Robotics, Computer Vision, and NLP

Max Fu and Rudy Corona

max.fu.letian@berkeley.edu

Berkeley
UNIVERSITY OF CALIFORNIA

Last week: how are robot controlled

- Existing working systems are **not** end to end reinforcement learning / imitation learning
- 2 key ideas
    - Abstraction
    - Hierarchy

Last week: how are robot controlled

- **Abstraction**
- Lowest level: joint position (torque PD Control)
- Middle: cartesian space control (jacobian + PID control) / model-predictive control
- Upper: motion planning (RRT, PRM) / SLAM
- Beyond: task planning (where to grasp? Ordering of task? affordance?) / where to go?

Last week: how are robot controlled

- **Modularity**
- Different modules work together to solve a problem
- “If we make each module work perfectly, then a problem will be solved!”
- In addition to hierarchical modules, you can expand horizontally!
    - Analyze failure modes and catch them individually

Summary from last week

- Advantages:
    - *"If each part works, then as whole it should work!" *after some integration tests*
    - Interpretability!
    - We FULLY know the robot's dynamics!
- Disadvantages:
    - **Slow / high latency**: planning / scouting out a scene takes sec/minutes. The robot needs to move now!
        - **Many of them are not reactive**: take a capture of the scene then the robot moves (what if the object is a moving target?)
    - We as grad students needs to implement each of the module!
    - **Data labelers (amazon turk, scale ai) is more scalable than grad students!**
    - But, the framework as whole does not scale with data
        - Each part may still scale with increasing data
    - NOT align with the bitter lesson [1]! **Too much inductive bias!**

[1] Sutton, R. (2019). The bitter lesson. *Incomplete Ideas (blog)*, *13*(1), 38.

The bitter lesson

The bitter lesson is based on the historical observations that 1) AI researchers have often tried to **build knowledge into their agents (*abstraction and modularity*)**, 2) this always helps in the short term, and is personally satisfying to the researcher, but 3) in the long run it **plateaus** and even inhibits further progress (***each module becomes increasing hard to improve/fix***), and 4) breakthrough progress eventually arrives by an opposing approach based on **scaling computation by search and learning** (***robot learning?***). The eventual success is tinged with bitterness, and often incompletely digested, because it is success over a favored, human-centric approach.

One thing that should be learned from the bitter lesson is the great power of general purpose methods, of **methods that continue to scale with increased computation even as the available computation becomes very great**. The two methods that seem to scale arbitrarily in this way are search and learning.

Sutton, R. (2019). The bitter lesson. *Incomplete Ideas (blog)*, *13*(1), 38.

How can we move away from abstraction?

- Predict one of
  - Lowest level: joint torque (>1000 Hz)
  - Lower level: joint position (+ PD Control with torque) (10-200 Hz)
  - Middle: cartesian space control (jacobian + PID control) (ideally > 10 Hz)
  - Upper: motion planning (RRT, PRM) (**a few seconds*)
  - Beyond: task planning (where to grasp? Ordering of task? affordance?) / where to go? (*this can take arbitrarily long*)

How can we move away from abstraction?

- Upper: motion planning (RRT, PRM) (**a few seconds*)
  - Just make it learning based / optimize it better!
  - Still not reactive though unless you keep planning!

Metric	BiTStar		MπNet		cuRobo v0.6.2			DiffusionSeeder				
Condition	$\bar{\delta}$	δ	δ'	δ	$N_{atp}=1$	10	100	$N_{iters}=25$	50	100	200	475
Plan Time (s)	0.52	0.48	0.50	1.95	0.049	0.082	0.207	**0.015**	0.017	0.020	0.027	0.045
Total Time (s)	0.69	0.65	0.50	1.95	0.079	0.112	0.237	**0.045**	0.047	0.050	0.057	0.075
Success Rate	26.6%	6.0%	27.4%	8.3%	66.2%	77.7%	77.9%	85.1%	85.8%	84.9%	85.1%	**86.2%**
Jerk (rad/s^3)	**47.2**	49.9	56.8	60.6	98.5	96.7	97.8	108.8	103.6	99.3	93.5	89.6
Motion Time (s)	1.84	1.98	5.35	7.71	**1.14**	1.17	1.18	1.26	1.26	1.27	1.30	1.26
Translation Err (mm)	3.89	4.05	8.66	3.92	**0.05**	0.06	0.06	0.98	0.95	0.91	0.50	0.78
Quaternion Err (°)	13.3	1.10	7.27	2.68	**0.63**	0.90	0.93	1.78	1.44	1.20	1.03	0.92

Table 1: Evaluation with partial observations of a depth image for BiTStar, MπNet, cuRobo and DiffusionSeeder. Mean values of each metric on successfully solved problems over the 1791 test problems are reported. As DiffusionSeeder-50 has a similar success to DiffusionSeeder-475 but is 60% faster at planning, we use DiffusionSeeder-50 when reporting primary results.

Huang, H., Sundaralingam, B., Mousavian, A., Murali, A., Goldberg, K., & Fox, D. DiffusionSeeder: Seeding Motion Optimization with Diffusion for Rapid Motion Planning. In 8th Annual Conference on Robot Learning.

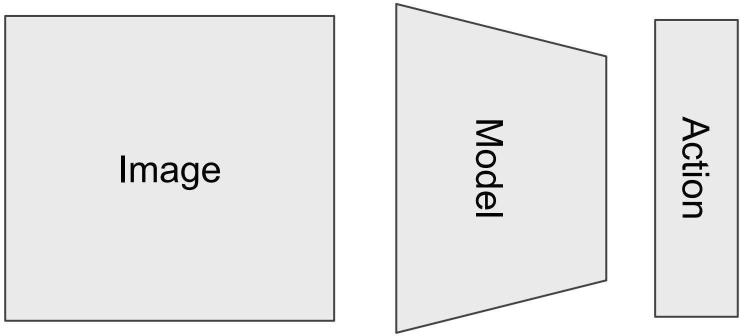
How can we move away from abstraction?

- Predict one of
  - Lowest level: joint torque (2000 Hz)
  - Lower level: joint position (+ PD Control with torque)
  - Middle: cartesian space control (jacobian + PID control) / model-predictive control
  - ~~Upper: motion planning (RRT, PRM) / SLAM~~
  - ~~Beyond: task planning (where to grasp? Ordering of task? affordance?) / where to go?~~
  - **Let's remove motion planning and task planning to make the policy more reactive and lower latency**
  - **Control from vision (+ proprioception)**

[1] Sutton, R. (2019). The bitter lesson. *Incomplete Ideas (blog)*, *13*(1), 38.

I. Imitation learning

Let's ignore language for now. How to get from vision to action?



ALVIN

Input: image

Output: steer angle

Fully connected network

305

ALVINN:
AN AUTONOMOUS LAND VEHICLE IN A NEURAL NETWORK

Dean A. Pomerleau
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213

ABSTRACT

ALVINN (Autonomous Land Vehicle In a Neural Network) is a 3-layer back-propagation network designed for the task of road following. Currently ALVINN takes images from a camera and a laser range finder as input and produces as output the direction the vehicle should travel in order to follow the road. Training has been conducted using simulated road images. Successful tests on the Carnegie Mellon autonomous navigation test vehicle indicate that the network can effectively follow real roads under certain field conditions. The representation developed to perform the task differs dramatically when the network is trained under various conditions, suggesting the possibility of a novel adaptive autonomous navigation system capable of tailoring its processing to the conditions at hand.

INTRODUCTION

Autonomous navigation has been a difficult problem for traditional vision and robotic techniques, primarily because of the noise and variability associated with real world scenes. Autonomous navigation systems based on traditional image processing and pattern recognition techniques often perform well under certain conditions but have problems with others. Part of the difficulty stems from the fact that the processing performed by these systems remains fixed across various driving situations.

Artificial neural networks have displayed promising performance and flexibility in other domains characterized by high degrees of noise and variability, such as handwritten character recognition [Jackel et al., 1988] [Pawlicki et al., 1988] and speech recognition [Waibel et al., 1988]. ALVINN (Autonomous Land Vehicle In a Neural Network) is a connectionist approach to the navigational task of road following. Specifically, ALVINN is an artificial neural network designed to control the NAVLAB, the Carnegie Mellon autonomous navigation test vehicle.

NETWORK ARCHITECTURE

ALVINN's current architecture consists of a single hidden layer back-propagation network

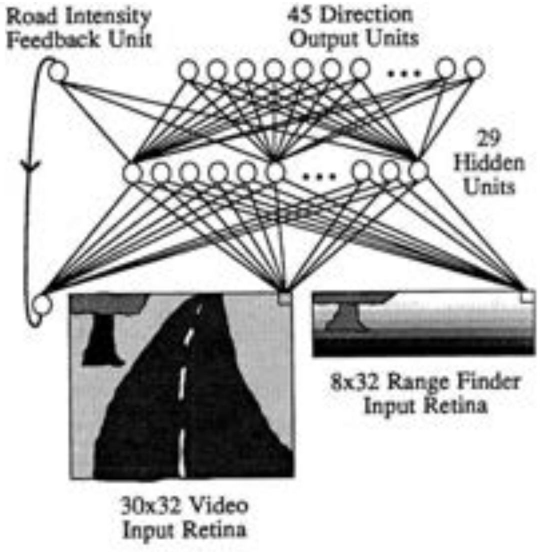


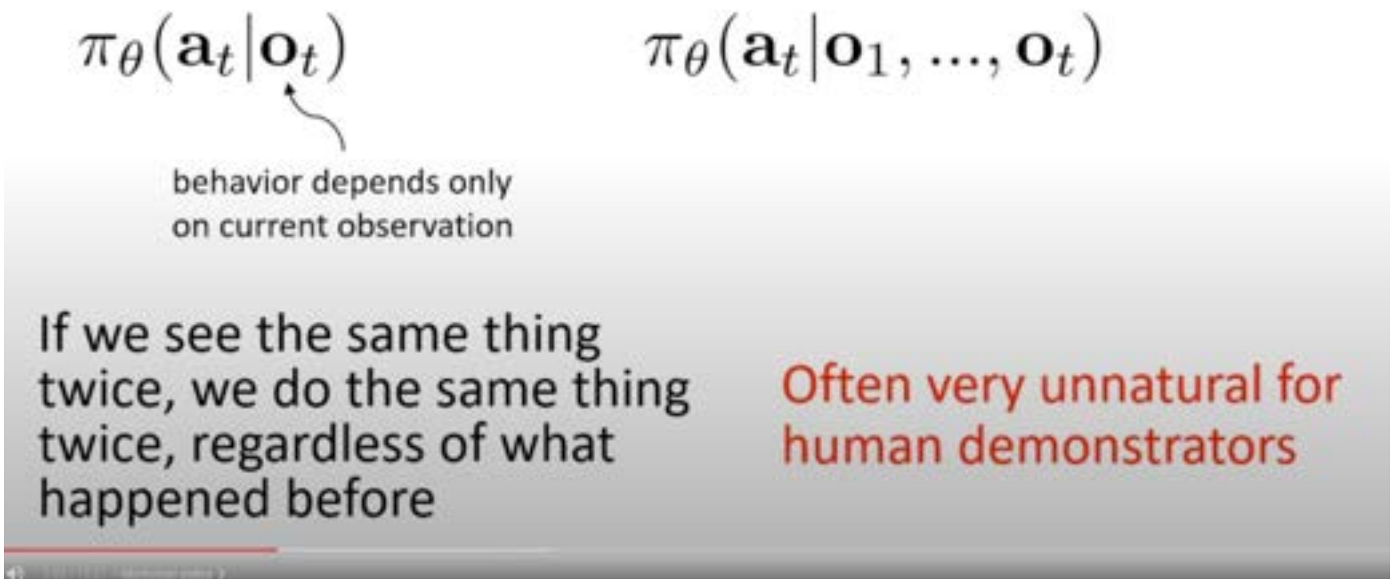
Figure 1: ALVINN Architecture

Pomerleau, D. A. (1988). Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, *1*.

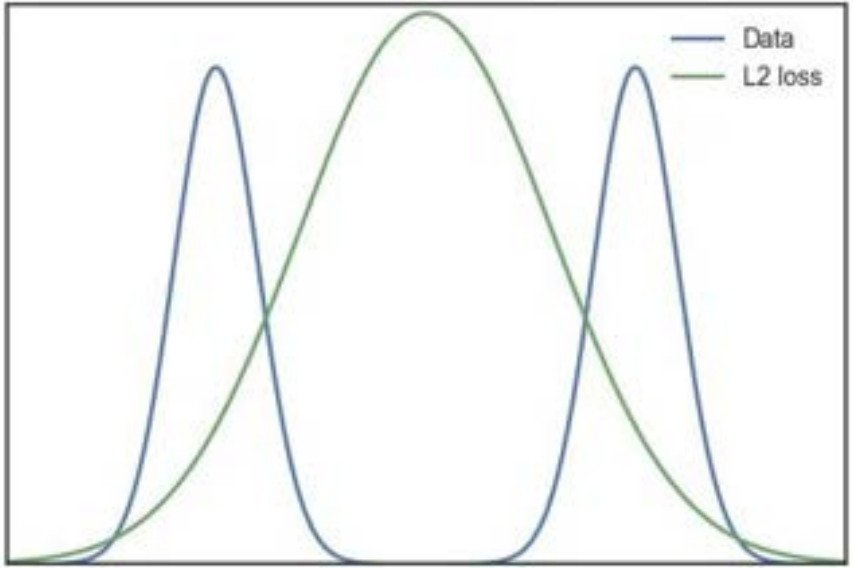
What are some problems

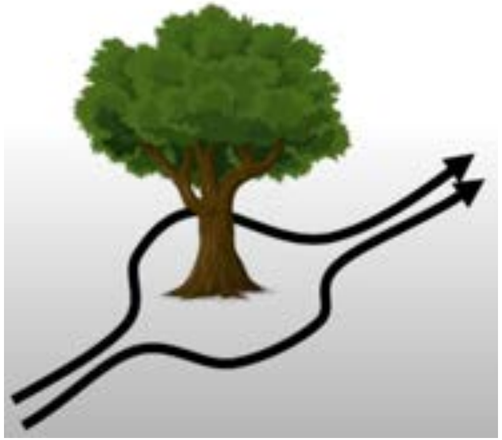
Demonstrations are non-markovian

Demonstrations are multimodal



Regressing Continuous Action





*Problem with L1 / L2 regression:
- Assumes unimodal gaussian prior
- Mode covering
- Gonna hit the tree!
- Naive solution: mixture of gaussians

Implicit BC

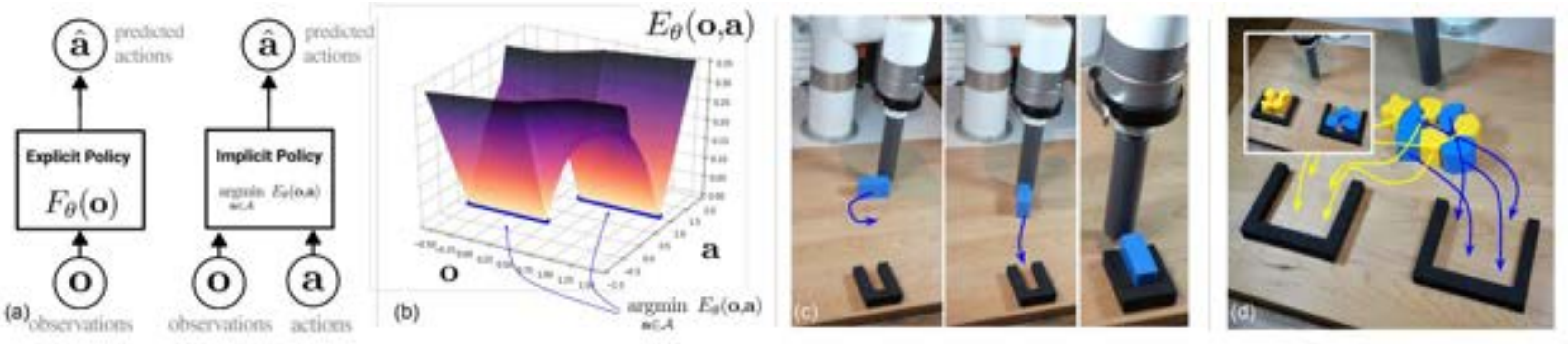


Figure 1. (a) In contrast to explicit policies, implicit policies leverage parameterized energy functions that take both observations (e.g. images) and actions as inputs, and optimize for actions that minimize the energy landscape (b). For learning complex, closed-loop, multimodal visuomotor tasks such as precise block insertion (c) and sorting (d) from human demonstrations, implicit policies perform substantially better than explicit ones.

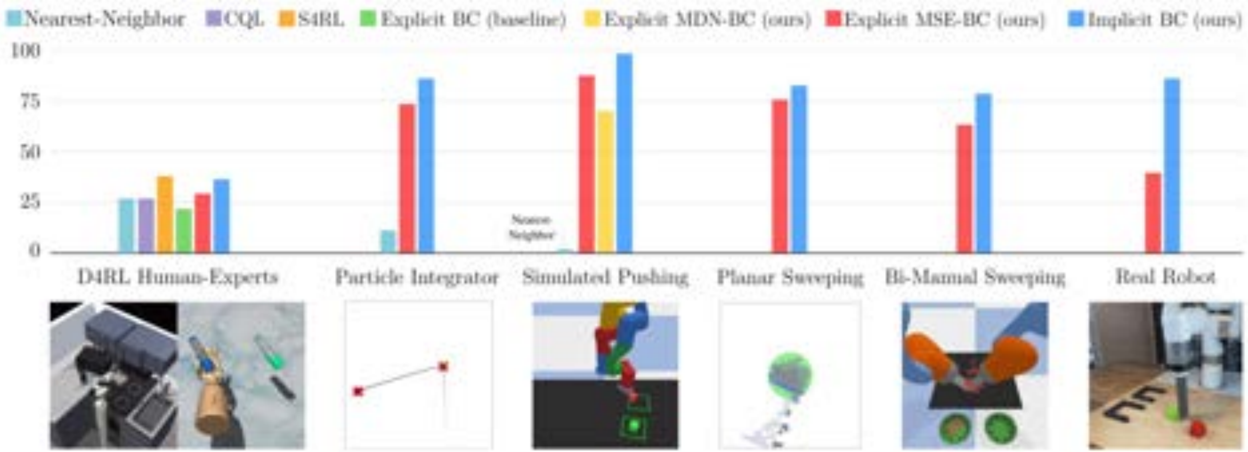
Florence, P., Lynch, C., Zeng, A., Ramirez, O. A., Wahid, A., Downs, L., ... & Tompson, J. (2022, January). Implicit behavioral cloning. In Conference on Robot Learning (pp. 158-168). PMLR.

