Open-World Robotic Control

Instructor: Kuan Fang

2024/11/12

The Plan for Today

- Task Decomposition for Open-World Robotic Control
- API Calling for Open-World Robotic Control
- Affordance Representations for Open-World Robotic Control

The Plan for Today

- **Task Decomposition for Open-World Robotic Control**
- API Calling for Open-World Robotic Control
- Affordance Representations for Open-World Robotic Control

Markov Decision Process

A Markov Decision Process (MDP) is defined by a tuple $\mathcal{M} =< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >$.

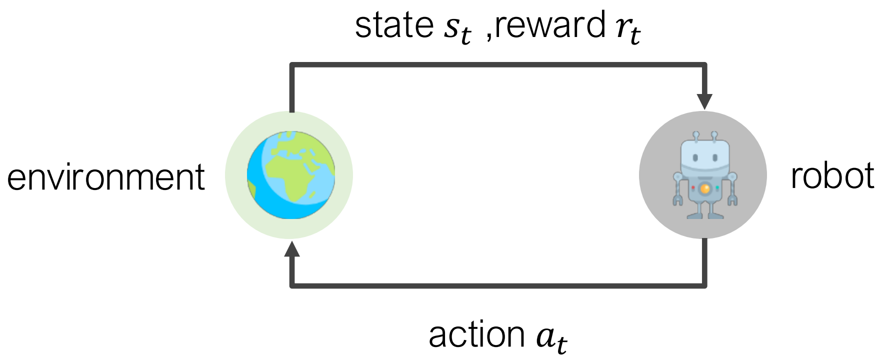
$\mathcal{S}$: state space ($s_t \in \mathcal{S}$)

$\mathcal{A}$: action space ($a_t \in \mathcal{A}$)

$\mathcal{P}$: transition probability $s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)$

$\mathcal{R}$: reward function $r_t \sim \mathcal{R}(s_t, a_t, s_{t+1})$

γ: a discount factor $\gamma \in [0, 1]$



A policy π maps state: $\mathcal{S} \rightarrow \mathcal{A}$

```
for i in range(1000):
    action = np.random.randn(env.robots[0].dof) # sample random action
    obs, reward, done, info = env.step(action)  # take action in the environment
    env.render()  # render on display
```

Goal-Conditioned MDP

A Goal-Conditioned Markov Decision Process is defined by a tuple

$\mathcal{M} =< \mathcal{S}, \mathcal{C}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >.$

$\mathcal{S}$: state space ($s_t \in \mathcal{S}$)

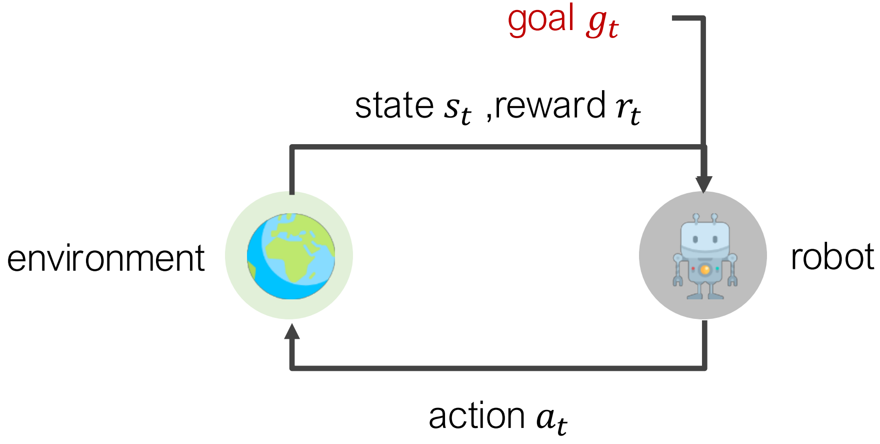
$\mathcal{C}$: goal space ($g_t \in \mathcal{C} \subset \mathcal{S}$)

$\mathcal{A}$: action space ($a_t \in \mathcal{A}$)

$\mathcal{P}$: transition probability $s_{t+1} \sim \mathcal{P}(\cdot | s_t, a_t)$

$\mathcal{R}$: reward function $r_t = -\mathbf{1}[s_t == g_t]$

γ: a discount factor $\gamma \in [0, 1]$



Language-Conditioned MDP

A Goal-Conditioned Markov Decision Process is defined by a tuple

$\mathcal{M} =< \mathcal{S}, \mathcal{C}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >.$

$\mathcal{S}$: state space ($s_t \in \mathcal{S}$)

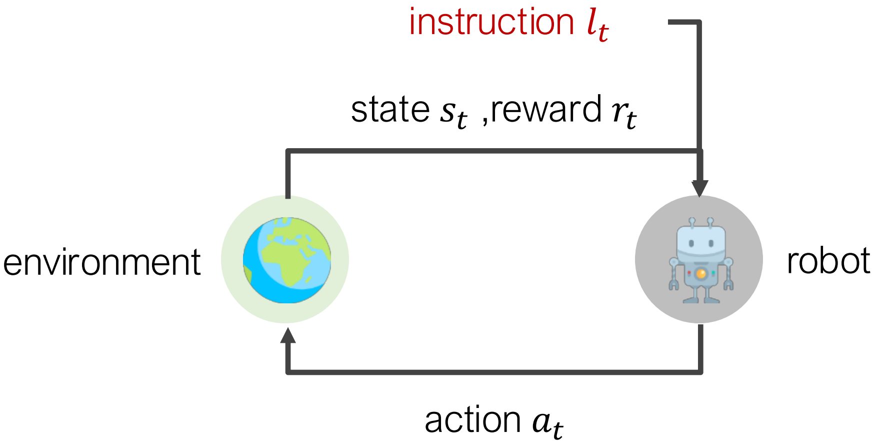
$\mathcal{C}$: instruction space ($l_t \in \mathcal{C}$)

$\mathcal{A}$: action space ($a_t \in \mathcal{A}$)

$\mathcal{P}$: transition probability $s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)$

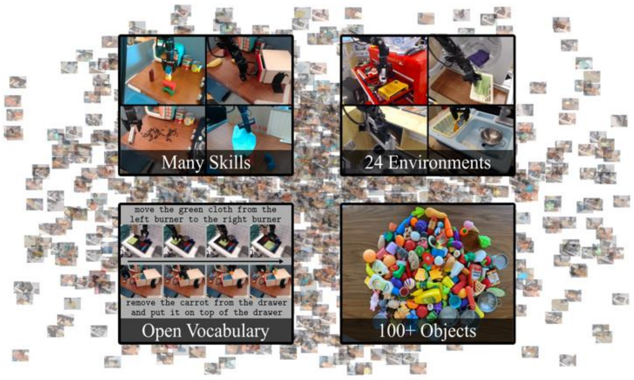
$\mathcal{R}$: reward function $r_t = ?$

γ: a discount factor $\gamma \in [0, 1]$



Learning to Follow Instructions

sweep the skittles into the bin after putting the mushroom in the container



demos with language labels

pre-train

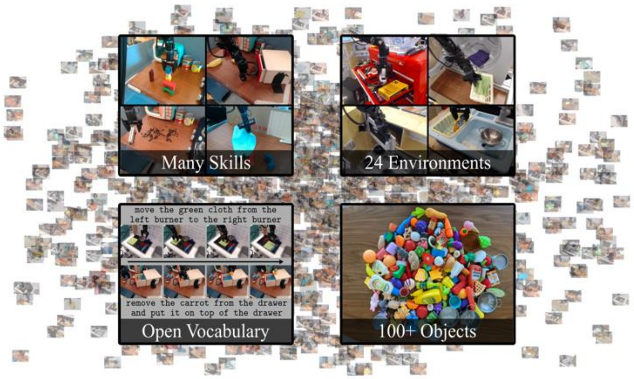
instruction

$$\pi(a|s, l; \theta)$$

generalize

new task

Language-Conditioned Imitation Learning



demos with language labels

pre-train $\longrightarrow$ $\pi(a|s, l; \theta)$

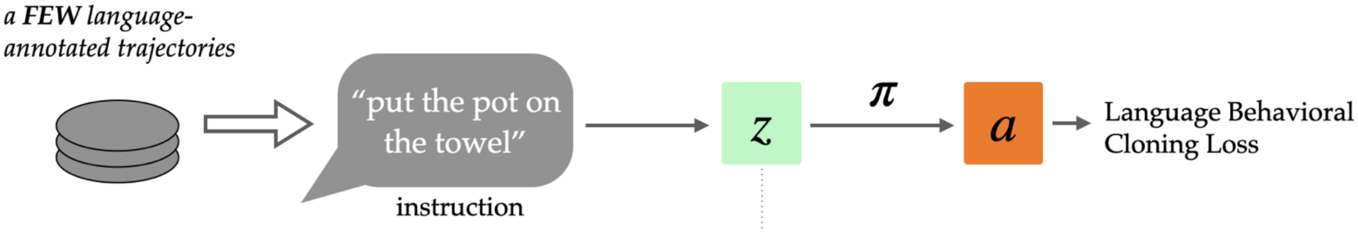
instruction

language labels are expensive to get

Language-Conditioned Behavior Cloning: Given a training dataset of (expert) behaviors $D = \{(s_i, a_i, l_i)\}_{i=1}^{N}$, train the policy $\pi_\theta(a_t|s_t, l_t)$ to imitate the behaviors:

