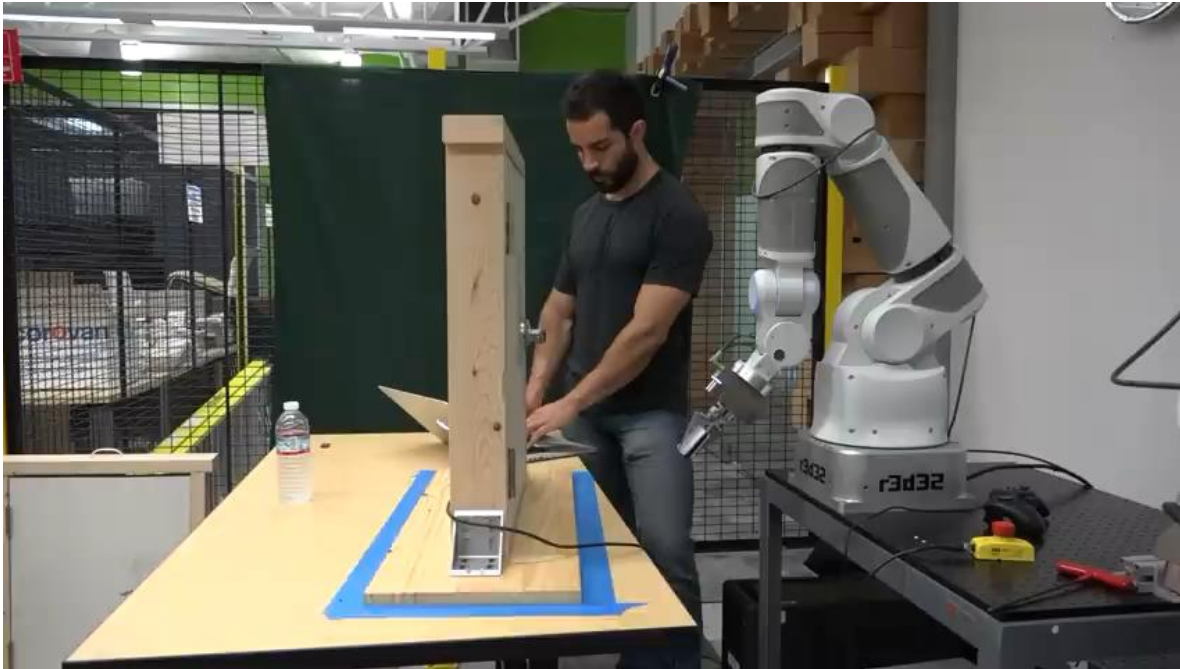


We won't cover in this course! Check Robot Perception and Learning / (Deep) Reinforcement Learning

Learning to Act by Cloning Behaviors



Collect Behaviors by Teleoperation



https://youtu.be/9-9udQtO1PY?si=AZCr7n_9V41BA2o9



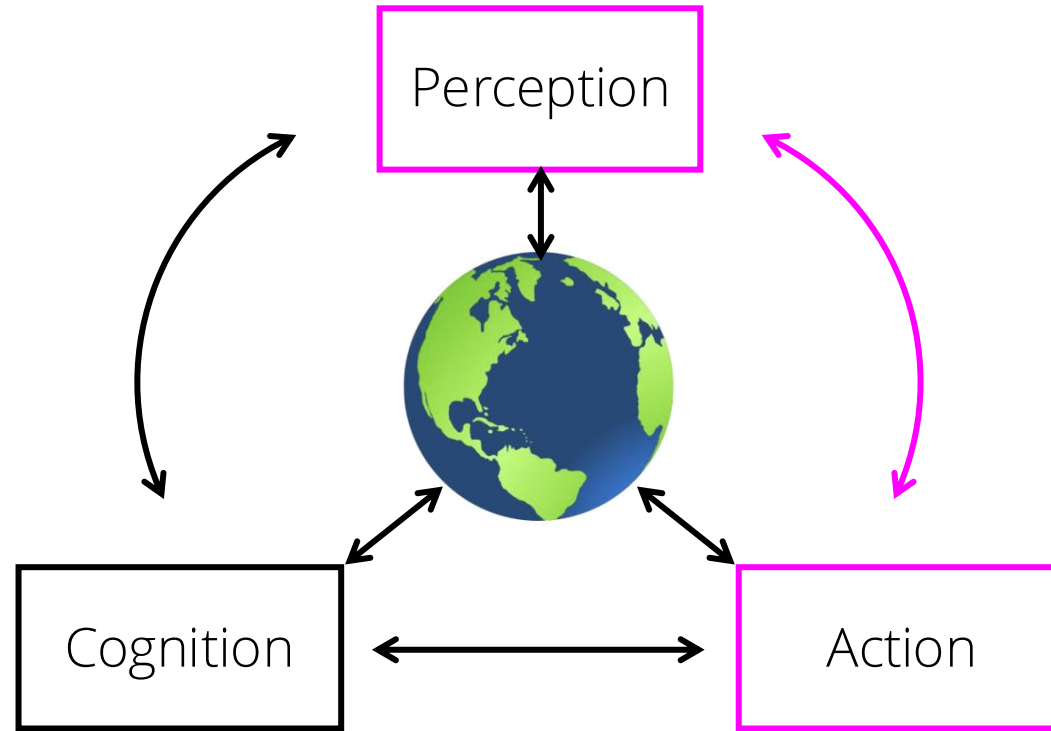
<https://youtu.be/Bhg3uOx9ZPw?si=et7L0endzGvGPJz->

Teleoperation is too expensive!

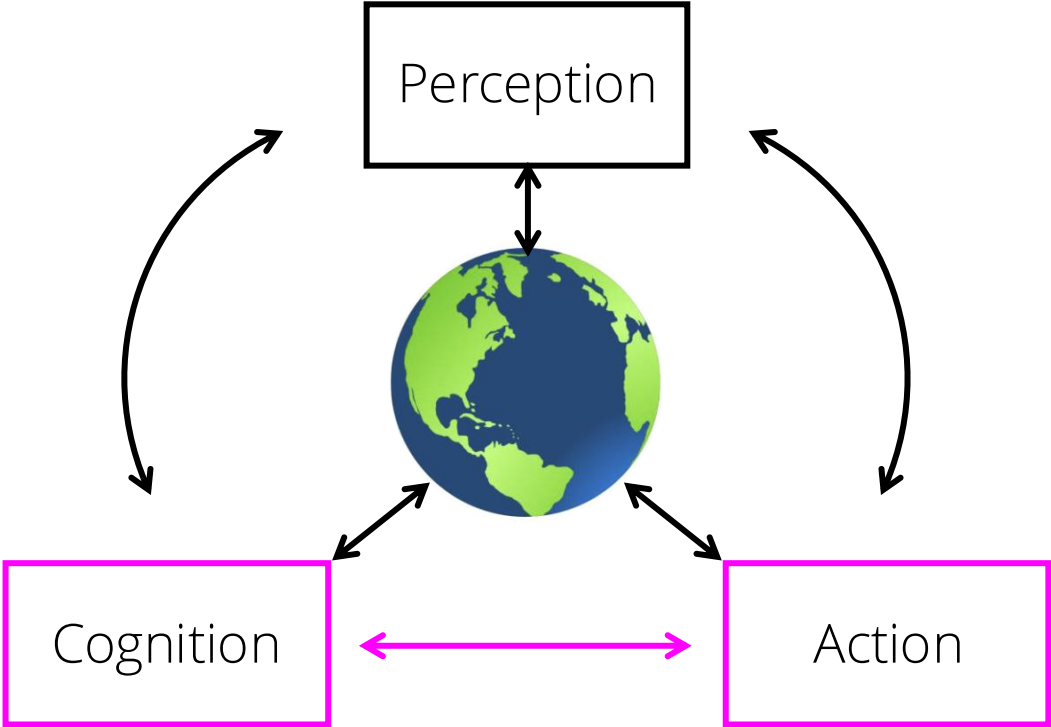
Learning to Act by Visual Imitation



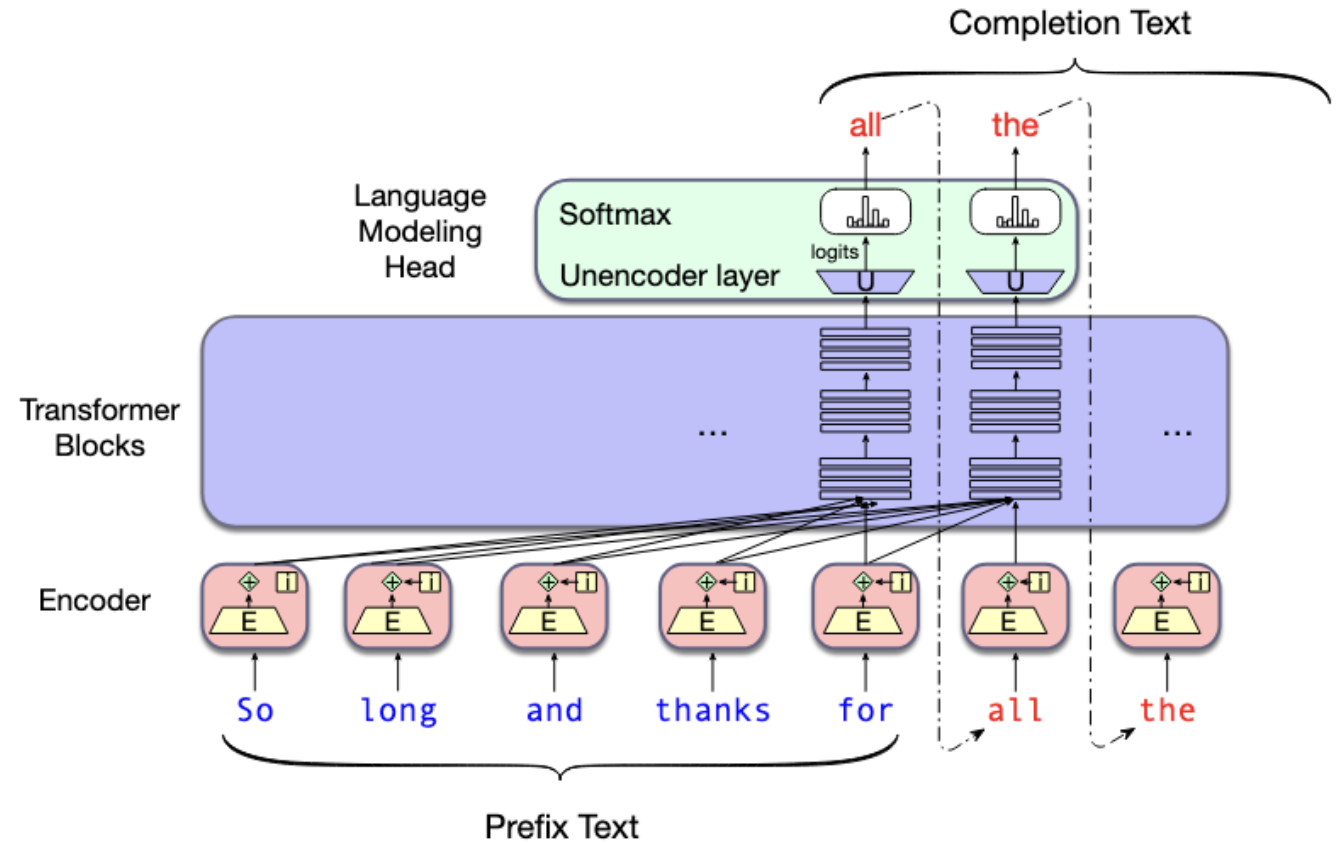
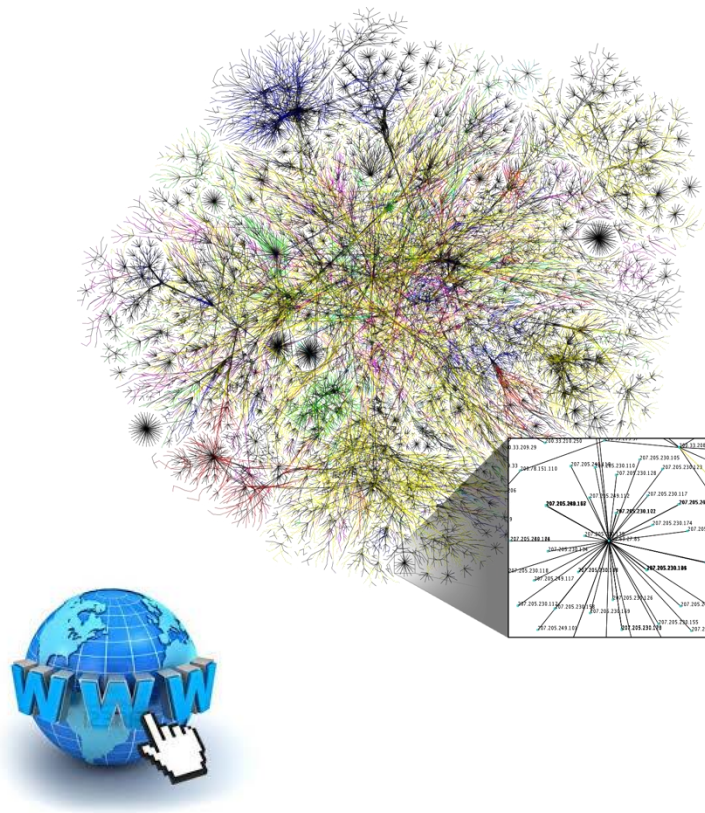
How Can Perception Facilitate Action?



How Can Cognition Facilitate Action?



Large Language Models Capture the Distribution of Internet Data



Large Language Models Show Strong Cognition



Gemini



Llama

Should basketball move its corner three line further out? Run a thorough analysis of existing sentiment with projections and visualizations of the impact of such a move and make a recommendation.

Breaking down NBA corner three analysis

- The user wants a thorough analysis of the potential impact of moving the NBA corner three-point line further out. I'll look into the latest NBA discussions and fan sentiment, review opinions from players and analysts, and gather recent data on shot value and frequency. Projections might include a heatmap visualization of shot distances and comparing the expected value difference if the line shifted. However, I don't have all the precise datasets at the moment—let's gather that info!
- I think we could start by gathering data on corner three vs. above-the-break shots. For instance, corner three accuracy is often around 39%, and non-corner about 35%. We'll need up-to-date stats like shot frequency for the 2024 season, and maybe a web search will help find that.

Next, I'd look into what discussions there have been about moving the corner three line, using articles or social media. From there, I'll create projections and prepare visualizations with bar charts, ultimately offering a recommendation.

Message ChatGPT



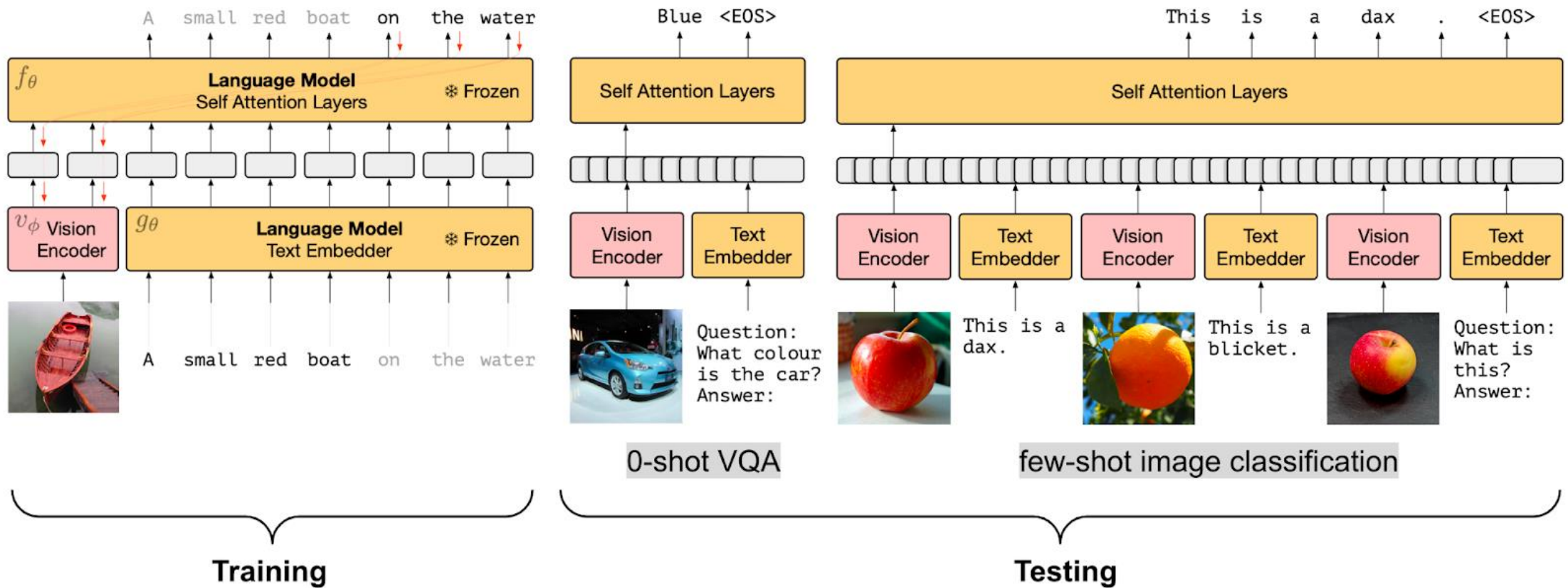
Search



Deep research



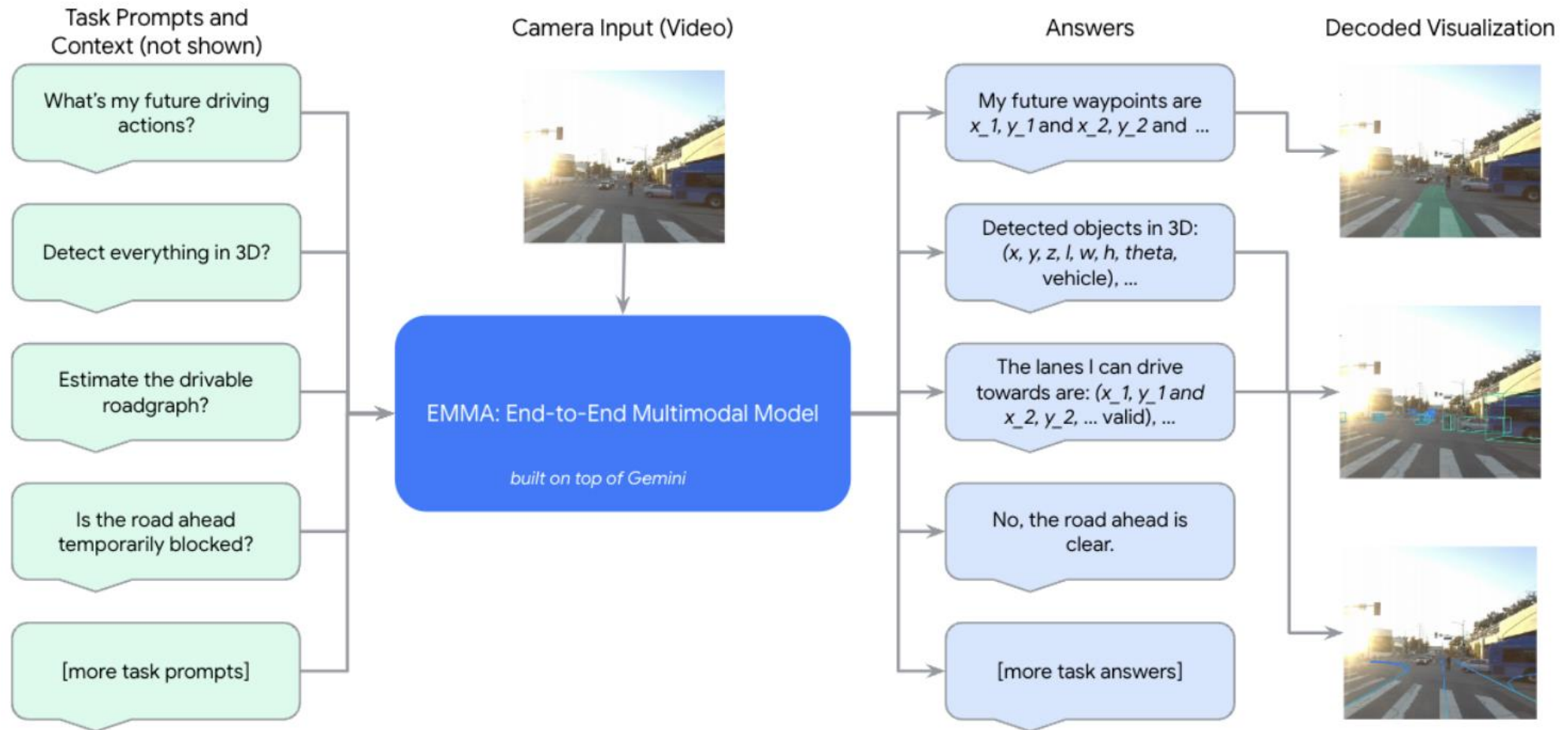
Vision-Language Models Capture the Distribution of Internet Text-Image Data



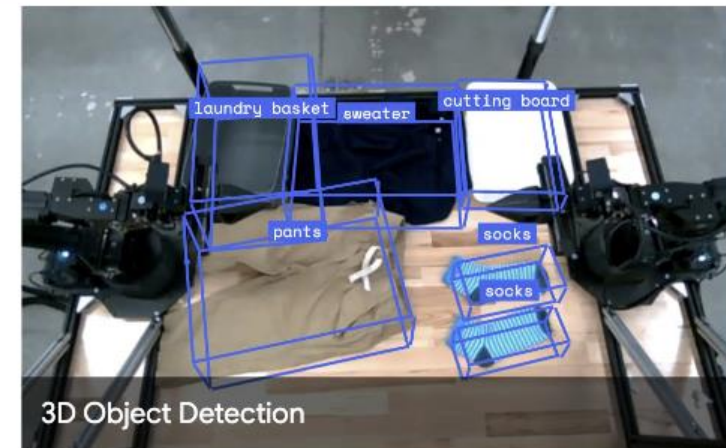
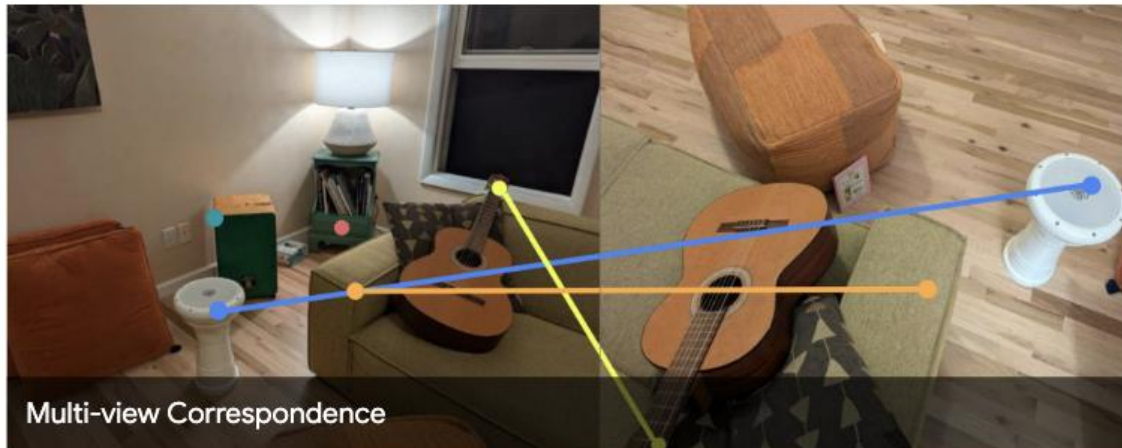
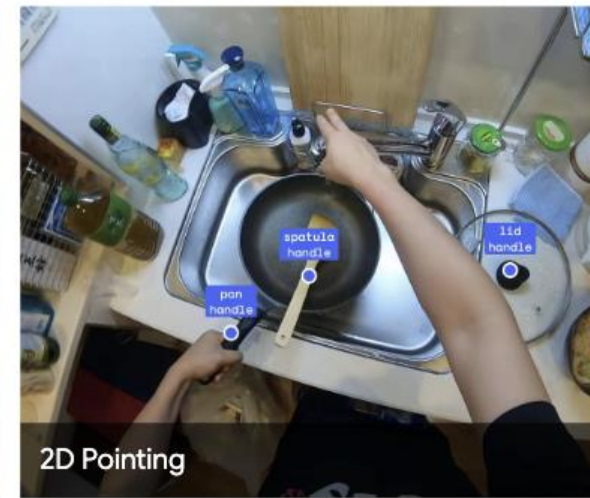
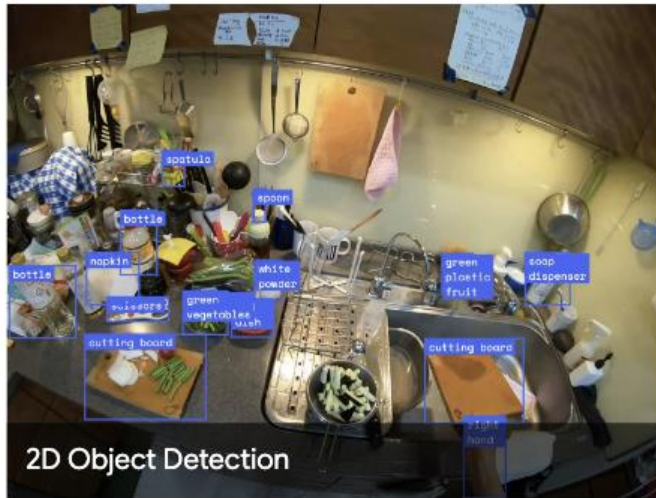
Large Language Models Show Strong Cognition and Perception






Large Language Models Show Strong Cognition and Perception





Large Language Models Show Strong Cognition and Perception



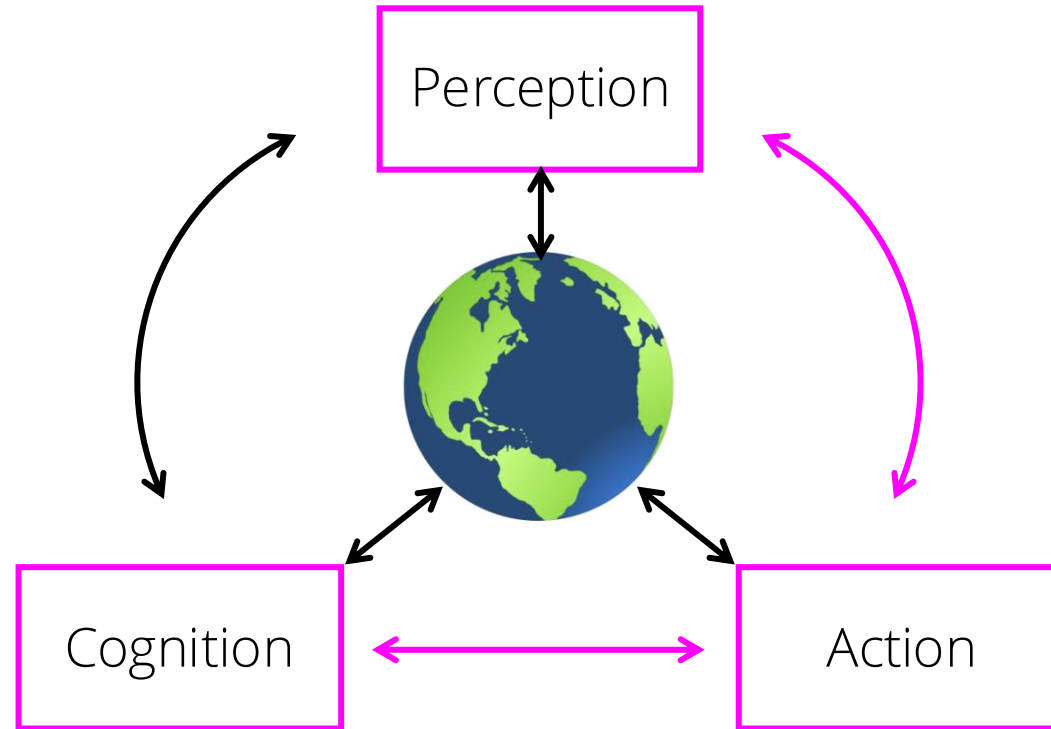
Large Language Models Show Strong Cognition and Perception

Trajectory Reasoning	Action Reasoning	Spatial Reasoning
 <p>Approximately which colored trajectory should the zipper follow to begin zipping up the suitcase?</p> <ul style="list-style-type: none">A. BlueB. PurpleC. GreenD. Red	 <p>How should the person move the wrench so that it is ready to rotate the hex screw closest to it?</p> <ul style="list-style-type: none">A. Forward and rightB. Up and leftC. Forward and leftD. None of the above	 <p>There are 4 sinks in the picture. Which arrow points to the one that is closest to the viewer?</p> <ul style="list-style-type: none">A. CyanB. BlueC. RedD. None of the arrows

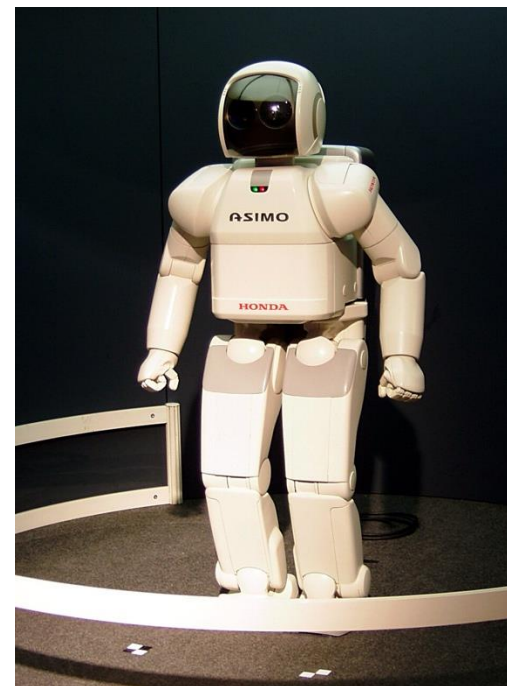
Large Language Models Show Strong Cognition and Perception

Question	Reasoning Trace	Question	Reasoning Trace
 <p>If the yellow robot gripper follows the yellow trajectory, what will happen?</p> <p>A. Robot puts the soda on the wooden steps. B. Robot moves the soda in front of the wooden steps. C. Robot moves the soda to the very top of the wooden steps. D. Robot picks up the soda can and moves it up.</p>	<p>Let's analyze the trajectory and the position of the soda can and the wooden steps.</p> <p>Step 1: Observe the starting point. The yellow trajectory starts at the green circle, which is around the soda can. This indicates the robot gripper is initially positioned to interact with the can.</p> <p>Step 2: Analyze the trajectory path. The yellow line moves upwards from the can, indicating a lifting motion. Then, it curves towards the wooden steps.</p> <p>Step 3: Observe the ending point. The trajectory ends at the purple circle, which is located on the first step of the wooden structure.</p> <p>Step 4: Combine the observations. The robot gripper starts at the can, moves upwards (picking up the can), and then moves towards the wooden steps, ending on the first step.</p> <p>Step 5: Deduce the outcome. Based on the trajectory, the robot will pick up the soda can and place it on the first step of the wooden structure.</p> <p>Final Answer: (A)</p>	 <p>How does the bottom dial need to move in order to match the top two dials?</p> <p>A. Half turn clockwise. B. Quarter turn counterclockwise. C. Quarter turn clockwise. D. No change needed</p>	<p>Let's analyze the position of the bottom dial compared to the top two dials.</p> <p>Step 1: Observe the top two dials. Both the top and middle dials have their indicator lines pointing straight up.</p> <p>Step 2: Observe the bottom dial. The bottom dial has its indicator line pointing to the left.</p> <p>Step 3: Determine the necessary movement. To make the bottom dial's indicator line point straight up like the other two, it needs to be rotated to the right.</p> <p>Step 4: Quantify the movement. A full circle is 360 degrees. A quarter turn is $360/4 = 90$ degrees. The bottom dial needs to move 90 degrees to the right.</p> <p>Step 5: Select the correct option. A 90-degree rotation to the right is a quarter turn clockwise.</p> <p>Final Answer: (C)</p>

How Can Cognition and Perception Facilitate Action?