Implicit BC

$$\mathcal{L}_{\text{InfoNCE}} = \sum_{i=1}^{N} -\log\left(\tilde{p}_\theta(\mathbf{y}_i \mid \mathbf{x}, \{\tilde{\mathbf{y}}_i^j\}_{j=1}^{N_{\text{neg.}}})\right), \quad \tilde{p}_\theta(\mathbf{y}_i \mid \mathbf{x}, \{\tilde{\mathbf{y}}_i^j\}_{j=1}^{N_{\text{neg.}}}) = \frac{e^{-E_\theta(\mathbf{x}_i, \mathbf{y}_i)}}{e^{-E_\theta(\mathbf{x}_i, \mathbf{y}_i)} + \sum_{j=1}^{N_{\text{neg}}} e^{-E_\theta(\mathbf{x}_i, \tilde{\mathbf{y}}_i^j)}}$$

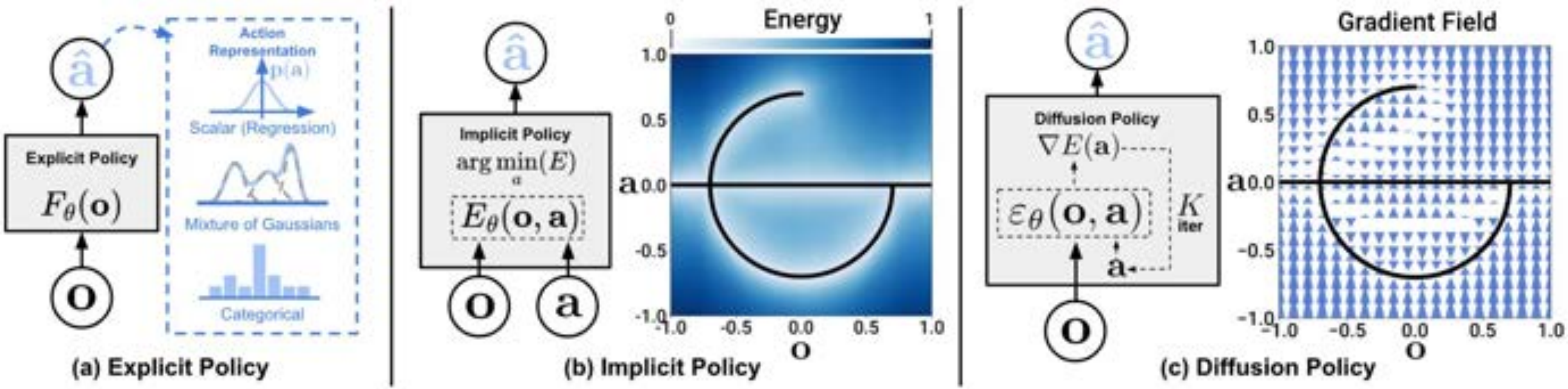
$$\hat{\mathbf{y}} = \operatorname{argmin}_{\mathbf{y}} \; E_\theta(\mathbf{x}, \mathbf{y})$$

Inference time: solve for **argmin** with sample based or gradient based methods



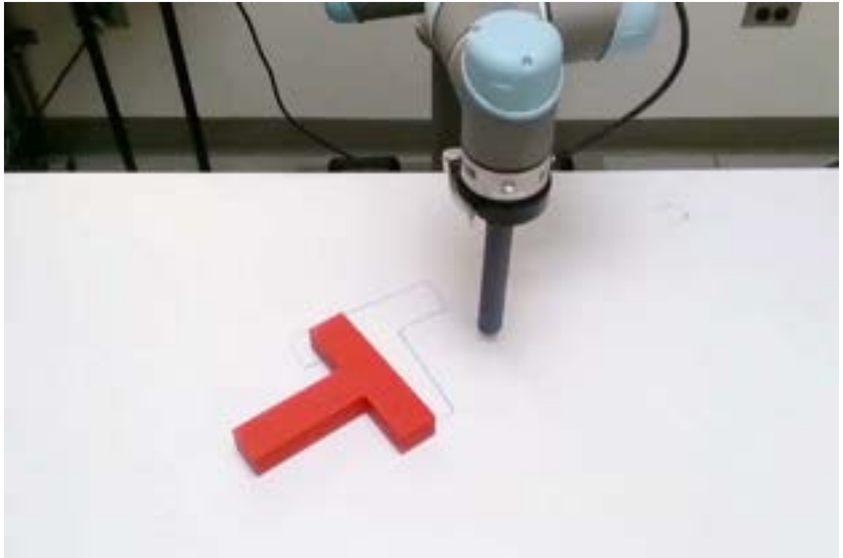
Florence, P., Lynch, C., Zeng, A., Ramirez, O. A., Wahid, A., Downs, L., ... & Tompson, J. (2022, January). Implicit behavioral cloning. In Conference on Robot Learning (pp. 158-168). PMLR.

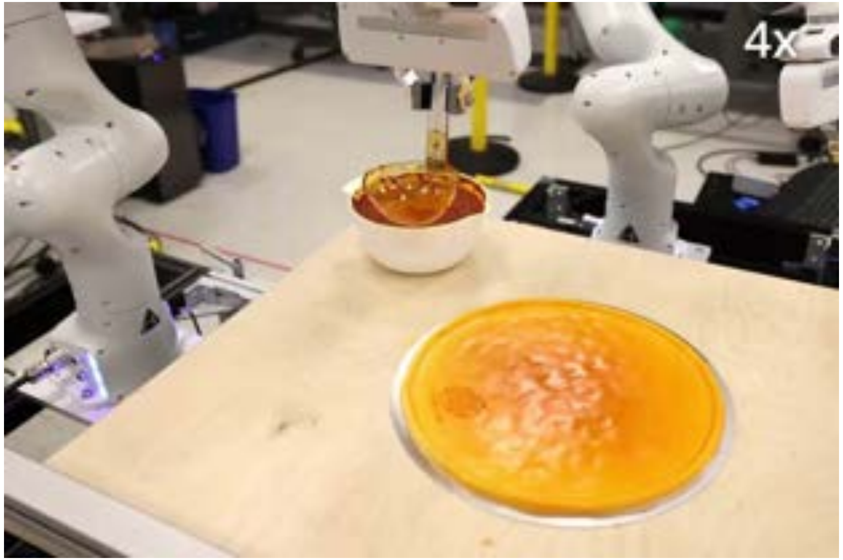
Diffusion Policy



Chi, C., Feng, S., Du, Y., Xu, Z., Cousineau, E., Burchfiel, B., & Song, S. (2023). Diffusion policy: Visuomotor policy learning via action diffusion. arXiv preprint arXiv:2303.04137.

Diffusion Policy



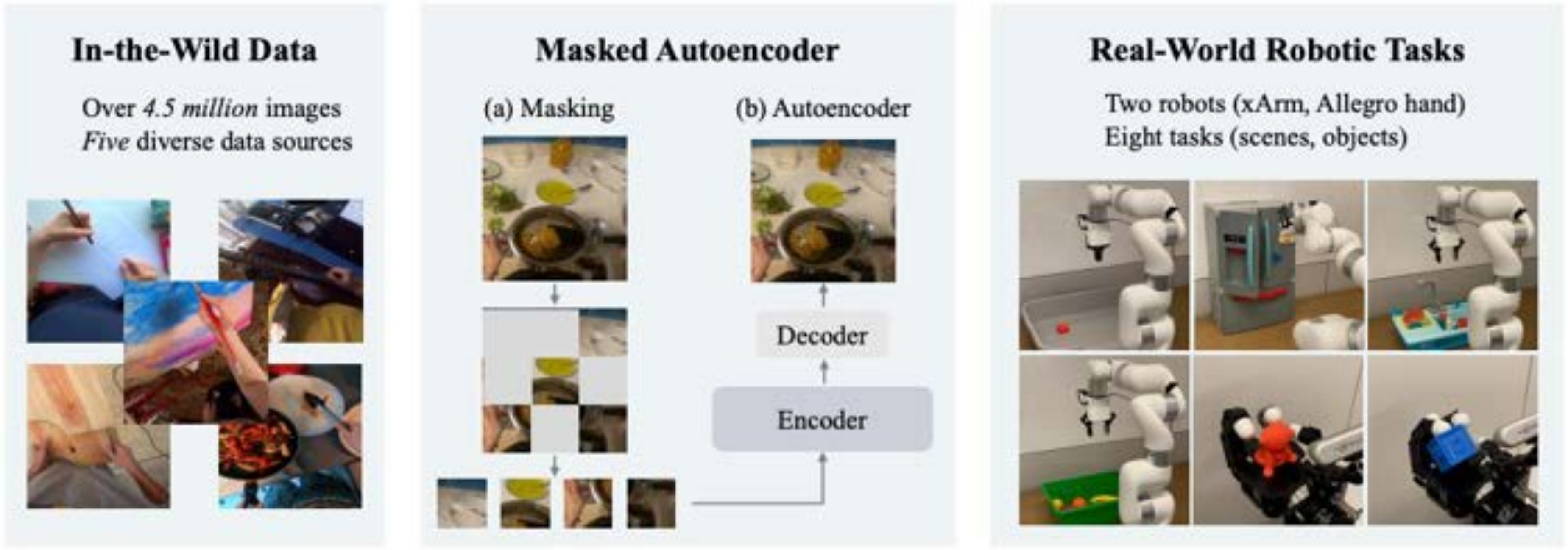


Chi, C., Feng, S., Du, Y., Xu, Z., Cousineau, E., Burchfiel, B., & Song, S. (2023). Diffusion policy: Visuomotor policy learning via action diffusion. arXiv preprint arXiv:2303.04137.

What’s an efficient training scheme? (pre-training?)

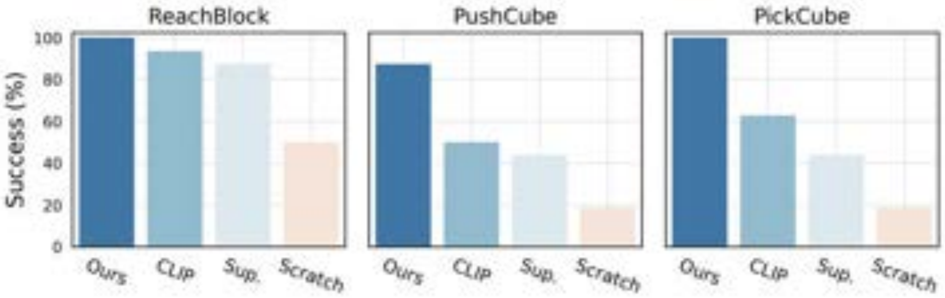
- Efficient multi-task learning -> reduce cost of transfer learning / task adaptation
- Visual Pre-training:
  - MAE -> the MVP
- Sequence pre-training:
  - BERT (efficient fine-tuning) -> RPT

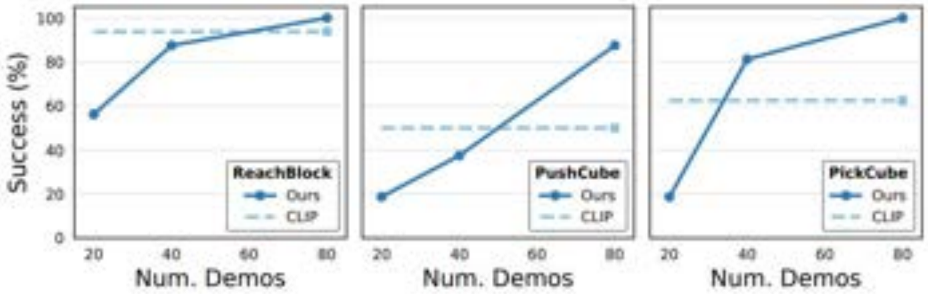
Masked Visual Pre-training for Motor Control



Radosavovic, I., Xiao, T., James, S., Abbeel, P., Malik, J., & Darrell, T. (2023, March). Real-world robot learning with masked visual pre-training. In Conference on Robot Learning (pp. 416-426). PMLR.

Masked Visual Pre-training for Motor Control



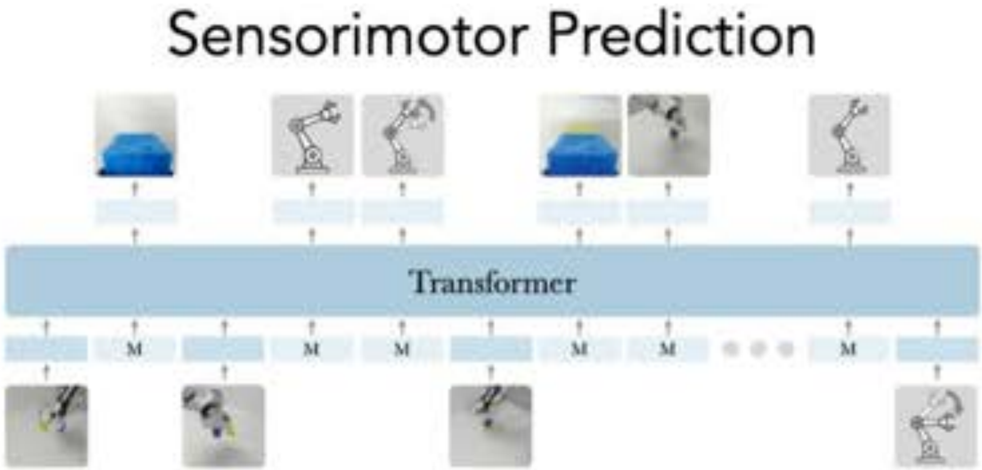


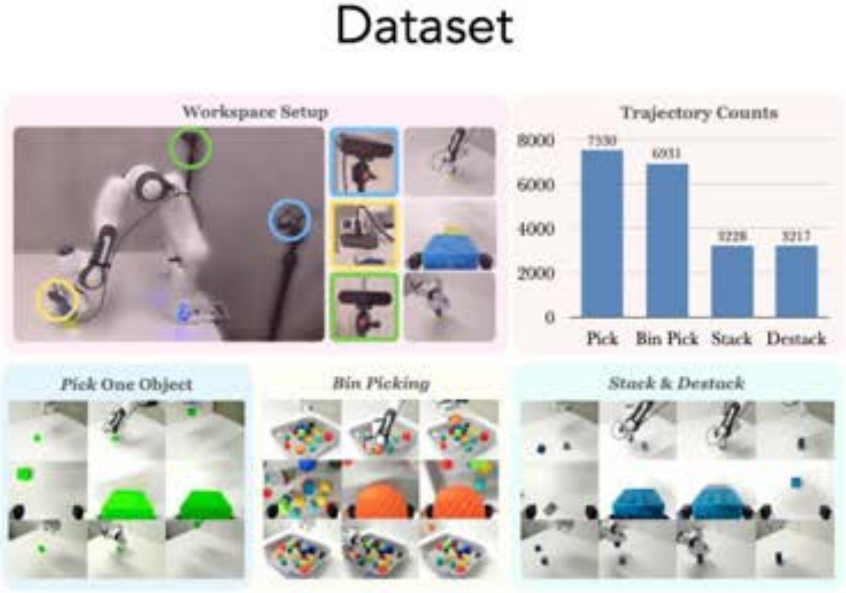
Radosavovic, I., Xiao, T., James, S., Abbeel, P., Malik, J., & Darrell, T. (2023, March). Real-world robot learning with masked visual pre-training. In Conference on Robot Learning (pp. 416-426). PMLR.

What's an efficient training scheme? (pre-training?)

- Efficient multi-task learning -> reduce cost of transfer learning / task adaptation
- Visual Pre-training:
    - MAE -> the MVP
- **Sequence pre-training:**
    - BERT (efficient fine-tuning) -> RPT

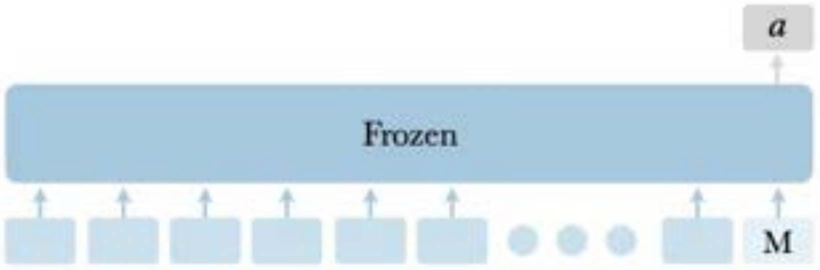
Robot Pretrained Transformer





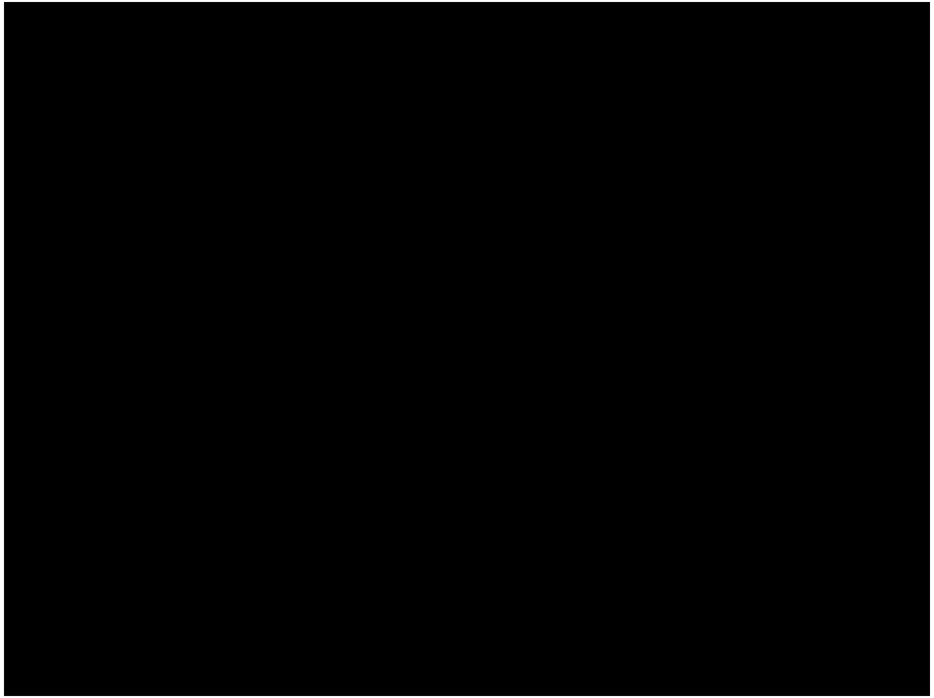
Radosavovic, I., Shi, B., Fu, L., Goldberg, K., Darrell, T., & Malik, J. (2023, December). Robot learning with sensorimotor pre-training. In *Conference on Robot Learning* (pp. 683-693). PMLR.