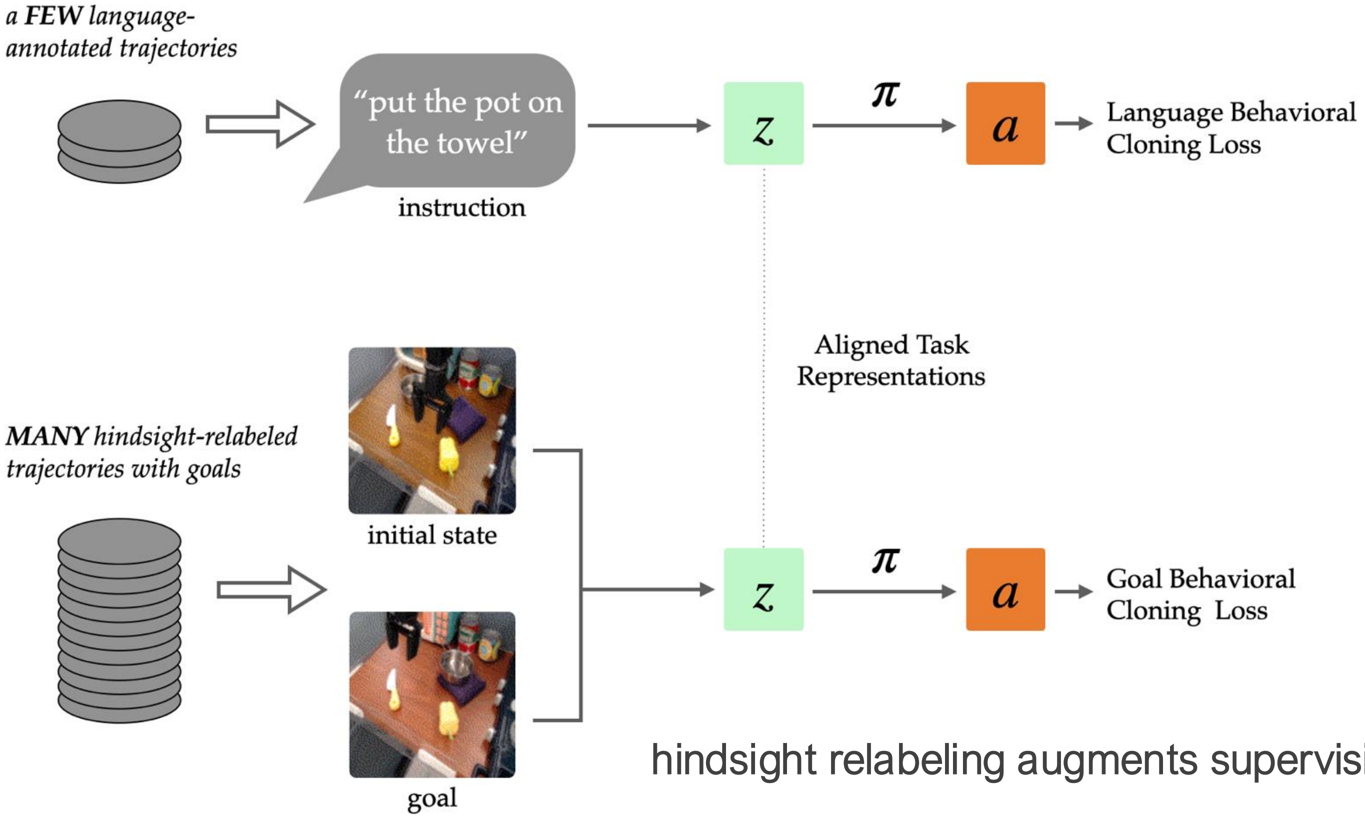
$$\theta^* = \arg\max_{\theta} \Sigma_D \log \pi_\theta(a_t|s_t, l_t)$$

Integrated Language-Conditioned and Goal-Conditioned BC



Myers et al. Goal Representations for Instruction Following: A Semi-Supervised Language Interface to Control. CoRL 2023

Integrated Language-Conditioned and Goal-Conditioned BC

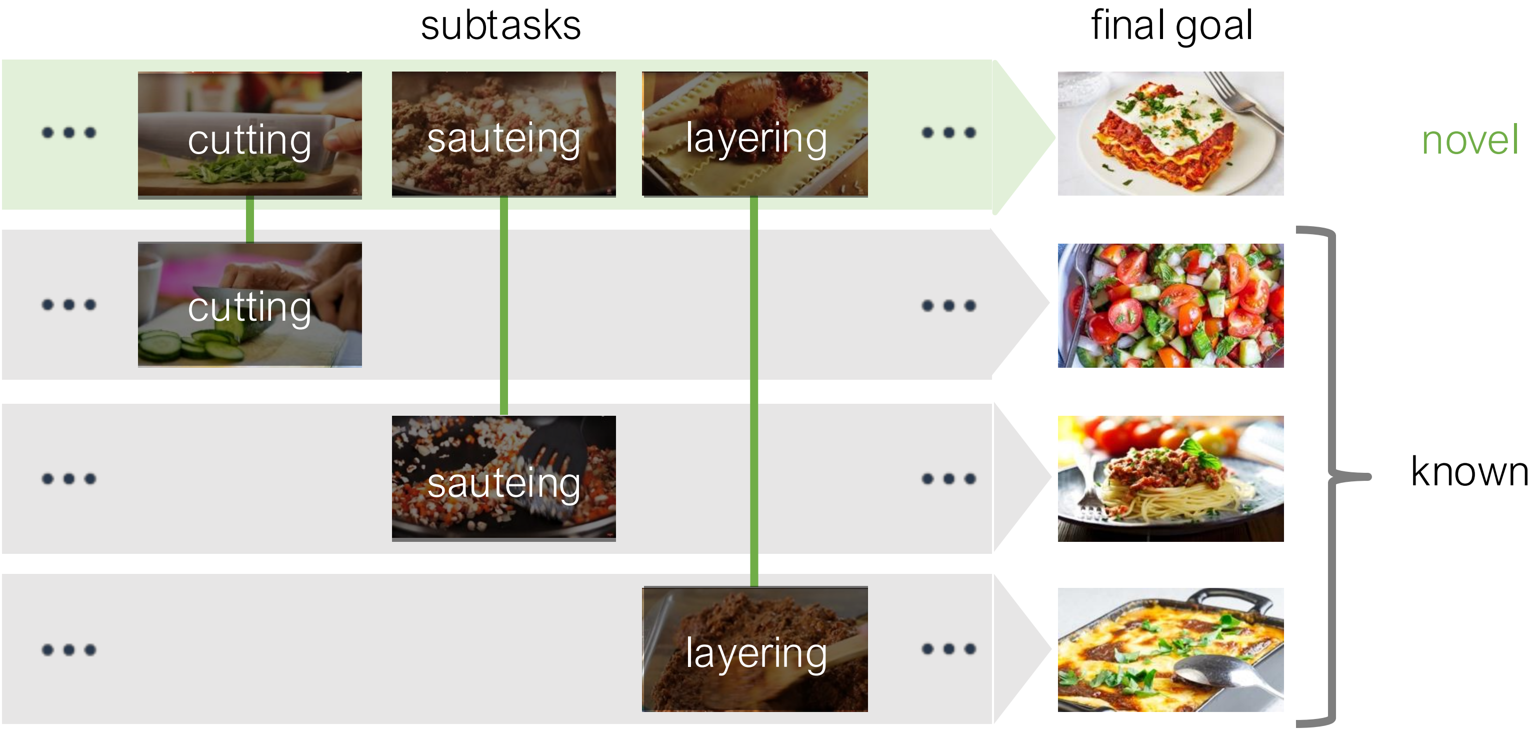


hindsight relabeling augments supervisions

Myers et al. Goal Representations for Instruction Following: A Semi-Supervised Language Interface to Control. CoRL 2023

Task Decomposition

Task decomposition enables robots to reuse and repurpose known skills.



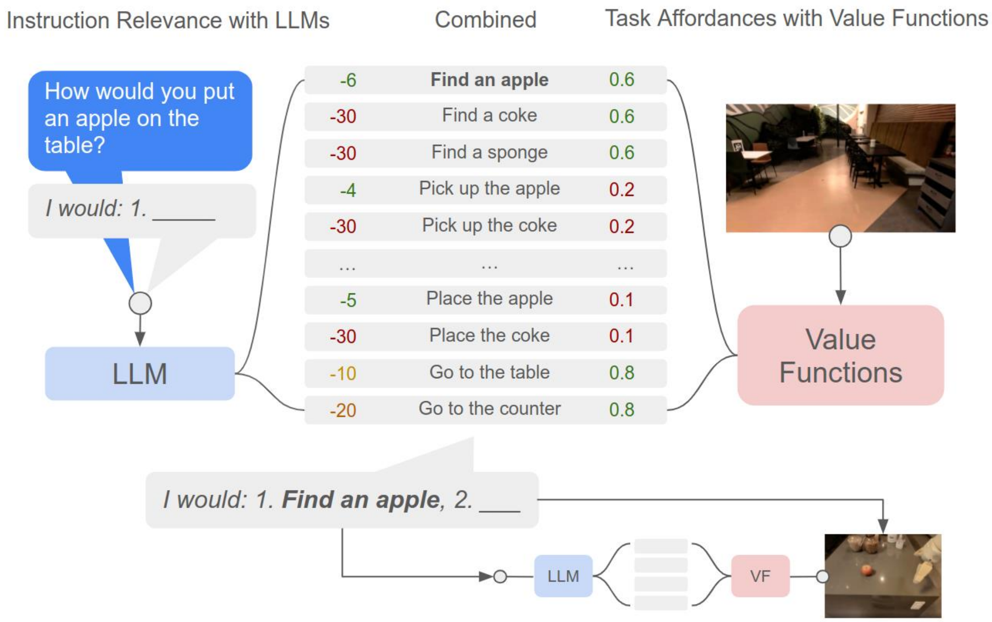
SayCan: “Do As I Can, Not As I Say”

Task decomposition needs to be grounded in the robot’s capabilities and the observed environment.

Ahn et al. CoRL 2023

SayCan: “Do As I Can, Not As I Say”

Combine probabilities from a language model with the probabilities from a value to select the skill (pre-trained or pre-defined) to perform.