How Can We Turn VLM into Robot Policies?





Asimo by Honda

Idea 0: Prompt VLM to Predict Actions

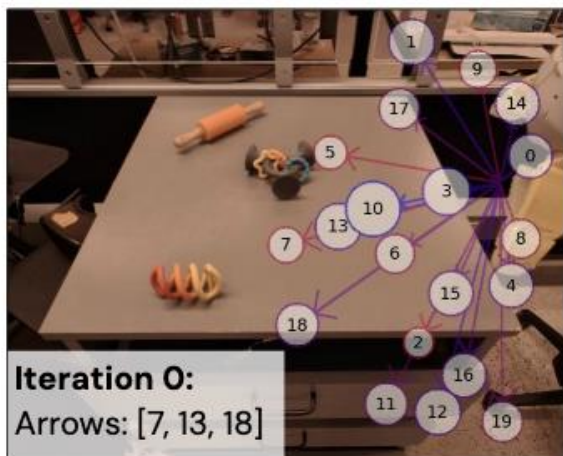
- The idea doesn't work well:
 - Robotic manipulation require high-precision actions, but discrete text is not an appropriate representation for continuous actions
 - VLMs are not trained to predict actions. The output is not grounded...

Idea 1: Prompt VLM to Select Marks

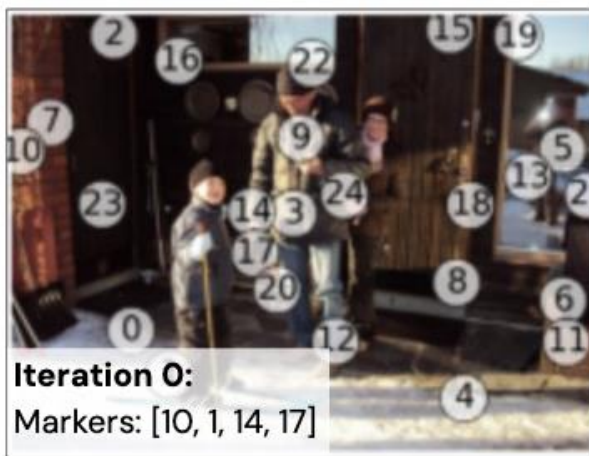
Input: Image	Input: Image + SoM
	
<p>Conversation</p> <p>User What is on the left side of the right laptop?</p> <p>GPT-4V On the left side of the right laptop, there is a cup or mug. ❌</p> <p>User I want to find a seat close to windows, where can I sit?</p> <p>GPT-4V You can sit on either of the two black chairs in front of the white desks, as they are closest to the windowed wall. ❌</p>	<p>Conversation + 🗺️</p> <p>User What is on the left side of the right laptop?</p> <p>GPT-4V On the left side of the right laptop (numbered as 9), there is a lamp, which is numbered as 12. ✅</p> <p>User I want to find a seat close to windows, where can I sit?</p> <p>GPT-4V You can sit on the chair numbered as 7. It's the closest seat to the windows (represented by the area numbered as 1). ✅</p>

Idea 2: Translate

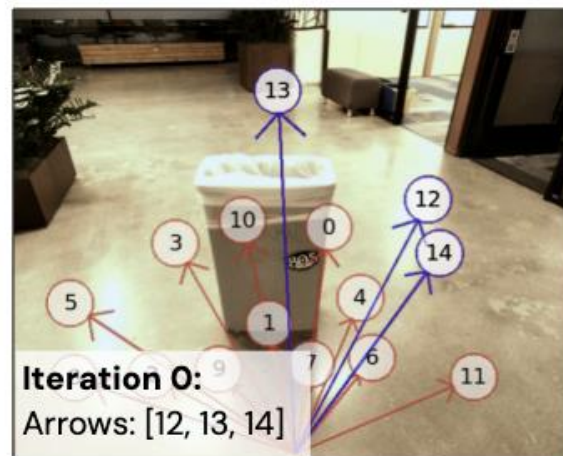
Task: What actions should the robot take to pick up the DNA chew toy?



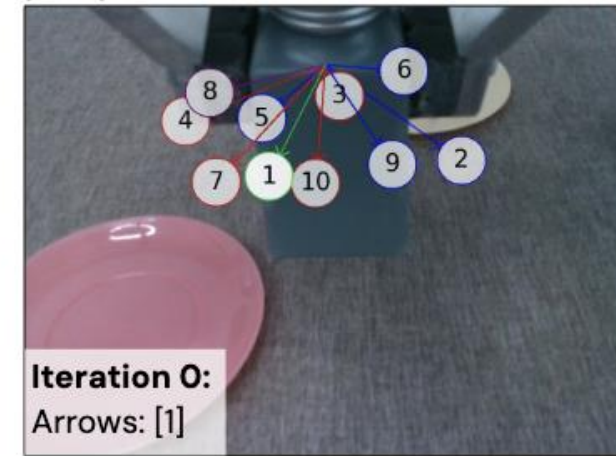
Task: What numbers overlay the "L kid"?



Task: What actions should the robot take to go to wooden bench without hitting the obstacle?

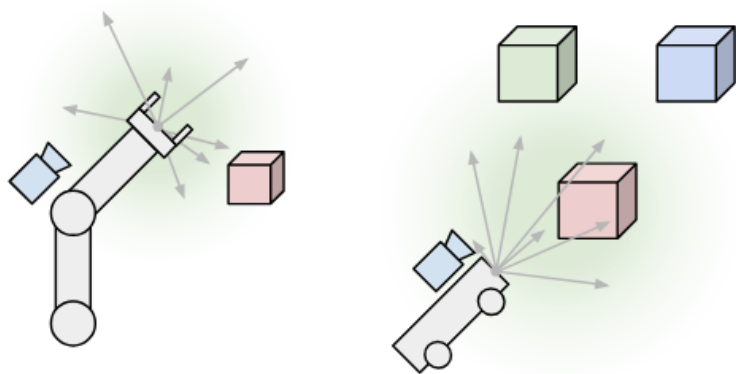


Task: What actions should the robot take to put the pepper shaker on the pink plate?



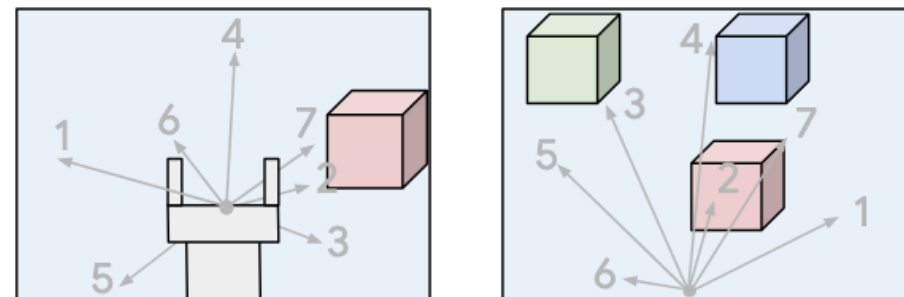
(a) Sample Actions

Sample candidate actions from action space $\mathbf{A}^{(i)}$



(b) Annotate Image

Project candidate actions into **image** and label

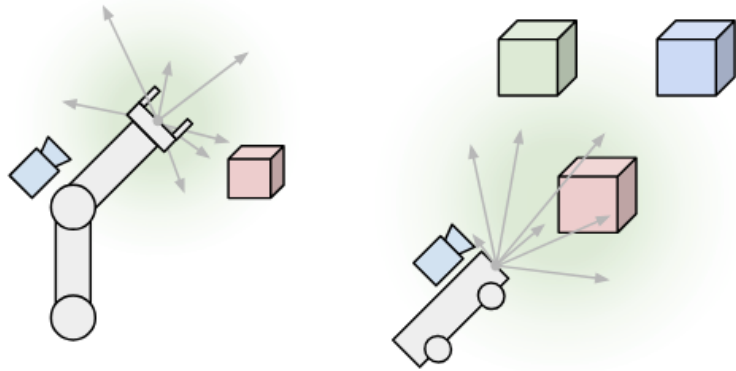


Sampled actions are too few, we need high-precision actions for robotic manipulation...

Evolutionary Search for Action Prediction

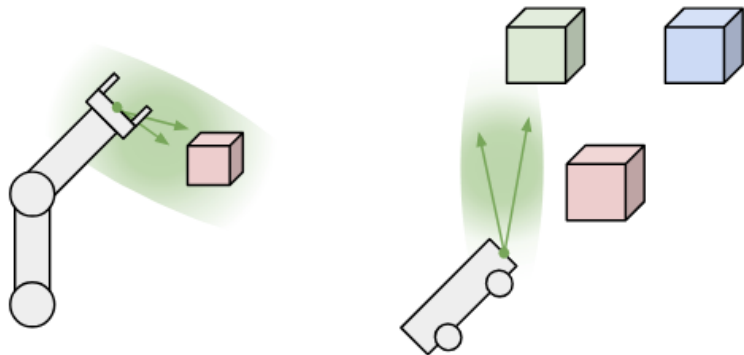
(a) Sample Actions

Sample candidate actions from action space $A^{(i)}$



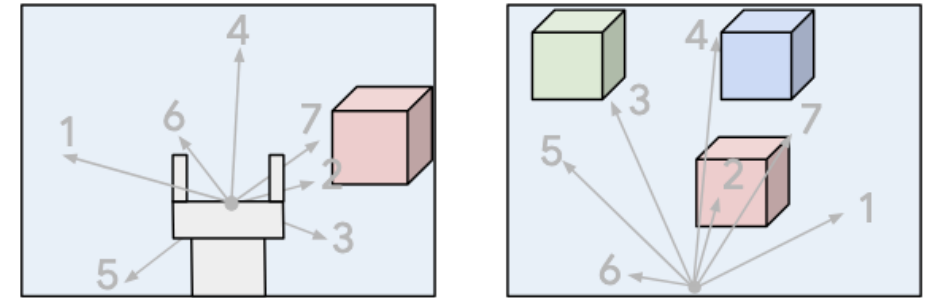
(d) Fit Distribution

Fit a selected action distribution $A^{(i+1)}$



(b) Annotate Image

Project candidate actions into **image** and label

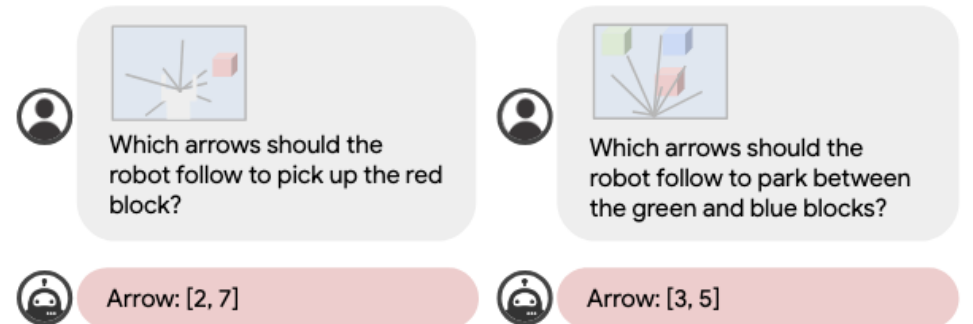


PIVOT

Prompting with Visual Iterative Optimization

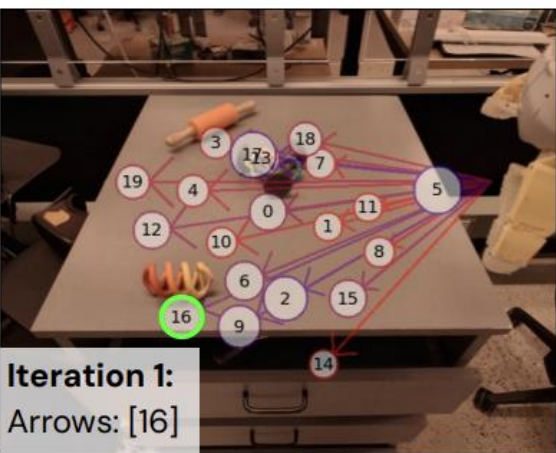
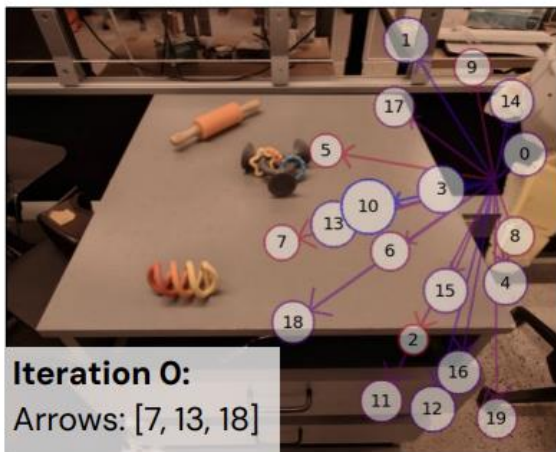
(c) Query VLM

Query **VLM** via VQA for best actions

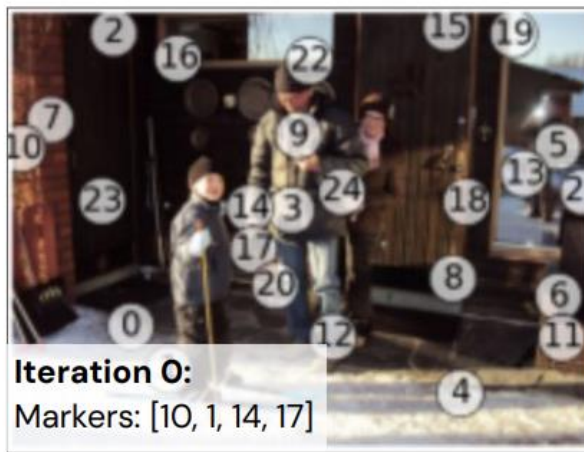


Evolutionary Search for Action Prediction

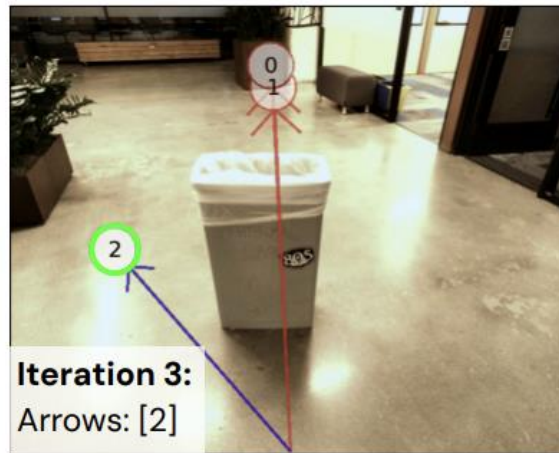
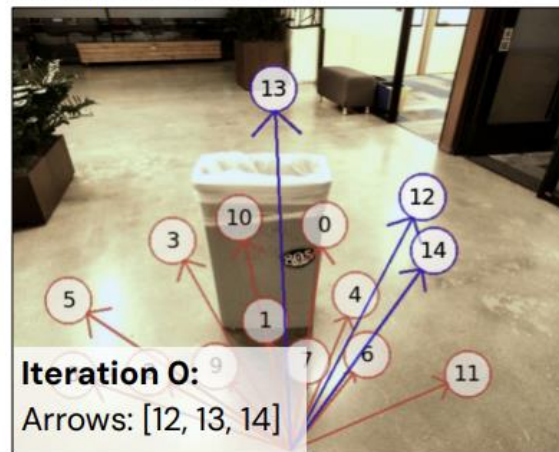
Task: What actions should the robot take to pick up the DNA chew toy?



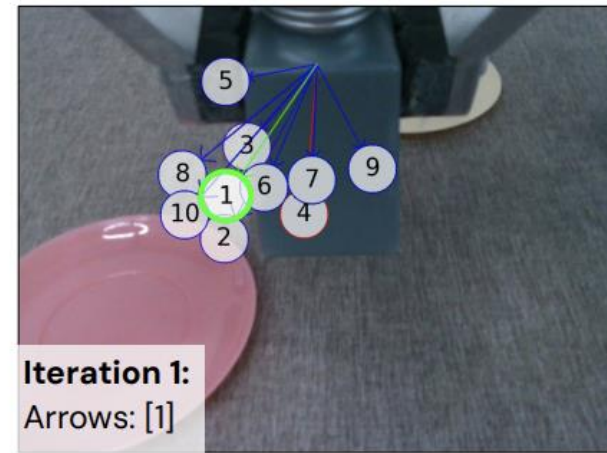
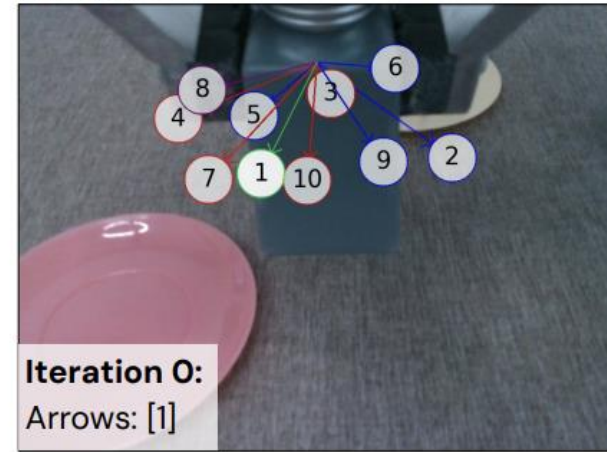
Task: What numbers overlay the "L kid"?



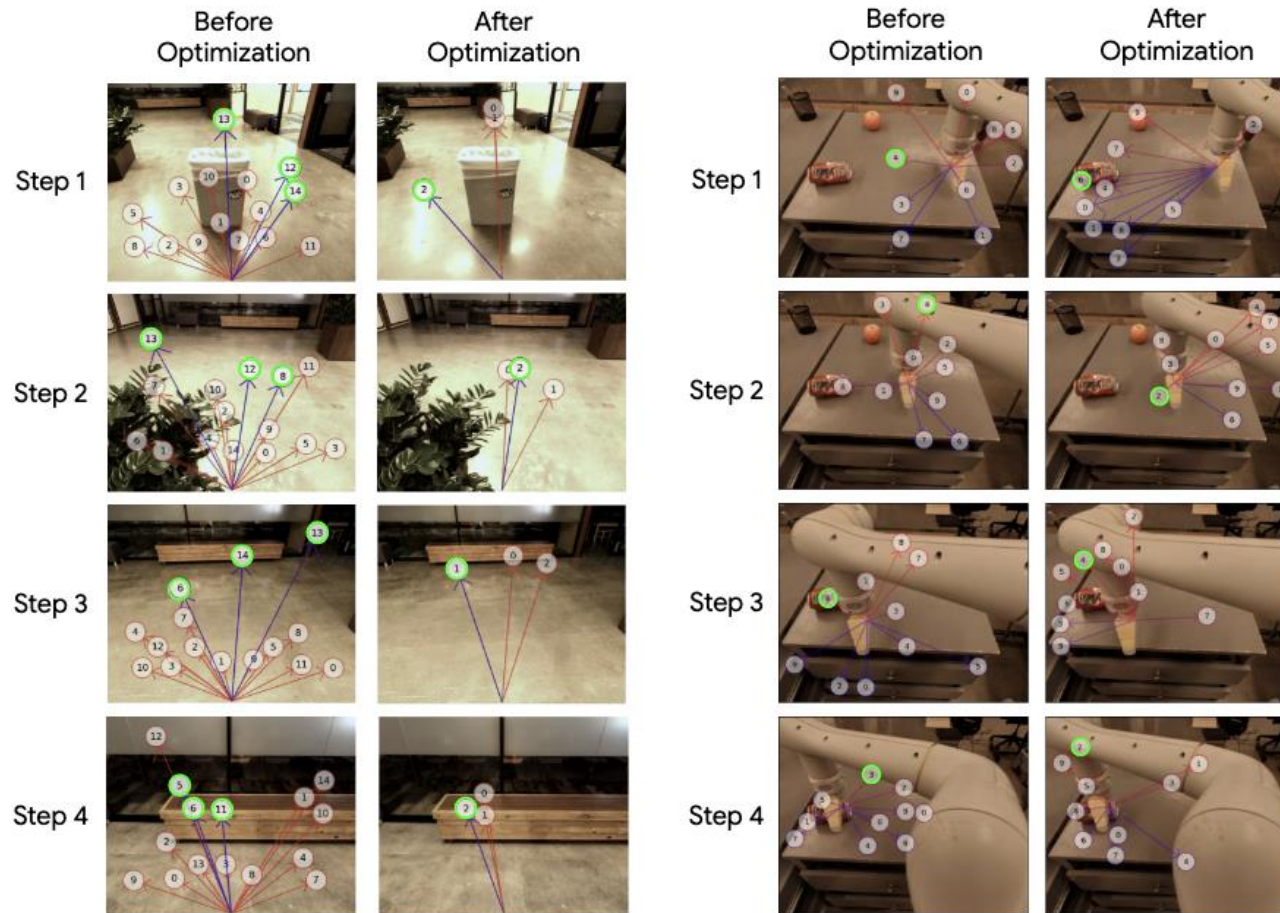
Task: What actions should the robot take to go to wooden bench without hitting the obstacle?



Task: What actions should the robot take to put the pepper shaker on the pink plate?



Results: Zero-shot Robotic Control



(a) Navigation: “Help me find a place to sit and write”
(b) Manipulation: “Pick up the coke can”

Results: Zero-shot Spatial Understanding

“Left Skier”

“Boy in Blue”

Step 1



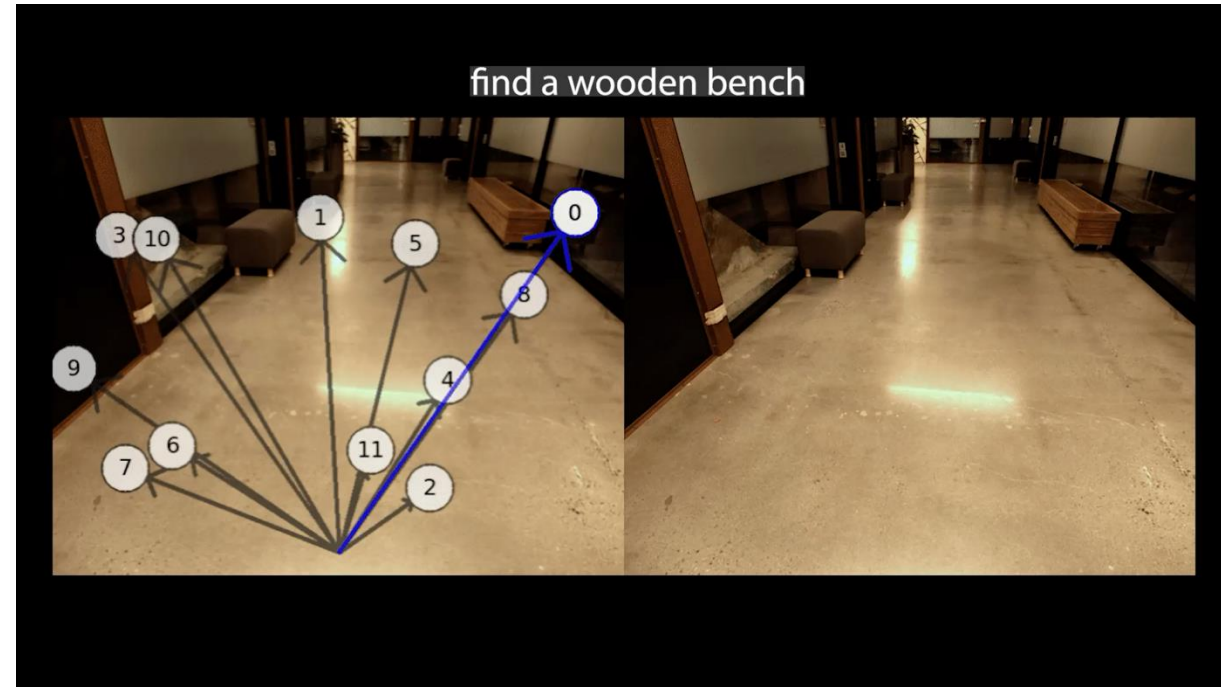
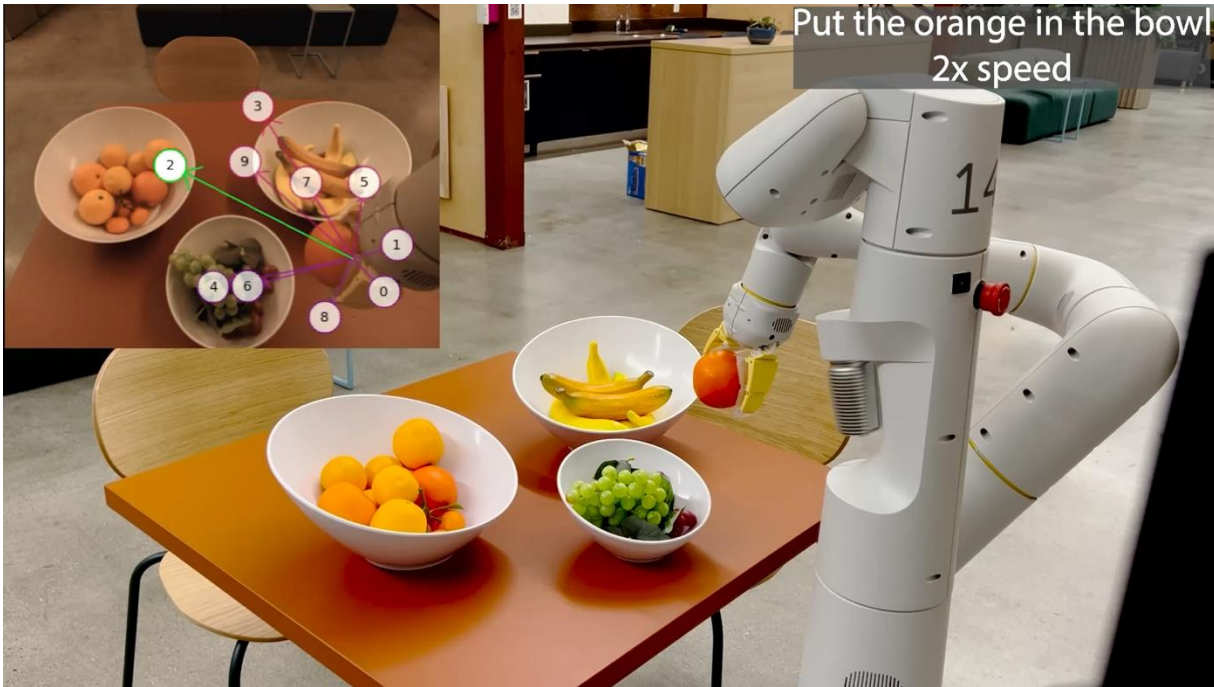
Step 2



Step 3



Results



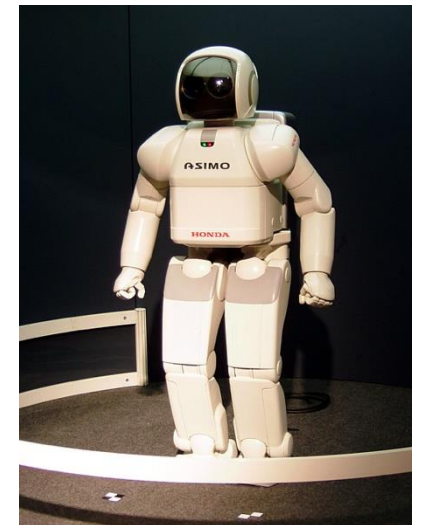
How to handle high-dimensional actions e.g rotation, gripper openness?