Robot Pretrained Transformer



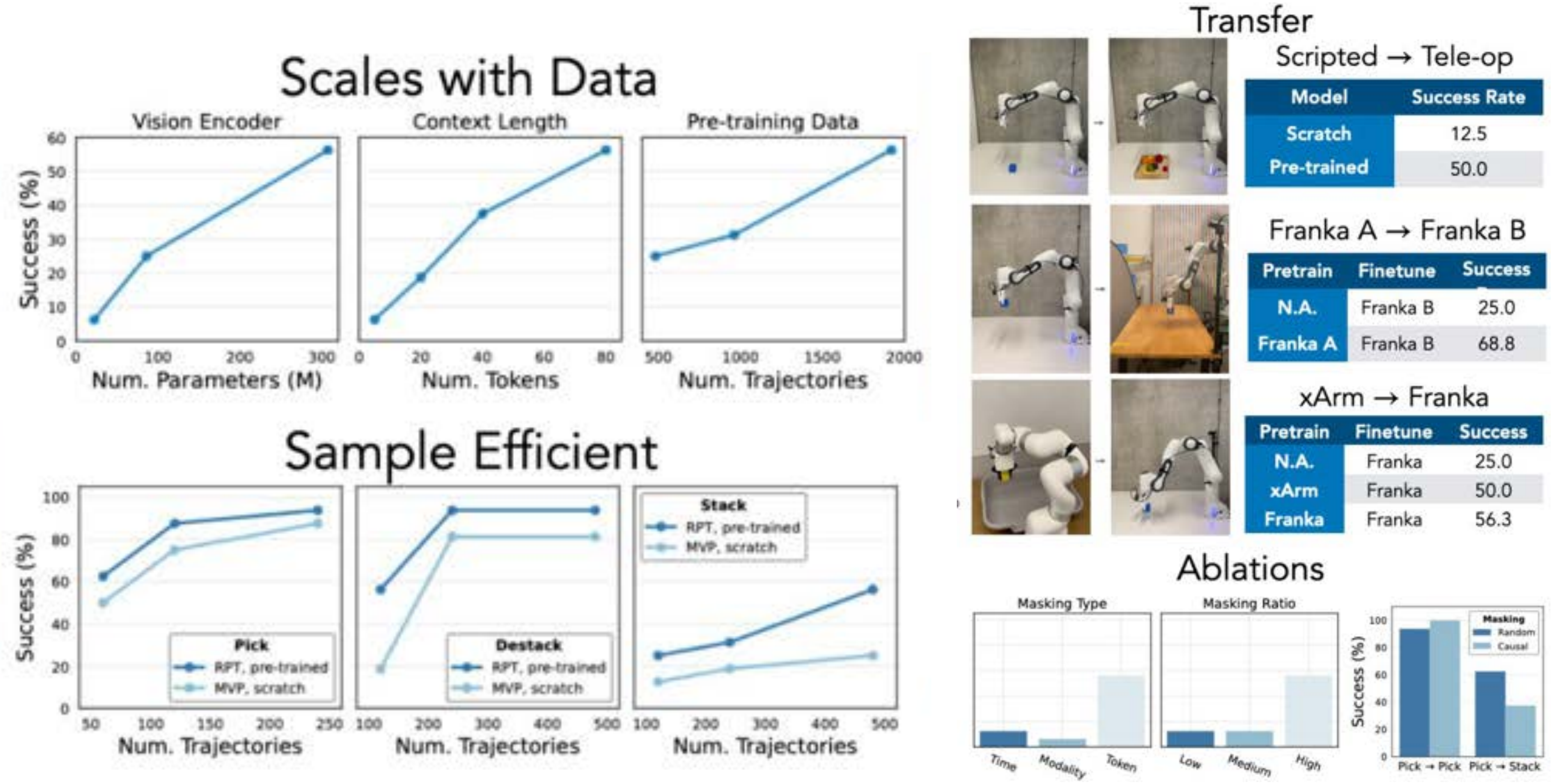
Task Specific Linear Probe

Task Specific Fine-tuning



Radosavovic, I., Shi, B., Fu, L., Goldberg, K., Darrell, T., & Malik, J. (2023, December). Robot learning with sensorimotor pre-training. In *Conference on Robot Learning* (pp. 683-693). PMLR.

Robot Pretrained Transformer

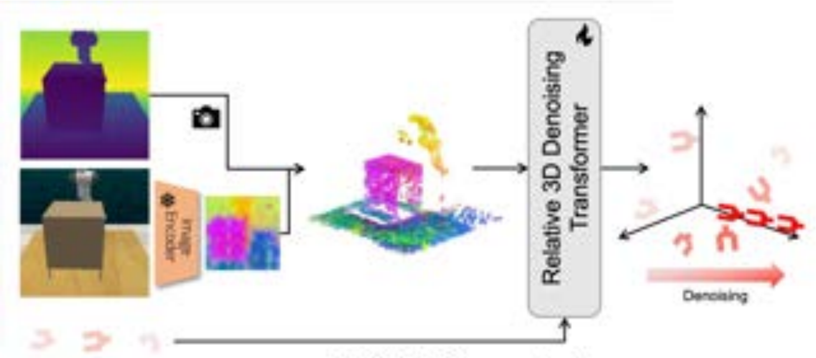


Radosavovic, I., Shi, B., Fu, L., Goldberg, K., Darrell, T., & Malik, J. (2023, December). Robot learning with sensorimotor pre-training. In *Conference on Robot Learning* (pp. 683-693). PMLR.

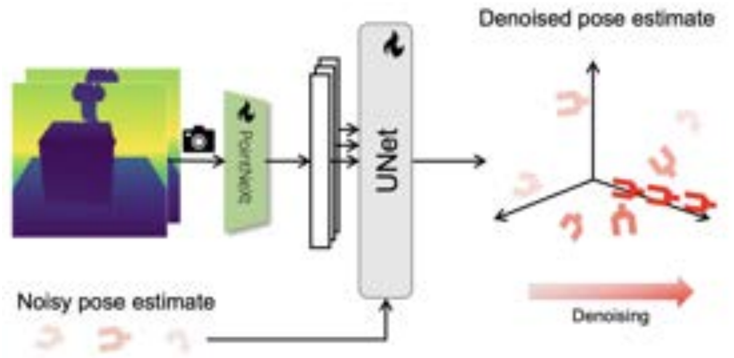
Similarly, we can raise to 3D representations

Or implicit 3D (multiview?)

Is 3D Necessary?: A case study in manipulation



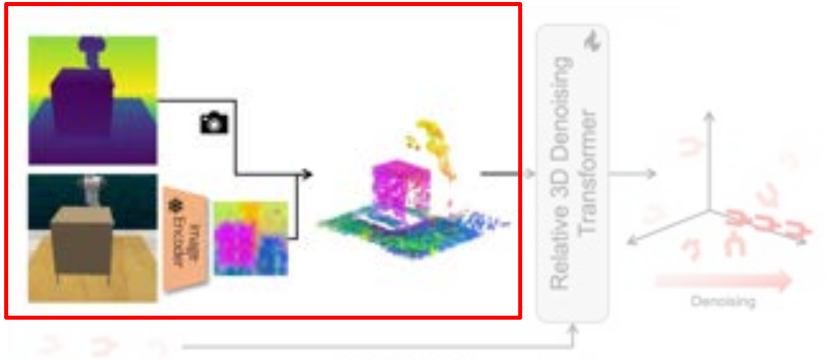
(d) 3D Diffuser Actor



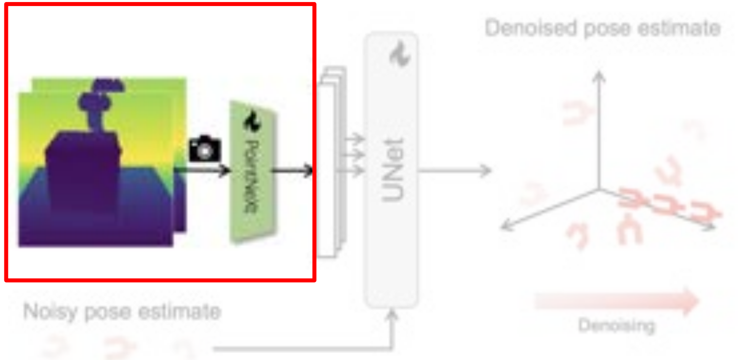
(c) 3D Diffusion Policy

3D Diffuser Actor: Policy Diffusion with 3D Scene Representations: Ke et al. 2024
3D Diffusion Policy: Generalizable Visuomotor Policy Learning via Simple 3D Representations: Ze et al. 2024

Is 3D Necessary?: A case study in manipulation



(d) 3D Diffuser Actor

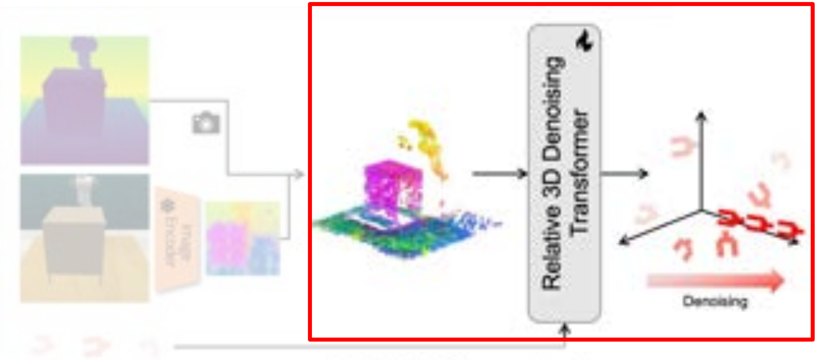


(c) 3D Diffusion Policy

Lifting to 3D

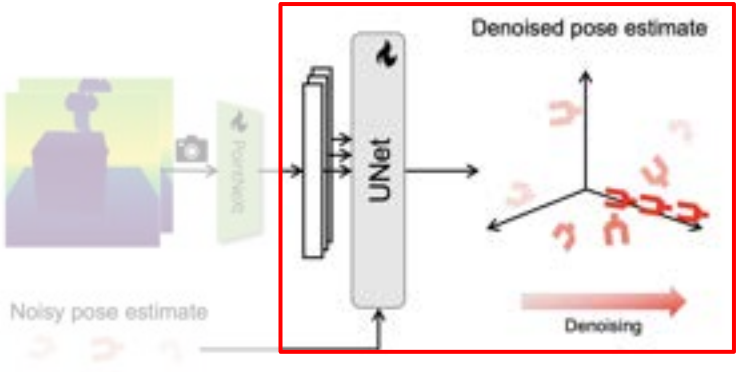
3D Diffuser Actor: Policy Diffusion with 3D Scene Representations: Ke et al. 2024
3D Diffusion Policy: Generalizable Visuomotor Policy Learning via Simple 3D Representations: Ze et al. 2024

Is 3D Necessary?: A case study in manipulation



(d) 3D Diffuser Actor

Prediction over
3D tokens.



(c) 3D Diffusion Policy

Prediction over
1D pooled tokens.

3D Diffuser Actor: Policy Diffusion with 3D Scene Representations: Ke et al. 2024
3D Diffusion Policy: Generalizable Visuomotor Policy Learning via Simple 3D Representations: Ze et al. 2024

Is 3D Necessary?: A case study in manipulation

	Train episodes	Task completed in a row 1	2	3	4	5	Avg. Len
3D Diffusion Policy [22]	Lang	$28.7_{\pm 0.4}$	$2.7_{\pm 0.4}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.31_{\pm 0.04}$
MCIL [71]	All	30.4	1.3	0.2	0.0	0.0	0.31
HULC [70]	All	41.8	16.5	5.7	1.9	1.1	0.67
RT-1 [49]	Lang	53.3	22.2	9.4	3.8	1.3	0.90
ChainedDiffuser [21] (60 keyposes)	Lang	$49.9_{\pm 0.01}$	$21.1_{\pm 0.01}$	$8.0_{\pm 0.01}$	$3.5_{\pm 0.0}$	$1.5_{\pm 0.0}$	$0.84_{\pm 0.02}$
RoboFlamingo [72]	Lang	82.4	61.9	46.6	33.1	23.5	2.48
SuSIE [45]	All	87.0	69.0	49.0	38.0	26.0	2.69
GR-1 [58]	Lang	85.4	71.2	59.6	49.7	40.1	3.06
3D Diffuser Actor (ours)	Lang	$\mathbf{93.8}_{\pm 0.01}$	$\mathbf{80.3}_{\pm 0.0}$	$\mathbf{66.2}_{\pm 0.01}$	$\mathbf{53.3}_{\pm 0.02}$	$\mathbf{41.2}_{\pm 0.01}$	$\mathbf{3.35}_{\pm 0.04}$

Table 4: **Zero-shot long-horizon evaluation on CALVIN** on 3 random seeds.

3D Diffuser Actor: Policy Diffusion with 3D Scene Representations: Ke et al. 2024
3D Diffusion Policy: Generalizable Visuomotor Policy Learning via Simple 3D Representations: Ze et al. 2024

Is 3D Necessary?: A case study in manipulation

	Train episodes	Task completed in a row					
		1	2	3	4	5	Avg. Len
3D Diffusion Policy [22]	Lang	28.7 ±0.4	2.7 ±0.4	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.31 ±0.04
MCIL [71]	All	30.4	1.3	0.2	0.0	0.0	0.31
HULC [70]	All	41.8	16.5	5.7	1.9	1.1	0.67
RT-1 [49]	Lang	53.3	22.2	9.4	3.8	1.3	0.90
ChainedDiffuser [21] (60 keyposes)	Lang	49.9 ±0.01	21.1 ±0.01	8.0 ±0.01	3.5 ±0.0	1.5 ±0.0	0.84 ±0.02
RoboFlamingo [72]	Lang	82.4	61.9	46.6	33.1	23.5	2.48
SuSIE [45]	All	87.0	69.0	49.0	38.0	26.0	2.69
GR-1 [58]	Lang	85.4	71.2	59.6	49.7	40.1	3.06
3D Diffuser Actor (ours)	Lang	93.8 ±0.01	80.3 ±0.0	66.2 ±0.01	53.3 ±0.02	41.2 ±0.01	3.35 ±0.04

Table 4: Zero-shot l[illegible] on 3 random seeds.

3D Diffuser Actor significantly better than 3D Diffusion Policy!

3D Diffuser Actor: Policy Diffusion with 3D Scene Representations: Ke et al. 2024
3D Diffusion Policy: Generalizable Visuomotor Policy Learning via Simple 3D Representations: Ze et al. 2024

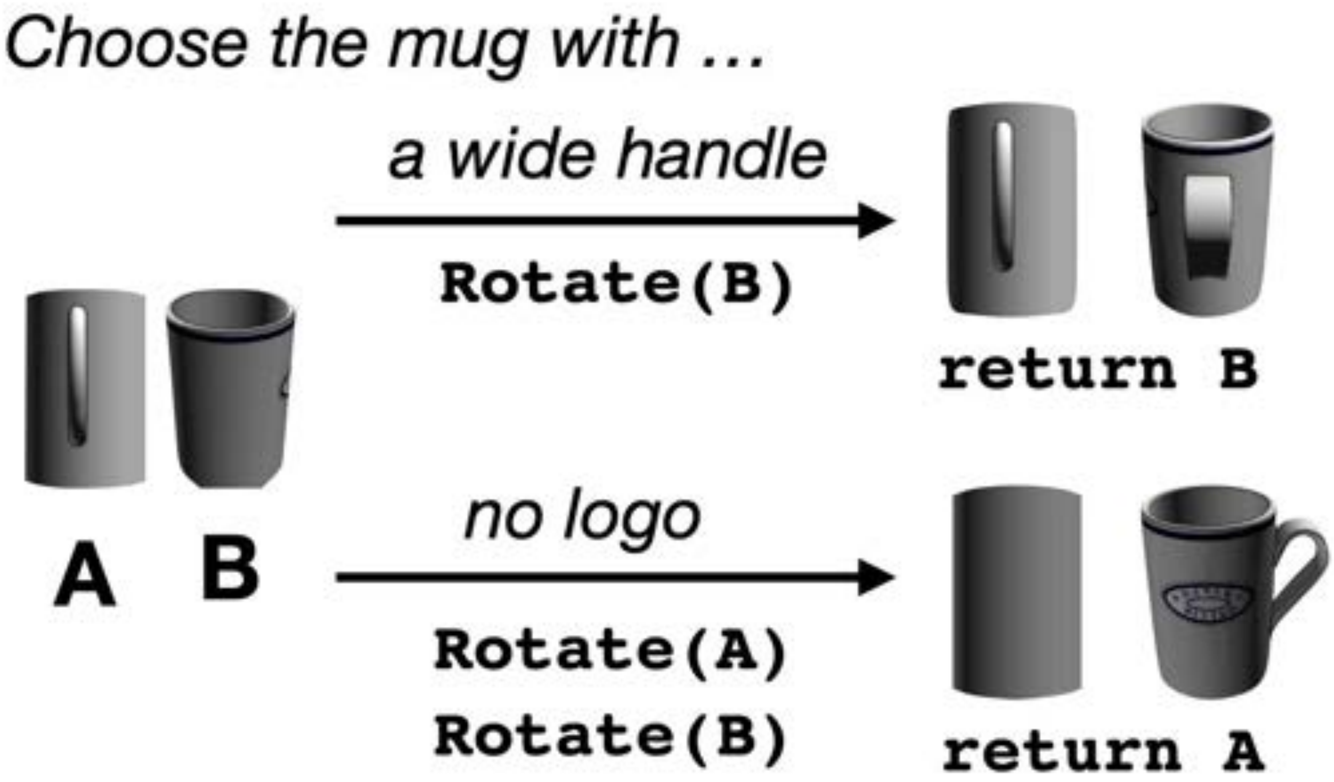
Is 3D Necessary?: A case study in manipulation

	Avg. Success.
2D Diffuser Actor	47.0
3D Diffuser Actor w/o RelAttn.	71.3
3D Diffuser Actor (ours)	**81.3**

Ablations show that 3D encoding is *critical* for performance.