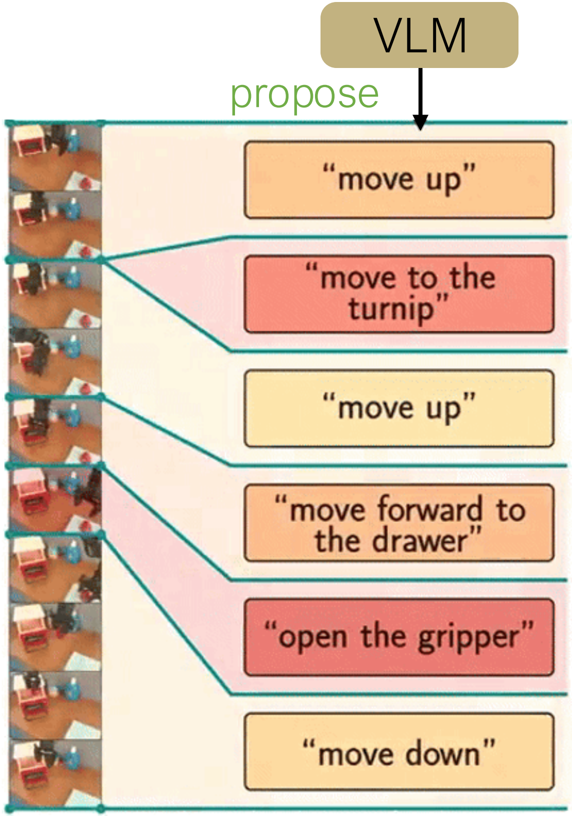


Ahn et al. CoRL 2023

SayCan: “Do As I Can, Not As I Say”

Ahn et al. CoRL 2023

Policy Adaptation via Language Optimization



$$\hat{a}_t \sim \pi(\cdot | s_t, c; \theta)$$

c_1	$\hat{a}_1$
c_2	$\hat{a}_2$
c_3	$\hat{a}_3$
c_4	$\hat{a}_4$
c_5	$\hat{a}_5$
c_6	$\hat{a}_6$

Myers*, Zheng*, Mees, Levine†, Fang†. Policy Adaptation via Language Optimization: Decomposing Tasks for Few-Shot Imitation. CoRL 2024

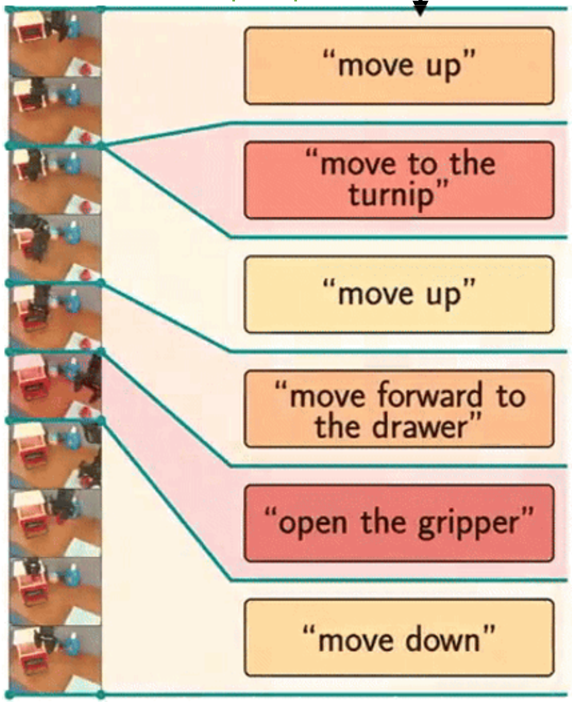
Policy Adaptation via Language Optimization

VLM

propose

$$\hat{a}_t \sim \pi(\cdot | s_t, c; \theta)$$

freeze



c_1 $\hat{a}_1$

c_2 $\hat{a}_2$

c_3 $\hat{a}_3$

c_4 $\hat{a}_4$

c_5 $\hat{a}_5$

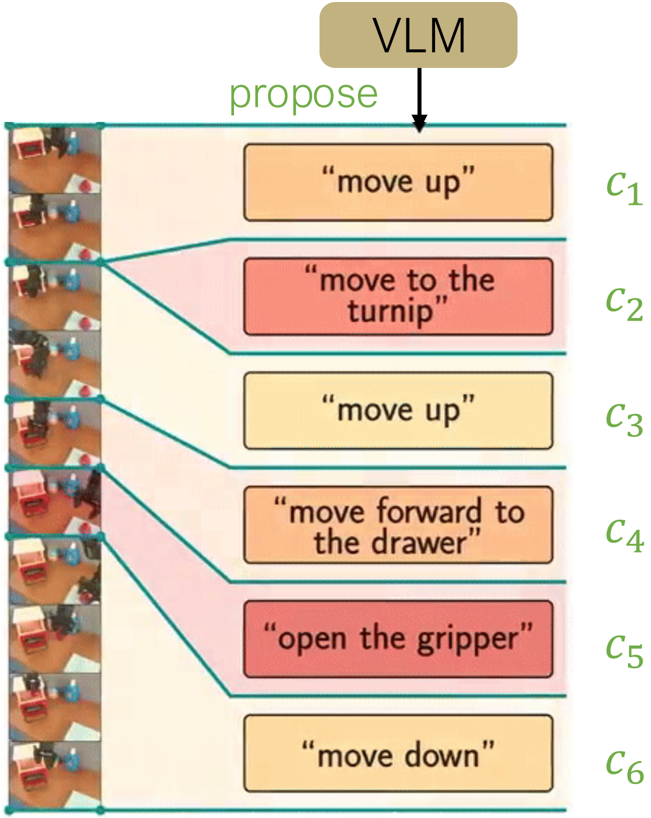
c_6 $\hat{a}_6$

Optimize instruction sequences using behavior cloning loss

$$c^* = \arg\min_c \sum_t \|\hat{a}_t - a_t\|^2$$

Myers*, Zheng*, Mees, Levine†, **Fang**†. Policy Adaptation via Language Optimization: Decomposing Tasks for Few-Shot Imitation. CoRL 2024

Policy Adaptation via Language Optimization



$$\hat{a}_t \sim \pi(\cdot | s_t, c; \theta)$$

freeze

$\hat{a}_1$

$\hat{a}_2$

$\hat{a}_3$

$\hat{a}_4$

$\hat{a}_5$

$\hat{a}_6$

Optimize instruction sequences using behavior cloning loss

$$c^*, u^* = \arg\min_{c,u} \sum_t \|\hat{a}_t - a_t\|^2$$

Jointly optimize the temporal segmentation

similar to prompt tuning in NLP

Myers*, Zheng*, Mees, Levine†, **Fang**†. Policy Adaptation via Language Optimization: Decomposing Tasks for Few-Shot Imitation. CoRL 2024

Given only 5 demos, PALO is able to robustly solve unseen, temporally extended tasks.

pour the contents of the scoop into the bowl

sweep the skittles into the bin after putting the mushroom in the container

put the beet toy/purple thing into the drawer

pry out the pot in the drawer using the ladle

PALO ✓

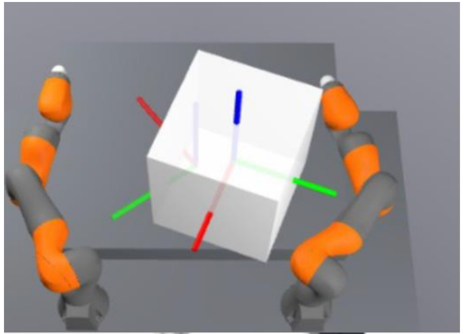
Policy Fine-Tuning ✗

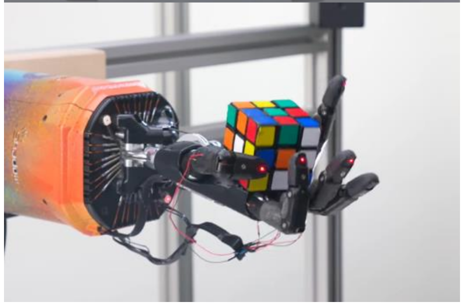
Myers*, Zheng*, Mees, Levine†, **Fang**†. Policy Adaptation via Language Optimization: Decomposing Tasks for Few-Shot Imitation. CoRL 2024

The Plan for Today

- Task Decomposition for Open-World Robotic Control
- **API Calls for Open-World Robotic Control**
- Affordance Representations for Open-World Robotic Control

Tools for physical understanding





physics simulator

motion planner

reinforcement learning

imitation learning

……

Lack Semantic Understanding

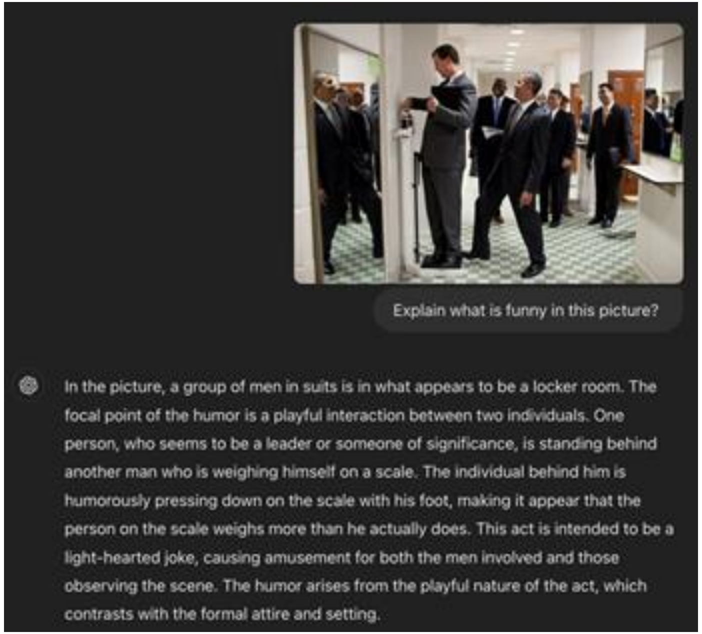
Akkaya et al. 2019, Suh et al. 2022

Tools for semantic understanding

Lack Physical Understanding

large language models

vision language models



Kirillov et al. 2023; Achiam et al. 2023

API Calls by LLMs

LLMs can solve new tasks, but struggle with basic functionality, such as arithmetic.