Idea 2: Prompt VLM to Predict Affordance and Translate Affordance into Actions

Affordance: Where and How to Interact with objects

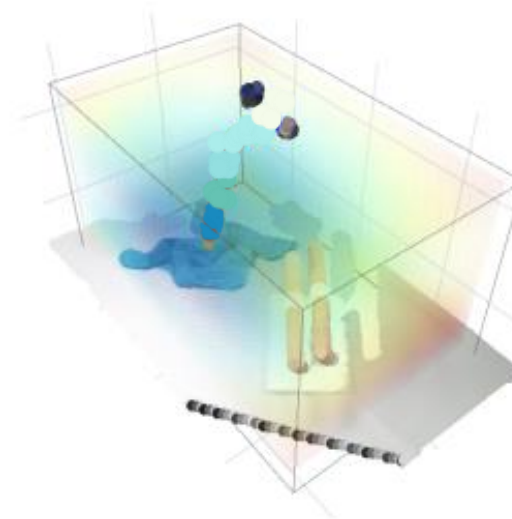
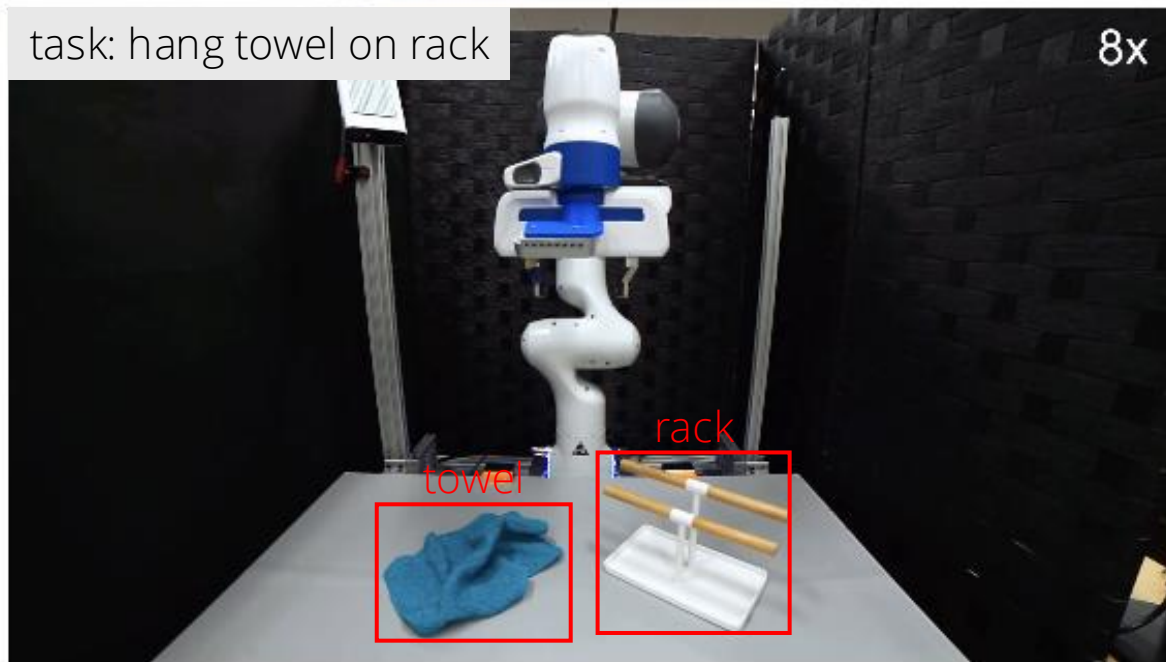


Gemini

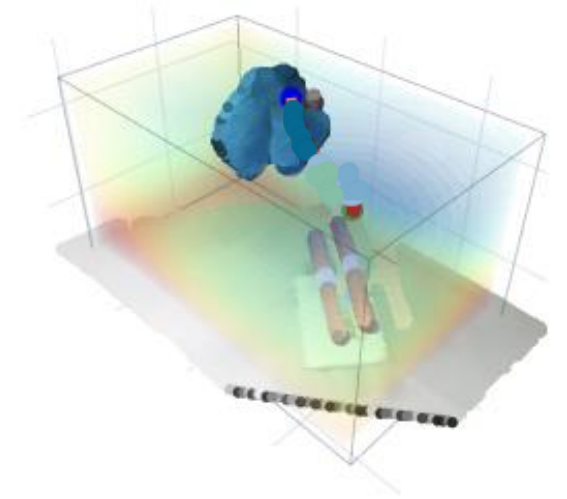


How to Infer Where to Interact with Objects?

- Object detection decides where Interact with objects



Interactive Value Map 1

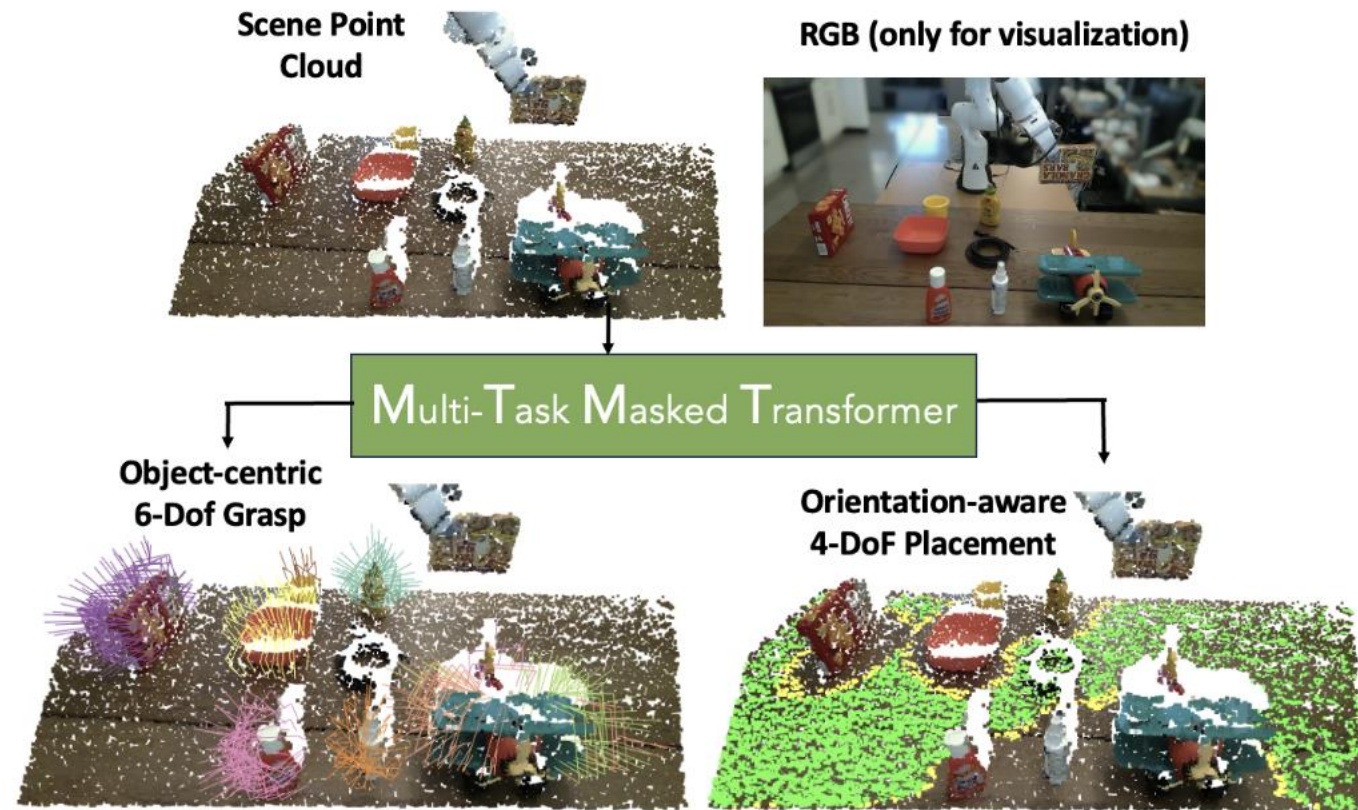


Interactive Value Map 2

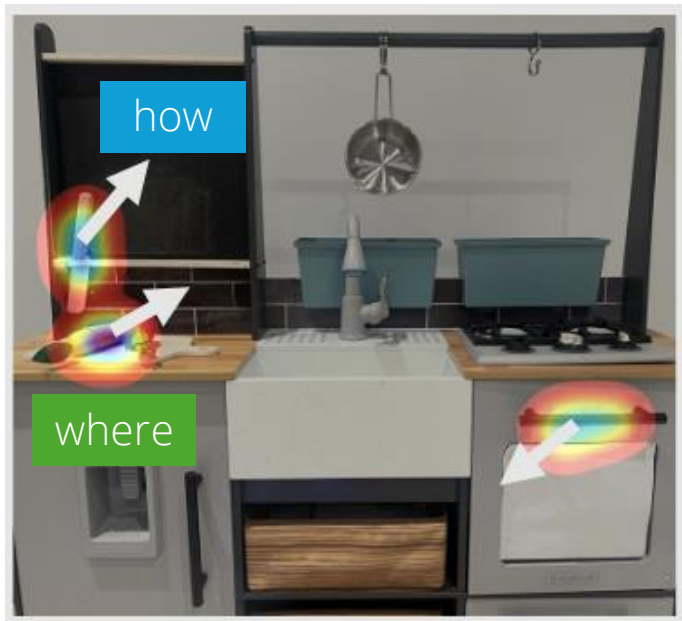
But the affordance encodes coarse 3d position , we need precise position, rotation and openness...

How to Infer Where to Interact with Objects?

- But the affordance encodes coarse 3d position , we need precise position, rotation and openness...



How to Infer How to Interact with Objects?

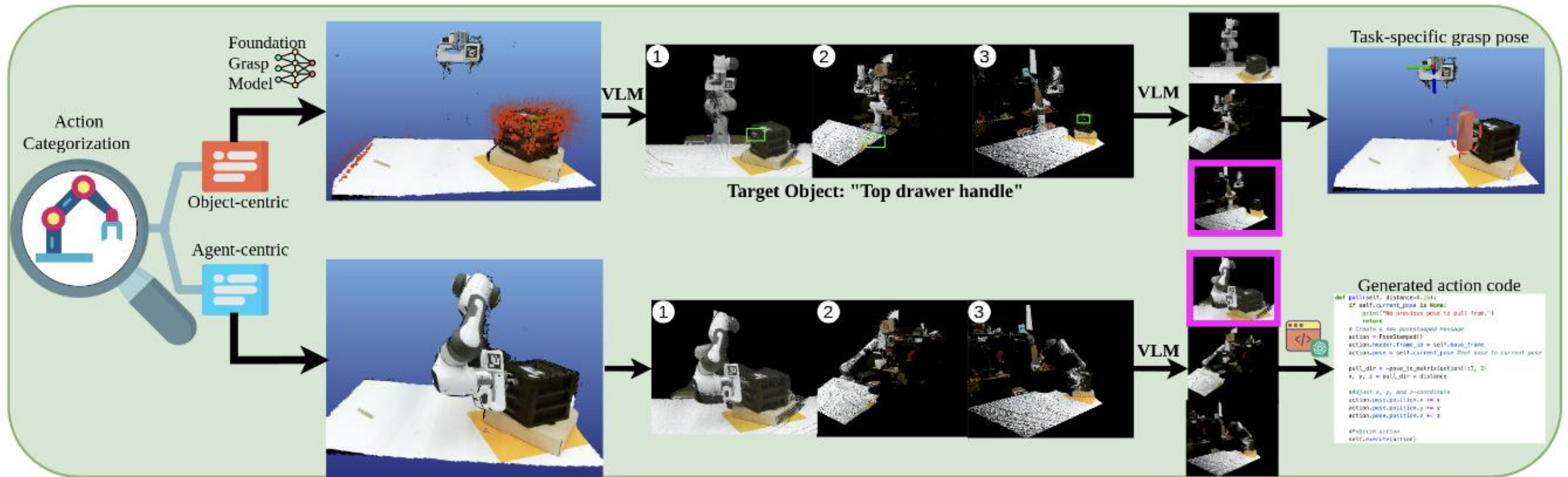


Writing codes to represent motion trajectories (how to interact with objects)

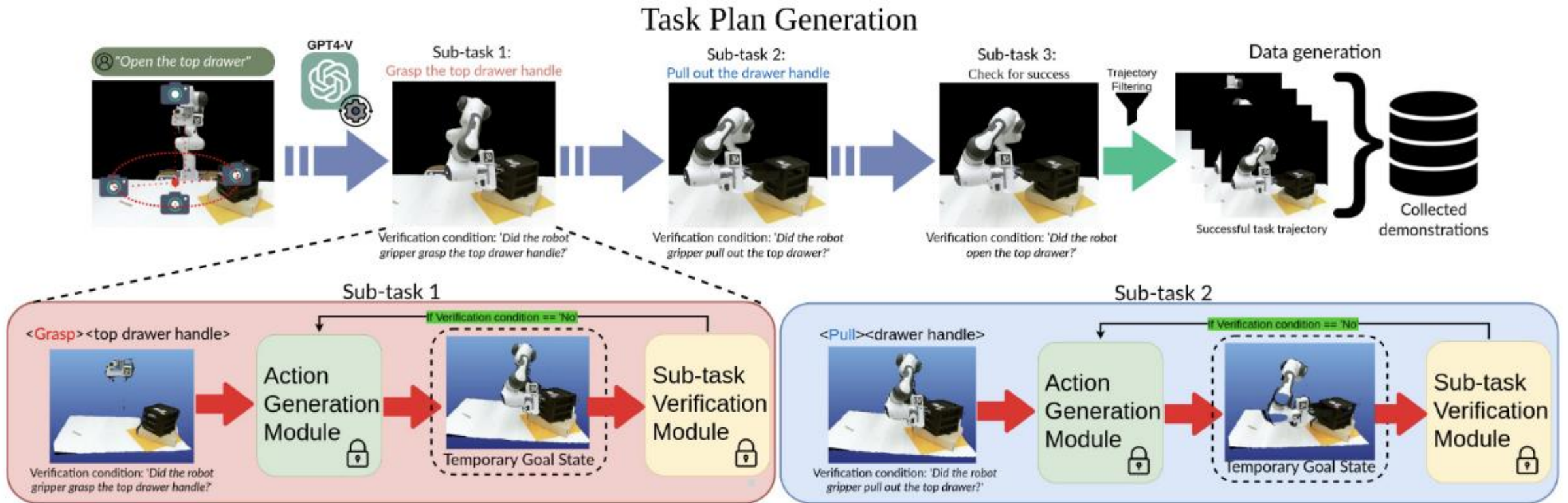
```
def pull(self, distance=0.15):  
    if self.current_pose is None:  
        print("No previous pose to pull from.")  
        return  
    # Create a new posestamped message  
    action = PoseStamped()  
    action.header.frame_id = self.base_frame  
    action.pose = self.current_pose #Set pose to current pose  
  
    pull_dir = -pose_to_matrix(action)[:3, 2]  
    x, y, z = pull_dir * distance  
  
    #Adjust x, y, and z-coordinate  
    action.pose.position.x += x  
    action.pose.position.y += y  
    action.pose.position.z += z  
  
    #Publish action  
    self.execute(action)
```



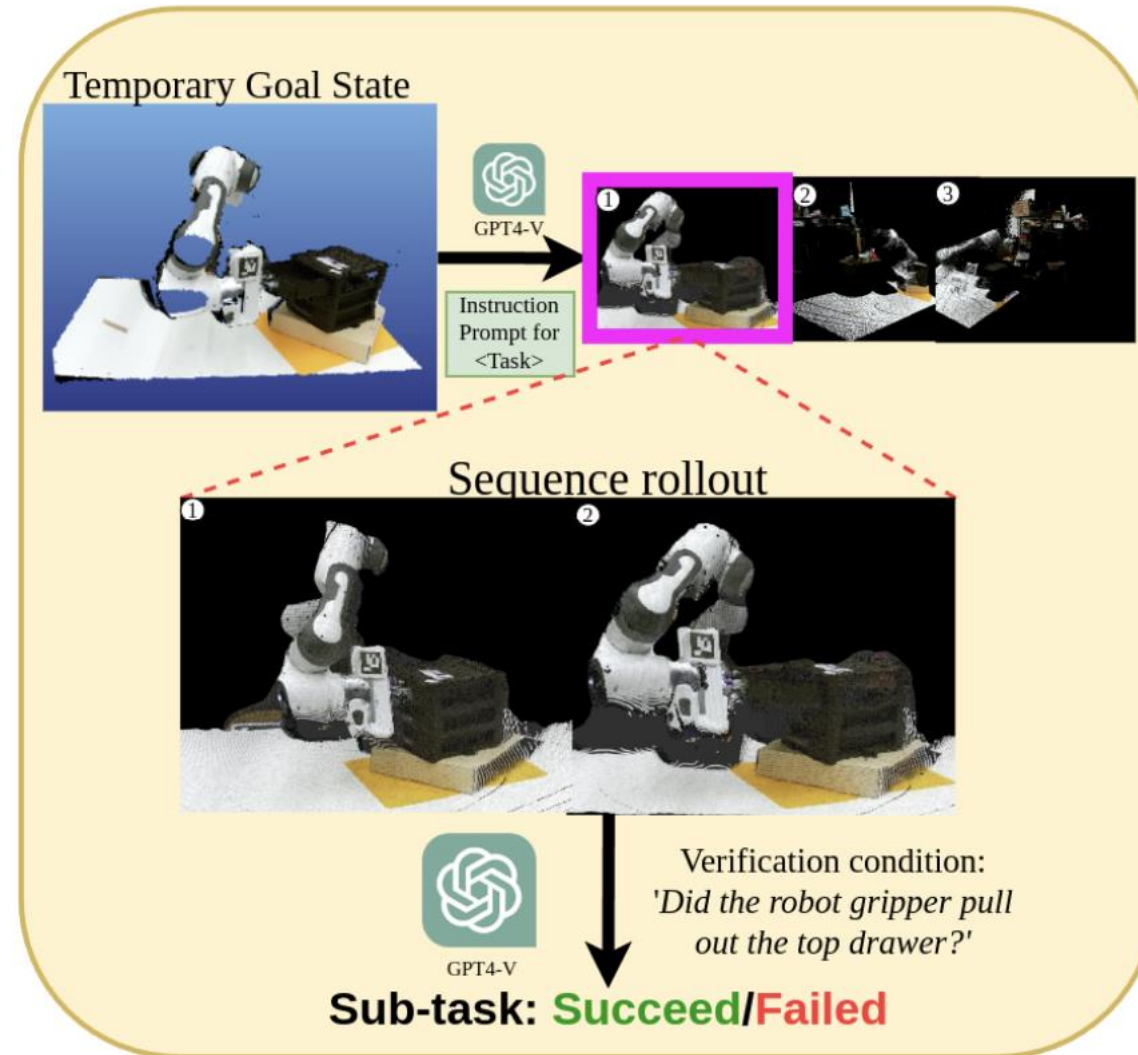
Prompt VLM to Predict Affordance and Translate Affordance into Actions



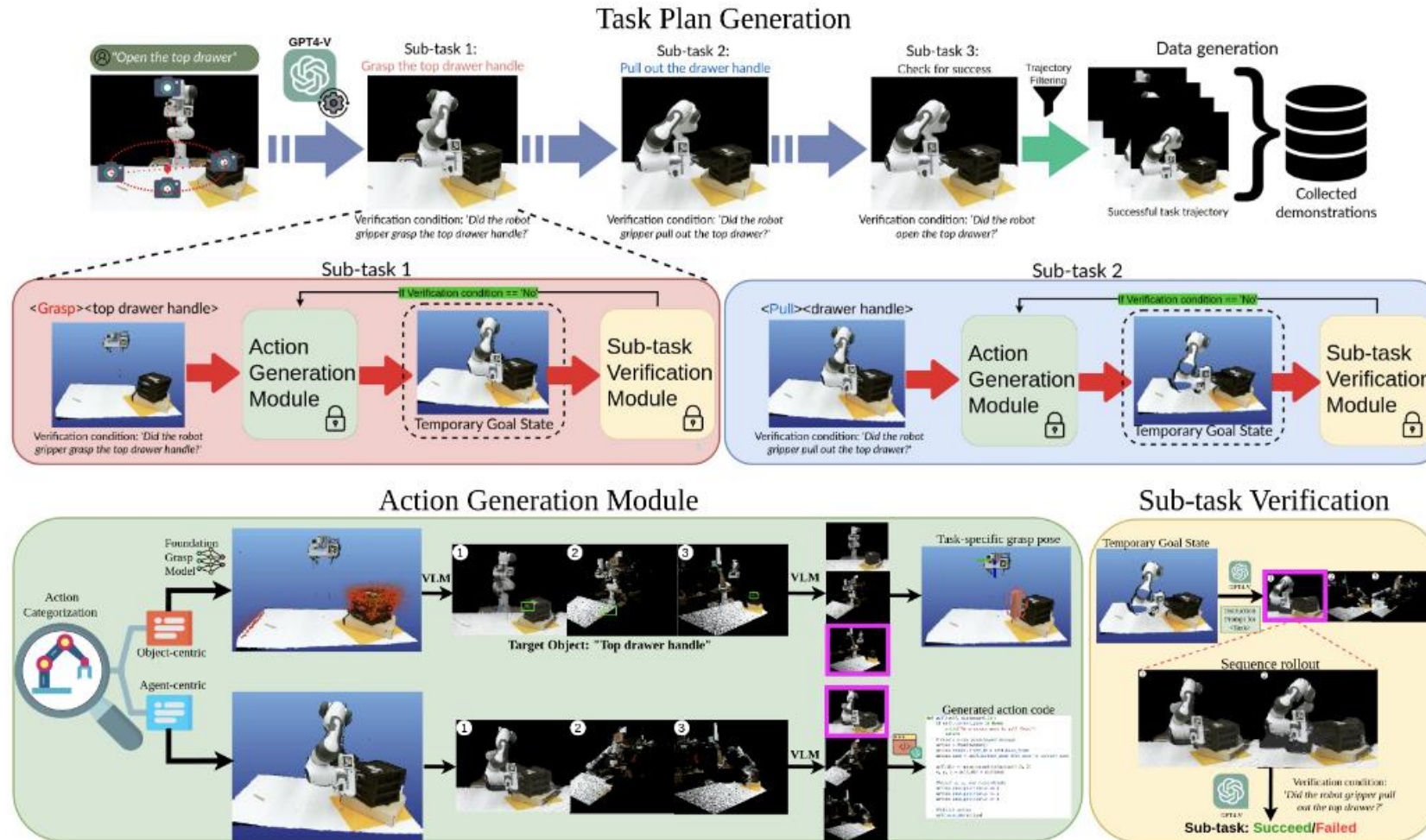
VLMs Can Do More: Task Planning



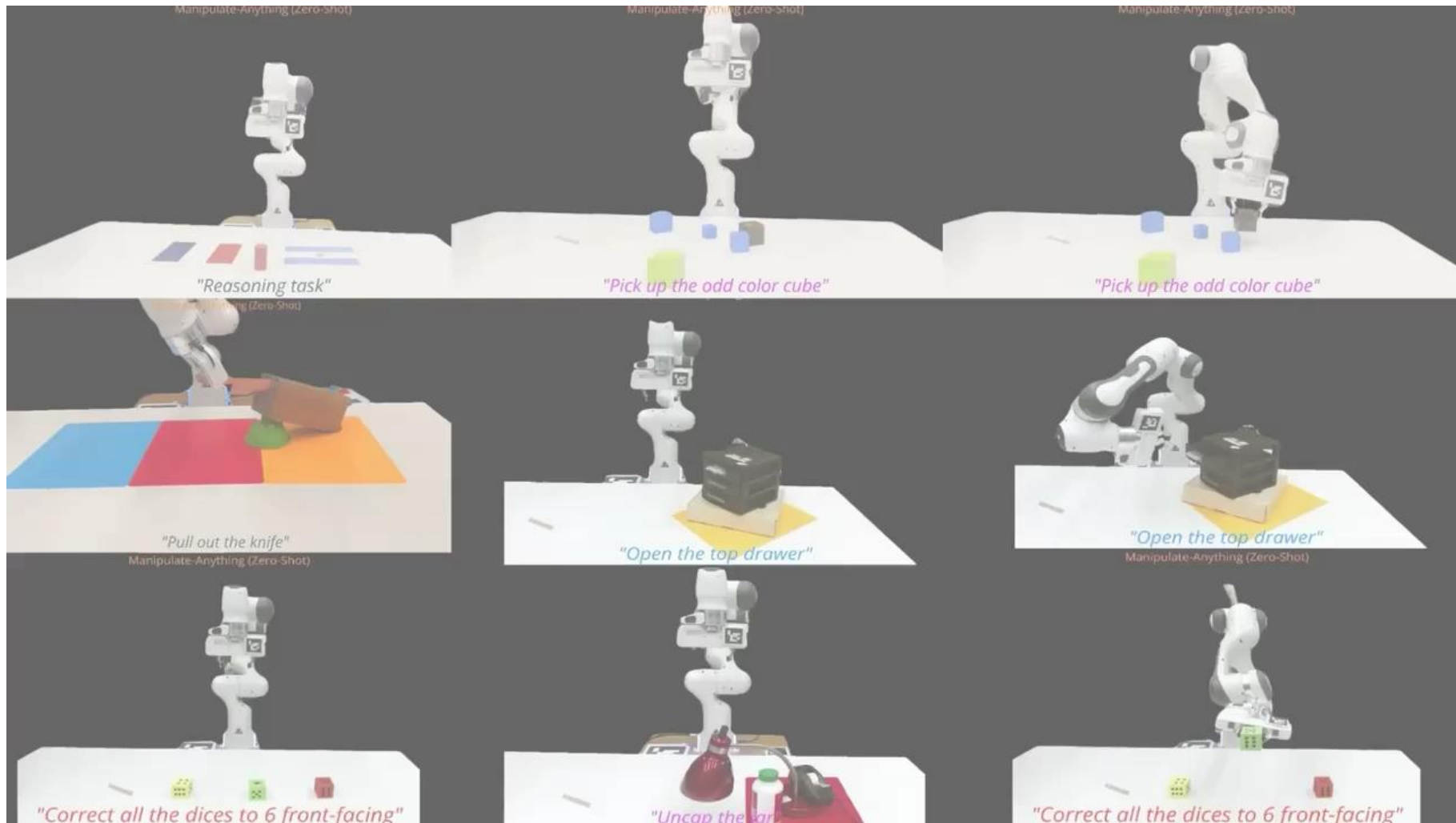
VLMs Can Do More: Success / Failure Detection



Putting Everything Together: VLMs Enable Zero-Shot Robotic Data Generation



Results: Zero-Shot Robotic Data Generation

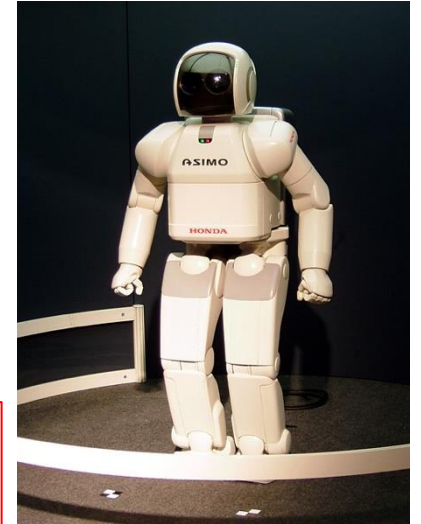
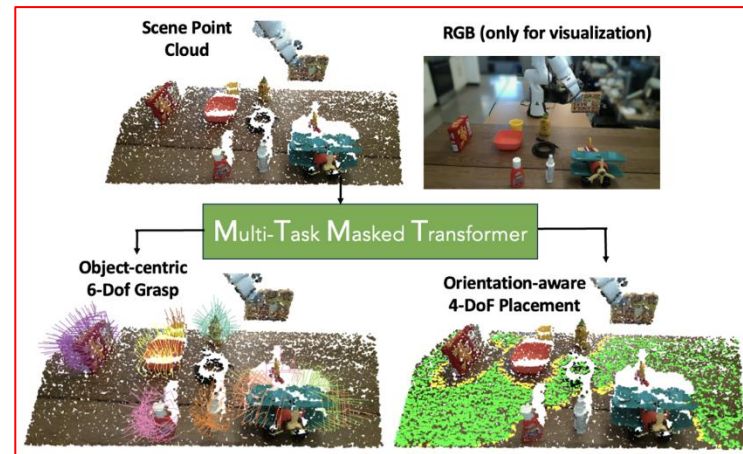