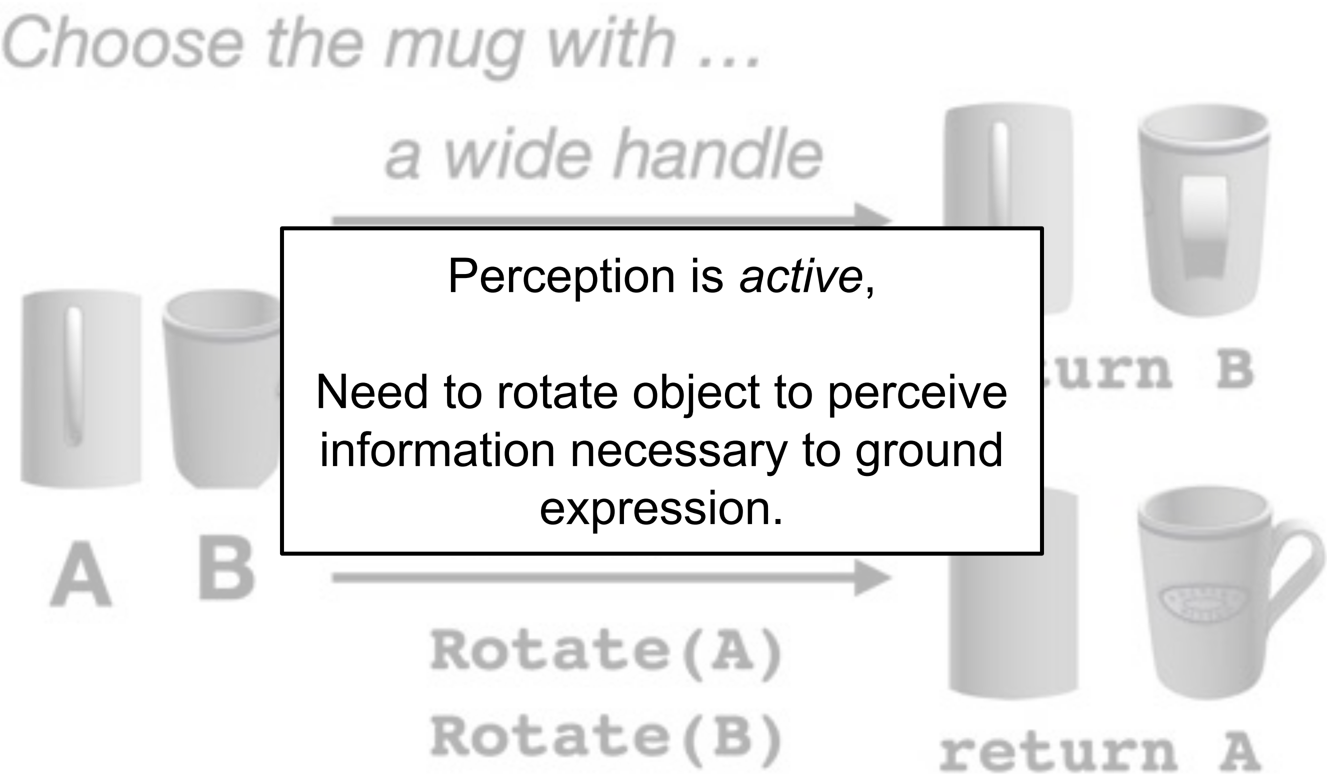
3D Diffuser Actor: Policy Diffusion with 3D Scene Representations: Ke et al. 2024

Is 3D Necessary?: A case study in *grounding*



Language Grounding with 3D Objects : Thomason et al. 2021

Is 3D Necessary?: A case study in *grounding*

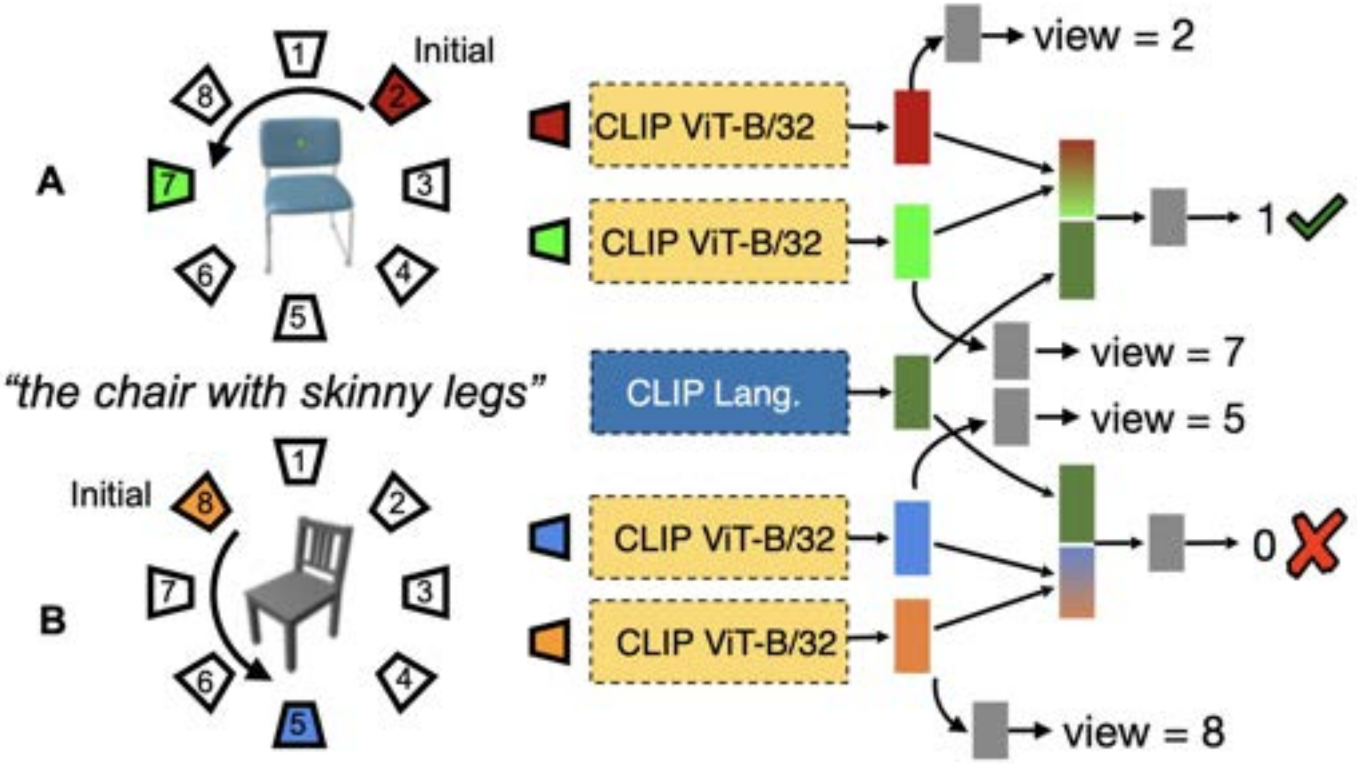


Perception is *active*,

Need to rotate object to perceive information necessary to ground expression.

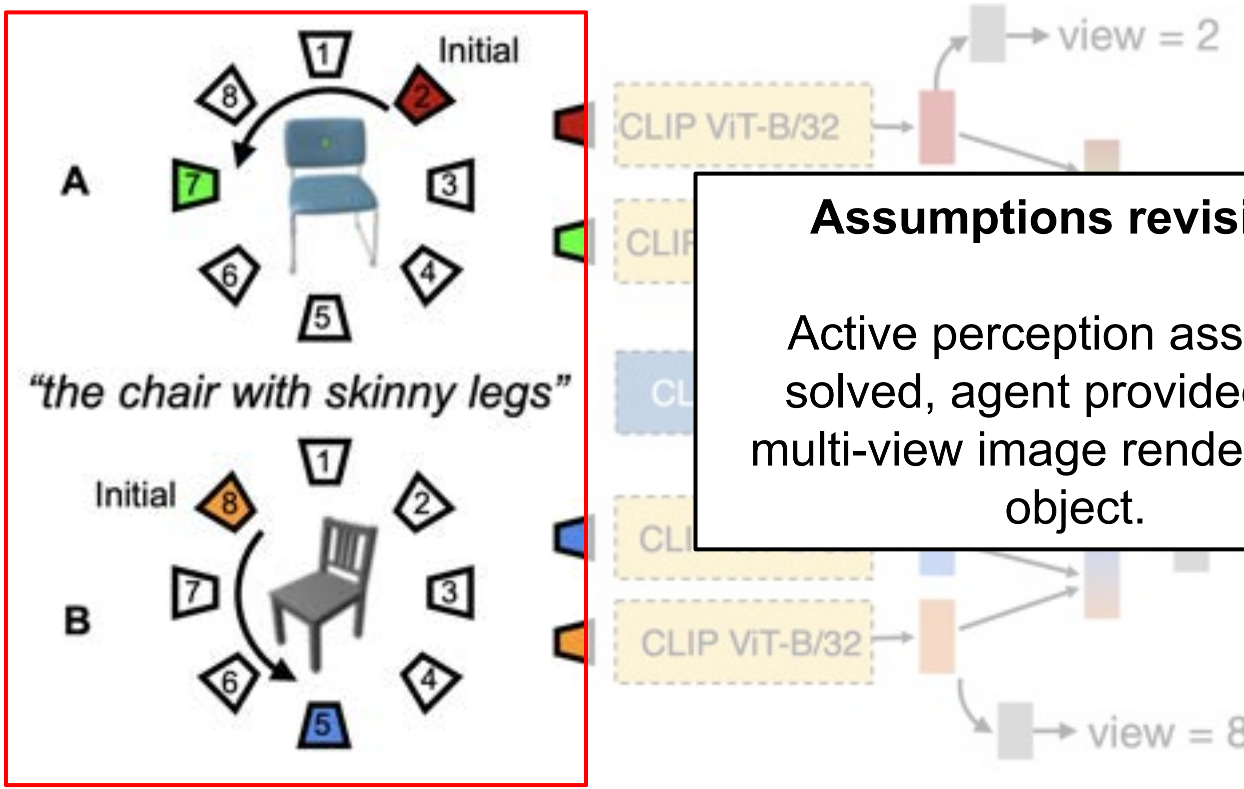
Language Grounding with 3D Objects : Thomason et al. 2021

Is 3D Necessary?: A case study in *grounding*



Language Grounding with 3D Objects : Thomason et al. 2021

Is 3D Necessary?: A case study in *grounding*

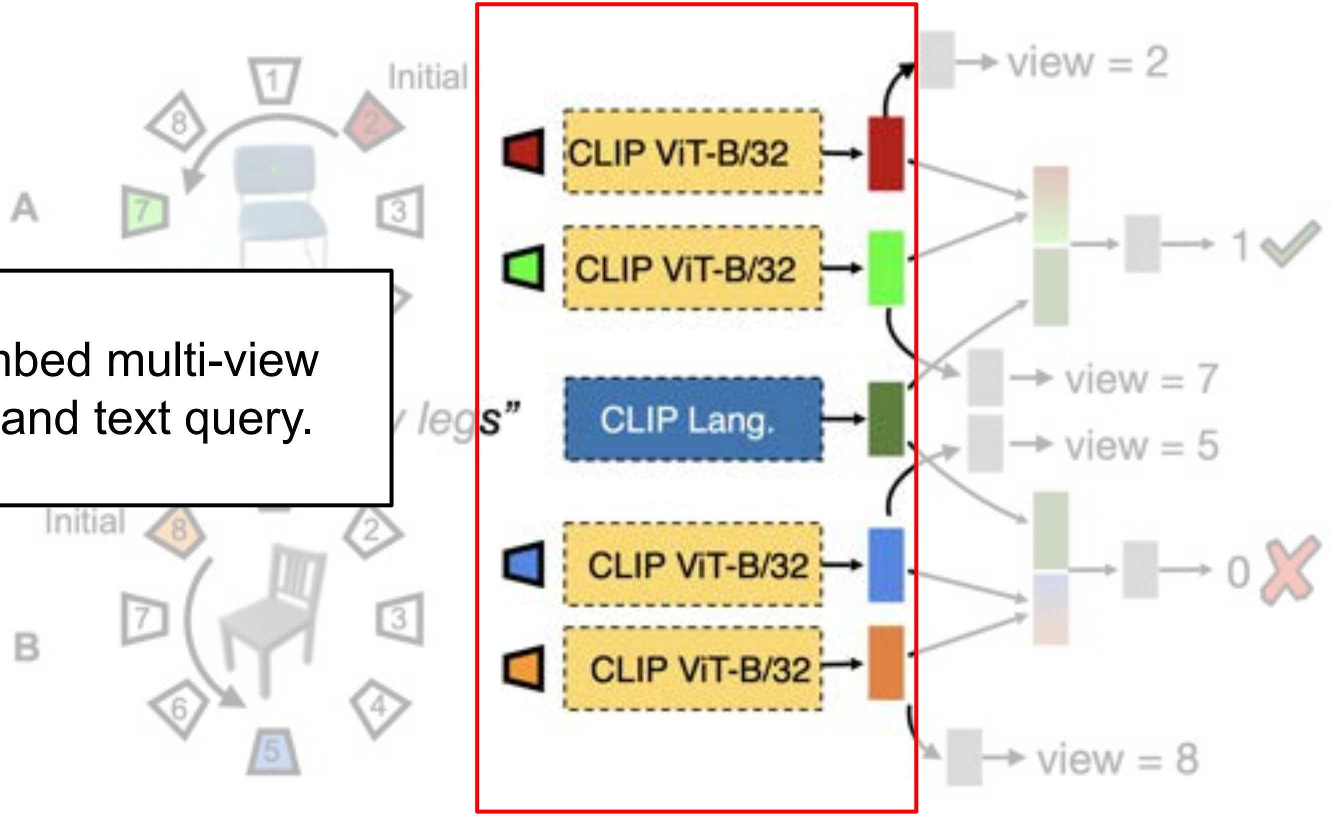


Assumptions revisited:

Active perception assumed solved, agent provided with multi-view image renderings of object.

Language Grounding with 3D Objects : Thomason et al. 2021

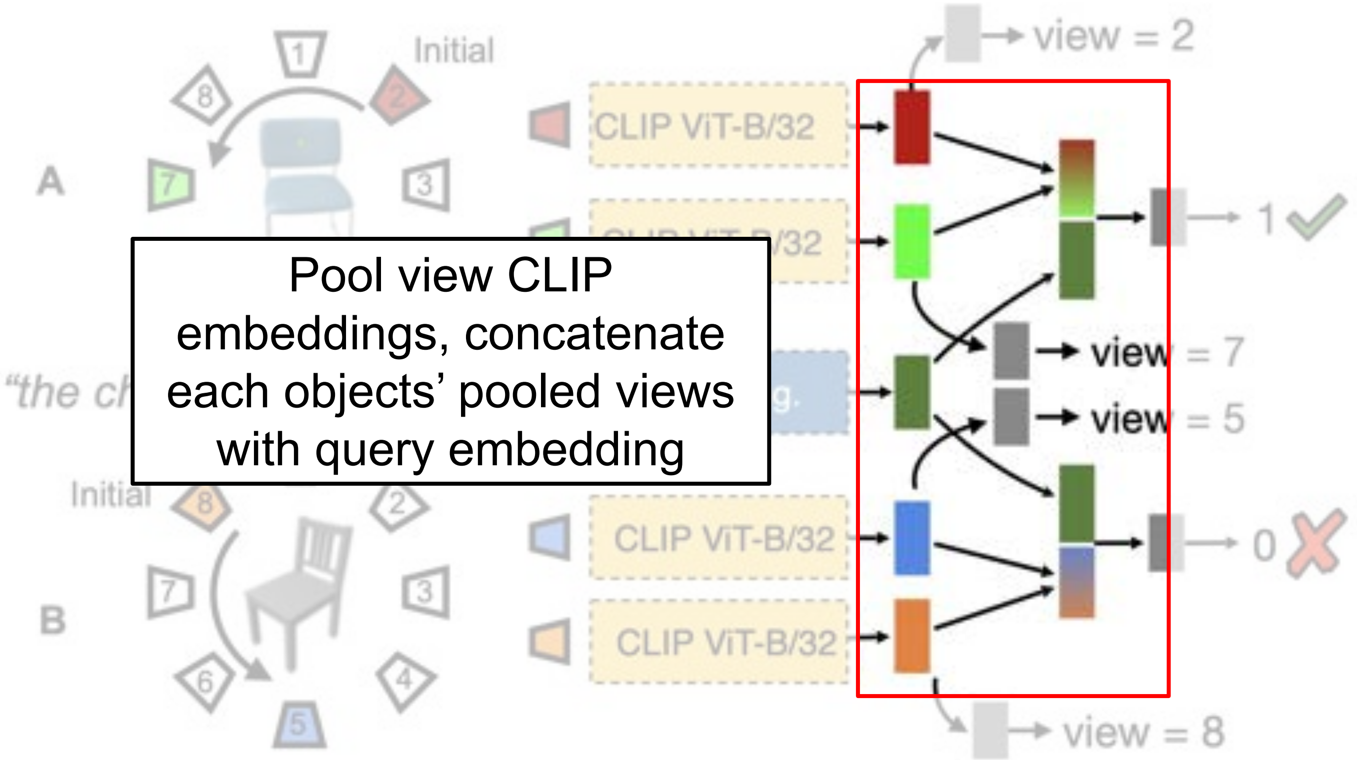
Is 3D Necessary?: A case study in *grounding*



CLIP embed multi-view images and text query.