Goal: Enable LLMs to call third-party APIs.

The New England Journal of Medicine is a registered trademark of **[QA("Who is the publisher of The New England Journal of Medicine?") → Massachusetts Medical Society]** the MMS.

Out of 1400 participants, 400 (or **[Calculator(400 / 1400) → 0.29]** 29%) passed the test.

The name derives from "la tortuga", the Spanish word for **[MT("tortuga") → turtle]** turtle.

The Brown Act is California's law **[WikiSearch("Brown Act") → The Ralph M. Brown Act is an act of the California State Legislature that guarantees the public's right to attend and participate in meetings of local legislative bodies.]** that requires legislative bodies, like city councils, to hold their meetings open to the public.

Improve performance with in-context examples

Your task is to add calls to a Question Answering API to a piece of text. The questions should help you get information required to complete the text. You can call the API by writing "[QA(question)]" where "question" is the question you want to ask. Here are some examples of API calls:

Input: Joe Biden was born in Scranton, Pennsylvania.

Output: Joe Biden was born in **[QA("Where was Joe Biden born?")]** Scranton, **[QA("In which state is Scranton?")]** Pennsylvania.

Input: Coca-Cola, or Coke, is a carbonated soft drink manufactured by the Coca-Cola Company.

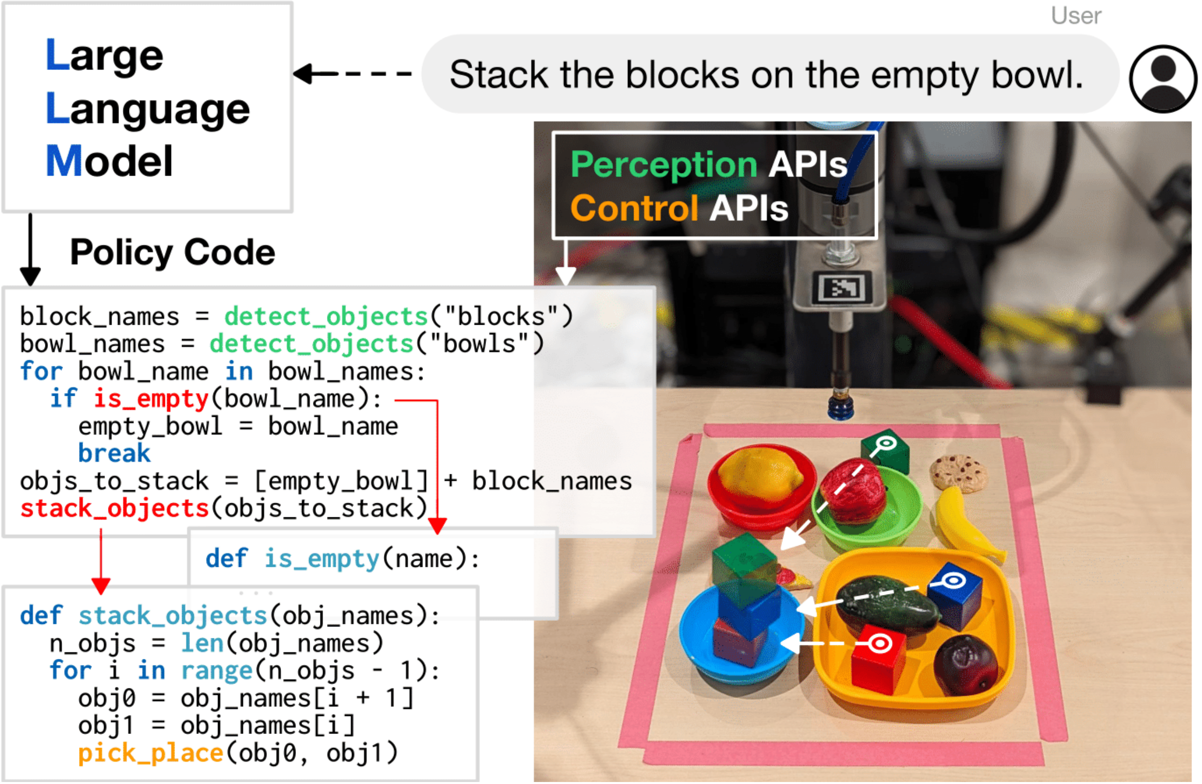
Output: Coca-Cola, or **[QA("What other name is Coca-Cola known by?")]** Coke, is a carbonated soft drink manufactured by **[QA("Who manufactures Coca-Cola?")]** the Coca-Cola Company.

Input: $\mathbf{x}$

Output:

Schick et al. 2023.

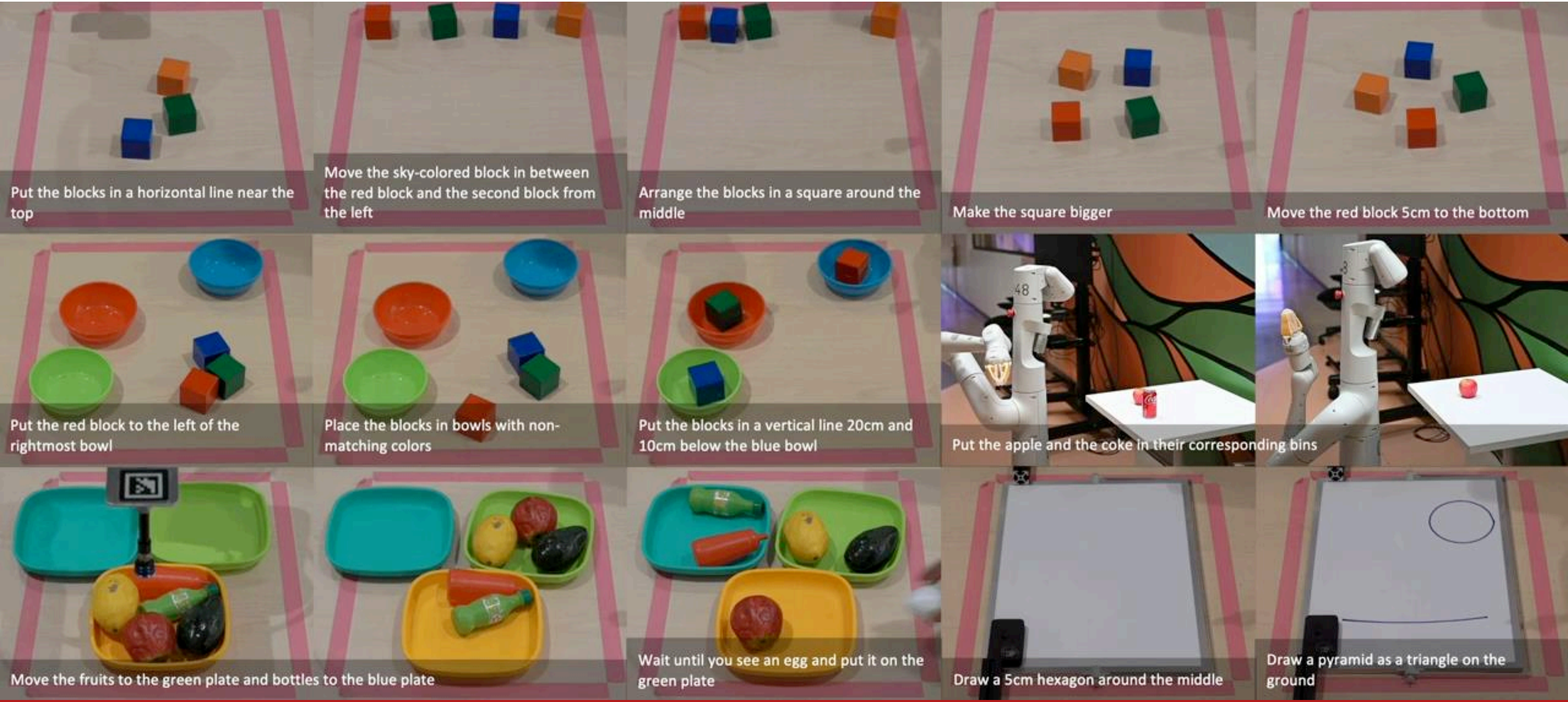
Code as Policies



- Generate control flows
- Generate calls of perception and control APIs
- Run the program

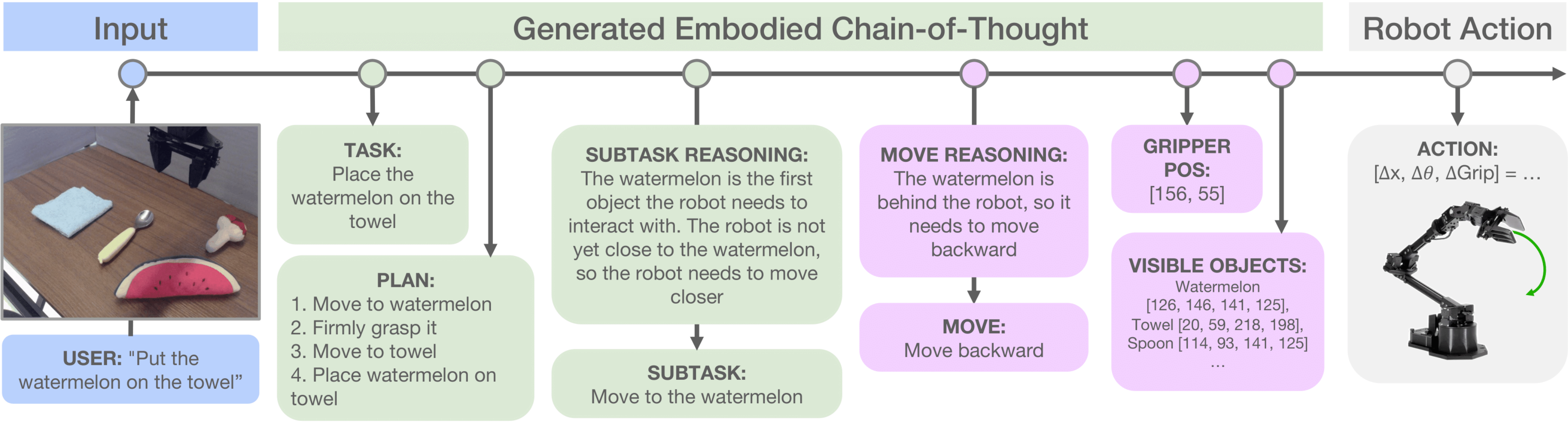
Liang et al. 2023.