Results: Generated Demonstrations Are as High-Quality as Expert Demonstrations

- Behavior cloning with different generated data

Data	Models	Take_umbrella	Sort_mustard	Open_wine	Lamp_on	Put_knife	Pick_&_lift
VoxPoser[3]	PerAct[34]	4.00±4.00	0.00±0.00	1.33±2.31	5.33±4.62	1.33±2.31	5.67±1.64
CAP[5]	PerAct[34]	13.33±10.06	-	-	8.00±16.00	9.33±6.11	46.67±2.31
Scaling-up [4]	PerAct[34]	4.00±4.00	0.00±0.00	81.33±12.86	76.00±4.00	5.33±2.31	53.33±10.06
MA (Ours)	PerAct[34]	84.00±6.93	53.33±6.11	86.67±6.11	89.33±6.11	8.00±4.00	33.33±2.31
RLBench[33]	PerAct[34]	58.67±50.80	53.33±34.02	86.67±12.86	84.00±13.86	30.67±10.07	62.67±9.24
VoxPoser[3]	RVT-2[59]	5.33±6.11	1.33±2.31	1.33±2.31	2.67±2.31	1.33±2.31	17.33±2.31
CAP[5]	RVT-2[59]	89.33±6.11	-	-	85.33±8.32	52.00±10.58	82.66±20.53
Scaling-up [4]	RVT-2[59]	94.67±4.62	24.00±4.00	62.67±2.31	21.33±2.31	53.33±2.31	80.00±6.93
MA (Ours)	RVT-2[59]	94.67±2.31	73.33±2.31	93.33±6.11	84.00±10.58	69.33±12.85	82.67±12.22
RLBench[33]	RVT-2[59]	97.33±2.31	69.33±8.33	88.00±8.00	93.33±4.62	72.00±10.58	64.00±10.58

Is the Method All We Need? No! The Performance is Bottlenecked by the Grasp Predictor



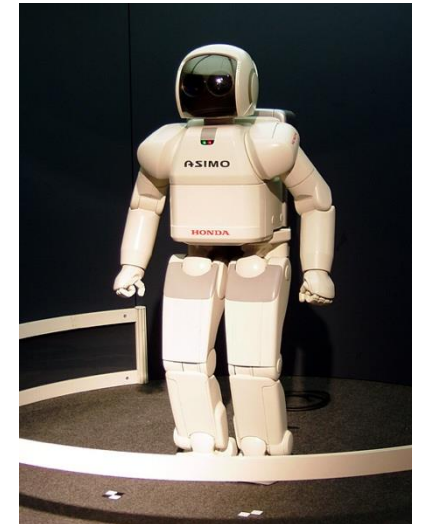
Idea 3: Prompt VLM to Generate Motion Constraints and Obtain Actions with Solvers

Motion Constraints

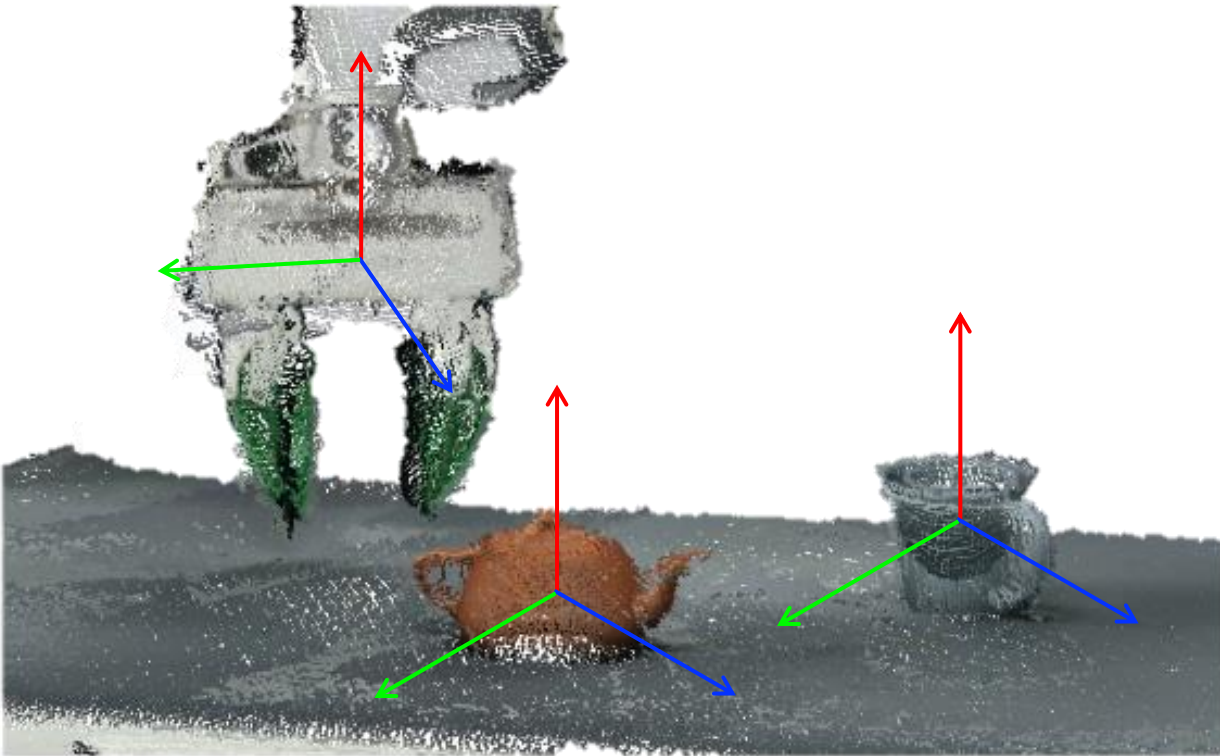


```
def subgoal_stage1_f1(k):  
    dist = norm(k[0]-k[1])  
    return dist  
  
def path_stage2_f1(k):  
    z_diff = k[1][2]-k[2][2]  
    return abs(z_diff)  
  
def subgoal_stage2_f1(k):  
    k[3][2] += 0.10  
    return norm(k[2]-k[3])  
  
...
```

Solvers



How to Represent Constraints?

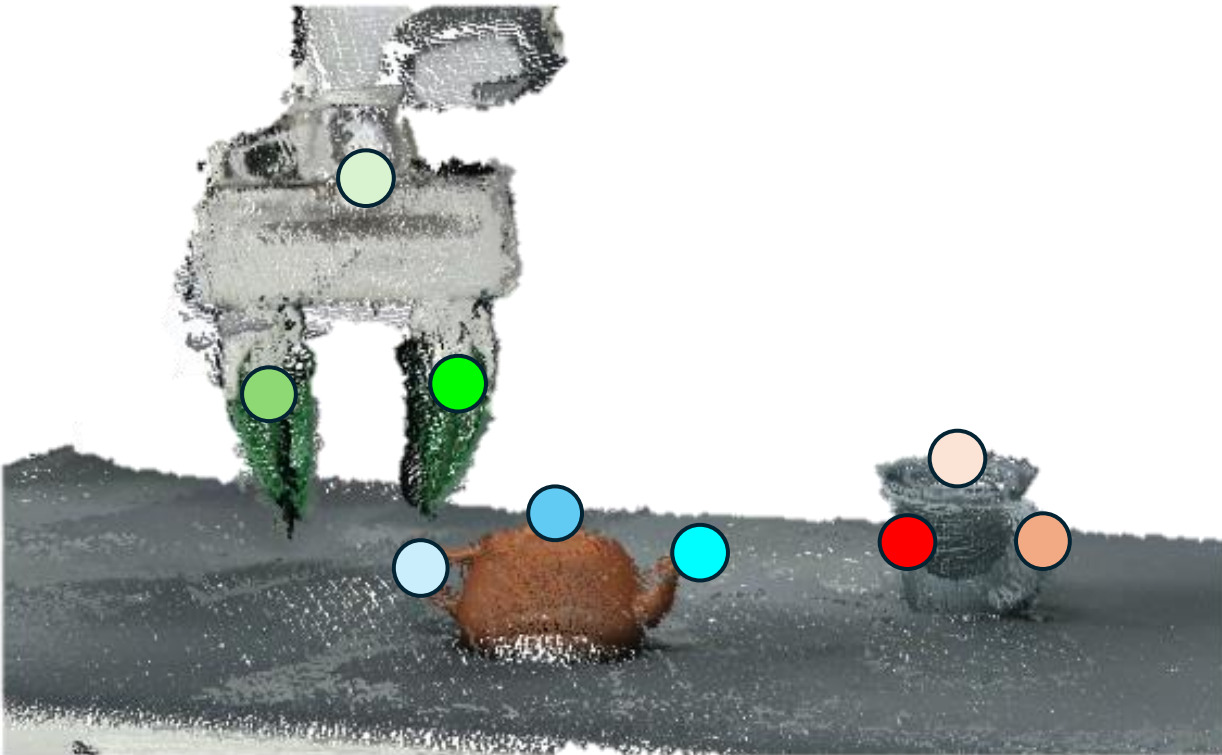


- Idea: Object-centric constraints

```
def pick_up_mug(pose_ee, pose_mug):  
    dist = norm(  
        pose_ee[:3] - pose_mug[:3])  
    return dist
```

- Problem:
 - Limited to rigid bodies
 - Understanding 3D rotation is more difficult than 3D position

How to Represent Constraints?

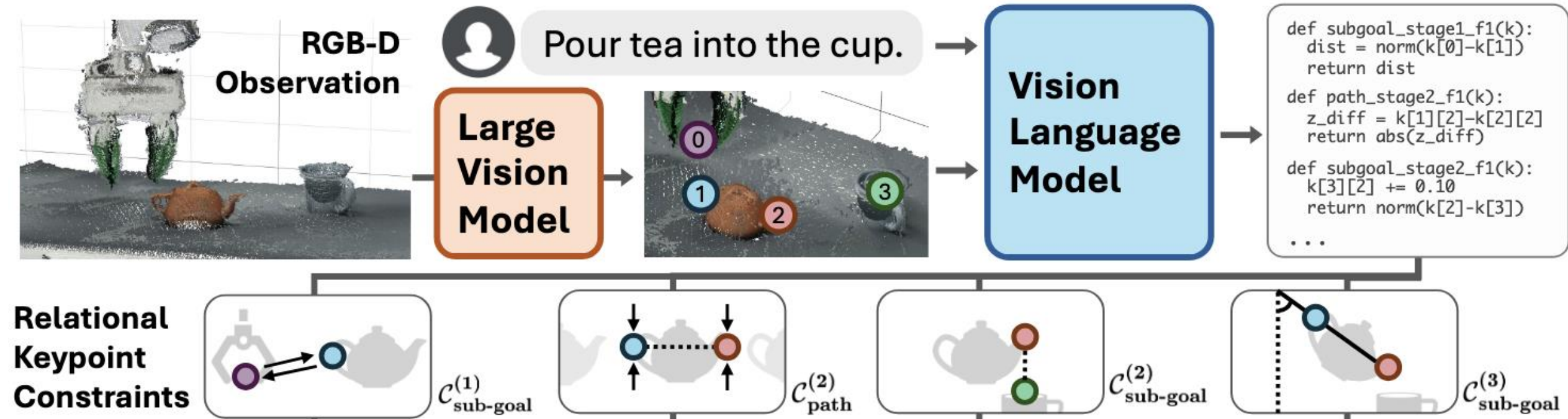


- Idea: Point-based constraints

```
def pick_up_mug(points_ee,
                points_mug):
    ee_inds = ...
    mug_inds = ...
    dist = norm(
        points_ee[ee_ind, :3]
        - pose_mug[mug_inds, :3])
    return dist
```

- Goodness:
 - Generalize to non-rigid bodies
 - Constraints on 3D rotation can be represented by constraints on points

How to Represent Constraints?



- Subgoal constraints characterize motion to achieve the current subgoal
- Path constraints characterize motion to reach the next subgoal

How to Detect Keypoints?



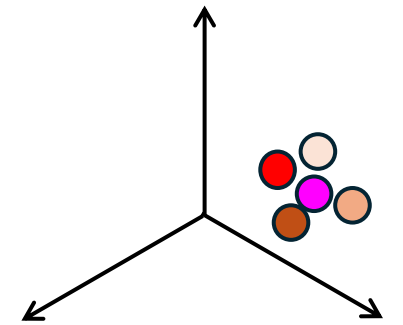
Segment an image



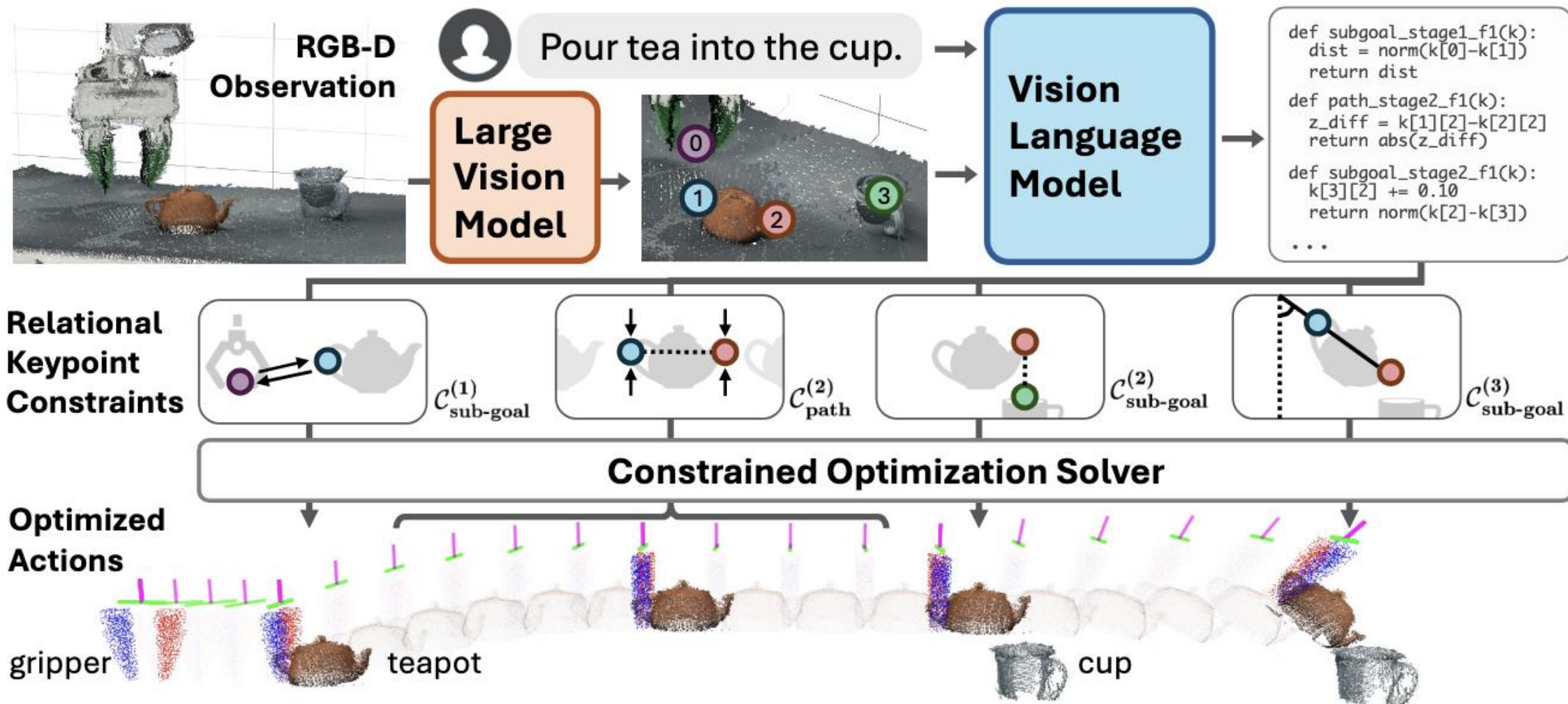
Obtain points within each mask by clustering pixel features



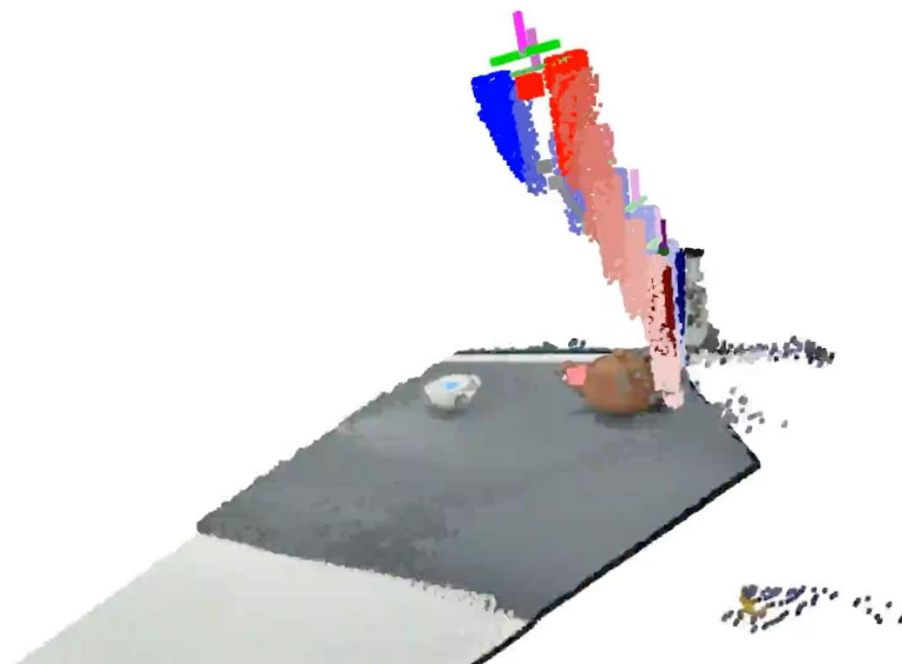
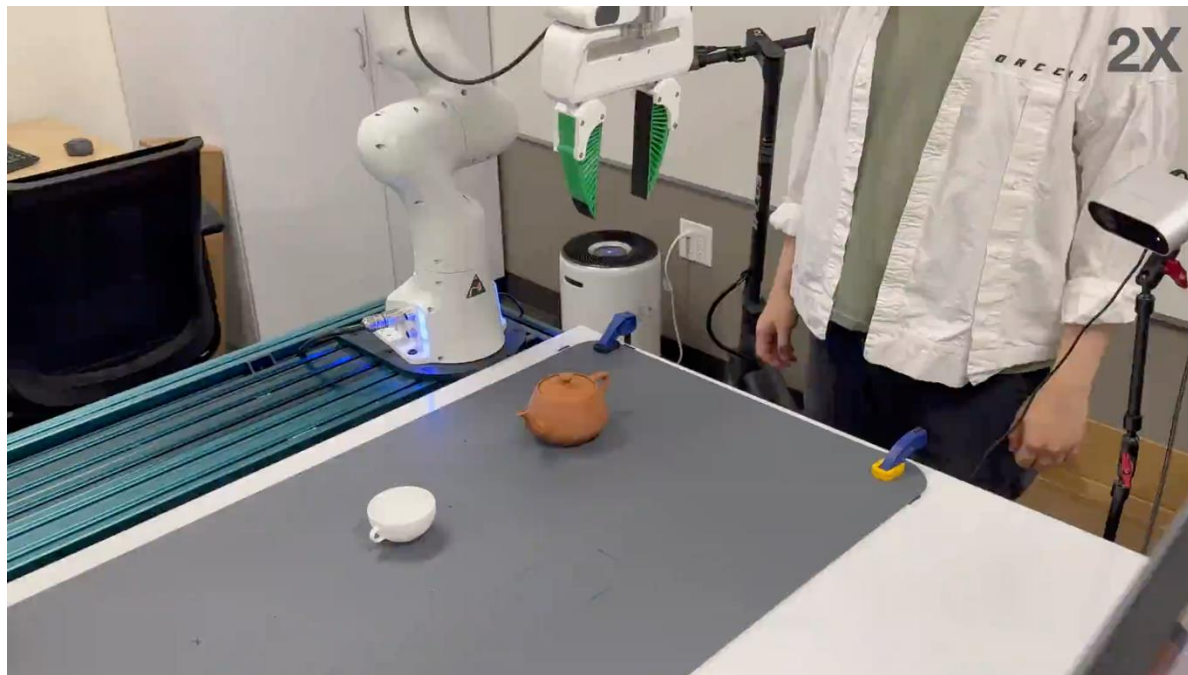
Lift points to 3D



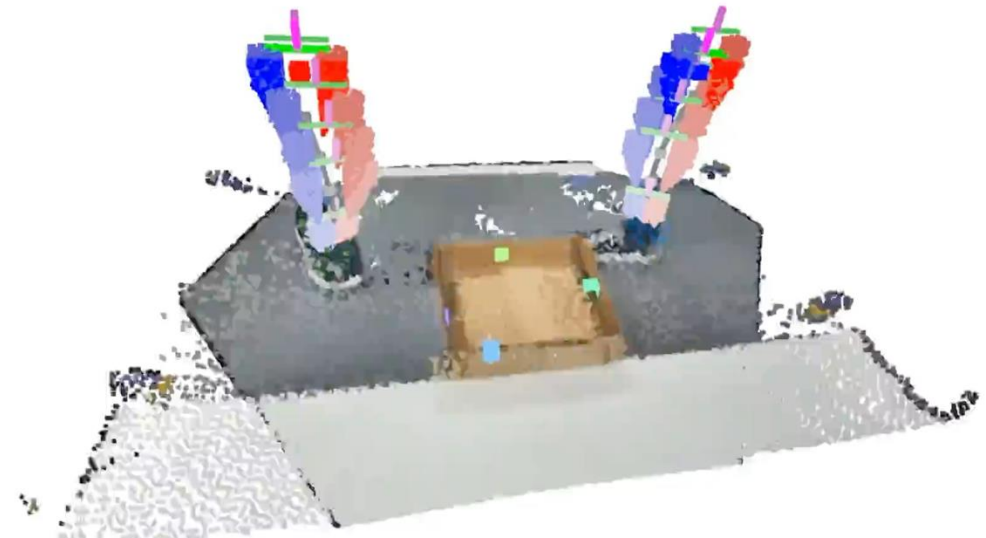
Obtain Actions by Solving the Motion Constraints



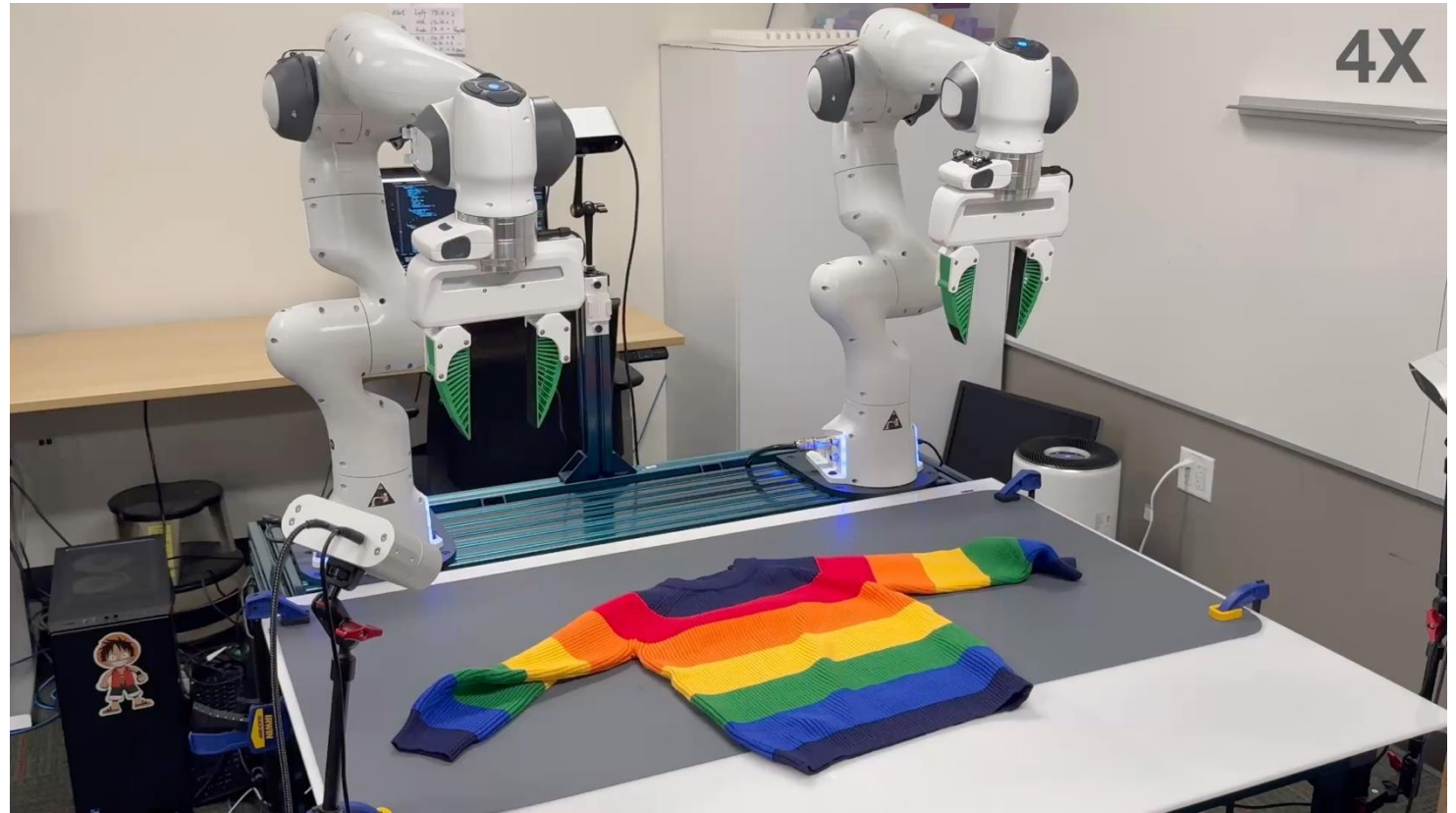
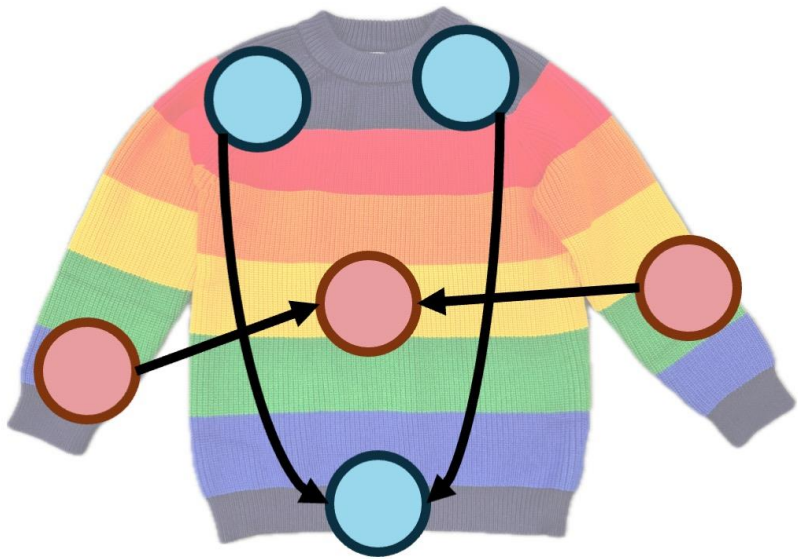
Results



Results



Results



Can We Directly Learn to Predict Actions?

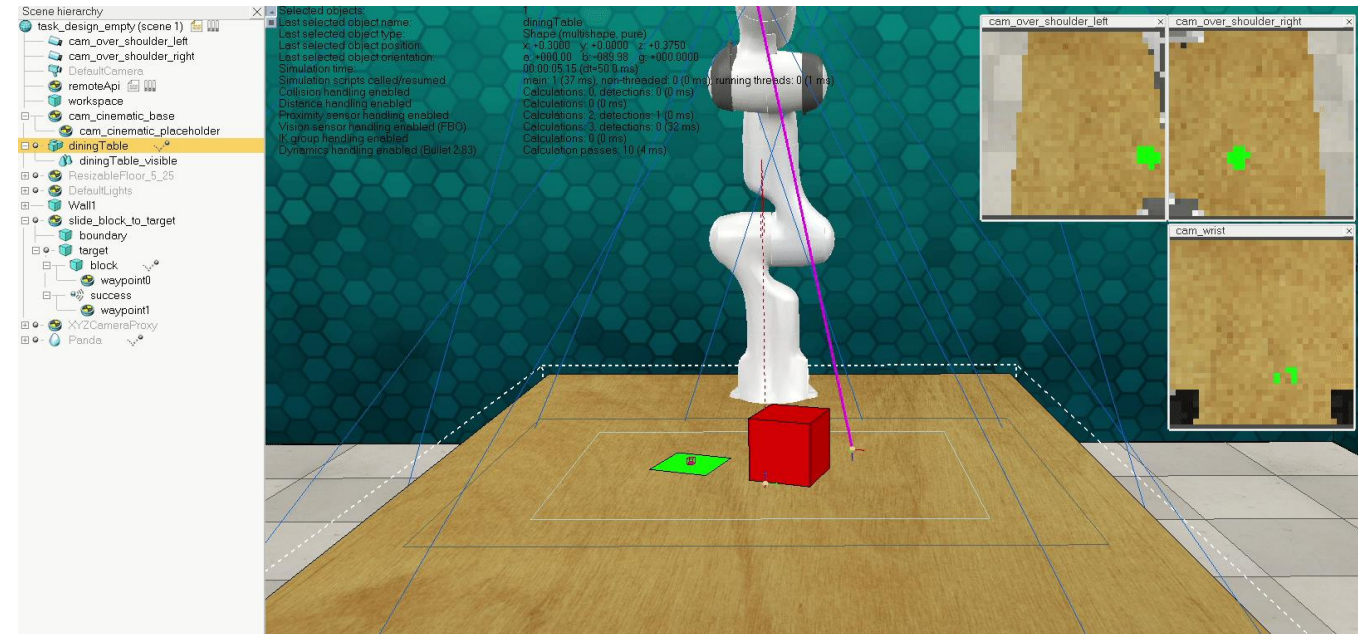
Collect Behaviors by Teleoperation



Collect Behaviors by Motion Planning



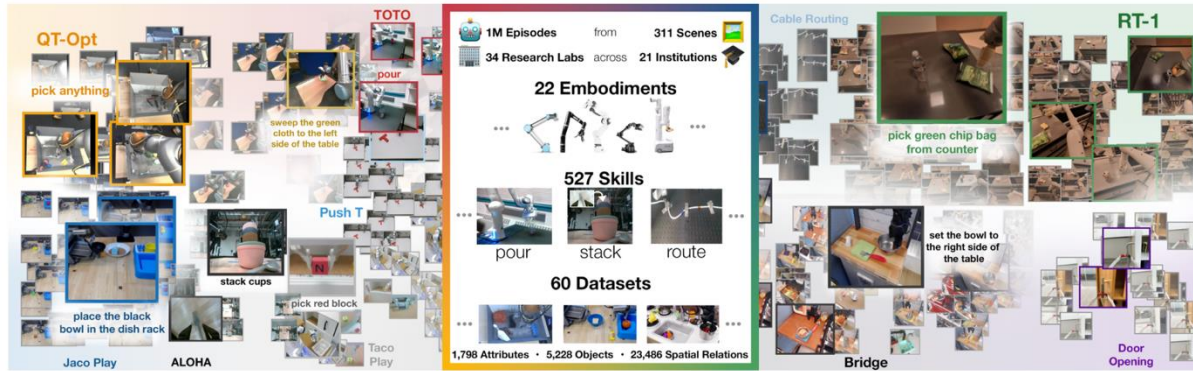
CuRoBo: CUDA Accelerated Robot Library



RLBench: Robot Learning Benchmark. James et al.