Language Grounding with 3D Objects : Thomason et al. 2021

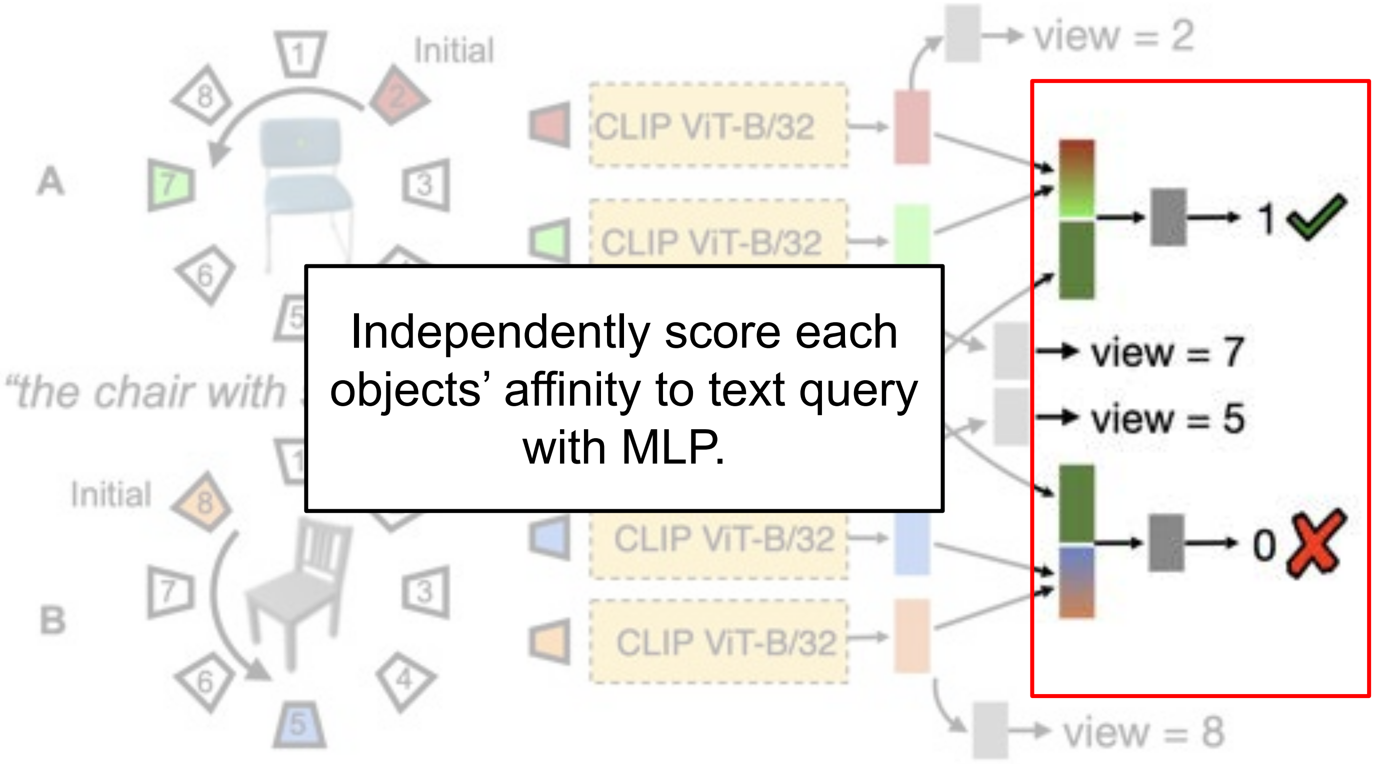
Is 3D Necessary?: A case study in *grounding*



Pool view CLIP embeddings, concatenate each objects' pooled views with query embedding

Language Grounding with 3D Objects : Thomason et al. 2021

Is 3D Necessary?: A case study in *grounding*



Language Grounding with 3D Objects : Thomason et al. 2021

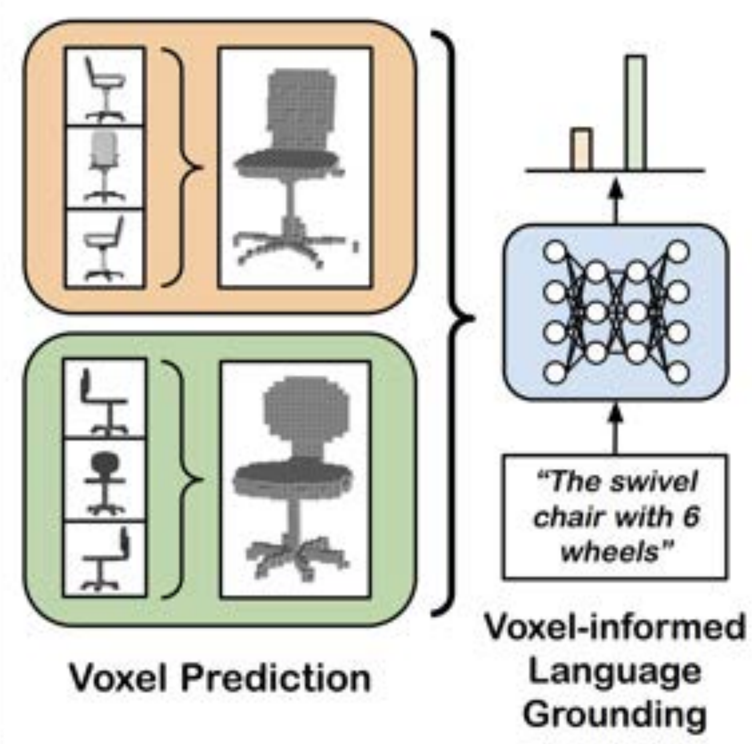
Is 3D Necessary?: A case study in *grounding*

Model	Views	Validation Visual⊂	Validation Blind⊂	Validation All	Test Visual⊂	Test Blind⊂	Test All
ViLBERT	All	**89.5**	**76.6**	**83.1**	80.2	**73.0**	**76.6**
CLIP	All	83.7 ±0.0	65.2 ±0.0	74.5 ±0.0	80.0 ±0.0	61.4 ±0.0	70.9 ±0.0
MATCH	All	89.2 ±0.9	75.2 ±0.7	82.2 ±0.4	**83.9** ±0.5	68.7 ±0.9	76.5 ±0.5
CLIP	Single	79.0 ±0.0	63.0 ±0.0	71.1 ±0.0	74.0 ±0.0	59.7 ±0.0	67.0 ±0.0
MATCH	Single	**88.4** ±0.4	**73.3** ±0.6	**80.9** ±0.4	**83.2** ±0.3	**68.0** ±0.5	**75.8** ±0.3
CLIP	Two	81.0 ±0.0	64.1 ±0.0	72.6 ±0.0	76.0 ±0.0	60.8 ±0.0	68.6 ±0.0
MATCH	Two	89.2 ±0.6	74.4 ±0.7	81.8 ±0.4	83.7 ±0.4	68.7 ±0.5	76.4 ±0.4
LAGOR	Two	**89.8** ±0.4	**75.3** ±0.7	**82.6** ±0.4	**84.3** ±0.4	**69.4** ±0.5	**77.0** ±0.5
Human (U)	All	94.0	90.6	92.3	93.4	88.9	91.2
Human (M)	All	100.0	100.0	100.0	100.0	100.0	100.0

Fares okay against human performance

Language Grounding with 3D Objects : Thomason et al. 2021

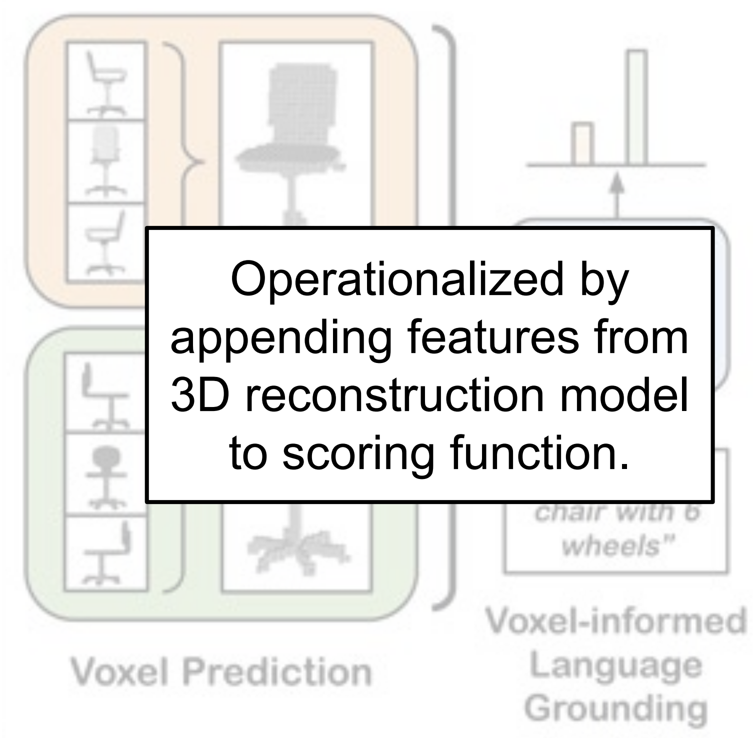
Is 3D Necessary?: A case study in *grounding*



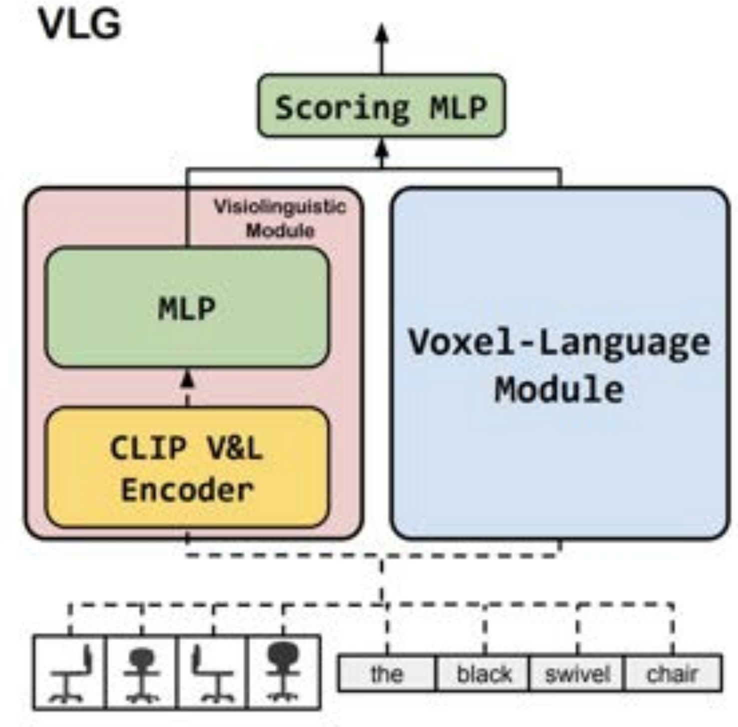
Idea: Use *explicit* 3D representation as anchor for grounding.

Voxel-informed Language Grounding: Corona et al. 2022

Is 3D Necessary?: A case study in *grounding*



Operationalized by appending features from 3D reconstruction model to scoring function.



Voxel-informed Language Grounding: Corona et al. 2022

Is 3D Necessary?: A case study in *grounding*

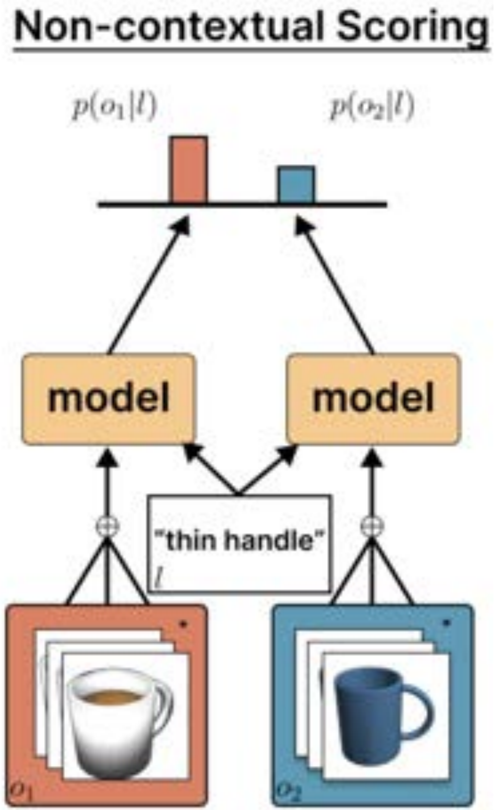
Improves performance over the standard model!

Did 3D *intrinsically* help?

	VALI			TEST		
Model	Visual	Bl			Blind	All
ViLBERT	89.5	76.6	83.1	80.2	73.0	76.6
MATCH	89.2 (0.9)	75.2 (0.7)	82.2 (0.4)	83.9 (0.5)	68.7 (0.9)	76.5 (0.5)
MATCH*	90.6 (0.4)	75.7 (1.2)	83.2 (0.8)	-	-	-
LAGOR	89.8 (0.4)	75.3 (0.7)	82.6 (0.4)	84.3 (0.4)	69.4 (0.5)	77.0 (0.5)
LAGOR*	89.8 (0.5)	75.0 (0.4)	82.5 (0.1)	-	-	-
VLG (Ours)	**91.2** (0.4)	**78.4**†(0.7)	**84.9**†(0.3)	**86.0**	71.7	**79.0**

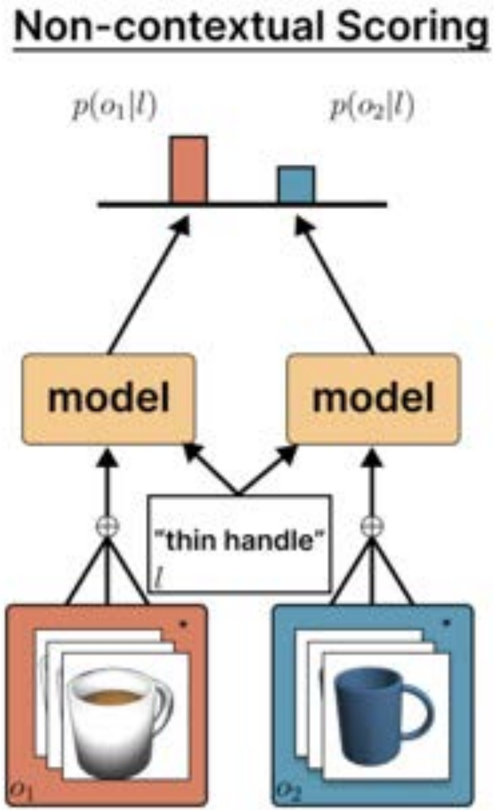
Voxel-informed Language Grounding: Corona et al. 2022

Is 3D Necessary?: A case study in *grounding*



Which One? Leveraging Context Between Objects and Multiple Views for Language Grounding: Mitra et al. 2024

Is 3D Necessary?: A case study in *grounding*



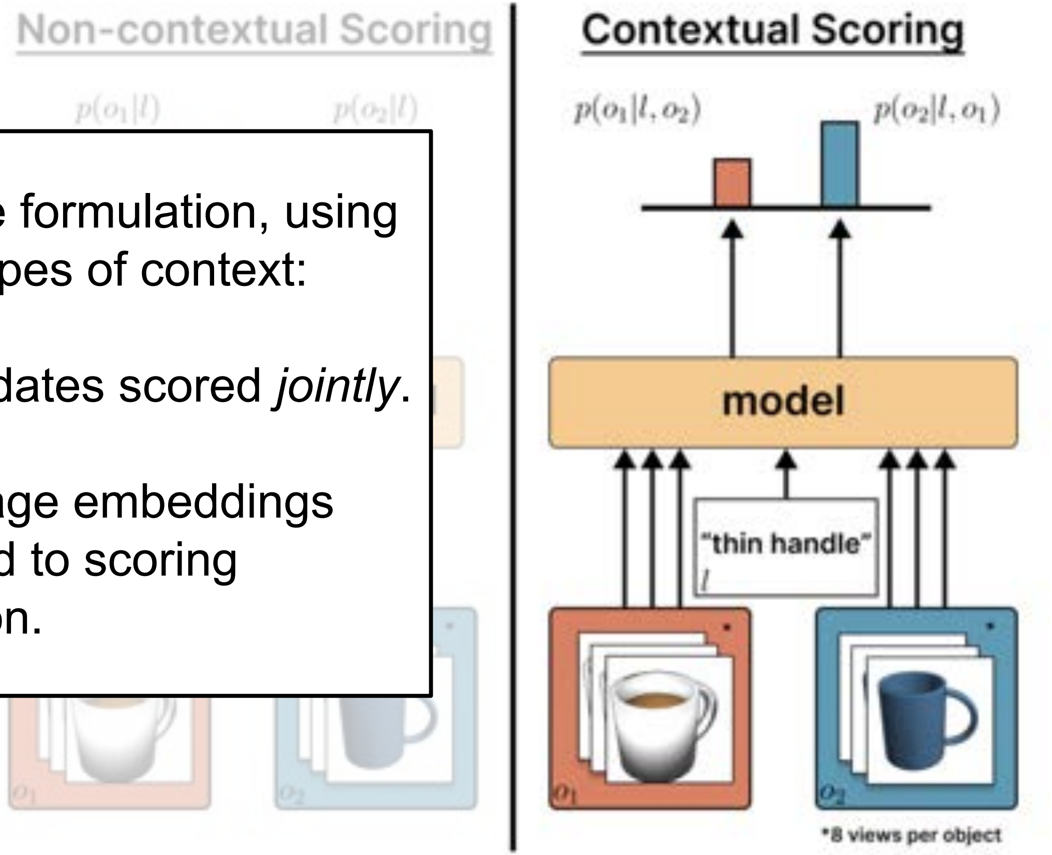
2 key structures of prior work:

1. Candidates scored *independently*.
2. Multi-view images *aggregated before* passing to scoring function.

Does this make sense for *comparative* referential expressions?