Code as Policies



Embodied Chain-of-Thought

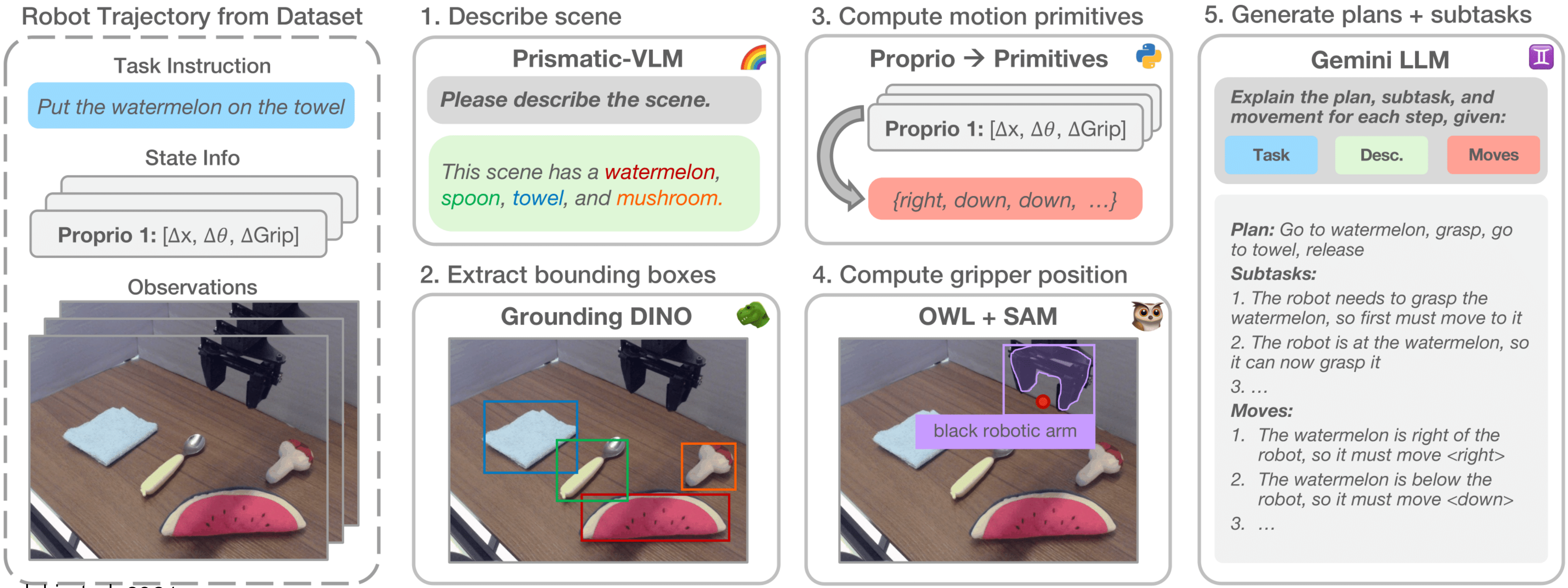
Train a vision-language-action policy to autoregressively generate textual reasoning in response to commands and observations before it chooses a robot action.



Zawalski et al. 2024.

Embodied Chain-of-Thought

a synthetic data generation pipeline that leverages numerous foundation models to extract features from robot demonstrations to put into corresponding textual

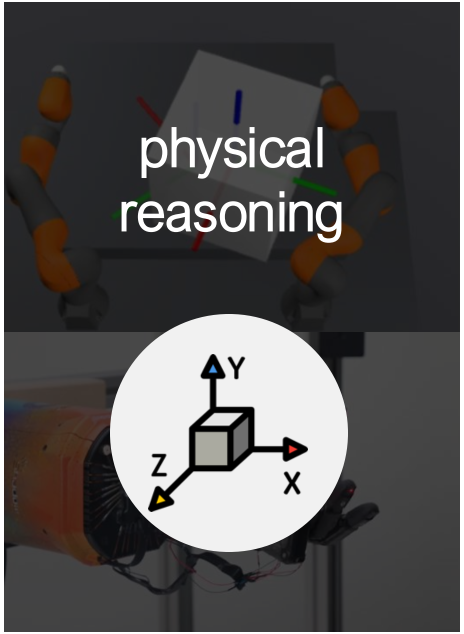


Zawalski et al. 2024.

The Plan for Today

- Task Decomposition for Open-World Robotic Control
- API Calling for Open-World Robotic Control
- **Affordance Representations for Open-World Robotic Control**

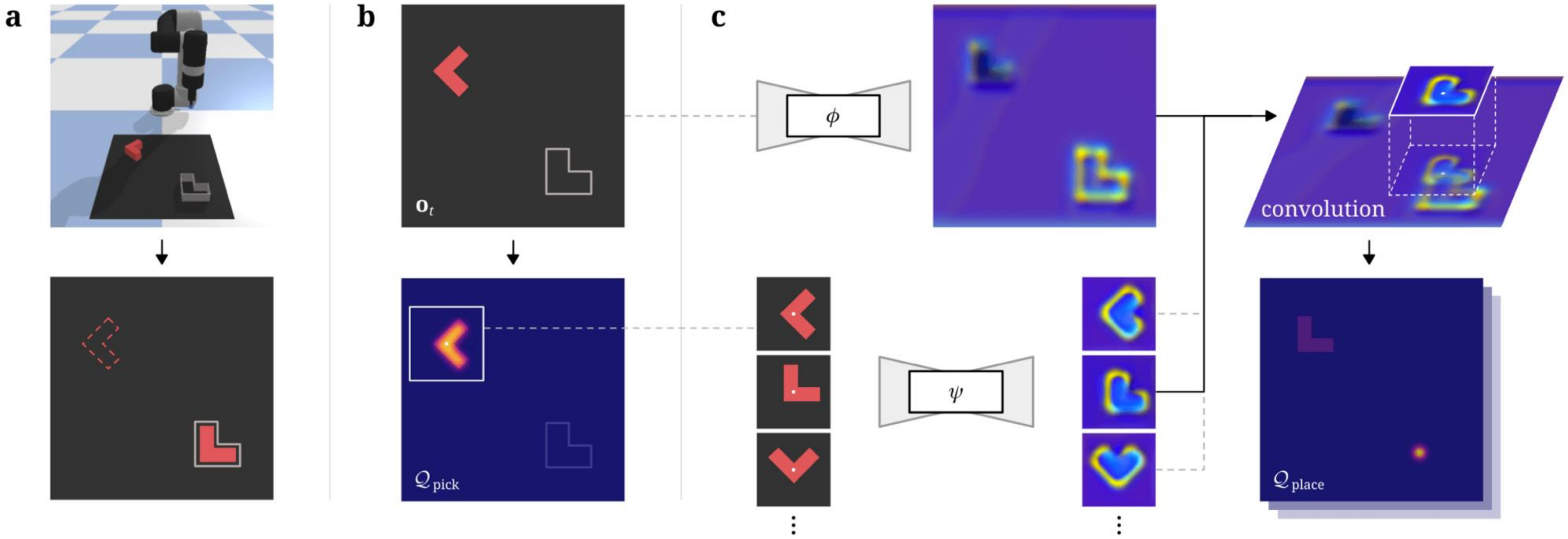
Bridge Semantic and Physical Reasoning with Affordances



spatially grounded visual affordances

Transporter Policy

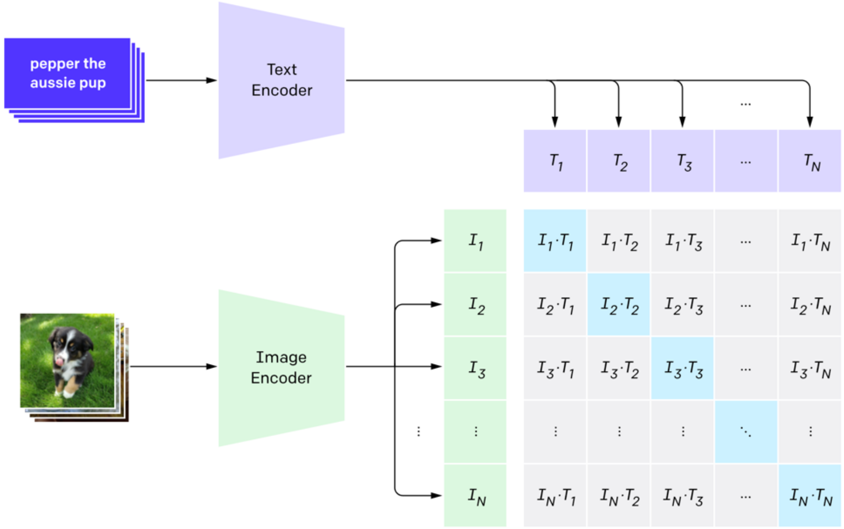
Rearrange deep features to infer spatial displacements from visual input for parameterizing robot actions



Zeng, et al. 2022

CLIP

Pair the texts and images, minimize the InfoNCE loss.



$$I(\mathbf{x};\mathbf{c}) = \sum_{\mathbf{x},\mathbf{c}} p(\mathbf{x},\mathbf{c}) \log \frac{p(\mathbf{x}|\mathbf{c})}{p(\mathbf{x})}$$

$$f(\mathbf{x},\mathbf{c}) \propto \frac{p(\mathbf{x}|\mathbf{c})}{p(\mathbf{x})}$$

$$\mathcal{L}_{\text{InfoNCE}} = -\mathbb{E}\left[\log \frac{f(\mathbf{x},\mathbf{c})}{\sum_{\mathbf{x}'\in X} f(\mathbf{x}',\mathbf{c})}\right]$$

Learning Transferable Visual Models From Natural Language Supervision. Radford et al. 2021

InfoNCE

Given a context vector c, draw one positive sample from the conditional distribution $p(x|c)$, and $N-1$ negative samples from the unconditional distribution $p(x)$.