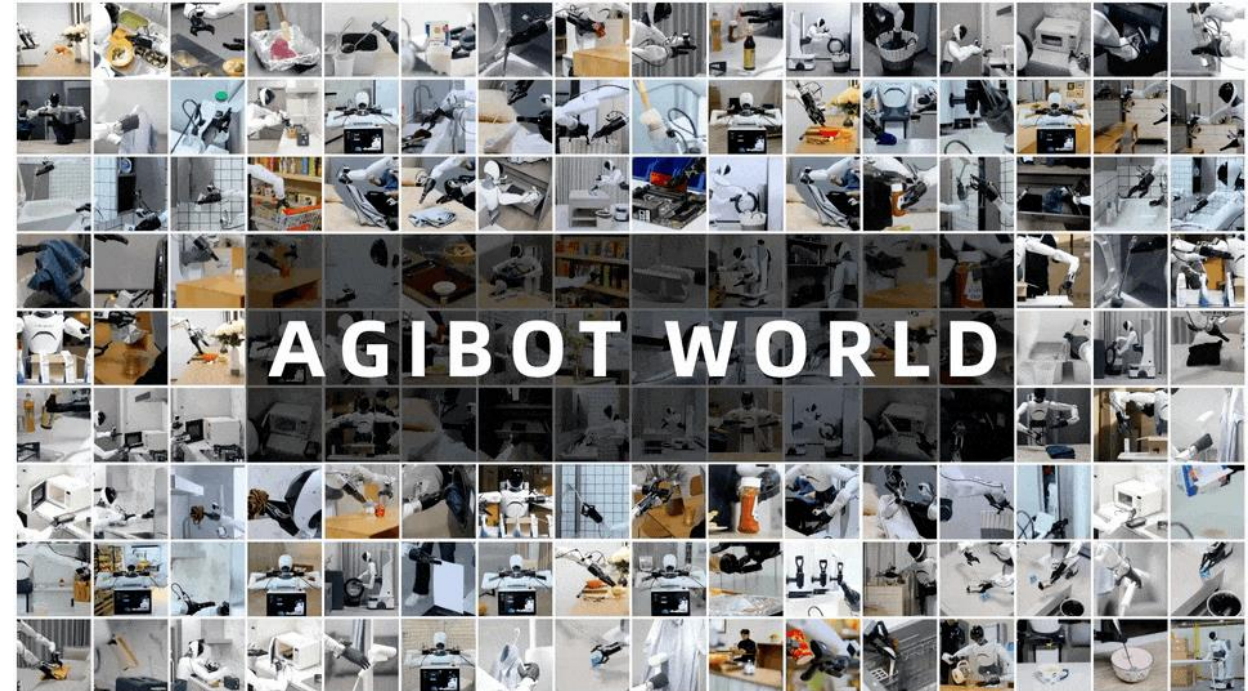
All You Need is to Collect More Demonstration Data

Open X-Embodiment

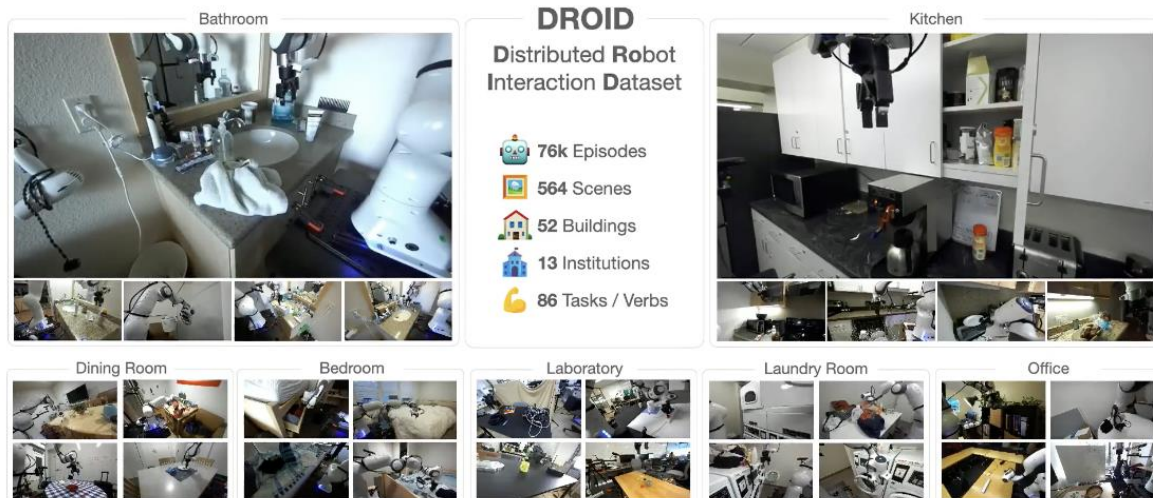


Open X-Embodiment dataset overview showing 1M Episodes from 311 Scenes across 34 Research Labs and 21 Institutions. It features 22 Embodiments, 527 Skills (pour, stack, route), and 60 Datasets. The dataset includes 1,798 Attributes, 5,228 Objects, and 23,486 Spatial Relations. Tasks shown include QT-Opt (pick anything), TOTO (pour), Cable Routing (RT-1), Bridge, Door Opening, and various household tasks like stack cups, push T, and place the black bowl in the dish rack.

AGIBOT World



DROID



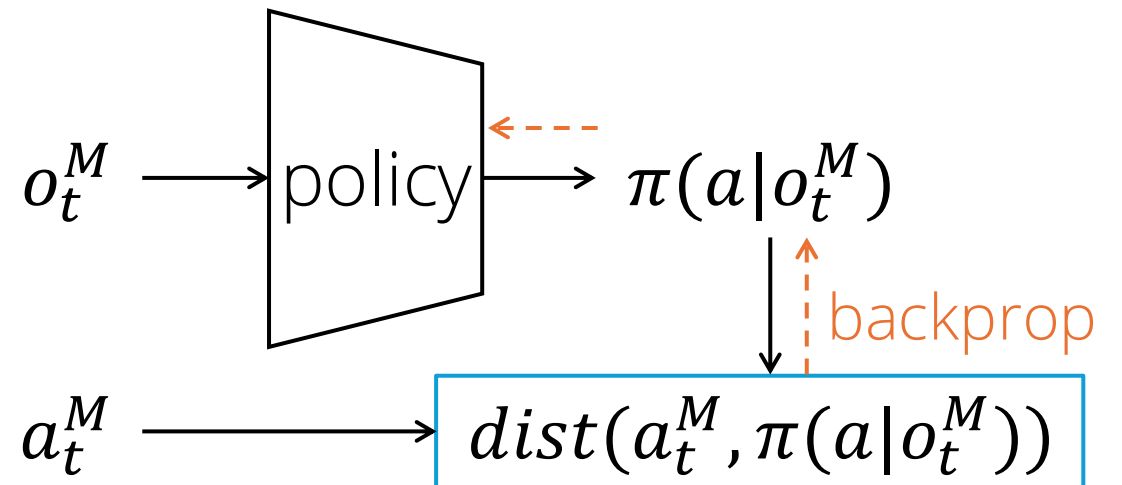
DROID Distributed Robot Interaction Dataset overview showing 76k Episodes, 564 Scenes, 52 Buildings, 13 Institutions, and 86 Tasks / Verbs. The dataset is divided into six environments: Bathroom, Kitchen, Dining Room, Bedroom, Laboratory, and Office. Each environment is represented by a grid of small images showing robot interactions.

Learning to Act by Cloning Behaviors from Demonstrations

Step 1: Collect demonstrations

$$\begin{array}{l} o_1^1, a_1^1, o_2^1, a_2^1, \dots, o_T^1, a_T^1 \\ o_1^2, a_1^2, o_2^2, a_2^2, \dots, o_T^2, a_T^2 \\ \dots \\ o_1^N, a_1^N, o_2^N, a_2^N, \dots, o_T^N, a_T^N \end{array}$$

Step 2: Train the policy to imitate



Multi-modality Behaviors: The Same Task can be Done in Different Ways

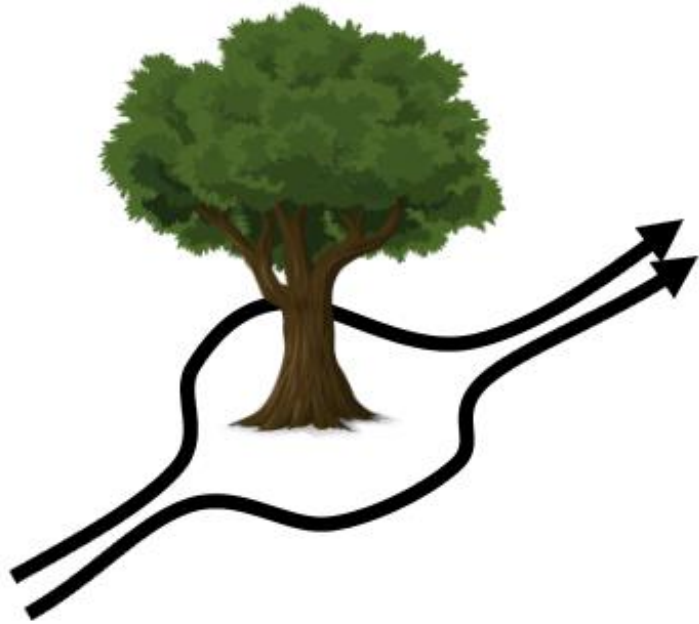
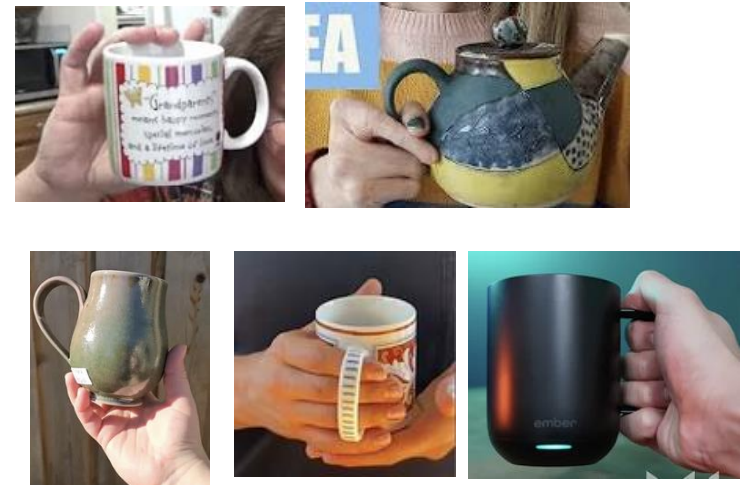


Image credit S. Levine]

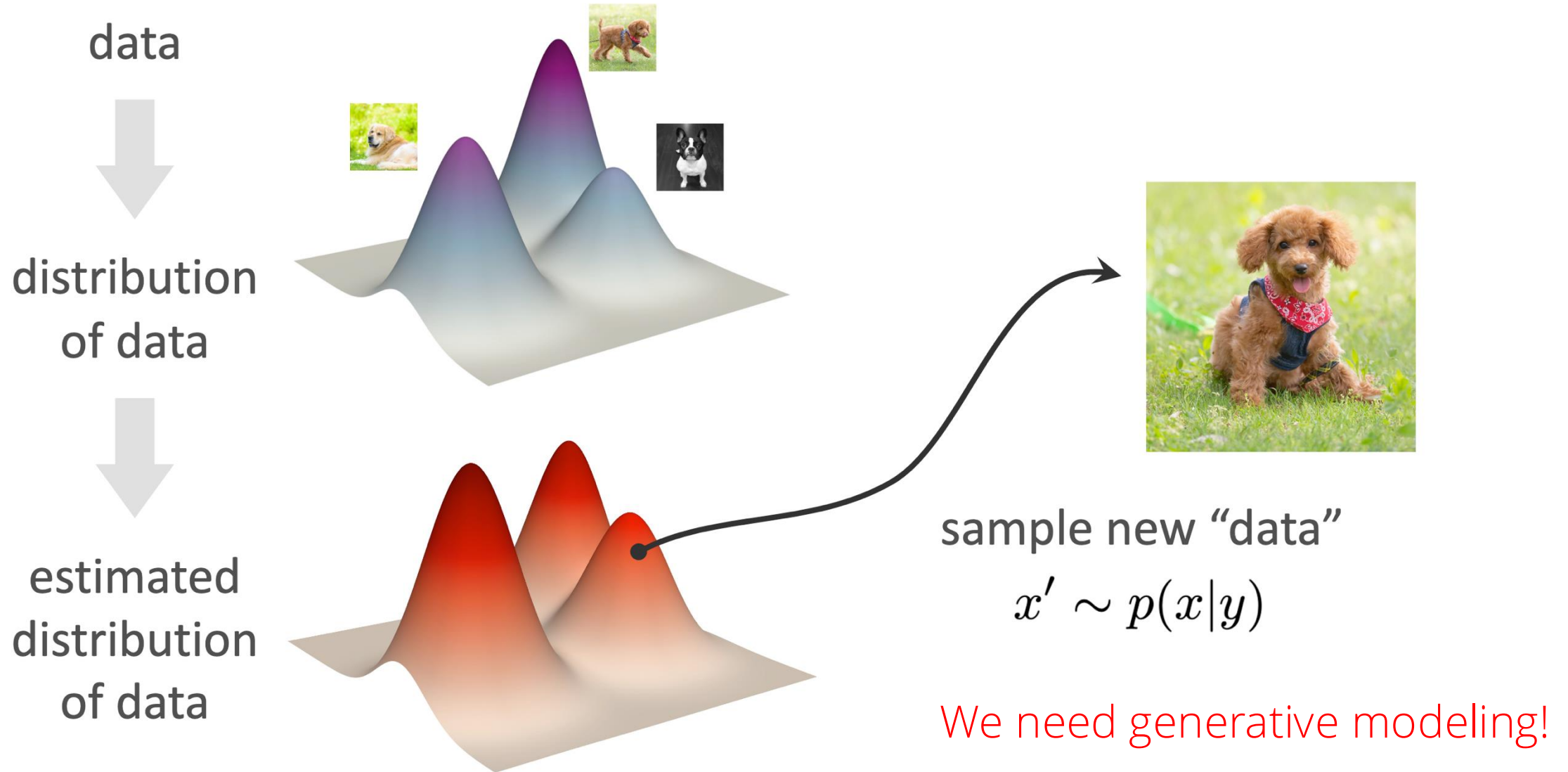


Diffusion policy: Visuomotor policy learning via action diffusion



How to model multi-modal behaviors?

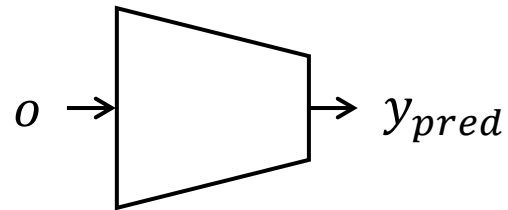
Generative models w/ probabilistic modeling



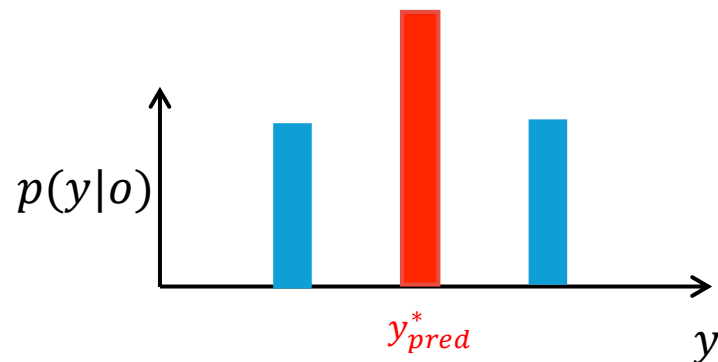
Recap: Capture Probabilistic Distribution

Deterministic

- predicts the same output given the same input

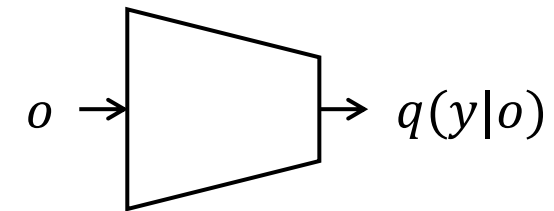


$$y_{pred}^* = \underset{y_{pred}}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^N |y_{pred} - y_n|^2$$

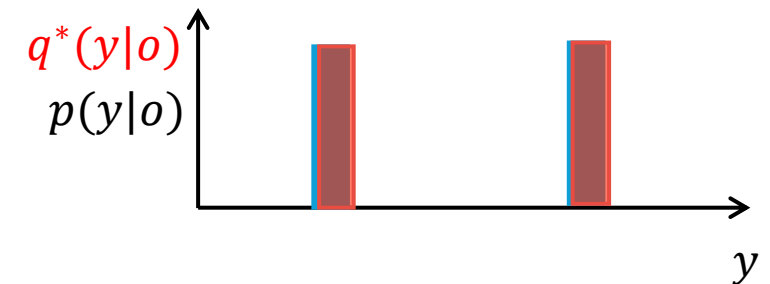


Probabilistic

- predicts the same likelihood of all output given the same input



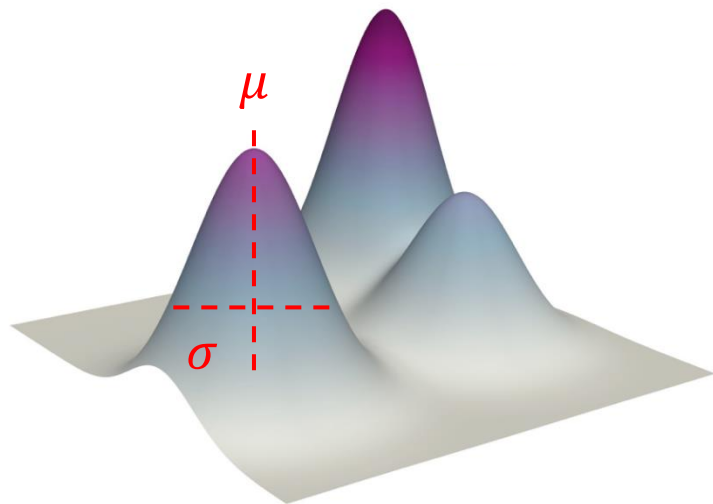
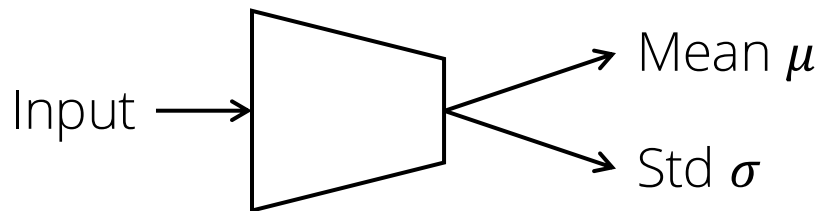
$$q^*(y|o) = \underset{q^*}{\operatorname{argmin}} D_{KL}(q(y|o), p(y|o))$$



Recap: Probabilistic Generative Models

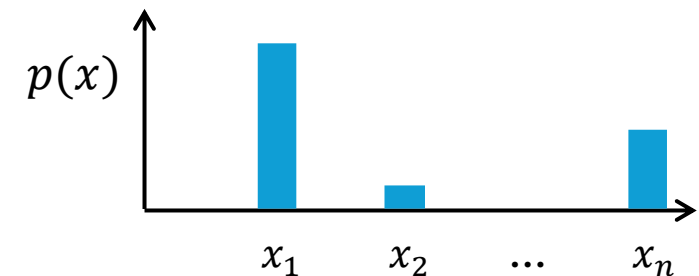
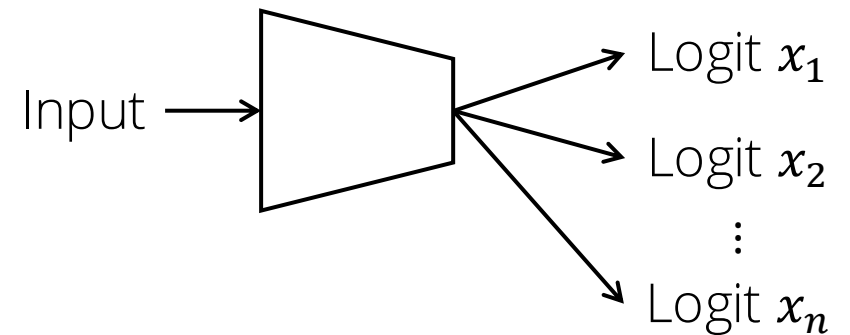
Gaussian

- Predict the mean and standard deviation to model the probability of sample "x"
- VAE, Diffusion models etc

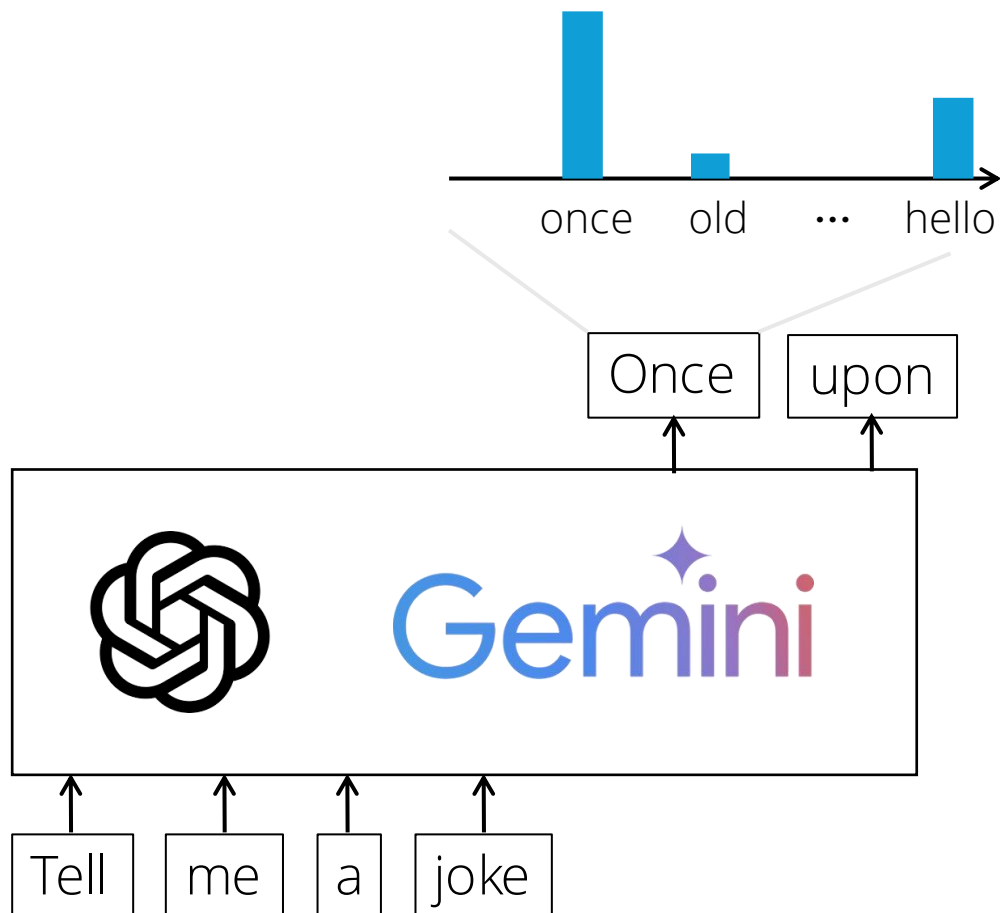


Categorical

- Predict the categorical probability of sample "x"
- Most LLMs etc



Recap: Discrete Autoregressive Transformer as Generative Models



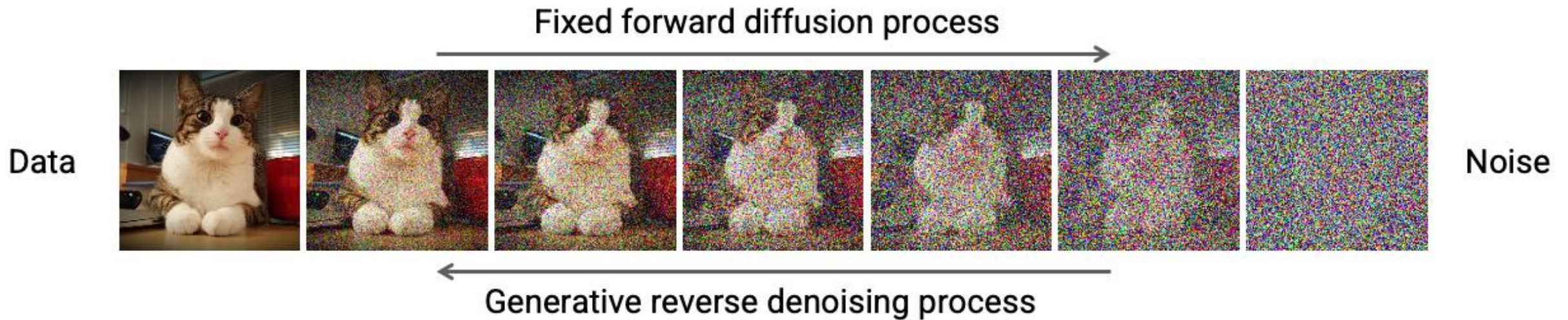
GPT-4.5

I'm going through a tough time after failing a test

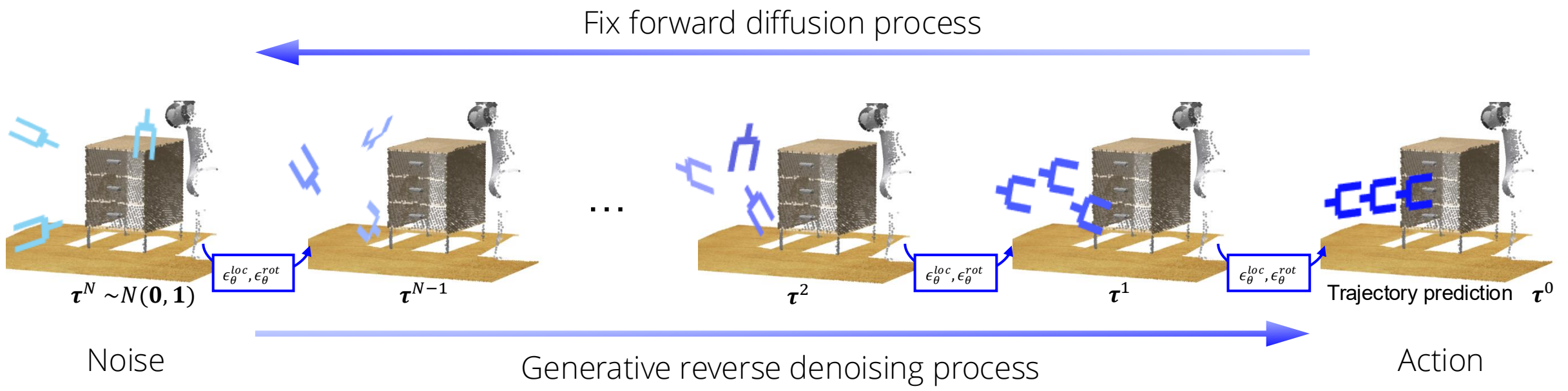
Aw, I'm really sorry to hear that. Failing a test can feel pretty tough and discouraging, but remember, it's just one moment—not a reflection of who you are or your capabilities. Want to talk about what happened, or do you just need a distraction? I'm here either way.