Which One? Leveraging Context Between Objects and Multiple Views for Language Grounding: Mitra et al. 2024

Is 3D Necessary?: A case study in *grounding*



Alternative formulation, using two types of context:

1. Candidates scored *jointly*.

2. All image embeddings passed to scoring function.

Which One? Leveraging Context Between Objects and Multiple Views for Language Grounding: Mitra et al. 2024

Is 3D Necessary?: A case study in *grounding*

Model	Considers Both Objects	Lang Attends to Ind. Views	VALIDATION ACC. Visual	Blind	All	TEST ACC. Visual	Blind	All
Human (U)	✓	✓	94.0	90.6	92.3	93.4	88.9	91.2
ViLBERT	✗	✓	89.5	76.6	83.1	80.2	73.0	76.6
MATCH	✗	✗	89.2(0.9)	75.2(0.7)	82.2(0.4)	83.9(0.5)	68.7(0.9)	76.5(0.5)
LAGOR	✗	✗	89.8(0.4)	75.3(0.7)	82.6(0.4)	84.3(0.4)	69.4(0.5)	77.0(0.5)
VLG	✗	~	**91.2(0.4)**	**78.4(0.7)**	**84.9(0.4)**	**86.0**	**71.7**	**79.0**
MAGiC	✓	✓	**92.1(0.4)**	**81.3(0.9)**	**86.8(0.5)**	**87.7**	**75.4**	**81.7**

Does even better, without requiring 3D!

Which One? Leveraging Context Between Objects and Multiple Views for Language Grounding: Mitra et al. 2024

Is 3D Necessary?: A case study in *grounding*

	VALIDATION ACC.		
Model	Visual	Blind	All
MATCH	90.6(0.5)	77.0(0.7)	83.9(0.4)
+ obj. context	90.5(0.5)	76.8(0.6)	83.7(0.3)
MAGiC	**92.1(0.4)**	**81.3(0.9)**	**86.8(0.5)**
- obj. context	91.1(0.5)	79.4(1.1)	85.3(0.5)
- mv. context	91.0(0.6)	79.5(0.8)	85.3(0.4)
- both contexts	90.5(0.6)	78.2(1.2)	84.4(0.6)

Which One? Leveraging Context Between Objects and Multiple Views for Language Grounding: Mitra et al. 2024

Is 3D Necessary?: A case study in *grounding*

	VALIDATION ACC.		
Model	[illegible]	[illegible]	All
MATCH	[illegible]	[illegible] 7)	83.9(0.4)
+ obj. context	[illegible]	[illegible] 6)	83.7(0.3)
MAGiC	[illegible]	[illegible] 9)	**86.8(0.5)**
- obj. context	[illegible]	[illegible] 1)	85.3(0.5)
- mv. context	[illegible]	[illegible] 8)	85.3(0.4)
- both contexts	90.5(0.6)	78.2(1.2)	84.4(0.6)

Ablation takeaways:

- *Both* contexts critical for performance gain.
- Transformer provides no significant gains over MLP without context addition.

Which One? Leveraging Context Between Objects and Multiple Views for Language Grounding: Mitra et al. 2024

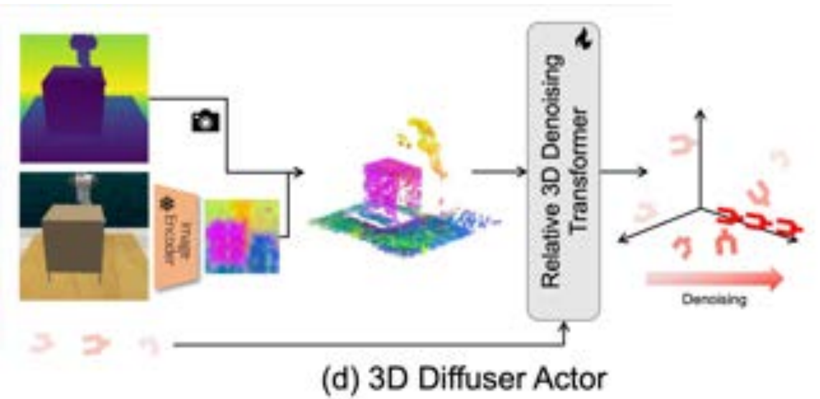
Is 3D Necessary?: A case study in *grounding*

	VALIDATION ACC.		
Model	Visual	Blind	All
MATCH	[illegible]	[illegible]	83.9(0.4)
+ obj. conte[illegible]	[illegible]	[illegible]	83.7(0.3)
MAGiC	[illegible]	[illegible]	**86.8(0.5)**
- obj. conte[illegible]	[illegible]	[illegible]	85.3(0.5)
- mv. context	91.0(0.6)	79.5(0.8)	85.3(0.4)
- both contexts	90.5(0.6)	78.2(1.2)	84.4(0.6)

Implication: 3D likely helped before because it provided information that was lost through *multi-view aggregation*.

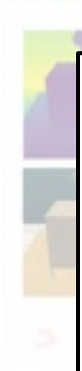
Which One? Leveraging Context Between Objects and Multiple Views for Language Grounding: Mitra et al. 2024

Why did 3D help one and not the other?



Which One? Leveraging Context Between Objects and Multiple Views for Language Grounding: Mitra et al. 2024

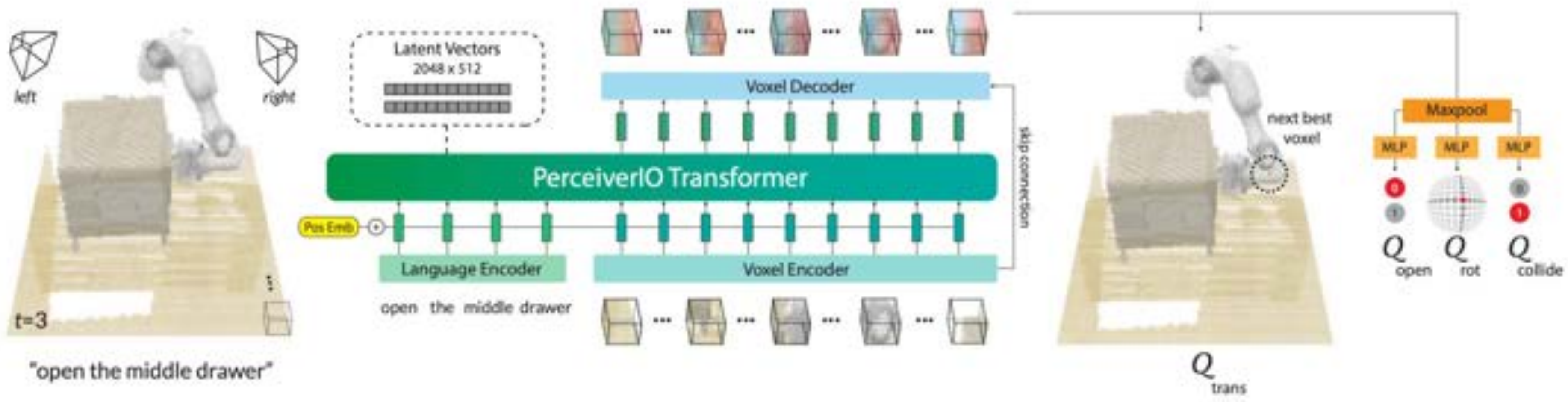
Why did 3D help one and not the other?



Active perception, i.e. *manipulation* of object was abstracted away from grounding task!

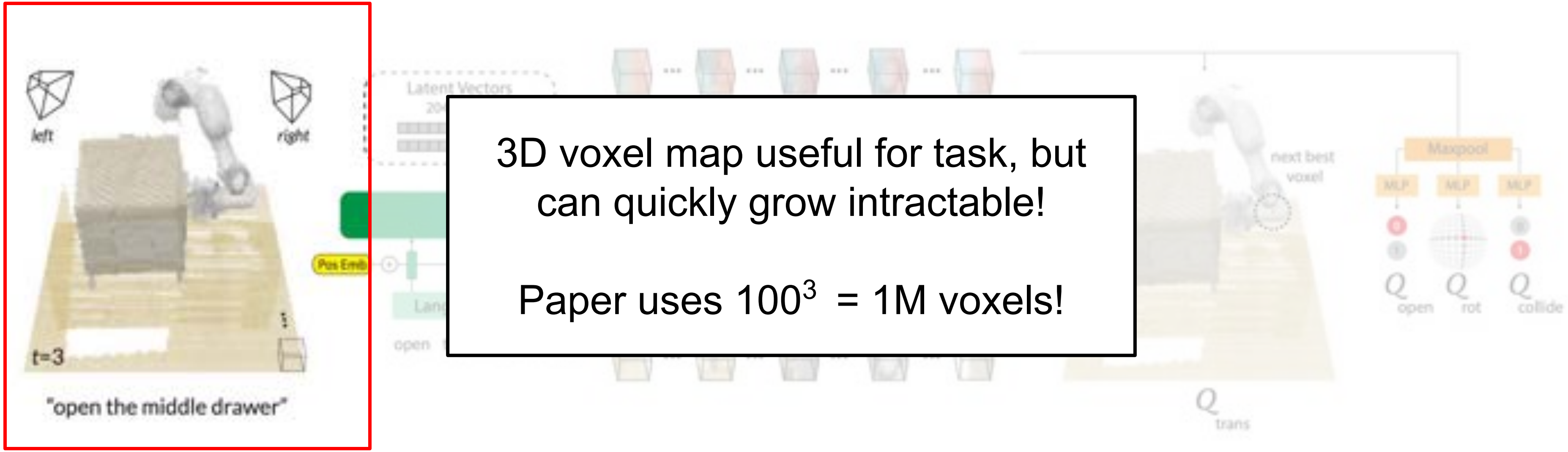
Grounding submodule did not need 3D, but end-to-end pipeline likely would!

Gallery of 3D Drawbacks



Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation: Shridhar et al. 2022

Gallery of 3D Drawbacks



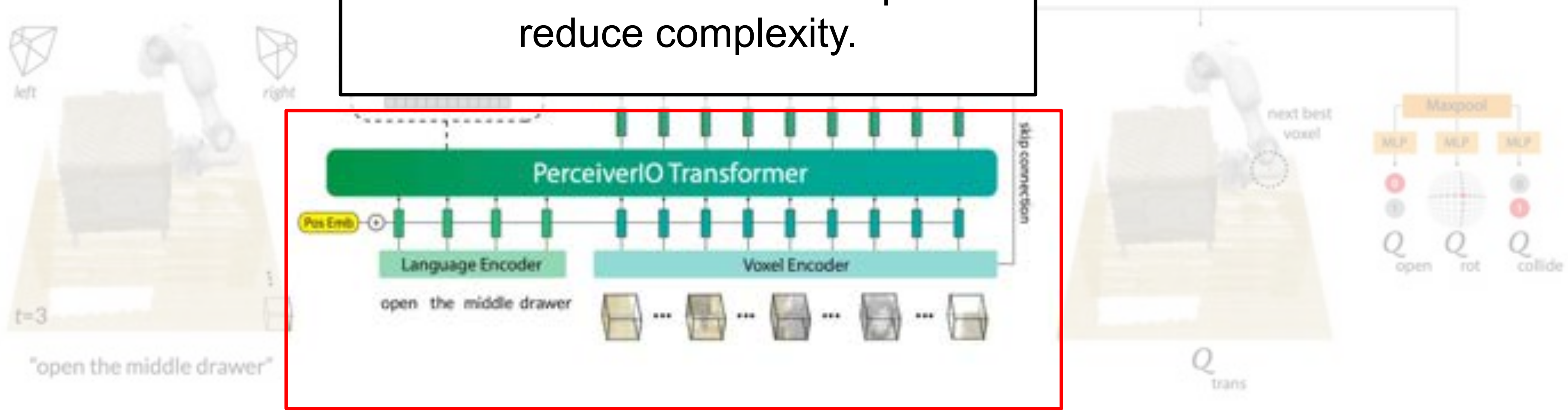
3D voxel map useful for task, but can quickly grow intractable!

Paper uses 100^3 = 1M voxels!

Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation: Shridhar et al. 2022

Gallery of 3D Drawbacks

Uses PercieverIO to subsample and reduce complexity.

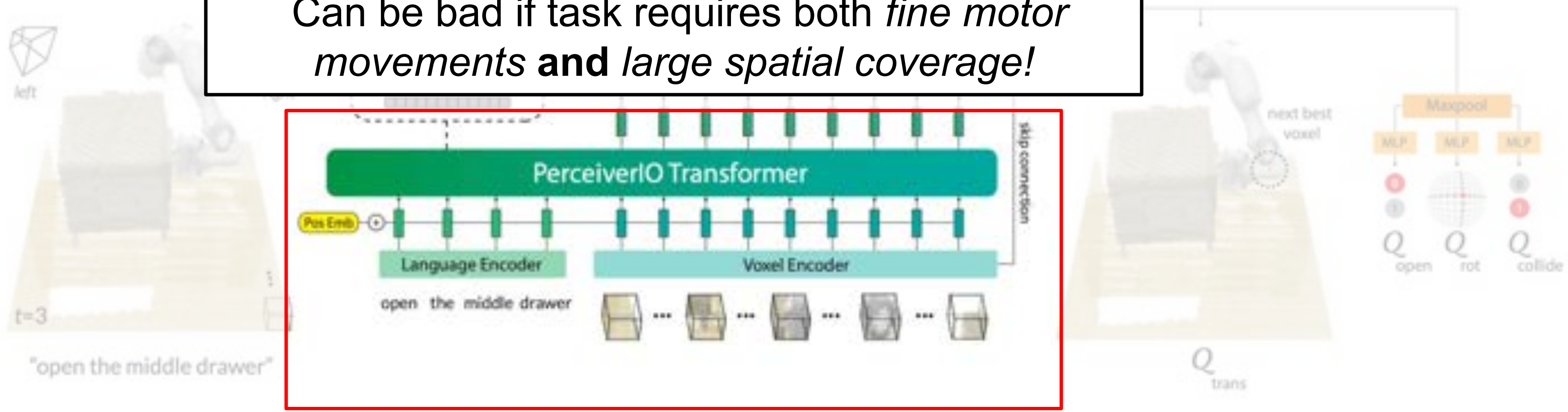


Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation: Shridhar et al. 2022

Gallery of 3D Drawbacks

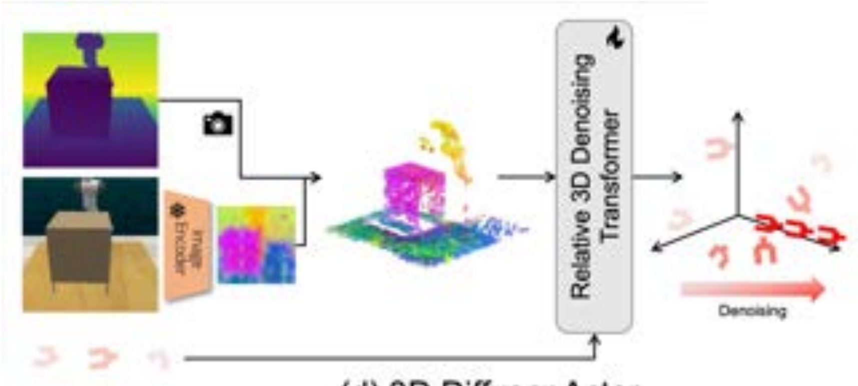
Like in video, implies tradeoff between *resolution/coverage* and *tractability*.

Can be bad if task requires both *fine motor movements* **and** *large spatial coverage!*

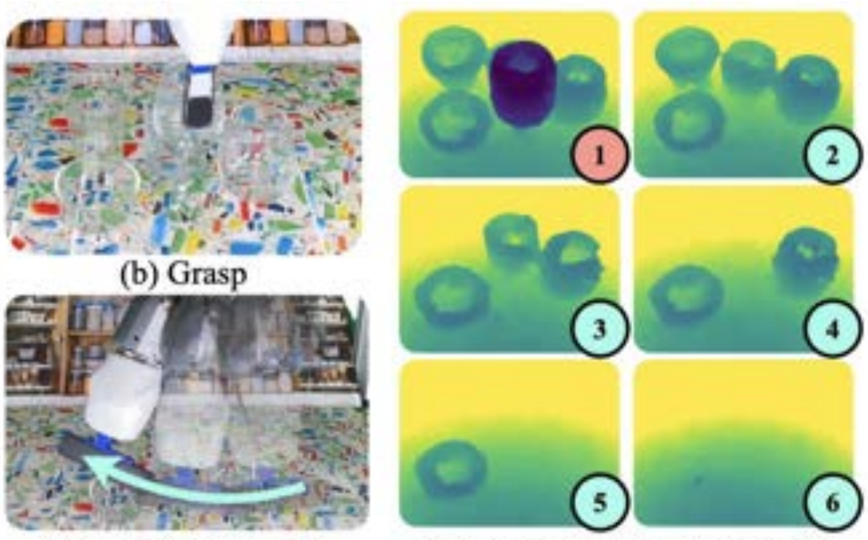


Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation: Shridhar et al. 2022

Gallery of 3D Drawbacks



(d) 3D Diffuser Actor



(b) Grasp

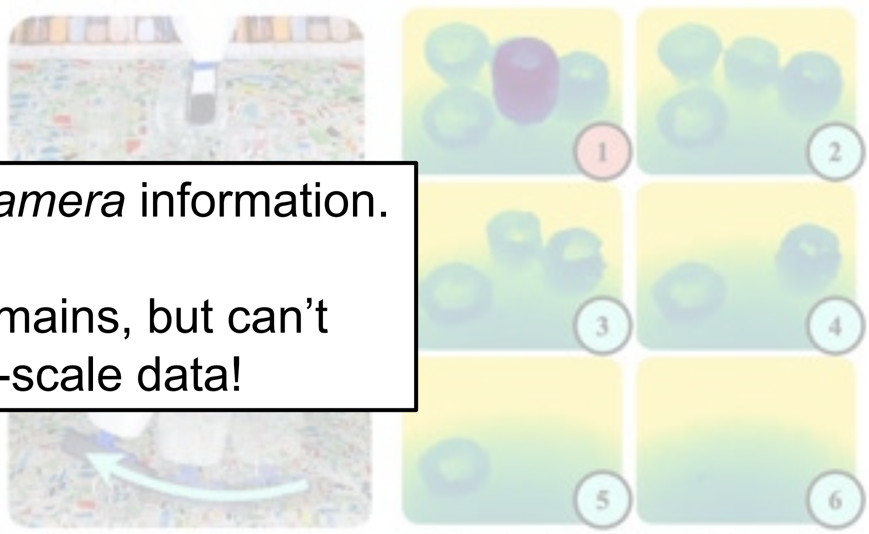
(c) Update Trajectory

(d) Consecutive NeRF Updates

3D Diffuser Actor: Policy Diffusion with 3D Scene Representations: Ke et al. 2024
Evo-NeRF: Evolving NeRF for Sequential Robot Grasping of Transparent Objects: (Kerr et al. 2022)

Gallery of 3D Drawbacks

(d) 3D Diffuser Actor



(c) Update Trajectory (d) Consecutive NeRF Updates

Require *calibrated camera* information.