Let all samples to be $X = \{x_i\}_{i=1}^{N}$. The probability of x_k to be the positive sample is:

$$p(k = \text{“}pos\text{”}|X, c) = \frac{p(x_k|c)\prod_{i\neq k} p(x_i)}{\sum_{j=1}^{N} p(x_j|c)\prod_{i\neq j} p(x_i)} = \frac{\frac{p(x_k|c)}{p(x_k)}}{\sum_{j=1}^{N}\frac{p(x_j|c)}{p(x_j)}}$$

van den Oord, et al. 2018

InfoNCE

Given a context vector c, draw one positive sample from the conditional distribution $p(x|c)$, and $N-1$ negative samples from the unconditional distribution $p(x)$.

Let all samples to be $X = \{x_i\}_{i=1}^{N}$. The probability of x_k to be the positive sample is:

$$p(k = \text{“}pos\text{”}|X, c) = \frac{p(x_k|c)\prod_{i\neq k} p(x_i)}{\sum_{j=1}^{N} p(x_j|c)\prod_{i\neq j} p(x_i)} = \frac{f_\theta(x_k, c)}{\sum_{j=1}^{N} f_\theta(x_j, c)}$$

$$f_\theta(x, c) \propto \frac{p(x|c)}{p(x)}$$

van den Oord, et al. 2018

InfoNCE

The InfoNCE loss optimizes the negative log probability of classifying the positive sample correctly:

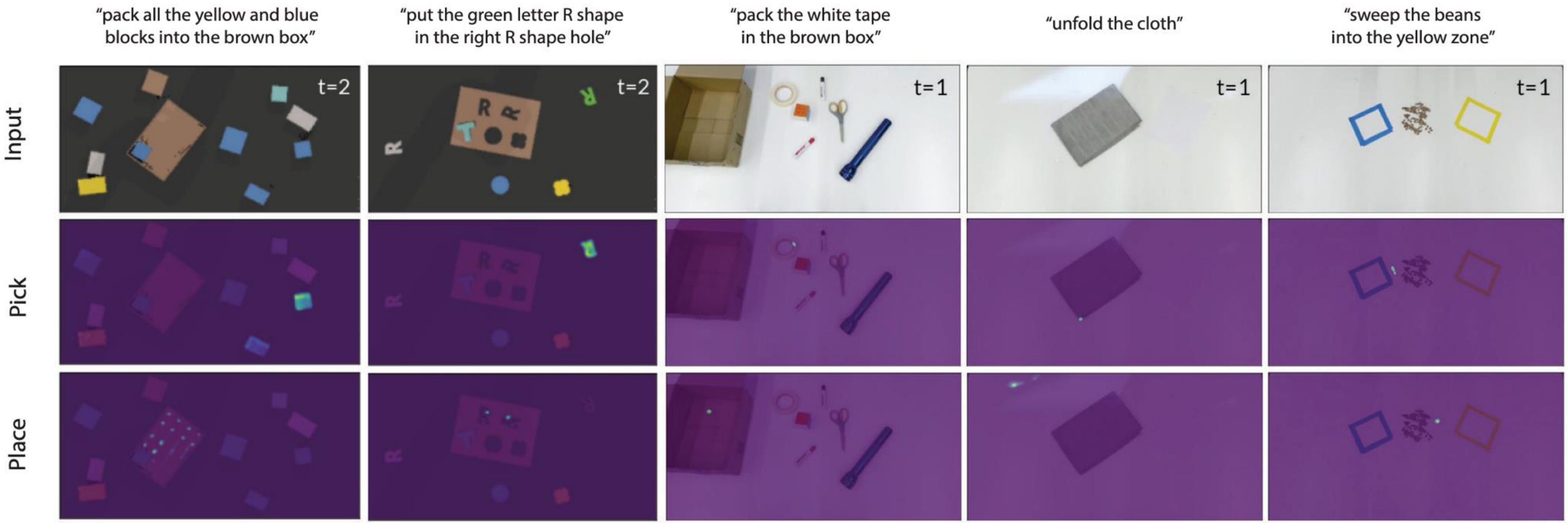
$$\mathcal{L}_{\text{InfoNCE}} = -\mathbb{E}[\log \frac{f_\theta(x, c)}{\sum_{x'} f_\theta(x'\ c)}]$$

$$f_\theta(x, c) \propto \frac{p(x|c)}{p(x)}$$

van den Oord, et al. 2018

CLIPort

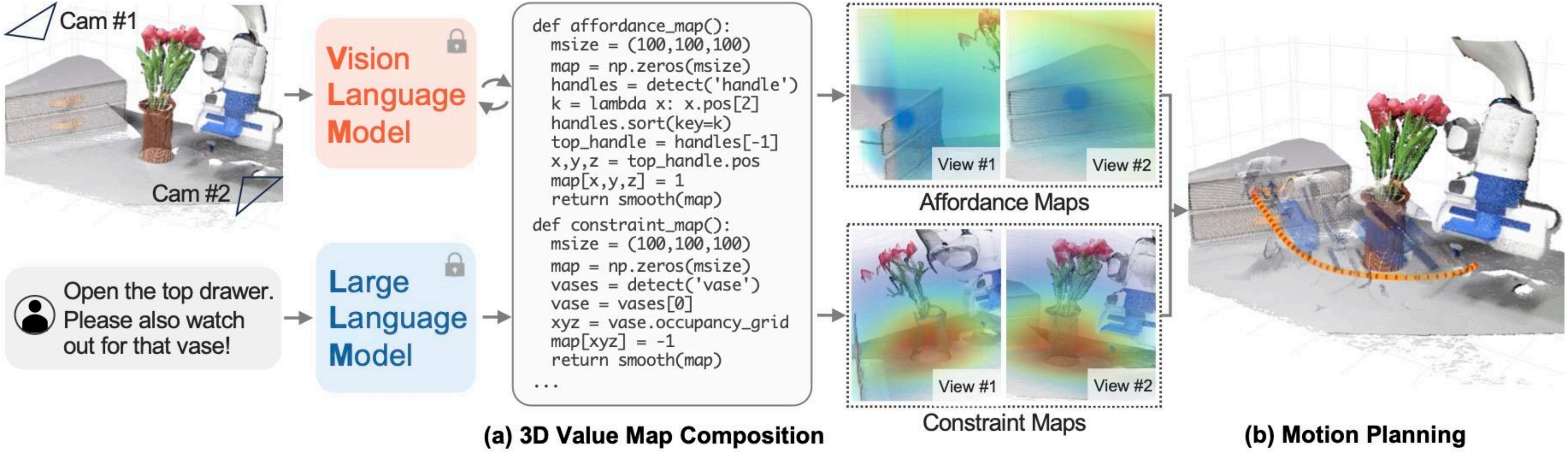
CLIPort combines the broad semantic understanding of CLIP with the spatial precision of Transporter.



Zeng, et al. 2022

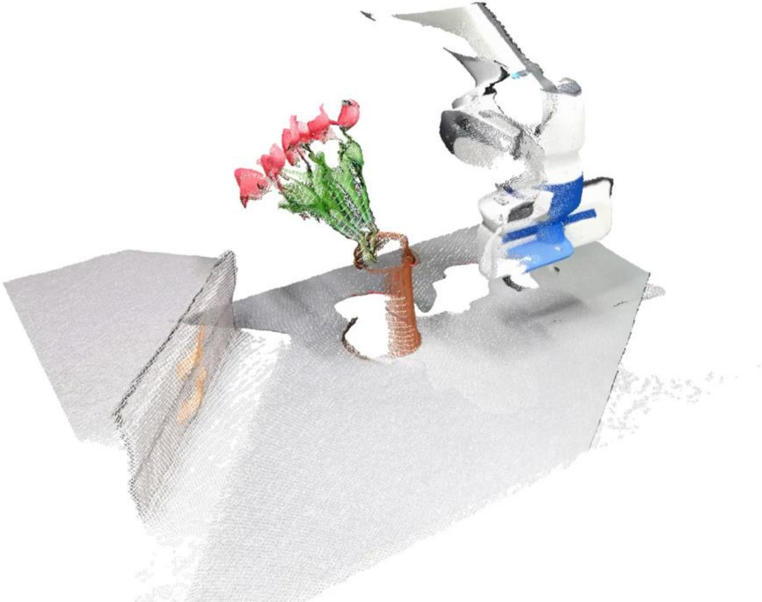
VoxPoser: Composable 3D Value Maps for Manipulation

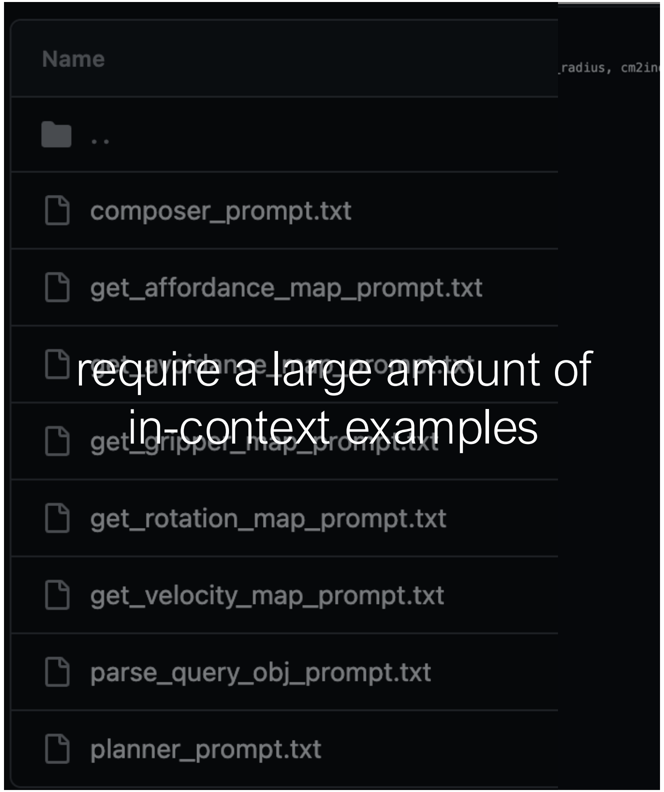
Given the RGB-D observation of the environment and a language instruction,
1. prompt LLMs to generate code to compute a value maps
2. plan for motion trajectories to maximize the values