<https://openai.com/index/introducing-gpt-4-5/>

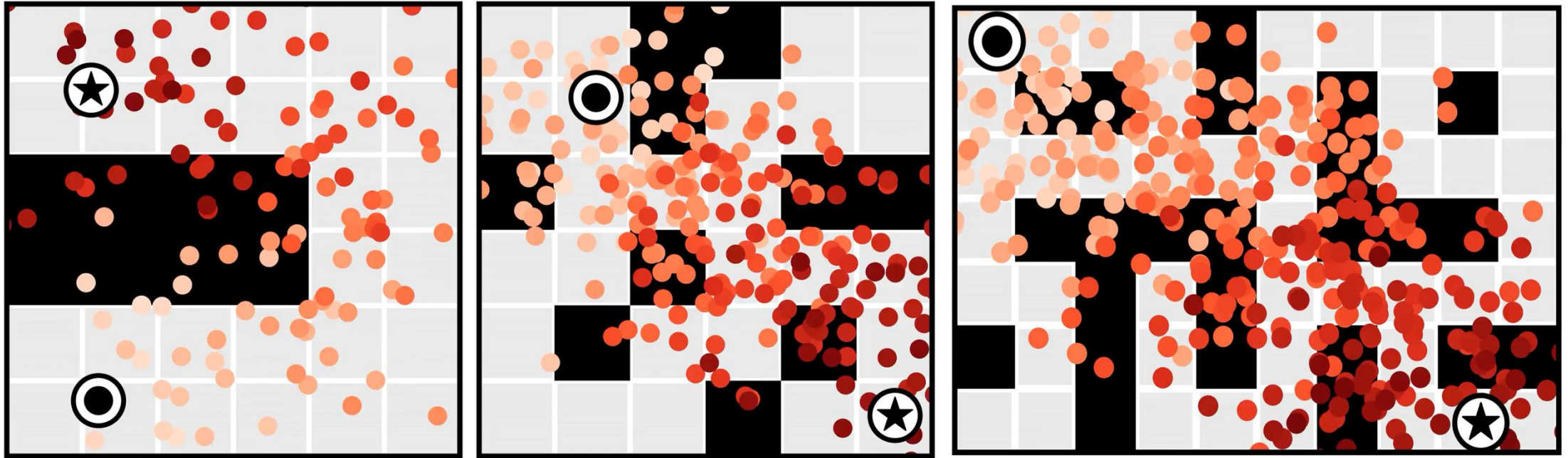
Recap: Diffusion Models as Generative Models



Learning Diffusion Models as Robot Policies

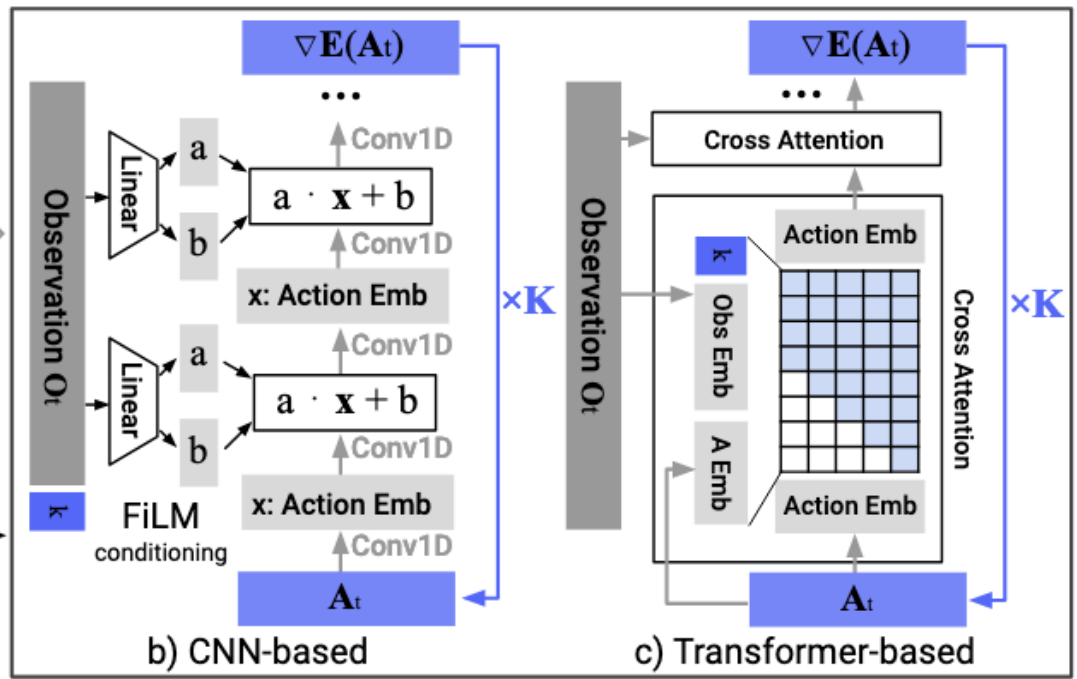
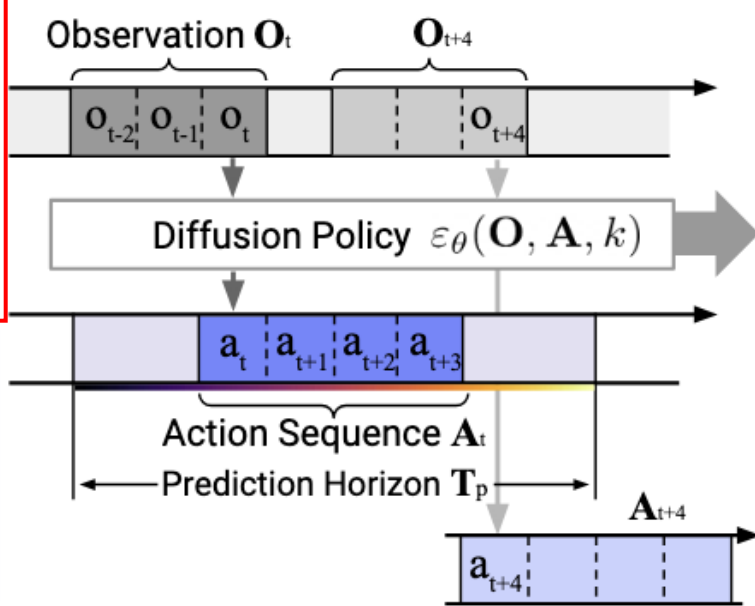
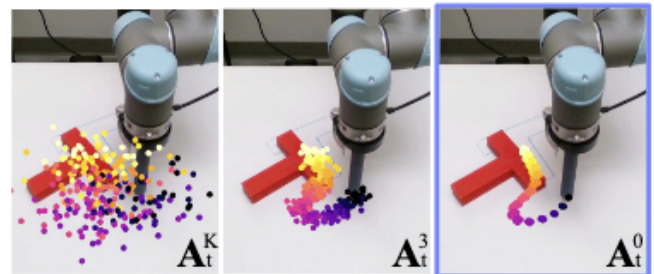


Learning Diffusion Models as Robot Policies



Diffusion Policy with 2D Inputs

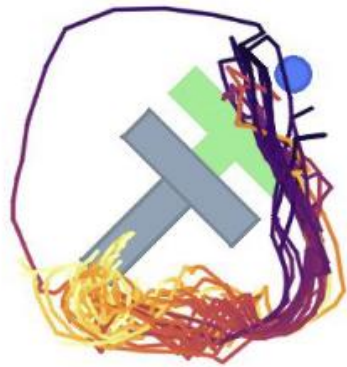
An input image is represented by a flattened feature vector



Diffusion Policy with 2D Inputs



Diffusion Policy



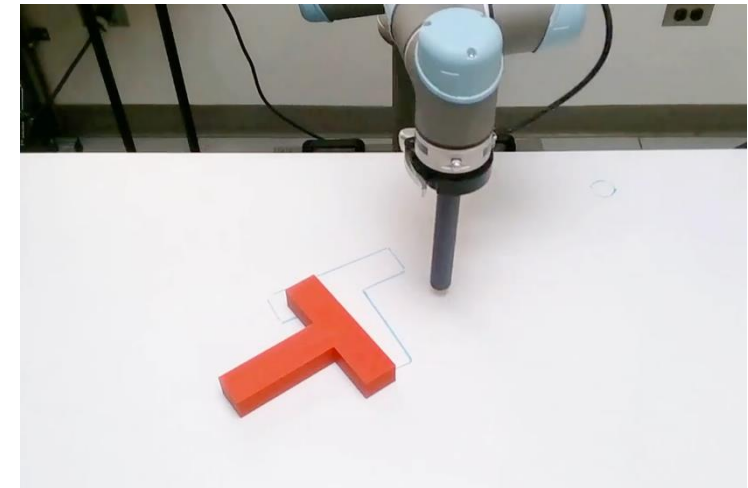
LSTM-GMM



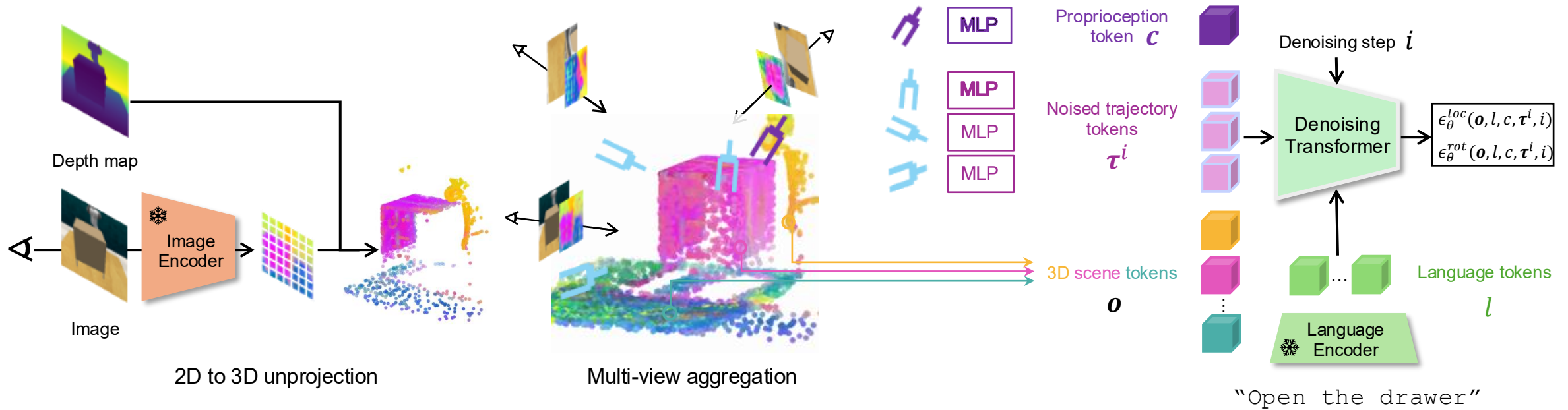
BET



IBC



Diffusion Policy with 3D Inputs



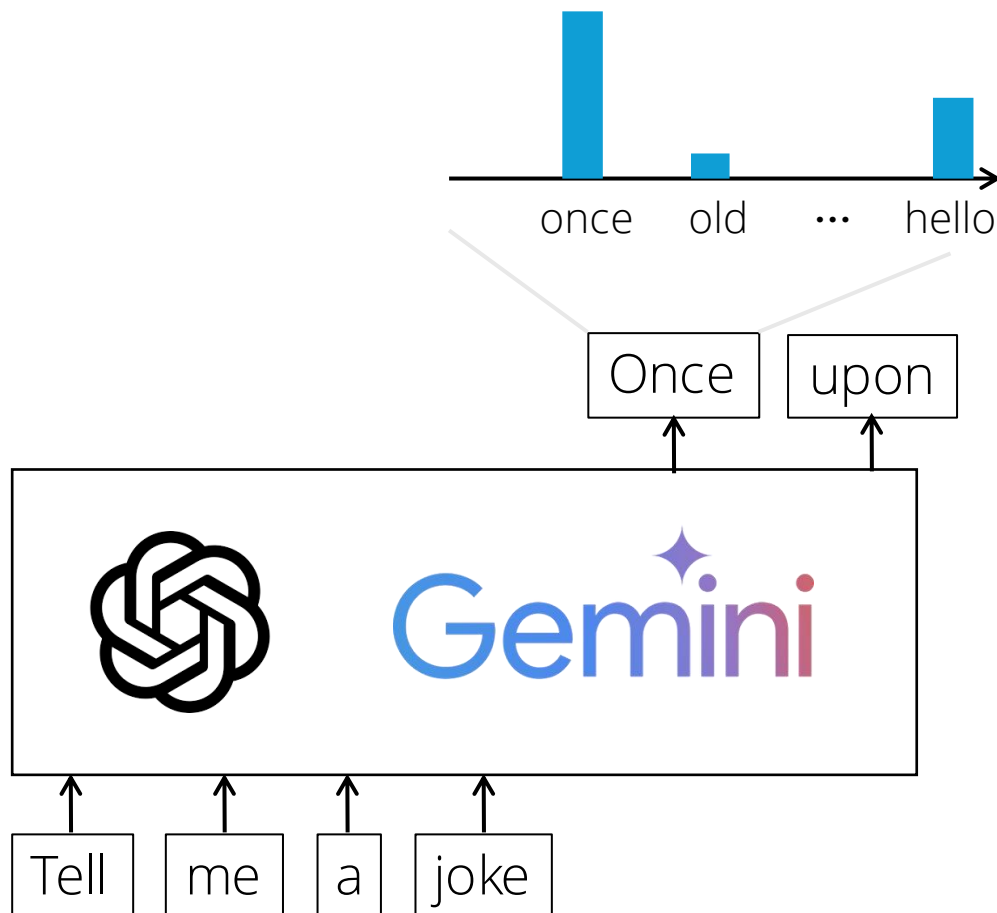
Diffusion Policy with 3D Inputs

Calvin

Real-world



Recap: Discrete Autoregressive Transformer as Generative Models



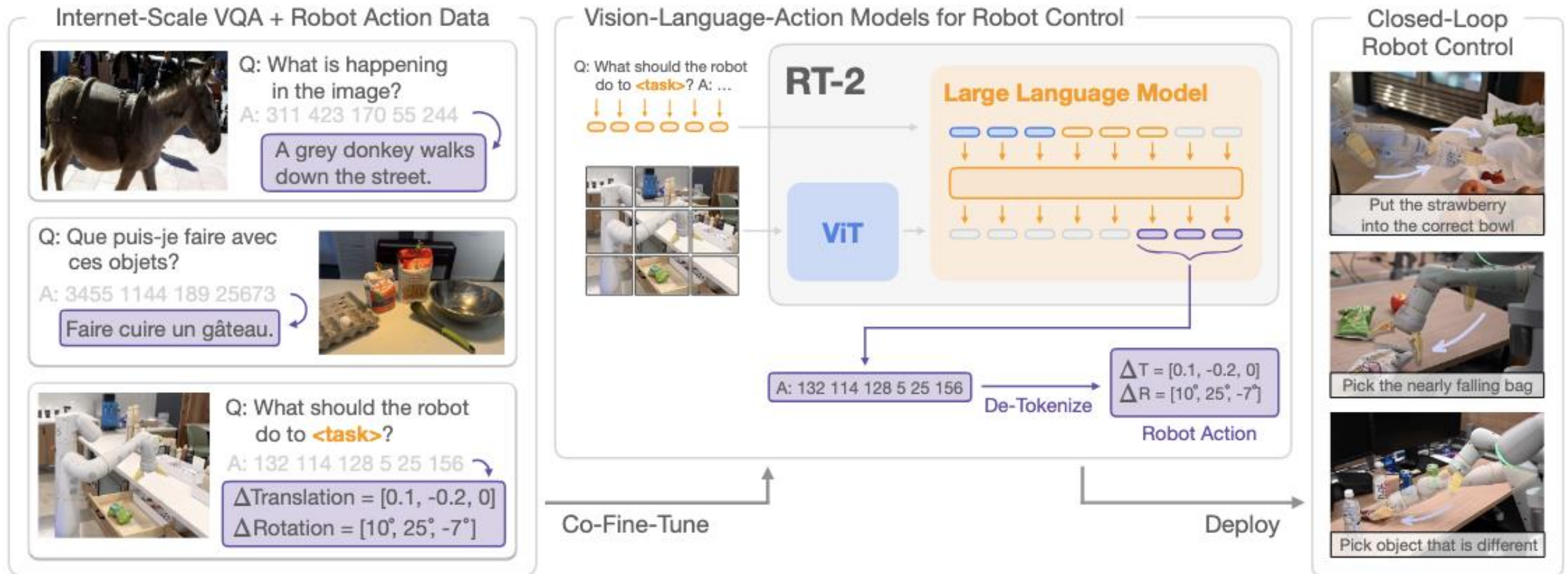
GPT-4.5

I'm going through a tough time after failing a test

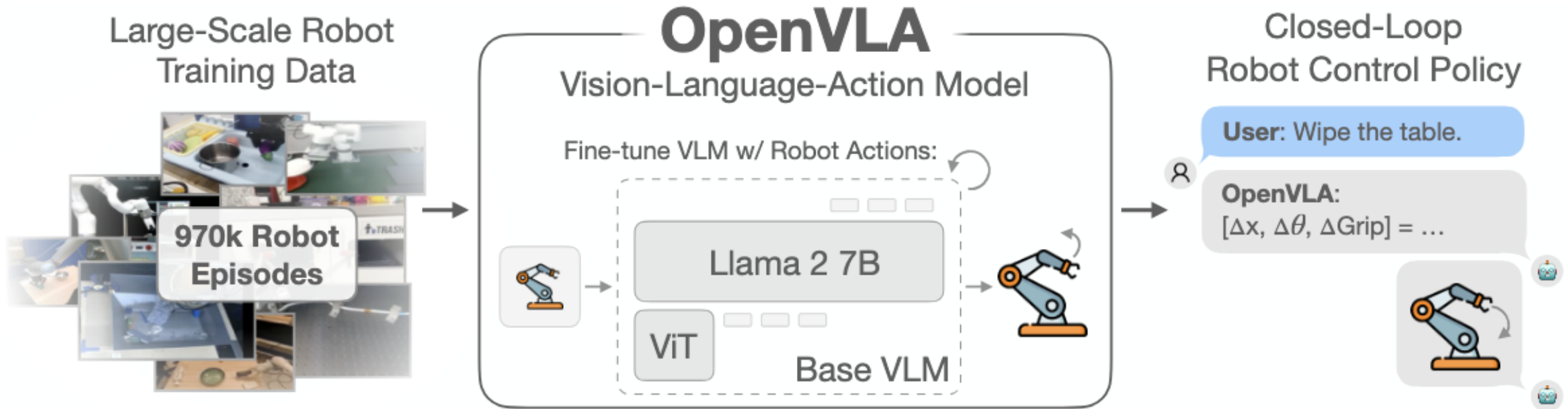
Aw, I'm really sorry to hear that. Failing a test can feel pretty tough and discouraging, but remember, it's just one moment—not a reflection of who you are or your capabilities. Want to talk about what happened, or do you just need a distraction? I'm here either way.

Can we have learn autoregressive transformers as robot policies?

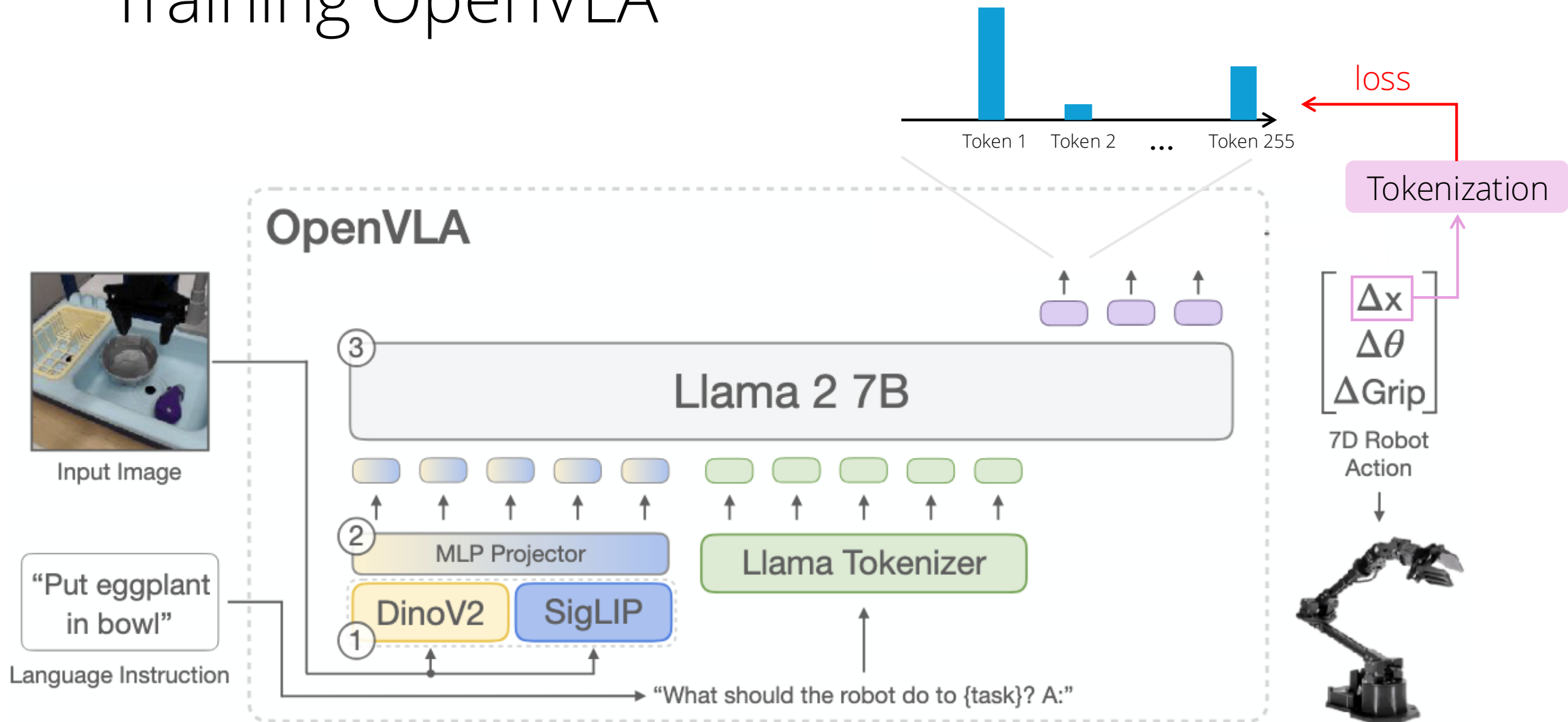
Idea: Transformer Vision-Language Models (VLMs) into Vision-Language-Action Models (VLAs)



Idea: Transformer Vision-Language Models (VLMs) into Vision-Language-Action Models (VLAs)



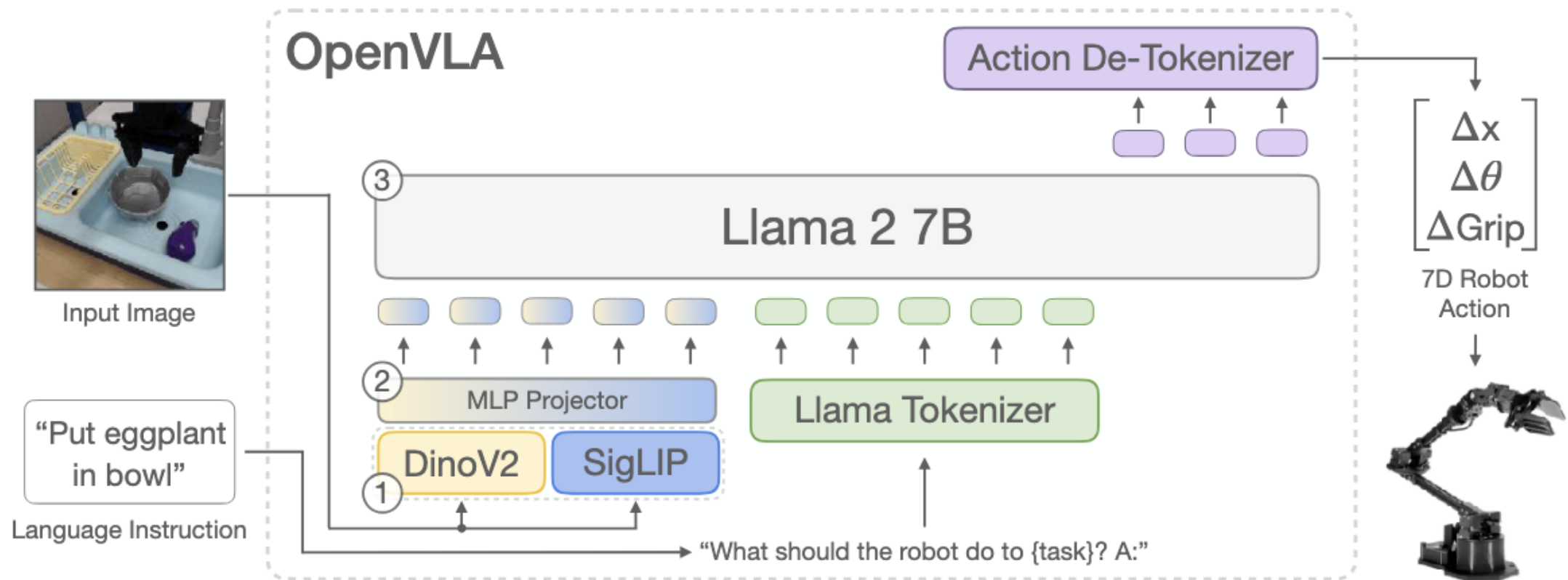
Training OpenVLA



How to tokenize real-valued actions?

Normalization and mapping to rarely used tokens

Inference with OpenVLA

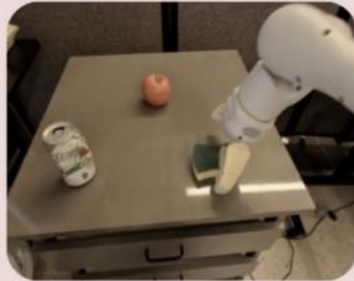


Is OpenVLA All We Need? No! It doesn't Generalize Well...

Train on real-world datasets and test on digital twins of these real-world environments

Real robot evaluation (train on real, evaluate in **real**)

Expensive and slow
Difficult to reproduce



Simulated evaluation (train on real, evaluate in **sim**)

Cheap and scalable
Fully reproducible



Real world



Digital twins



Is OpenVLA All We Need? No! It doesn't Generalize Well...