OK in robotics domains, but can’t leverage web-scale data!

3D Diffuser Actor: Policy Diffusion with 3D Scene Representations: Ke et al. 2024
Evo-NeRF: Evolving NeRF for Sequential Robot Grasping of Transparent Objects: (Kerr et al. 2022)

Gallery of 3D Drawbacks

ShapeNet: An Information-Rich 3D Model Repository: Chang et al. 2015
Objaverse-XL: A Universe of 10M+ 3D Objects: Deitke et al. 2023

Gallery of 3D Drawbacks

50K 3D Objects

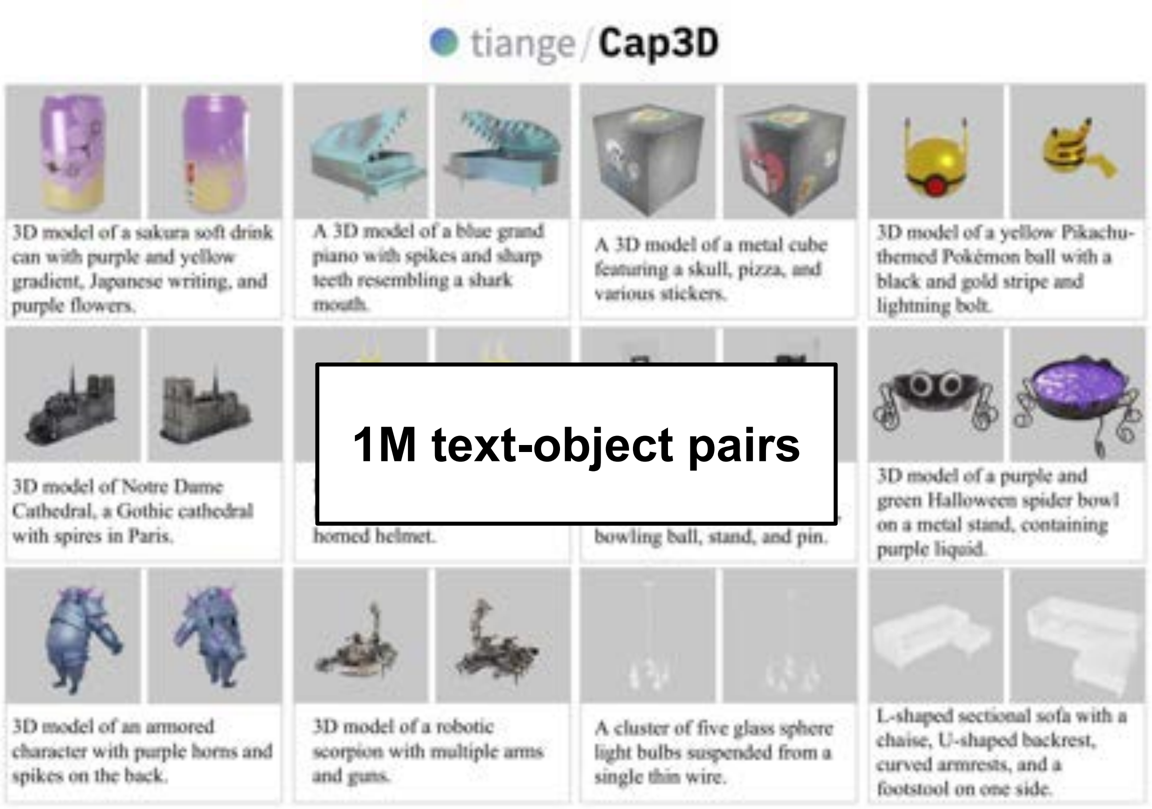
10M 3D Objects

ShapeNet: An Information-Rich 3D Model Repository: Chang et al. 2015
Objaverse-XL: A Universe of 10M+ 3D Objects: Deitke et al. 2023

Gallery of 3D Drawbacks

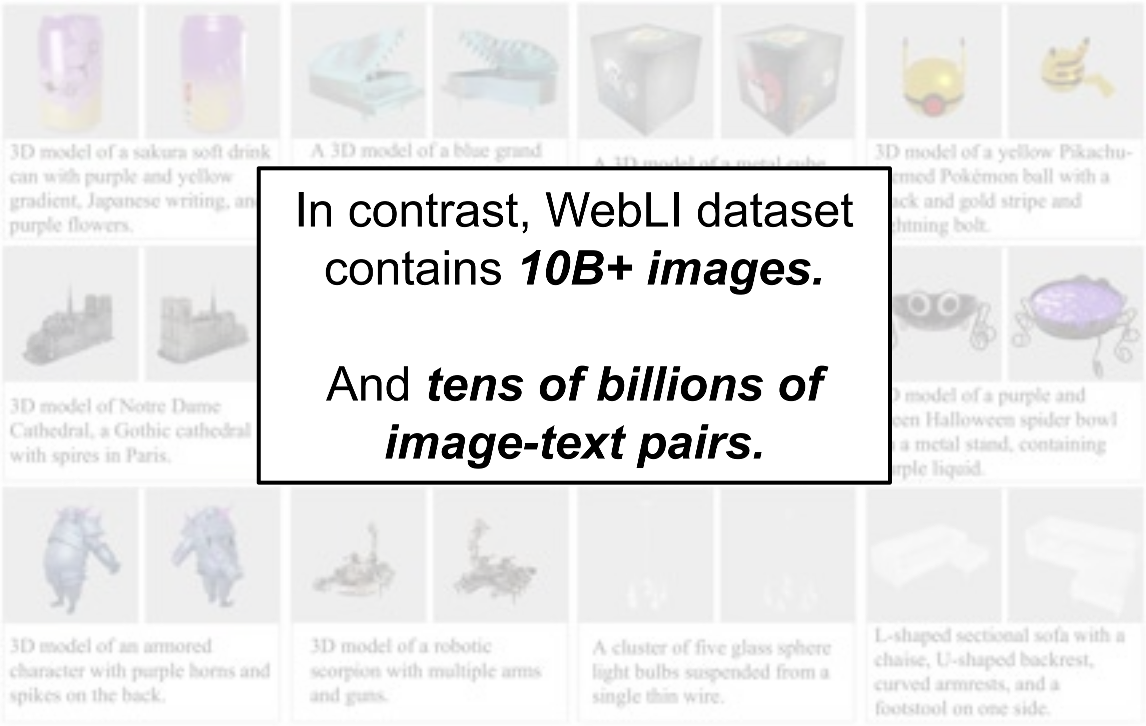
Scalable 3D Captioning with Pretrained Models: Luo et al. 2023

Gallery of 3D Drawbacks



Scalable 3D Captioning with Pretrained Models: Luo et al. 2023

Gallery of 3D Drawbacks



In contrast, WebLI dataset contains ***10B+ images.***

And ***tens of billions of image-text pairs.***

Scalable 3D Captioning with Pretrained Models: Luo et al. 2023

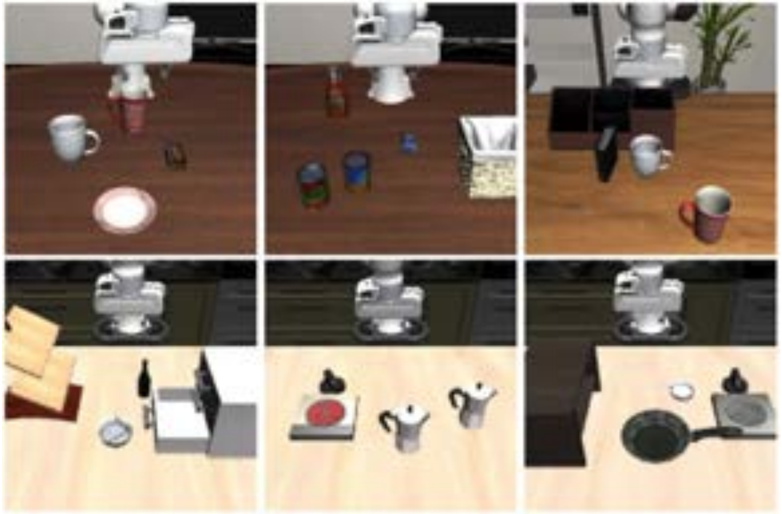
II. Towards Multi-Task Learning (Aka Robot Foundation Models)

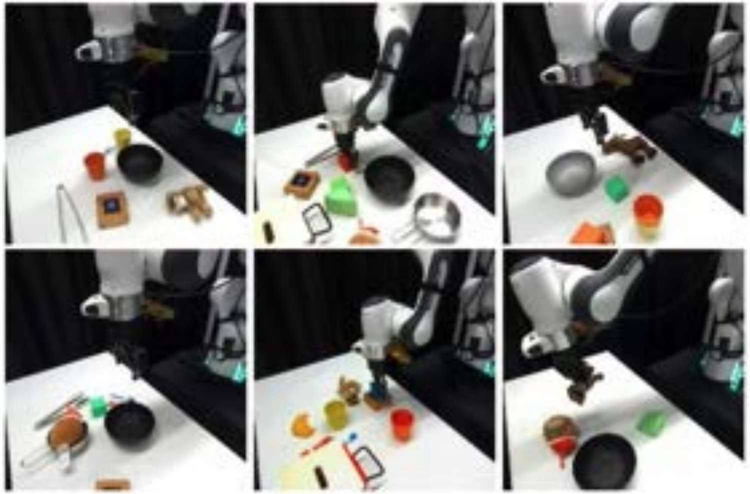
- We have covered many algorithms for learning a single task.
- **What's the challenge of real multi-task learning?**
- Multi-task conditioning?
    - Goal, language, or (?)

Challenges of Multi-task Learning

Definition:

Given a *task condition*, the policy performs the correct task amongst many training tasks.

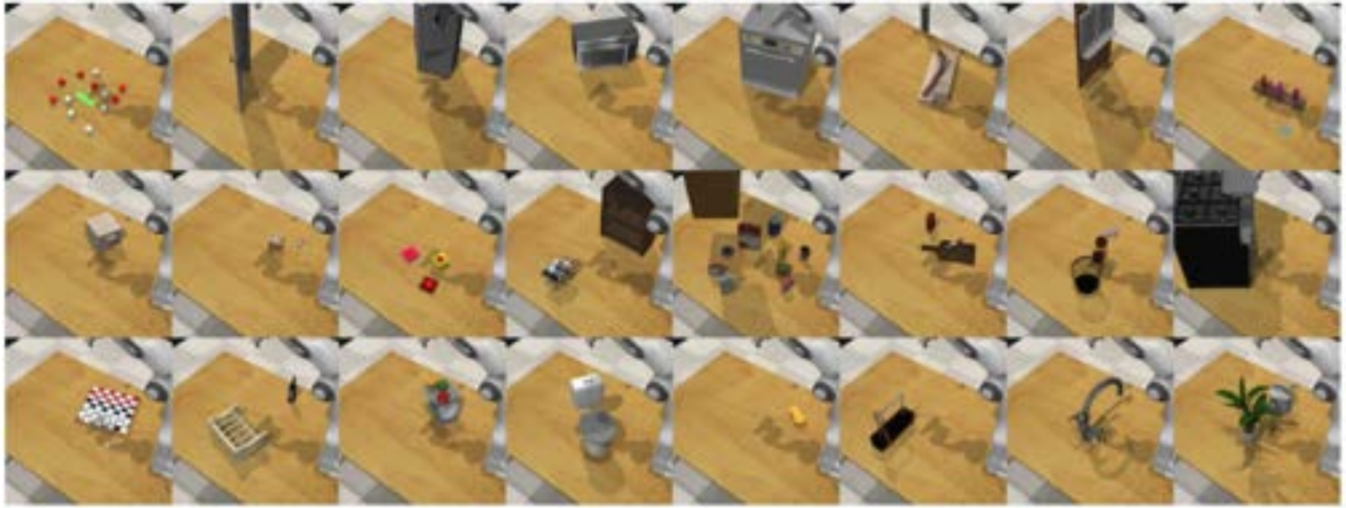




Challenges of Multi-task Learning

It is easy to overfit to a certain task, especially in simulation!

Why? There's just one object. No need to generalize!



James, S., Ma, Z., Arrojo, D. R., & Davison, A. J. (2020). Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, *5*(2), 3019-3026.

II. Towards Multi-Task Learning (Aka Robot Foundation Models)

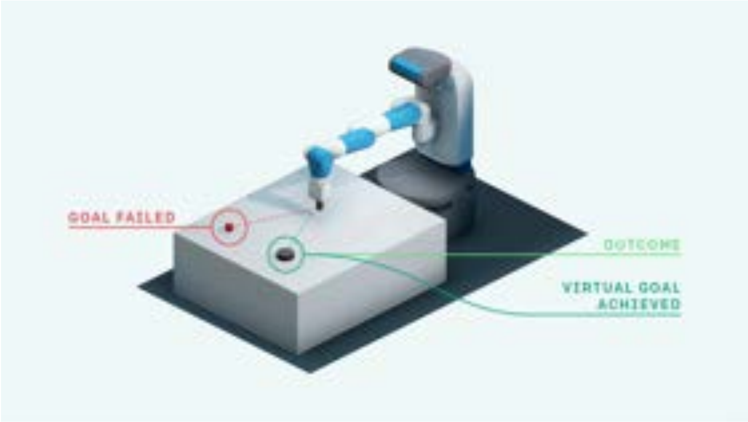
- We have covered many algorithms for learning a single task.
- What’s the challenge of real multi-task learning?
- **Multi-task conditioning?**
  - Goal, language, or (?)

Goal Condition

Goal observation condition: sparsity

1. Goal relabeling to improve data utility:
   a. Hindsight Experience Replay
2. Imagine subgoals for task completion
   a. SuSIE

Hindsight Experience Replay

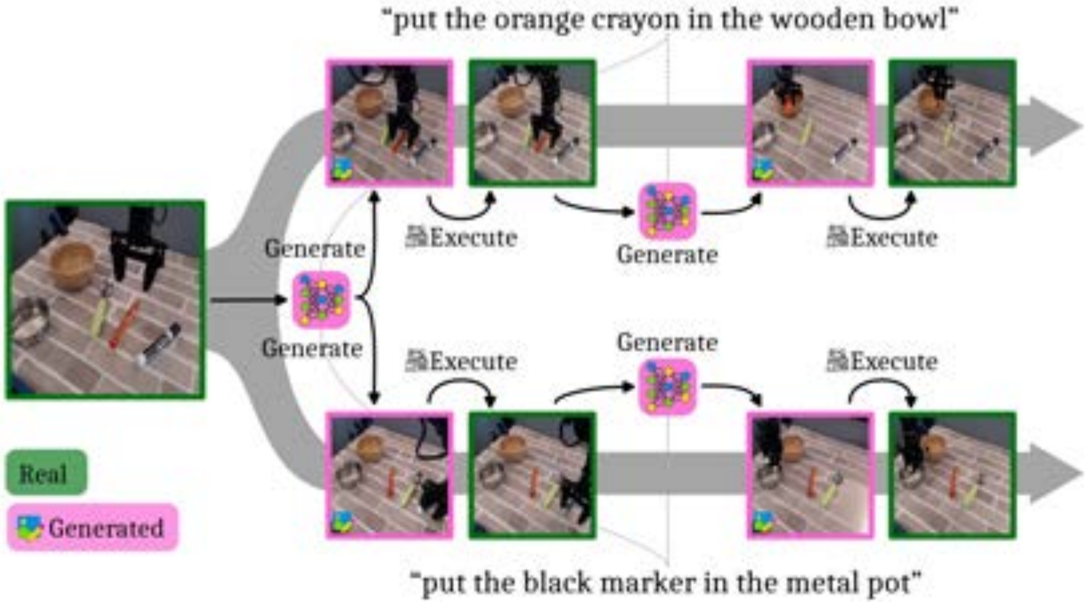


Usable for both RL and BC

Initialize $\mathbb{A}$ ▷ e.g. initialize neural networks
Initialize replay buffer R
for episode $= 1, M$ **do**
    Sample a goal g and an initial state s_0.
    for $t = 0, T-1$ **do**
        Sample an action a_t using the behavioral policy from $\mathbb{A}$:
            $a_t \leftarrow \pi_b(s_t||g)$ ▷ || denotes concatenation
        Execute the action a_t and observe a new state s_{t+1}
    end for
    for $t = 0, T-1$ **do**
        $r_t := r(s_t, a_t, g)$
        Store the transition $(s_t||g, a_t, r_t, s_{t+1}||g)$ in R ▷ standard experience replay
        Sample a set of additional goals for replay $G := \mathbb{S}($**current episode**$)$
        for $g' \in G$ **do**
            $r' := r(s_t, a_t, g')$
            Store the transition $(s_t||g', a_t, r', s_{t+1}||g')$ in R ▷ HER
        end for
    end for
    for $t = 1, N$ **do**
        Sample a minibatch B from the replay buffer R
        Perform one step of optimization using $\mathbb{A}$ and minibatch B
    end for
end for

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., ... & Zaremba, W. (2017). Hindsight experience replay. Advances in neural information processing systems, 30.