Huang, et al. 2023

VoxPoser: Composable 3D Value Maps for Manipulation



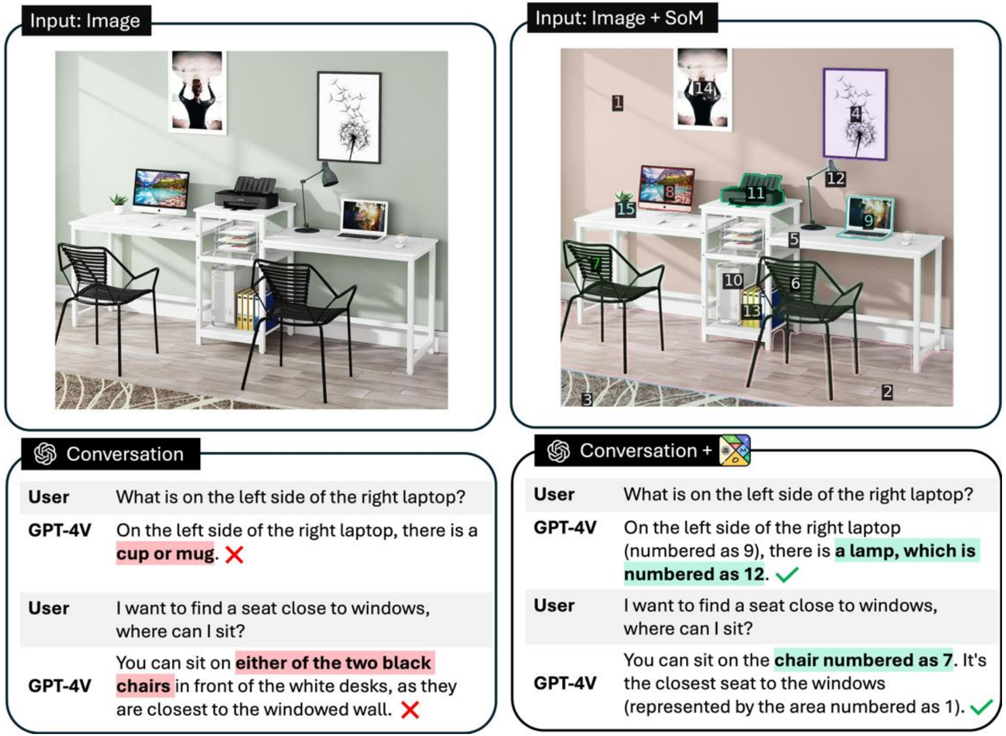


require a large amount of in-context examples

Huang, et al. 2023

Set-of-Mark Prompting

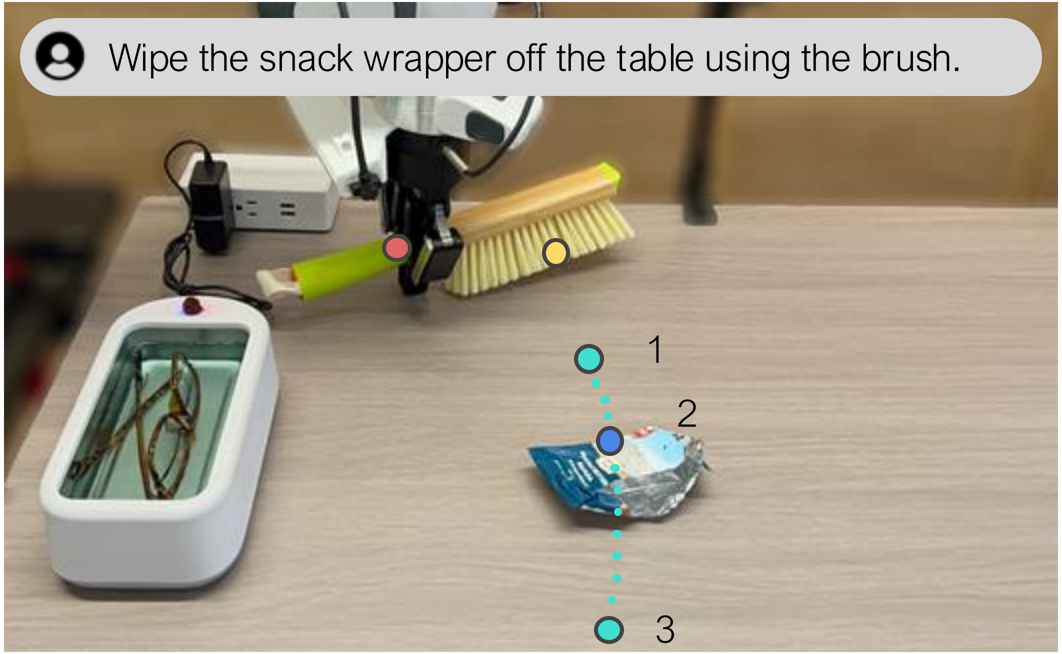
Simply overlaying IDs on image regions unleashes visual grounding and corrects answers for GPT-4V



Set-of-Mark Prompting Unleashes Extraordinary Visual Grounding in GPT-4V. Yang et al. 2023

MOKA: Marking Open-world Keypoint Affordances

Use a set of keypoints to specify the motion trajectory for solving the task.



grasp function target waypoints

- Separate semantics and motions
- Predictable on 2D images.
- Can specify diverse motions.
- Agnostic to the embodiment.

Fang, Liu, Abbeel, Levine. RSS 2024

MOKA: Marking Open-world Keypoint Affordances

Challenge: Directly predicting keypoint coordinates requires fine-grained spatial reasoning.

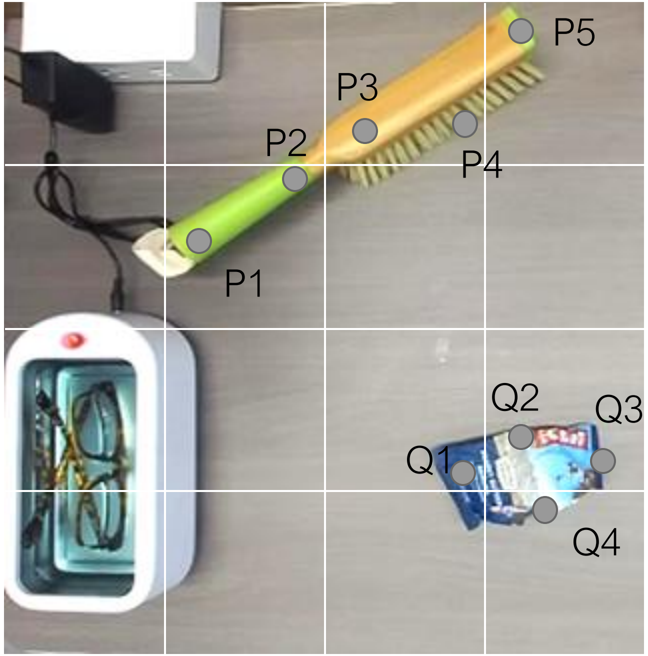
grasp function target waypoints

Fang, Liu, Abbeel, Levine. RSS 2024

MOKA: Marking Open-world Keypoint Affordances

To facilitate reasoning for the VLM, MOKA annotates a set of marks on the input image.

grasp function target waypoints



marks

MOKA: Marking Open-world Keypoint Affordances

Without any training on any robot data, the VLM can solve the commanded manipulation task.

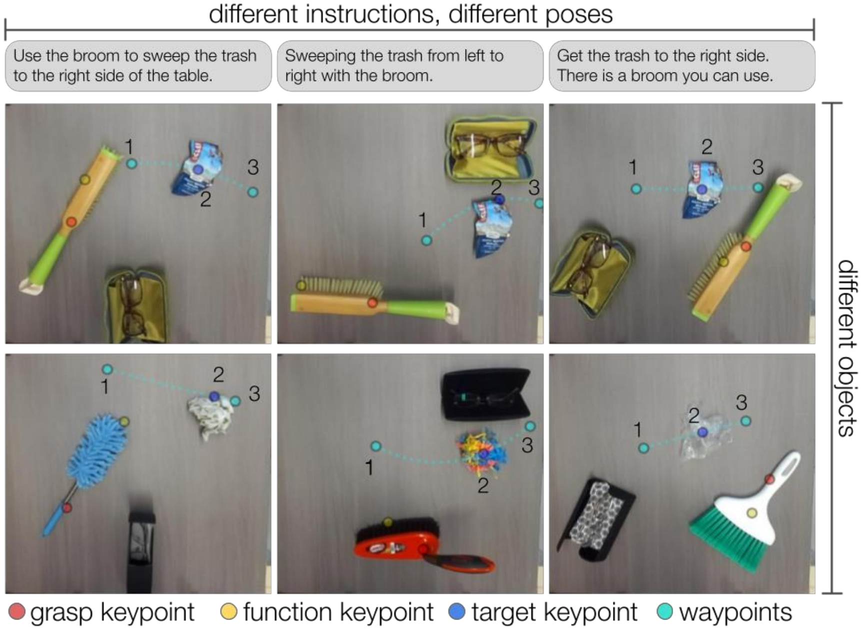
grasp function target waypoints

marks

MOKA: Marking Open-world Keypoint Affordances

Without any training on any robot data, the VLM can solve the commanded manipulation task.