Real robot evaluation (train on real, evaluate in **real**)

Expensive and slow
Difficult to reproduce



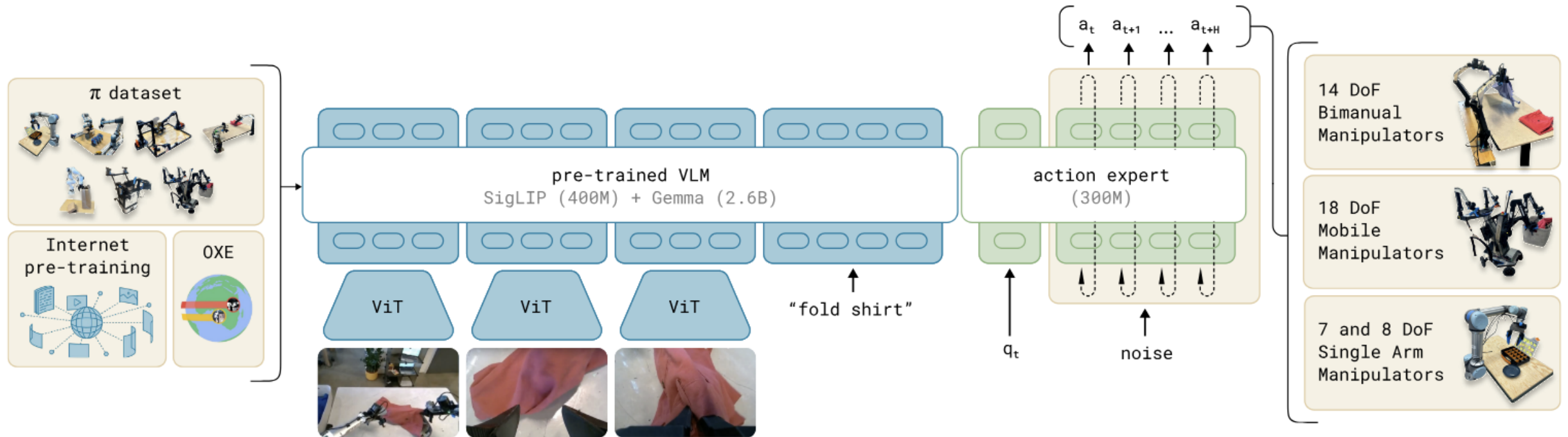
Simulated evaluation (train on real, evaluate in **sim**)

Cheap and scalable
Fully reproducible

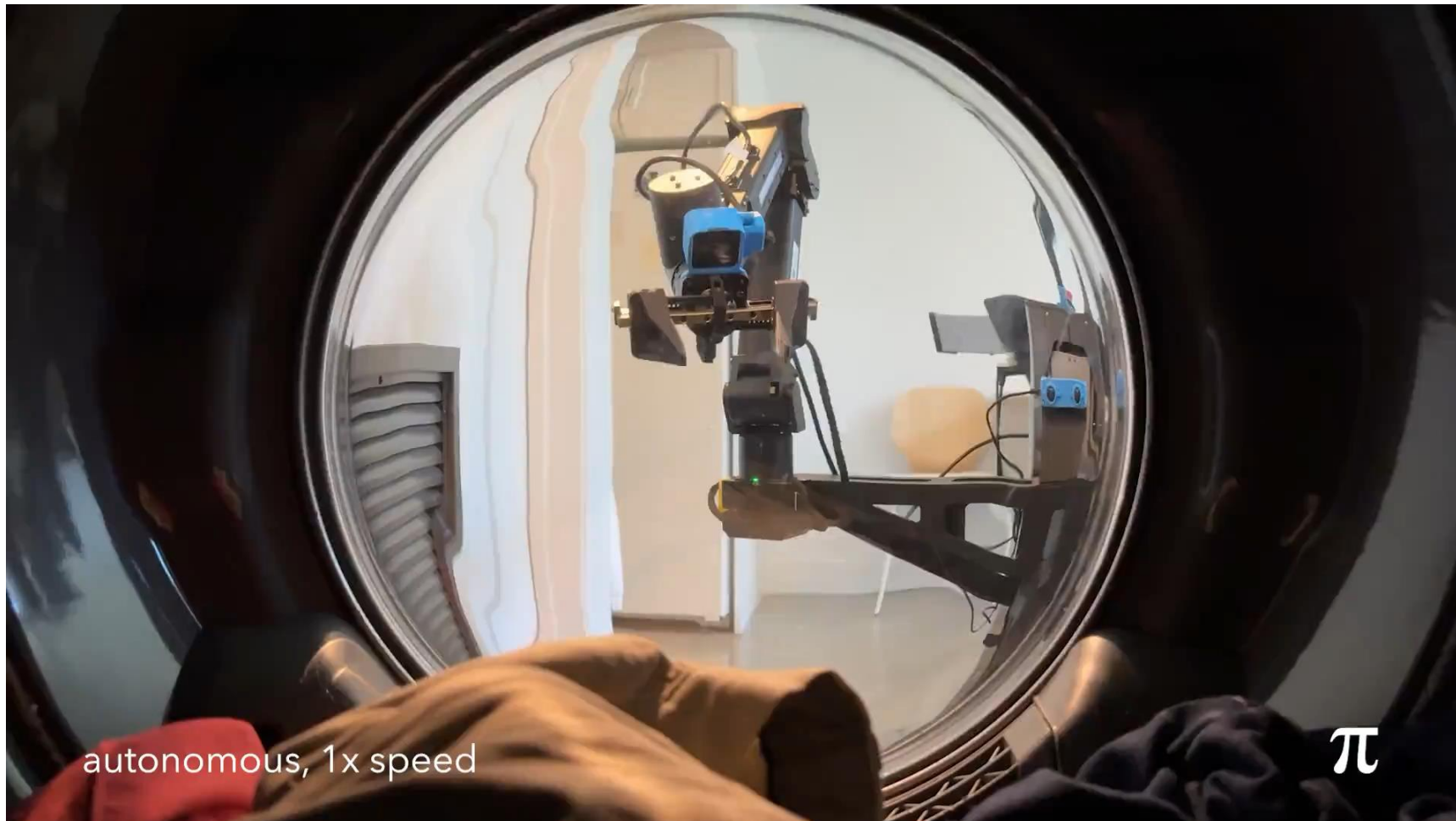


Model	Put Spoon on Towel		Put Carrot on Plate		Stack Green Block on Yellow Block		Put Eggplant in Yellow Basket		#Overall Average
	Grasp Spoon	Success	Grasp Carrot	Success	Grasp Green Block	Success	Grasp Eggplant	Success	
RT-1-X [13]	16.7%	0%	20.8%	4.2%	8.3%	0%	0.0%	0%	1.1%
Octo-Base [48]	34.7%	12.5%	52.8%	8.3%	31.9%	0%	66.7%	43.1%	16.0%
Octo-Small [48]	77.8%	47.2%	27.8%	9.7%	40.3%	4.2%	87.5%	56.9%	30.0%
OpenVLA [30]	4.1%	0%	33.3%	0%	12.5%	0%	8.3%	4.1%	1.0%
RoboVLM (zero-shot) [32]	37.5%	20.8%	33.3%	25.0%	8.3%	8.3%	0.0%	0%	13.5%
RoboVLM (fine-tuning) [32]	54.2%	29.2%	25.0%	25.0%	45.8%	12.5%	58.3%	58.3%	31.3%
SpatialVLA (zero-shot)	25.0%	20.8%	41.7%	20.8%	58.3%	25.0%	79.2%	70.8%	34.4%
SpatialVLA (fine-tuning)	20.8%	16.7%	29.2%	25.0%	62.5%	29.2%	100.0%	100.0%	42.7%

What's More Can We Do? VLA + Diffusion Models



Very Impressive Real-World Robotic Demo



Can We Incorporate Reasoning in VLAs?

For LLMs, Intermediate Reasoning Enhances Text Generation Tasks

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ❌

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✅

Idea: Incorporate Reasoning in VLAs

- Train VLAs to perform reasoning before generating actions
- Problems:
 - What to reason?
 - Where is the training data?

Regular VLA Policy

User: Clean the sink.

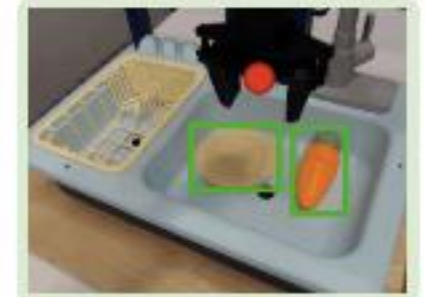
OpenVLA:
[Δx , $\Delta \theta$, ΔGrip] = ...



VLA w/ Embodied Chain-of-Thought

User: Clean the sink.

CoT OpenVLA:
Object bounding boxes:
Gripper position: ●

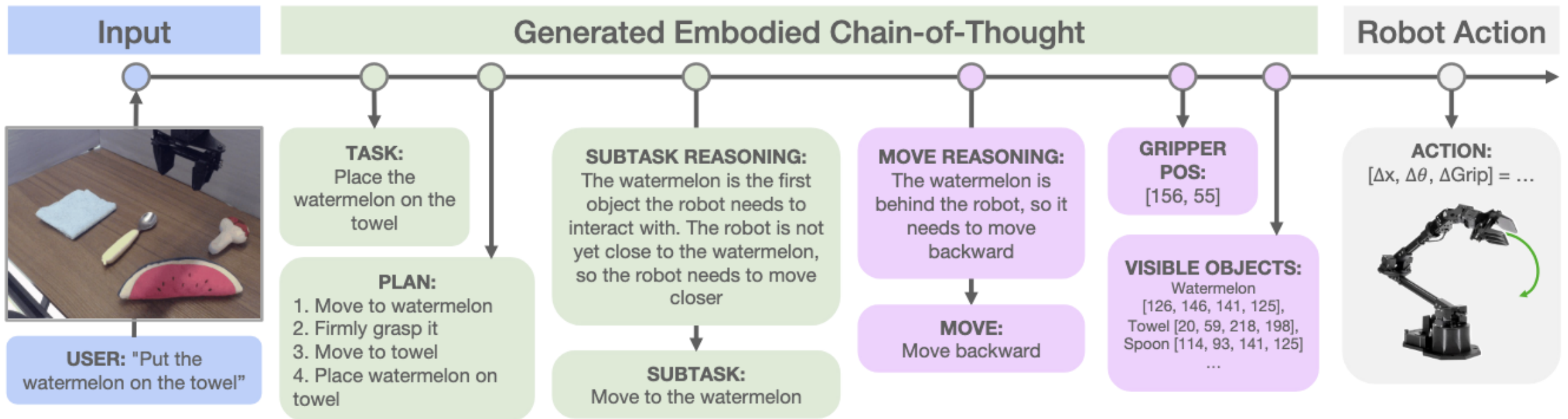


CoT OpenVLA:
Task Plan: 1. ..., 2. ...
Next Task: pick up carrot
Next Move: move right

CoT OpenVLA:
[Δx , $\Delta \theta$, ΔGrip] = ...

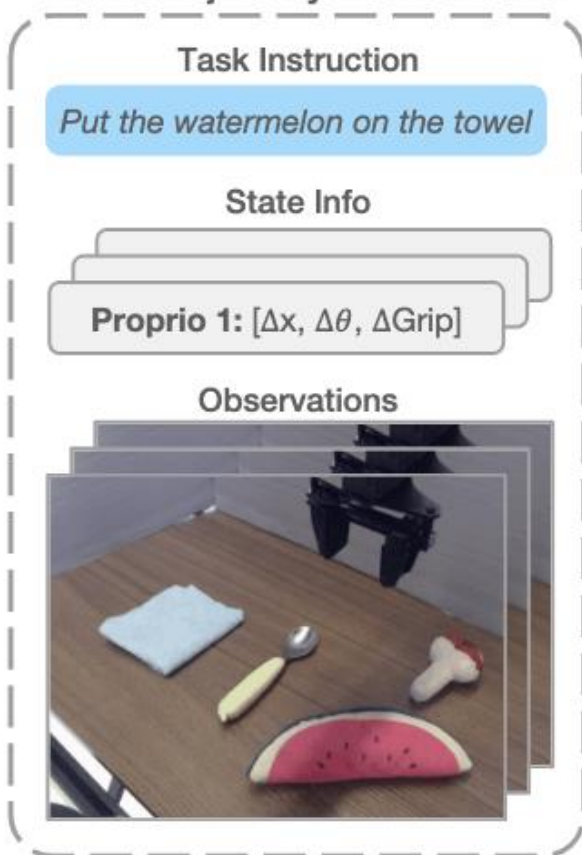


Embodied Chain-of-Thought

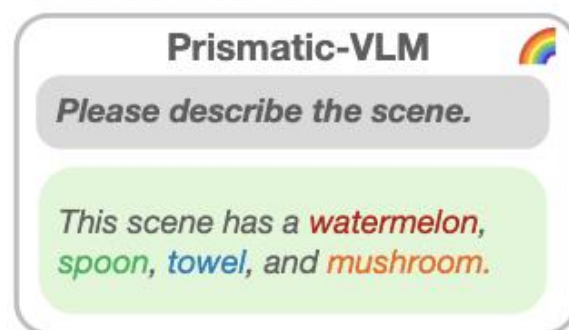


Prepare for Training Data

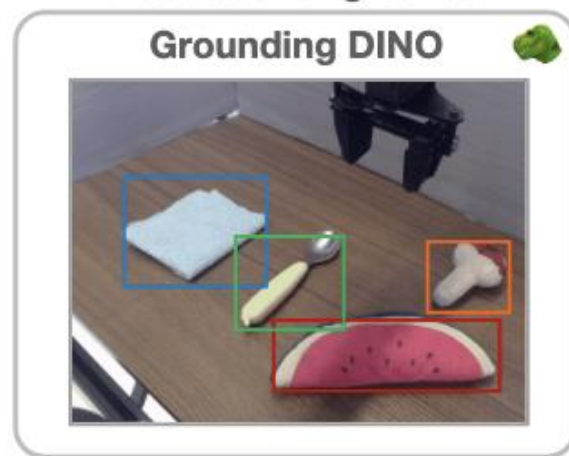
Robot Trajectory from Dataset



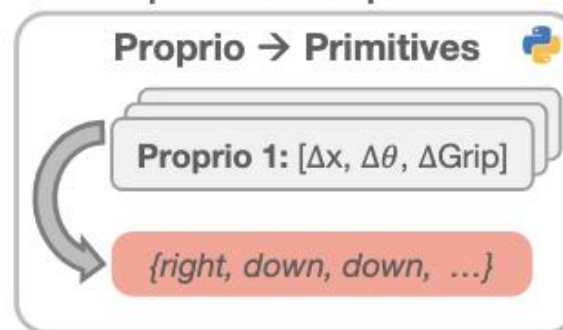
1. Describe scene



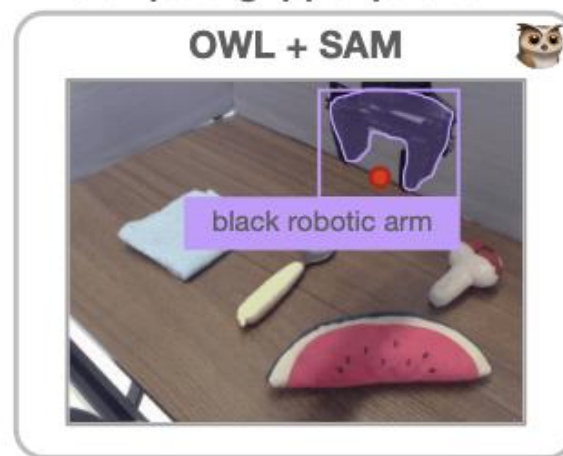
2. Extract bounding boxes



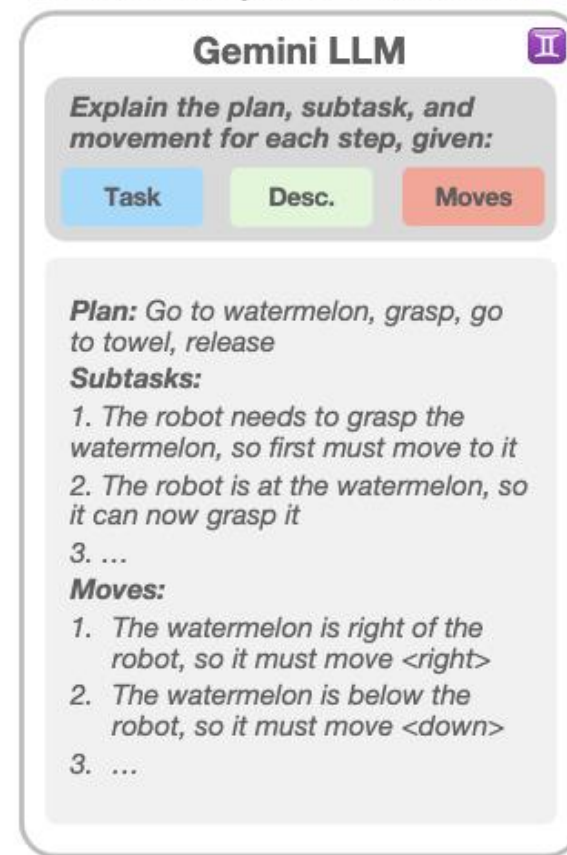
3. Compute motion primitives



4. Compute gripper position



5. Generate plans + subtasks



Embodied Chain-of-Thought

“Move the mushroom to the detergent”



TASK: Move the mushroom to the **detergent**

PLAN: Move to the mushroom, grasp it, move to the **detergent**, release the mushroom

SUBTASK: The mushroom is the closest object to the robot, and needs to be grasped first → Move to the mushroom

MOVE: The mushroom is to the right of the robot, so the robot needs to move right → Move right

“Put the leftmost object on the middle object”



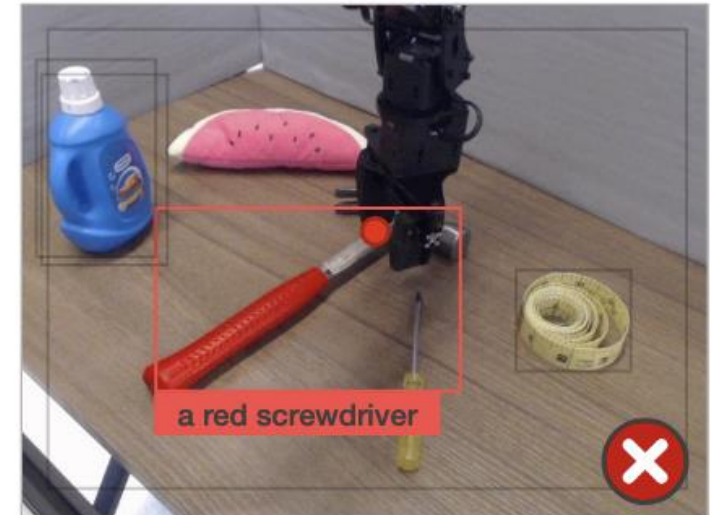
TASK: Put the **pink stuffed animal** on the **blue towel**

PLAN: Move to the **pink stuffed animal**. Pick up the **pink stuffed animal**. Move to the **blue towel**. Put the **pink stuffed animal** on the **blue towel**.

SUBTASK: The **pink stuffed animal** is held by the robotic arm and the blue towel is the middle object → Put the **pink stuffed animal** on the **blue towel**

MOVE: The **pink stuffed animal** is on the left of the **blue towel** and needs to be moved right → Move right

“Pick up the screwdriver”



TASK: Pick up the **screwdriver** and move it to the container

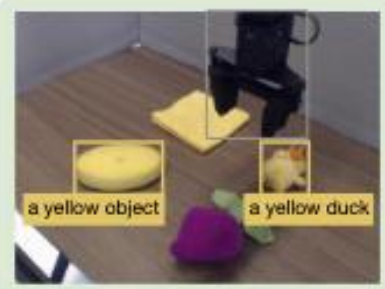
PLAN: Move to the **screwdriver**, pick it up, and move it to the container


SUBTASK: The **screwdriver** is in the top-left corner, so the robot needs to move forward and down to reach it → Move to the **screwdriver**

MOVE: The robot has reached the **screwdriver** → Stop

Intermediate Reasoning Enables Interactive Human-Policy Correction

User: Pick up any object that is **not** yellow.

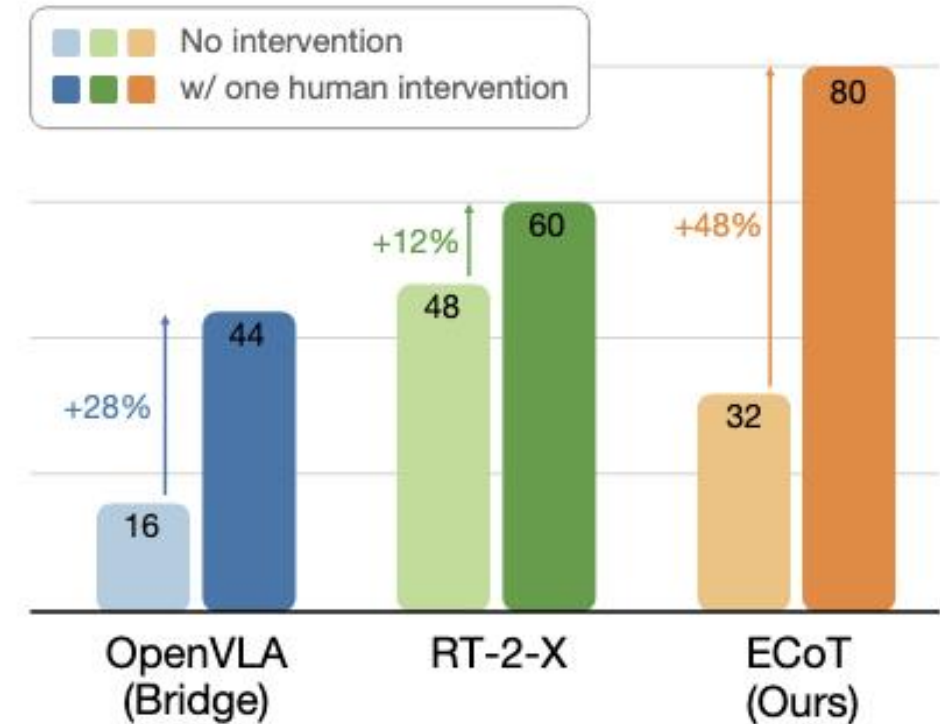



CoT OpenVLA: 
Next Task: find a **yellow** object
Next Move: move **left**

User: Wait! There is a purple object in front of the arm!

ChatGPT: Correct reasoning...

CoT OpenVLA:
Next Task: reach the **purple object**
Next Move: move **forward**



Results: Stronger Generalization to Out-of-distribution Data

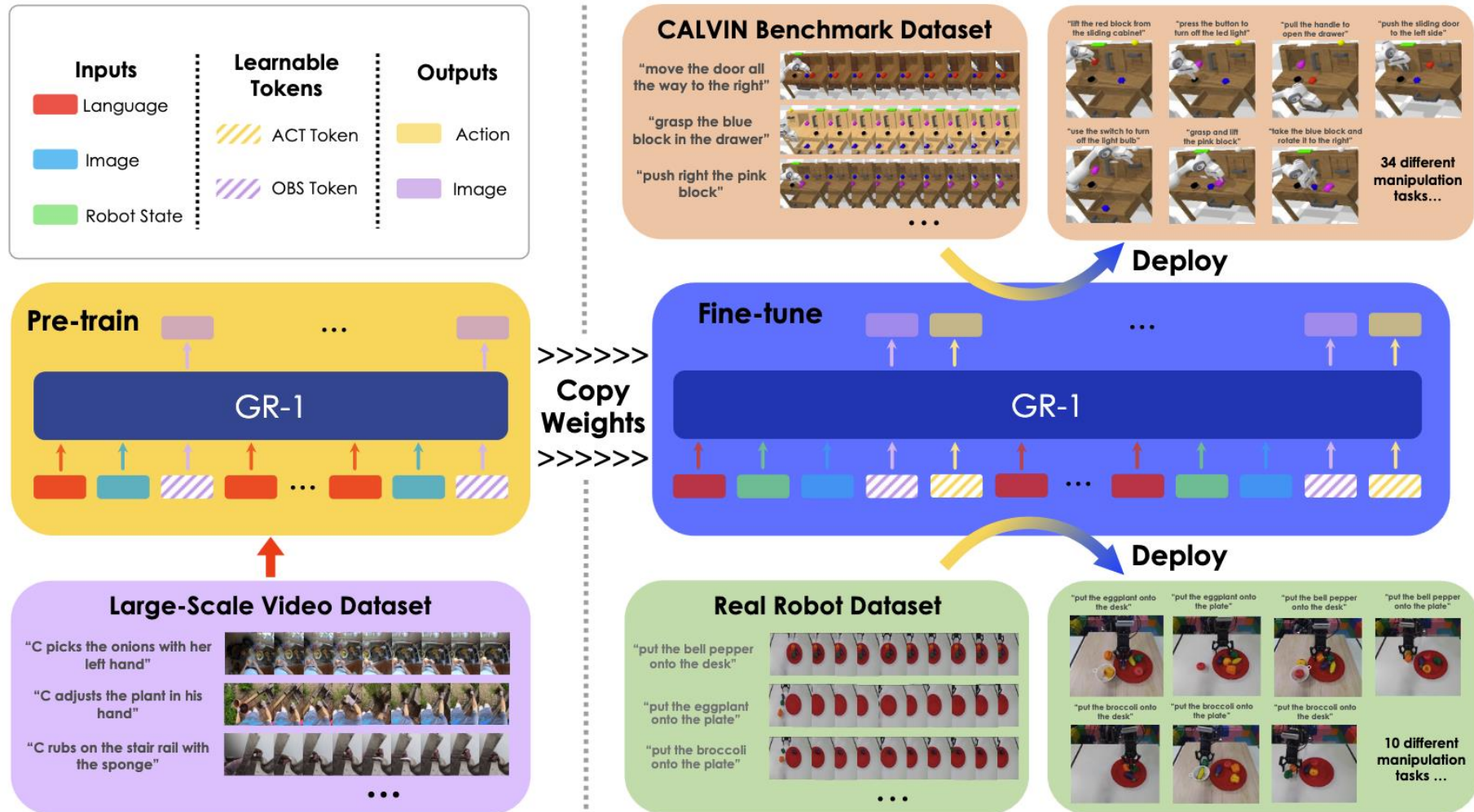
Type	Task	Algorithm (ID View)					Algorithm (OOD View)				
		Octo	OpenVLA (Bridge)	RT-2-X	Naive CoT	ECoT (Ours)	Octo	OpenVLA (Bridge)	RT-2-X	Naive CoT	ECoT (Ours)
ID	Put mushroom in pot	29%	88%	94%	71%	100%	35%	59%	76%	76%	65%
	Put spoon on towel	60%	90%	80%	60%	80%	20%	80%	80%	60%	80%
	Put carrot on plate	70%	80%	90%	90%	100%	40%	90%	90%	100%	90%
	Wipe [plate / pan] with towel	13%	50%	38%	38%	50%	0%	50%	0%	13%	63%
Spatial Relations	Put mushroom in [left / right / middle] container	0%	22%	17%	22%	33%	0%	17%	22%	55%	67%
	Put purple object in [left / right / middle] container	0%	28%	17%	50%	56%	0%	22%	11%	55%	39%
	Put [right / left] object on middle object	0%	13%	0%	50%	63%	0%	25%	25%	50%	63%
OOD Objects	Pick up [screwdriver / hammer / measuring tape / detergent / watermelon]	30%	20%	80%	50%	50%	30%	20%	80%	50%	50%
	Move mushroom to [measuring tape / detergent]	0%	10%	70%	20%	100%	10%	0%	90%	40%	90%
	Put mushroom in tall cup	0%	80%	0%	70%	30%	10%	20%	0%	20%	30%
	Place watermelon on towel	20%	30%	60%	60%	70%	50%	10%	90%	30%	40%
OOD Instructions	Pick up any object that is not [yellow / a duck / a sponge / a towel]	50%	33%	58%	50%	42%	17%	17%	67%	25%	67%
	Put the edible object in the bowl	13%	25%	13%	25%	88%	0%	13%	25%	25%	100%
	Put the object used for [eating / drinking] on towel	25%	38%	38%	38%	75%	13%	0%	25%	38%	75%
Aggregate		21% ± 3.3%	44% ± 3.9%	47 ± 4.0%	48 ± 4.0%	66% ± 3.8%	16% ± 2.9%	30% ± 3.6%	48 ± 4.0%	48% ± 4.0%	64 ± 3.9%

Can We Incorporate Visually Grounded Reasoning in VLAs?