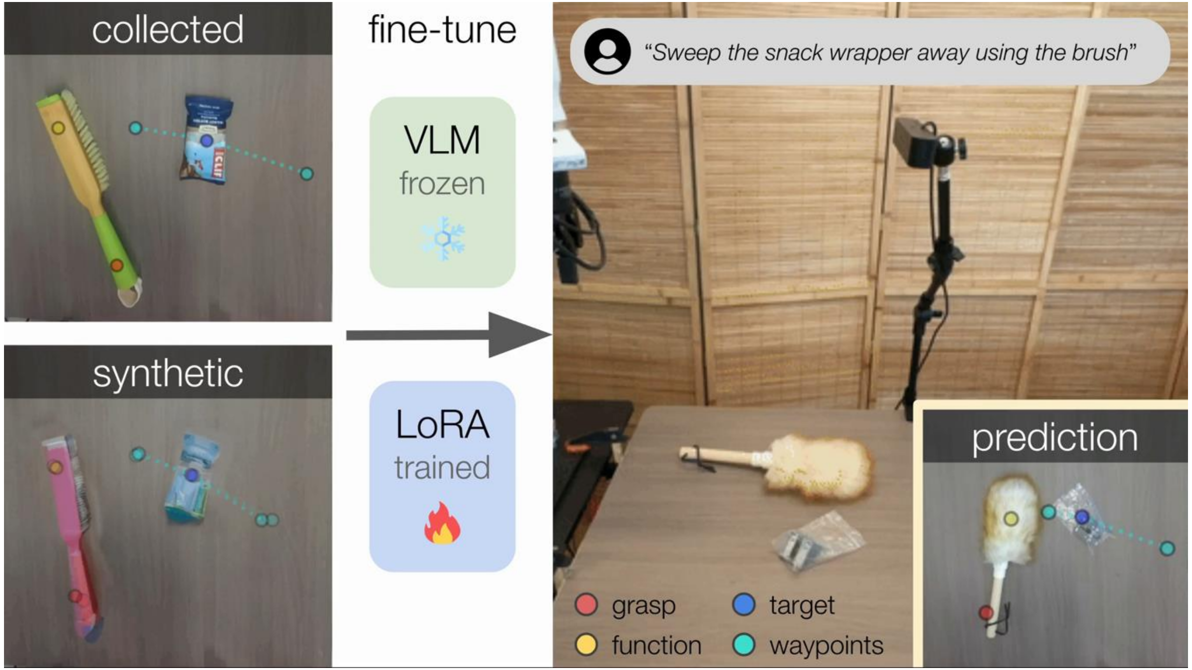
The prediction is robust to different instructions, poses, and objects.



Fang*, Liu*, Abbeel, Levine. Multi-Task Domain Adaptation. RSS 2024

KALIE: Keypoint Affordance Learning from Imagined Environments

How can we fine-tune VLM for robotic control without extensive robot data?



Fang, Liu, Abbeel, Levine. RSS 2024

KALIE: Keypoint Affordance Learning from Imagined Environments

Directly applying generative models to generate new images will result in artifacts and misaligned information.

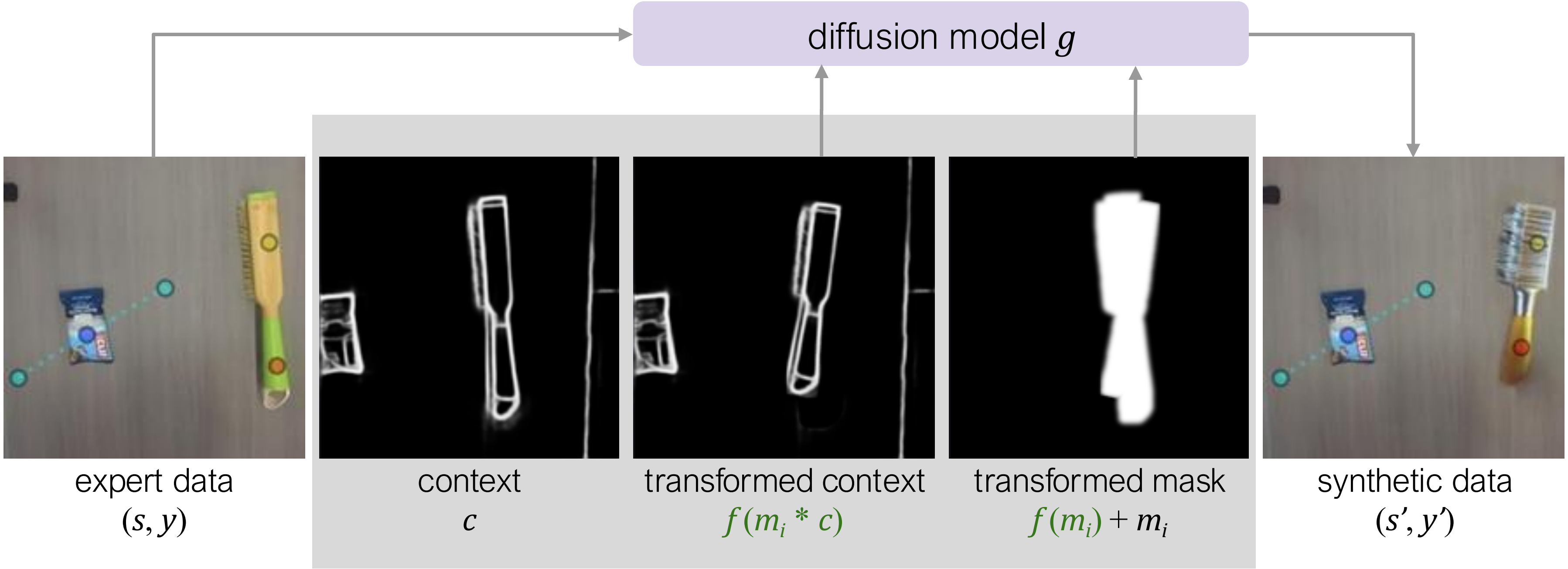
input w/o original w/o context

How can we generates synthetic data with high diversity while staying faithful to the task semantics and keypoint annotation?

Tang, Rajkumar, Zhou, Walke, Levine, **Fang**. Fine-Tuning Vision-Language Models for Open-World Manipulation without Robot Data. In Submission

KALIE: Keypoint Affordance Learning from Imagined Environments

KALIE uses a **context image** as additional inputs to the diffusion model, which specifies the geometric properties of the object to be inpainted.

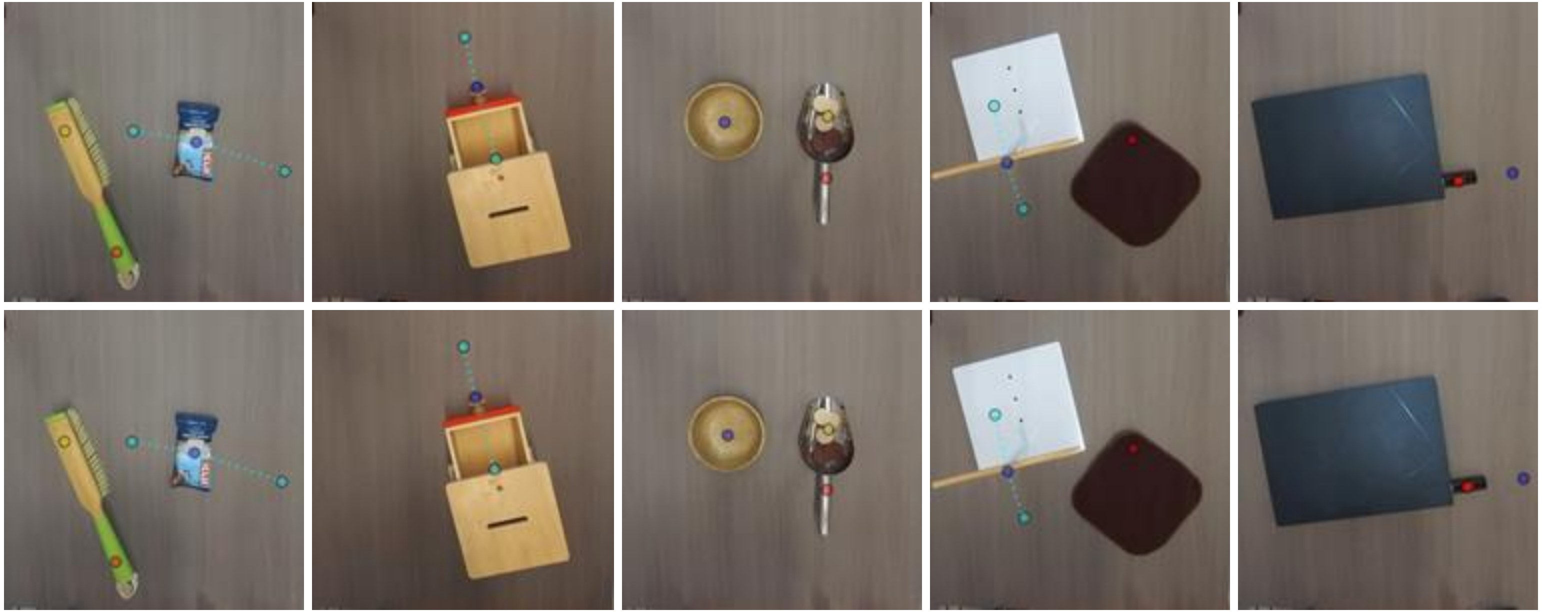


KALIE: Keypoint Affordance Learning from Imagined Environments

- Employ conditional diffusion models to **diversify** the training data.
- **Fine-tune** the VLM to predict affordances through low-rank adaptation.

collected data

synthetic data



The Plan for Today

- Task Decomposition for Open-World Robotic Control
- API Calling for Open-World Robotic Control
- Affordance Representations for Open-World Robotic Control