Videos Elaborate the Execution of a Task



World Action Models: Reasoning with Videos in VLAs

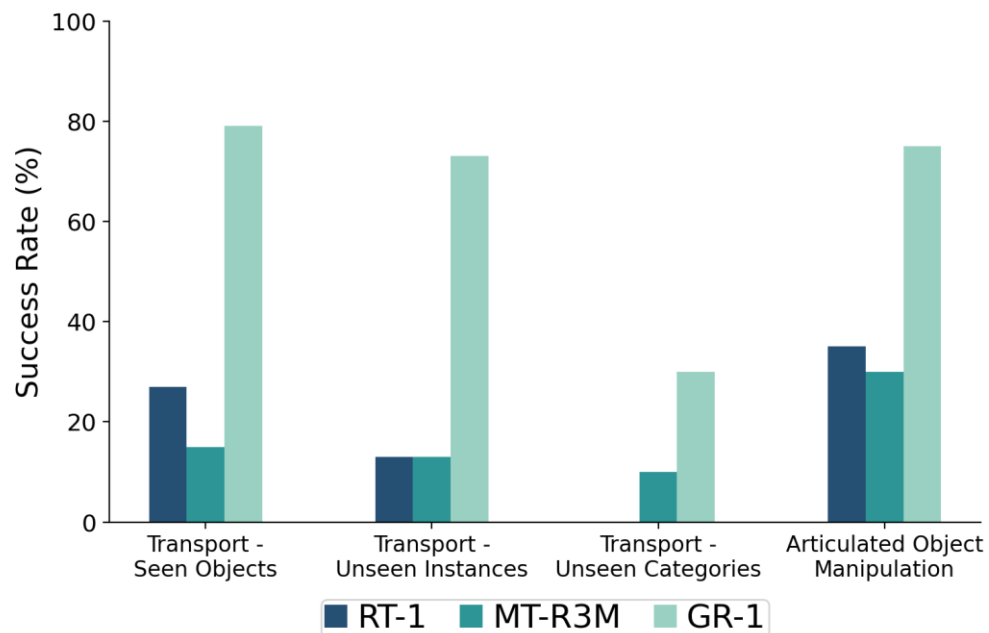


Advantage: Pre-training on Action-free Videos

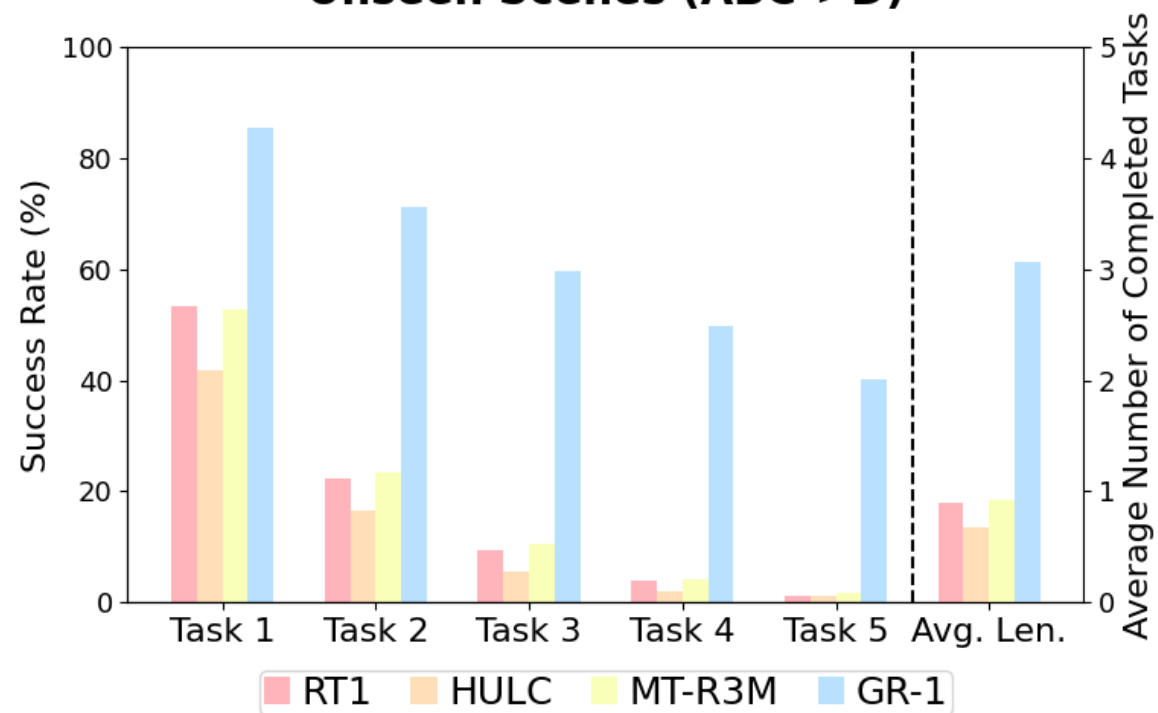


Results

Results on Real Robot Experiments

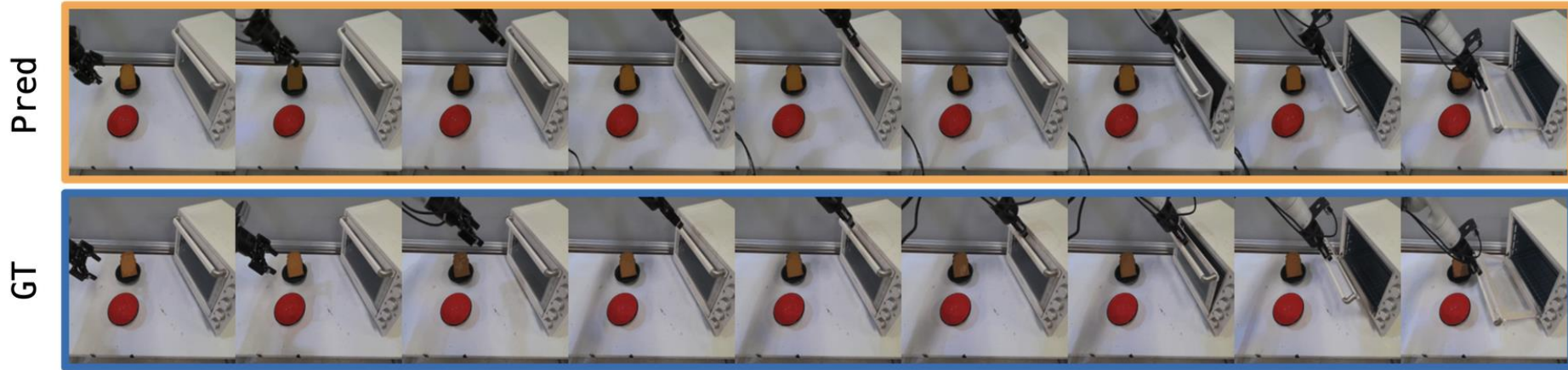


Zero-Shot Generalization to Unseen Scenes (ABC->D)

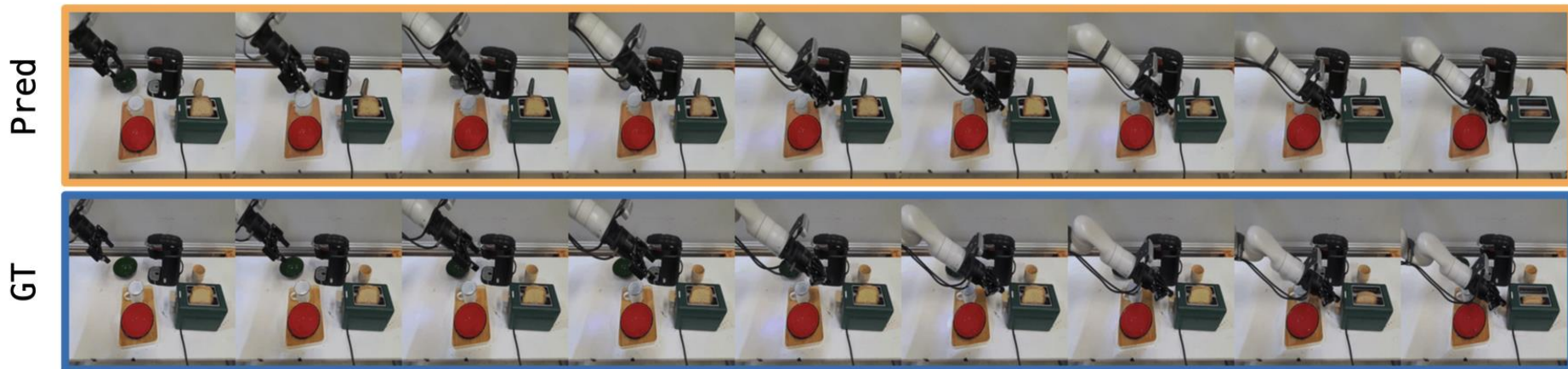


Results

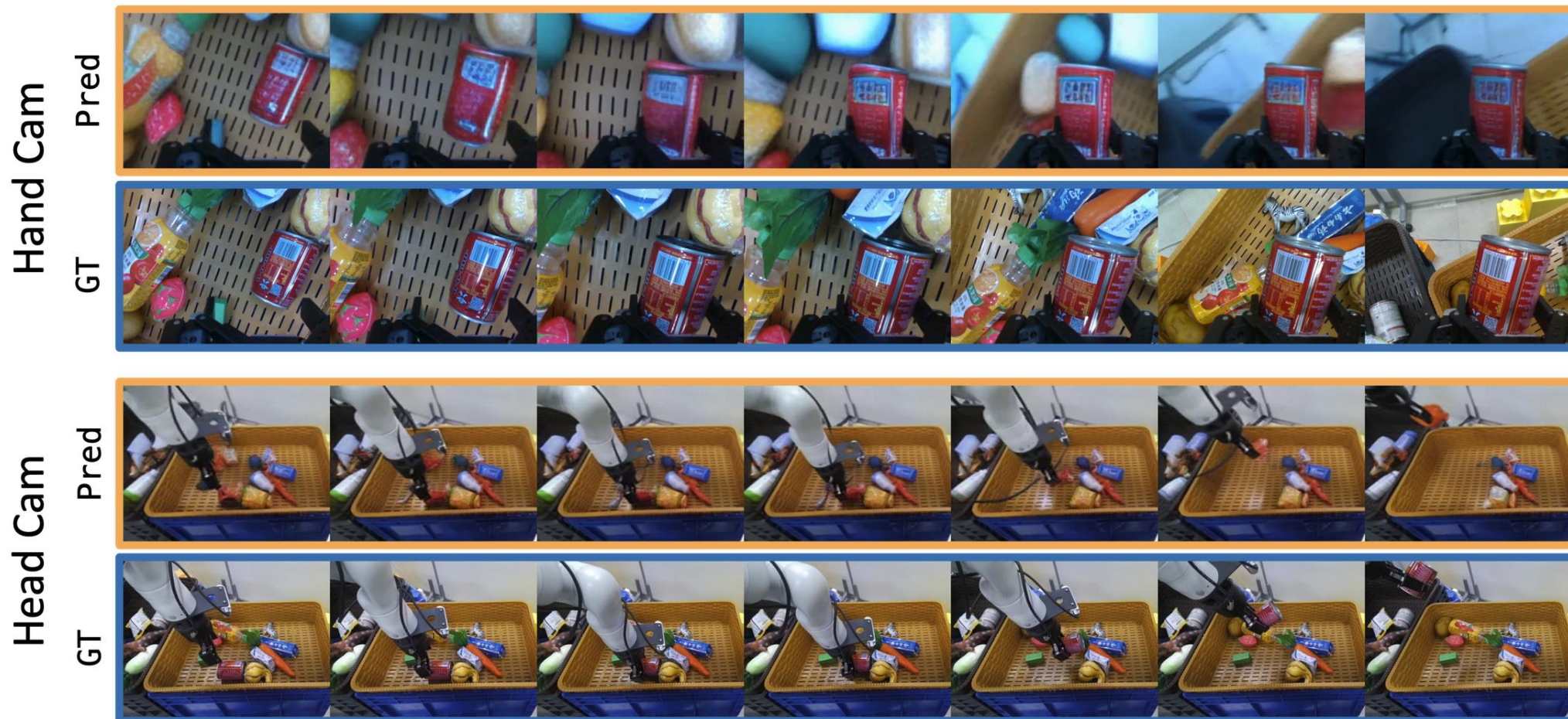
open the oven



press the toaster switch



Results



Problem: Videos Make Physical Interpretation
Indirect and Ambiguous

Can We Incorporate Other Kind of Visually Grounded Reasoning in VLAs?

What Do We Learn from Watching a Demonstration?



Human video

What Do We Learn from Watching a Demonstration?



Where to interact with the object

Human video

How to interact with the object

Frame without human



Contact points

Frame without human



Wrist trajectory

Visually Grounded Reasoning with 2D Waypoints

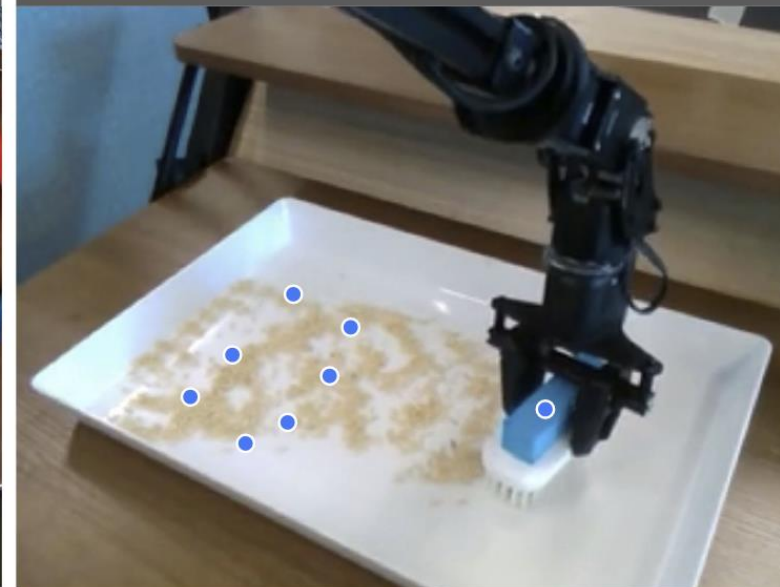
Point to the left hand and the handle of the blue screwdriver, and a trajectory of 6 points connecting them.



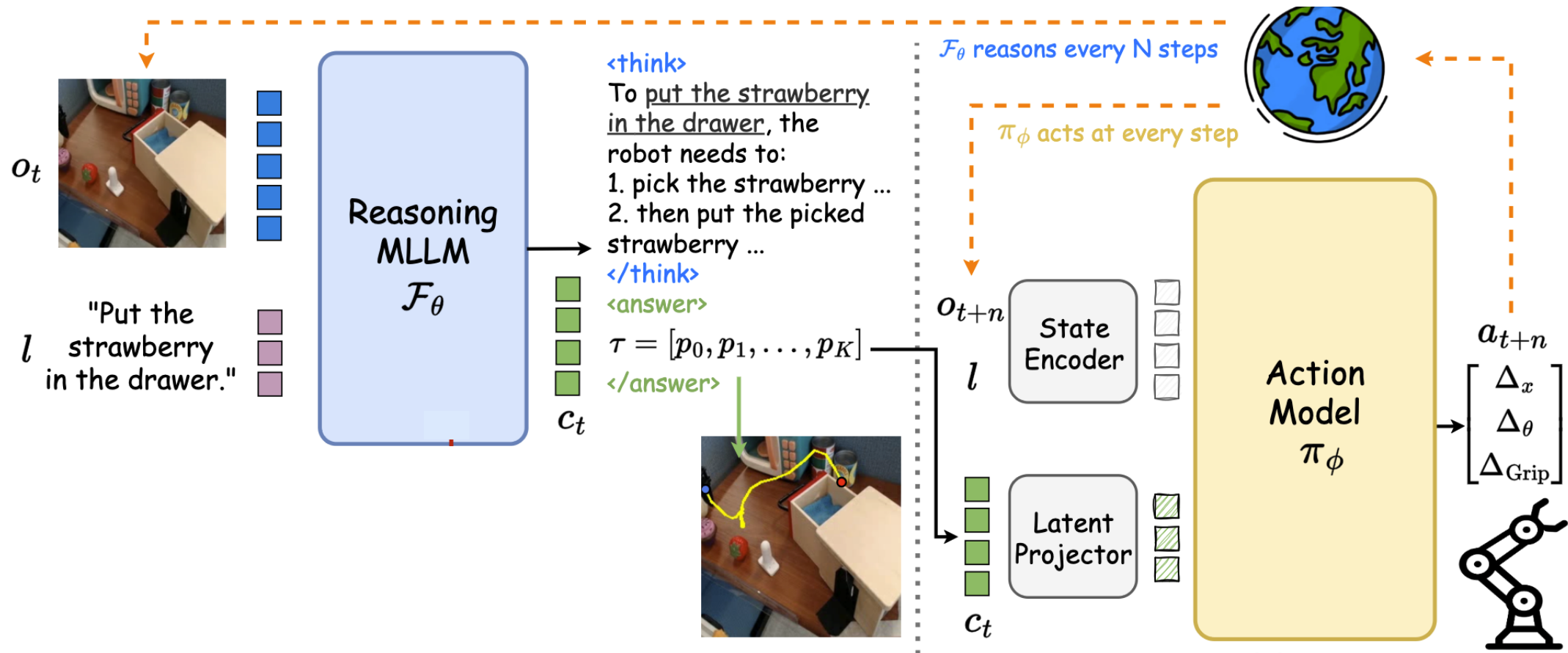
Point to the right hand and the handles of the scissor ... and a trajectory of 5 points from the right hand to the handles of the scissor



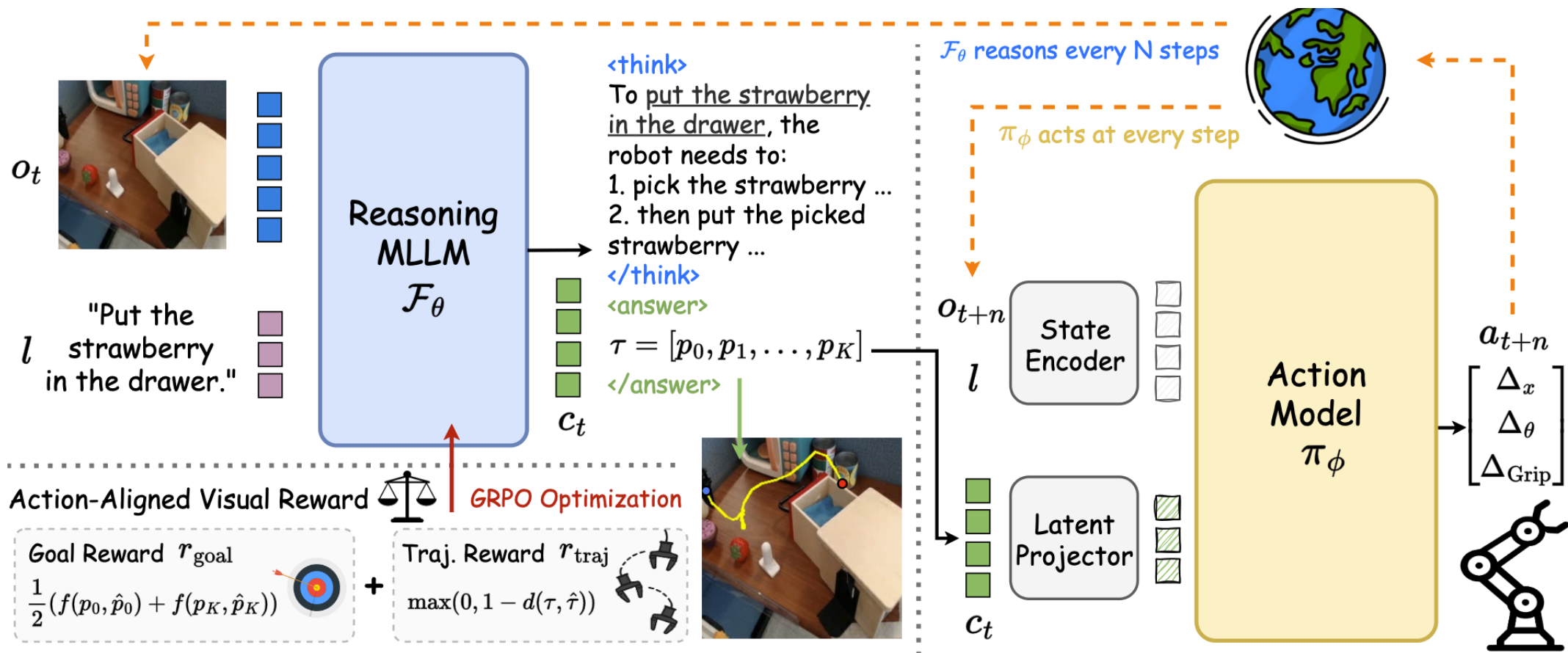
Point to the blue brush and a list of points covering the region of particles



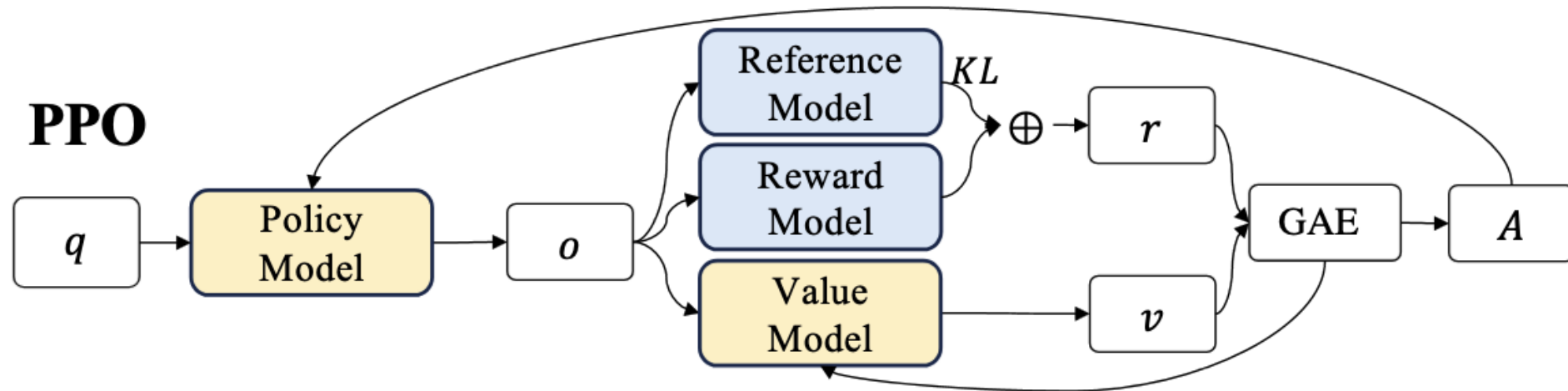
Idea: Reasoning with 2D Waypoints in VLAs



How to Train VLMs for Reasoning?



Training with RL: PPO

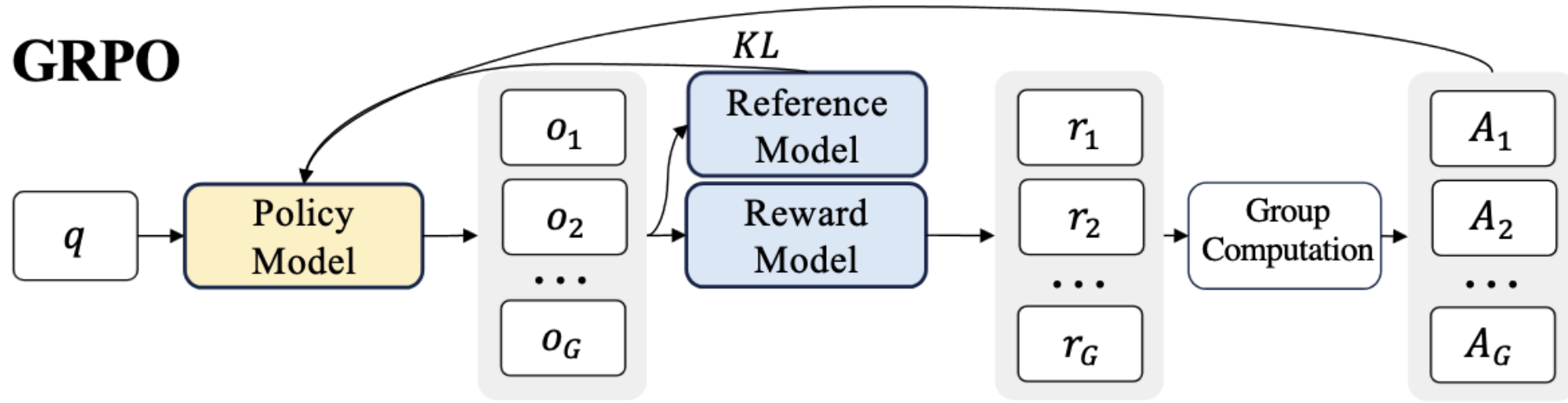


Measure how much better the current action is, compared to the average "learned value" ←

$$\mathcal{J}_{PPO}(\theta) = \mathbb{E}[q \sim P(Q), o \sim \pi_{\theta_{old}}(O|q)] \frac{1}{|o|} \sum_{t=1}^{|o|} \min \left[\frac{\pi_{\theta}(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})} A_t, \text{clip} \left(\frac{\pi_{\theta}(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})}, 1 - \epsilon, 1 + \epsilon \right) A_t \right]$$

Limit the deviation of updated policy from the original "reference policy" ←

Training with RL: GRPO



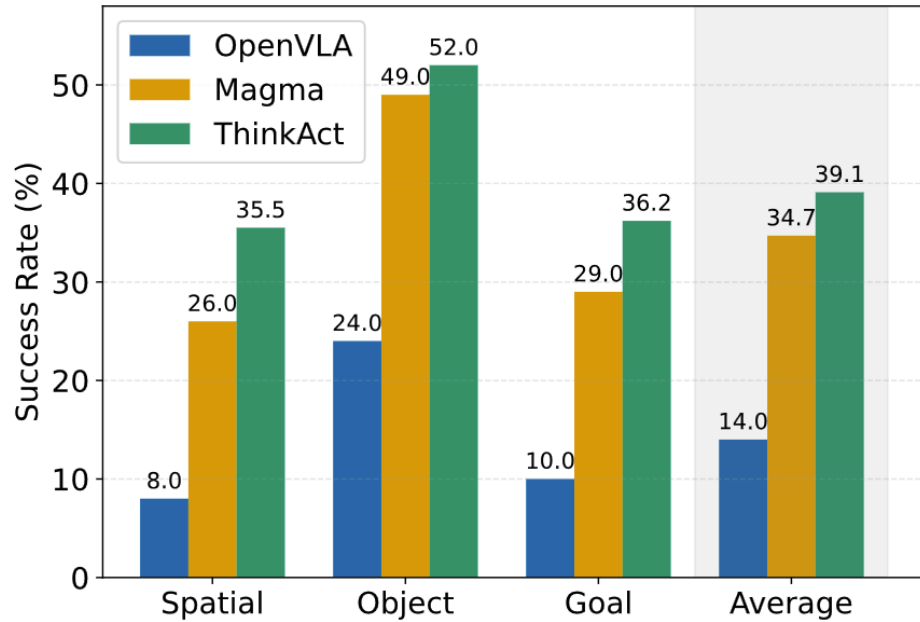
Measure how much better the current action is, compared to the average "sampled value" ←

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^G \sim \pi_{\theta_{old}}(O|q)]$$

$$\frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left\{ \min \left[\frac{\pi_{\theta}(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})} \hat{A}_{i,t}, \text{clip} \left(\frac{\pi_{\theta}(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})}, 1 - \epsilon, 1 + \epsilon \right) \hat{A}_{i,t} \right] - \beta \mathbb{D}_{KL} [\pi_{\theta} || \pi_{ref}] \right\}$$

Limit the deviation of updated policy from the original "reference policy" ←

Results



Dataset	Split / Metric	GPT-4V	LLaVA-Video	InternVL2.5	InternVL3	NVILA	Qwen2.5-VL	Qwen2.5-VL*	Magma	ThinkAct (Ours)
EgoPlan-Bench2	Daily life	36.7	38.0	36.2	38.5	35.8	31.4	47.9	32.1	50.1
	Work	27.7	29.9	28.7	32.9	28.7	26.7	46.3	25.7	49.8
	Recreation	33.9	39.0	34.4	36.1	37.2	29.5	44.3	34.4	44.8
	Hobbies	32.5	37.4	35.4	37.2	35.4	28.6	44.2	29.3	45.2
	Overall	32.6	35.5	33.5	36.2	33.7	29.1	45.7	29.8	48.2
RoboVQA	BLEU-1	32.2	35.4	40.5	44.3	42.7	47.8	65.3	38.6	69.1
	BLEU-2	26.5	32.1	33.3	36.5	39.7	41.2	57.3	31.5	61.8
	BLEU-3	24.7	30.0	29.6	31.6	37.6	36.2	52.2	28.1	56.0
	BLEU-4	23.9	29.0	27.5	28.9	36.1	33.7	48.0	26.7	52.4
	Overall	26.8	31.6	32.7	35.3	39.0	39.7	55.7	31.2	59.8
OpenEQA	Obj. State	63.2	69.1	70.2	68.9	66.1	63.2	62.4	59.9	70.0
	Obj. Recog.	43.4	42.6	47.2	49.1	49.5	46.2	45.2	43.8	47.2
	Func. Reason.	57.4	50.3	56.2	54.6	51.0	51.2	52.3	50.0	53.2
	Spatial	33.6	46.2	44.1	43.3	43.1	41.2	42.8	39.3	47.6
	Attri. Recog.	57.2	64.1	64.9	74.4	69.3	63.0	65.0	58.3	71.1
	World Know.	50.7	60.5	56.5	53.1	59.4	54.3	54.2	53.3	58.6
	Obj. Loc.	42.0	38.2	41.9	45.0	39.9	36.5	41.9	38.9	45.9
	Overall	49.6	53.0	54.4	55.5	54.0	50.8	52.0	49.1	56.2

Results

Simpler-Google

"Move the 7up can near the apple"

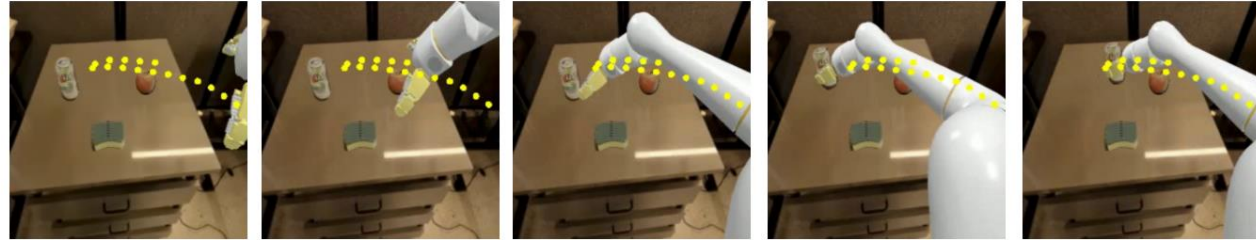
<think>

The image shows a 7up can, a sponge, and an apple on a table. ...

First, I need to **locate the 7up can and the apple**. ... Now, I need to **move the 7up can**. ...

I can now **place it near the apple**. ... The can is now right next to the apple. ...

</think>



Simpler-Bridge

"Put eggplant into yellow basket"

<think>

First, the eggplant is in the sink, and the basket is in the top left corner. ...

The gripper will **move toward the eggplant**, **lift it**, and **carry it to the basket**. ...

Okay, I'm ready to give the final trajectory: move to eggplant, lift it, and place it in basket.

</think>



LIBERO-Long

"Pick up the book and place it in the back compartm."

<think>

Let's **break down the task step by step**:

First, identify the book on the left side.

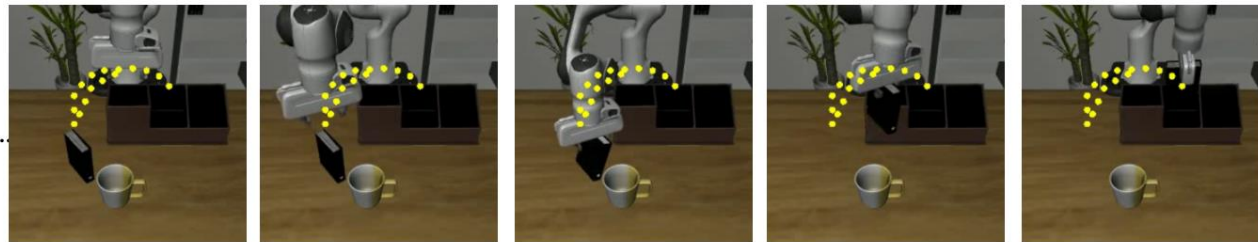
The back compartment is on the right.

Use the robot's arm near the book to **pick it up** ...

Move the book smoothly **from left to right** ...

Place it in the compartment. ...

</think>



Can We Incorporate Physically Grounded Reasoning in VLAs?

Yes! This Rewinds Lecture 4

- 2D Motion Representations
- 3D Motion Representations
 - 4D Volume Rendering with NERF
 - Explicit Motion Field with NERF
 - 4D Volume Rendering with 3DGS
 - Explicit Motion Field with 3DGS
 - Implicit Motion Field with 3DGS
 - 4D Foundation Models
 - Multi-view Video Generation (we'll cover in the next lecture)

A Quick Summary of This Class

Covered in Robot Perception
and Learning

Covered in Embodied Vision!