SUbgoal Synthesis via Image Editing (SuSIE)



Finetune InstructPix2Pix

"Given language instruction and current observation, predict some image that is k steps away."

Black, K., Nakamoto, M., Atreya, P., Walke, H., Finn, C., Kumar, A., & Levine, S. (2023). Zero-shot robotic manipulation with pretrained image-editing diffusion models. arXiv preprint arXiv:2310.10639.

SUbgoal Synthesis via Image Editing (SuSIE)

	Task	LCBC	MOO	UniPi	RT-2-X	SuSIE (Ours)
Scene A	Eggplant on plate	0.9	0.4	0.0	0.3	**1.0**
	Carrot on plate	0.4	0.3	0.0	0.6	**0.9**
	Eggplant in pot	0.6	**0.7**	0.0	0.4	**0.7**
	Average	0.63	0.47	0.0	0.43	**0.87**
Scene B	Bell pepper in pot	0.1	0.0	0.0	0.0	**0.5**
	Bell pepper in bowl	0.3	0.1	0.1	0.0	**0.5**
	Average	0.20	0.05	0.05	0.00	**0.50**
Scene C	Toothpaste in bowl	0.0	0.0	0.0	0.5	**0.6**
	Crayon in bowl	0.0	0.0	0.0	0.9	**1.0**
	Spoon in bowl	0.1	0.3	0.1	0.7	**0.9**
	Bowl to top	0.6	0.1	0.1	0.9	**1.0**
	Average	0.18	0.10	0.05	0.75	**0.88**

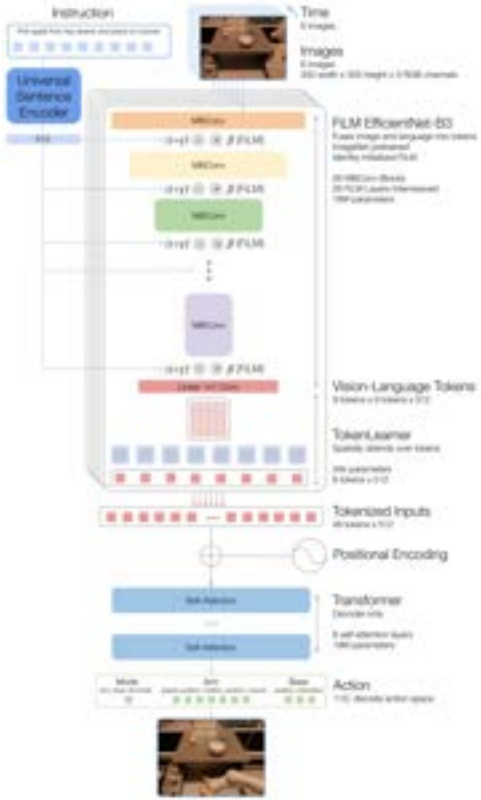
Black, K., Nakamoto, M., Atreya, P., Walke, H., Finn, C., Kumar, A., & Levine, S. (2023). Zero-shot robotic manipulation with pretrained image-editing diffusion models. arXiv preprint arXiv:2310.10639.

Language Condition

Two line of thoughts

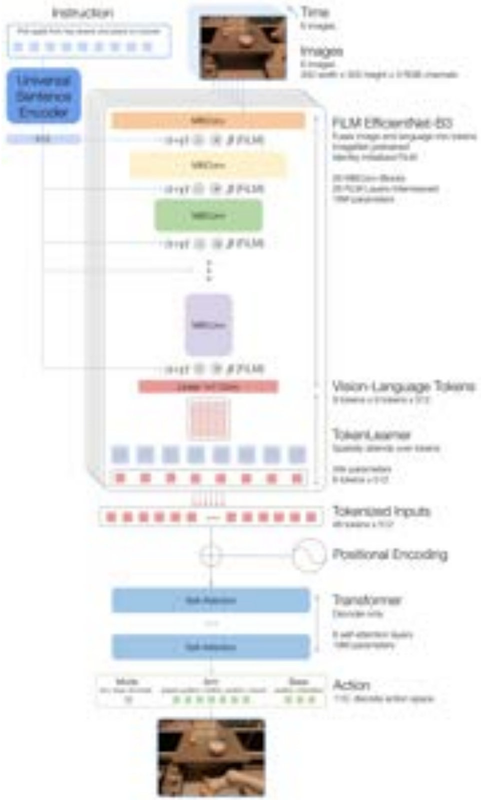
1. Fuse language + vision, policy learns from the shared latent
    a. **RT-1**
    b. **Early Fusion VLA**
2. Multi-modal sequence modeling / VQA:
    a. Gato
    b. RT-2
    c. **OpenVLA**
    d. LLarva
3. Both Language + Goal
    a. Octo

Robot Transformer



Brohan, Anthony, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan et al. "Rt-1: Robotics transformer for real-world control at scale." arXiv preprint arXiv:2212.06817 (2022).

Robot Transformer



Not tokenized images (Extracted by EfficientNetB3)

Training Data

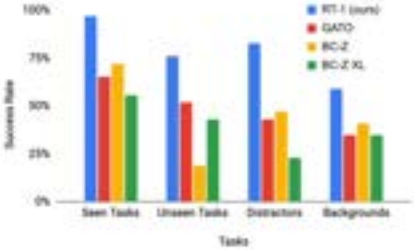
- Trained on 130k tele-operation demonstrations over 13 robots and 744 tasks.

Skill	Count	Description	Example Instruction
Pick `Object`	130	Lift the object off the surface	pick iced tea can
Move `Object` Near `Object`	337	Move the first object near the second	move pepsi can near rxbar blueberry
Place `Object` Upright	8	Place an elongated object upright	place water bottle upright
Knock `Object` Over	8	Knock an elongated object over	knock redbull can over
Open / Close `Drawer`	6	Open or close any of the cabinet drawers	open the top drawer
Place `Object` into `Receptacle`	84	Place an object into a receptacle	place brown chip bag into white bowl
Pick `Object` from `Receptacle` and Place on the Counter	162	Pick an object up from a location and then place it on the counter	pick green jalapeno chip bag from paper bowl and place on counter
Additional tasks	9	Skills trained for realistic, long instructions	pull napkin out of dispenser
Total	744		

Evaluation Data

- Evaluated on real-world randomized scenes and over 3000 total rollouts in the environment it was trained on as well as two new office kitchen environments.

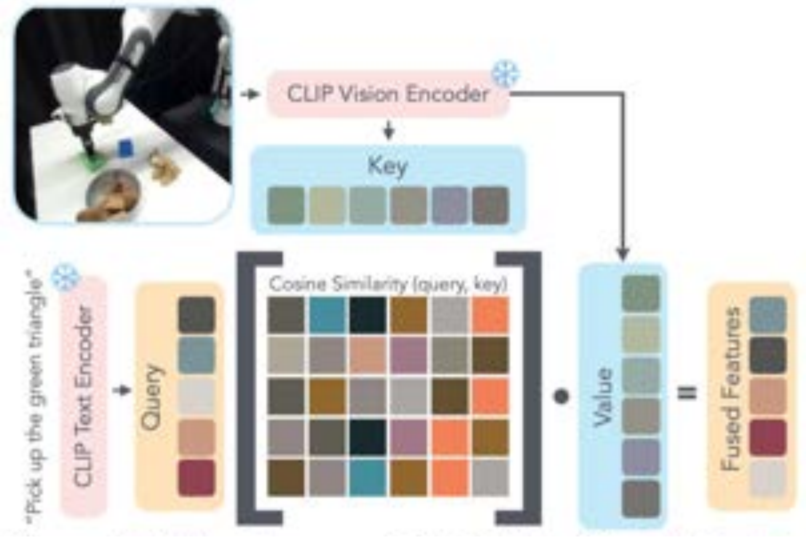
Model	Seen Tasks	Unseen Tasks	Distractors	Backgrounds
Gato (Reed et al., 2022)	65	52	43	35
BC-Z (Jang et al., 2021)	72	19	47	41
BC-Z XL	56	43	23	35
RT-1 (ours)	**97**	**76**	**83**	**59**

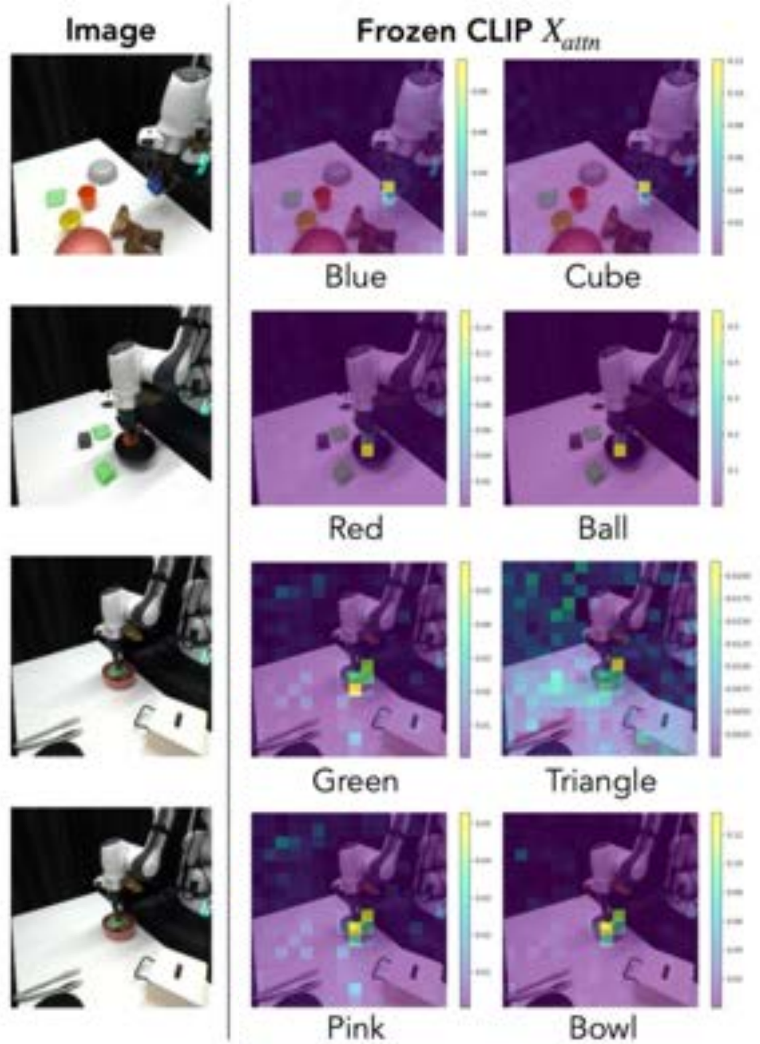


Brohan, Anthony, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan et al. "Rt-1: Robotics transformer for real-world control at scale." arXiv preprint arXiv:2212.06817 (2022).

Early Fusion VLA

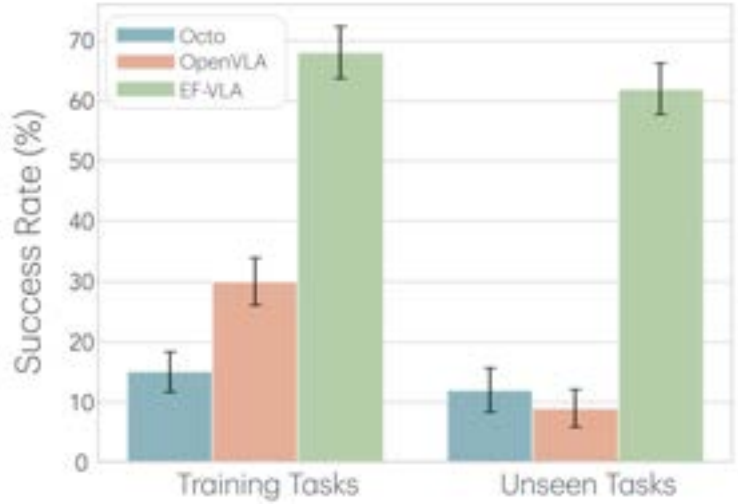
Do you really have to relearn language vision alignment with FiLM?
CLIP already knows that!

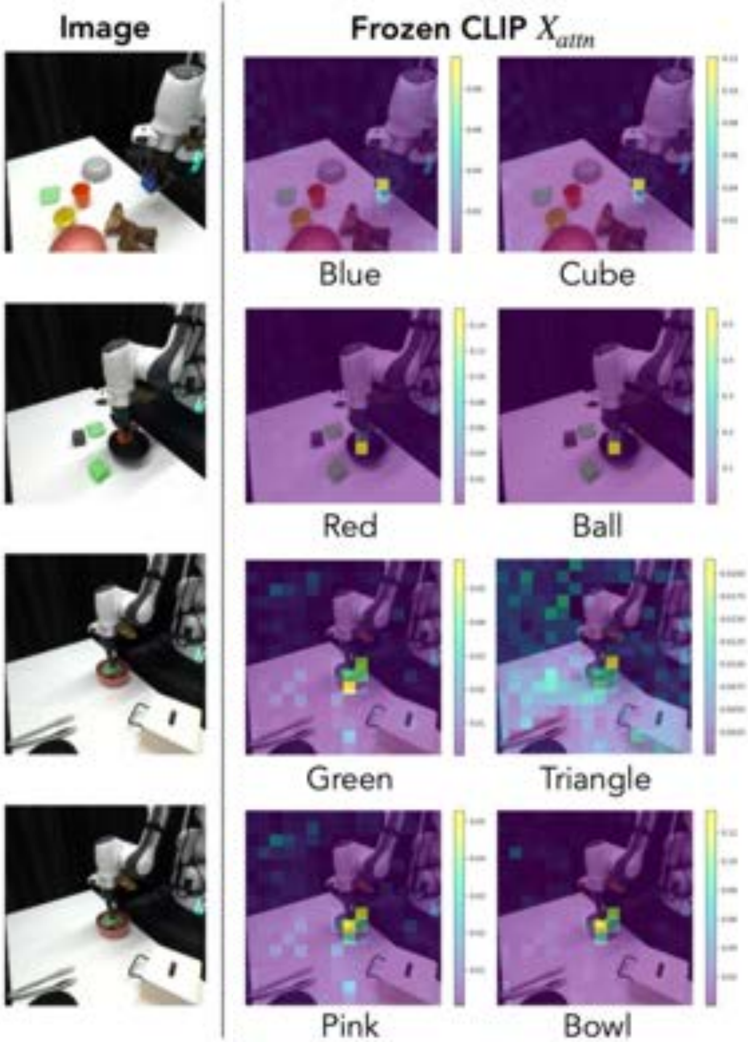




Early Fusion VLA

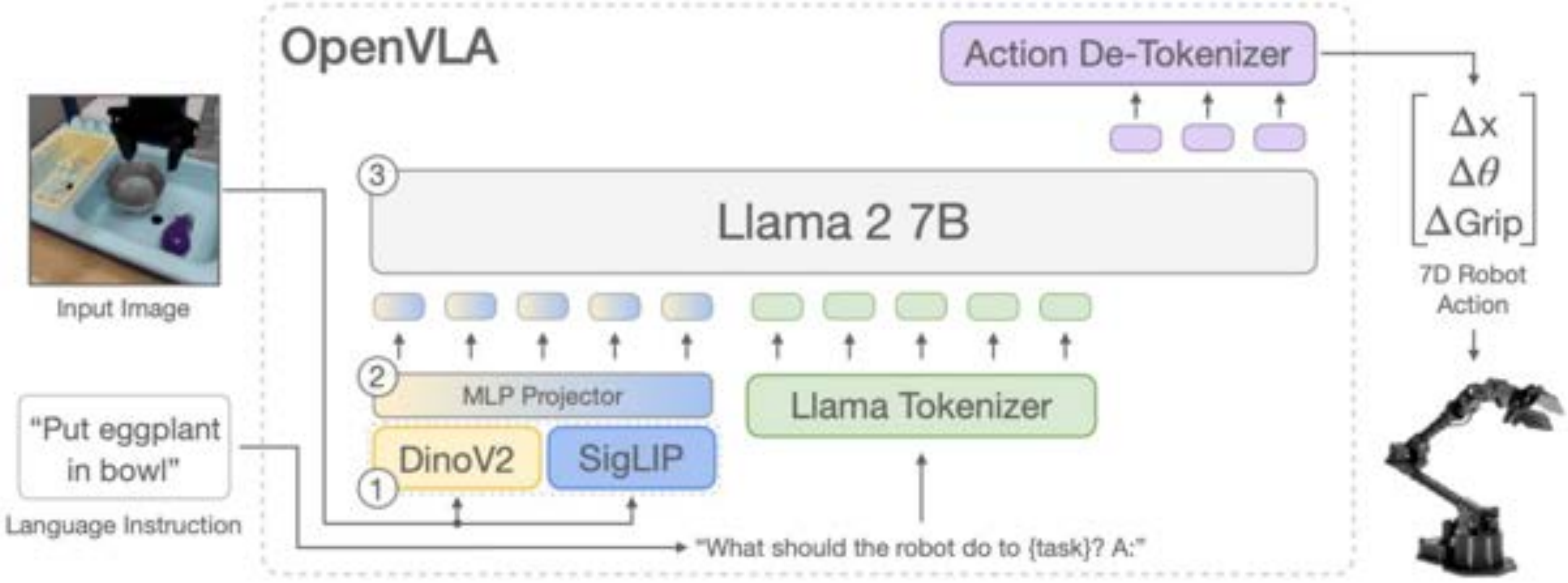
Do you really have to relearn language vision alignment with FiLM?
CLIP already knows that!





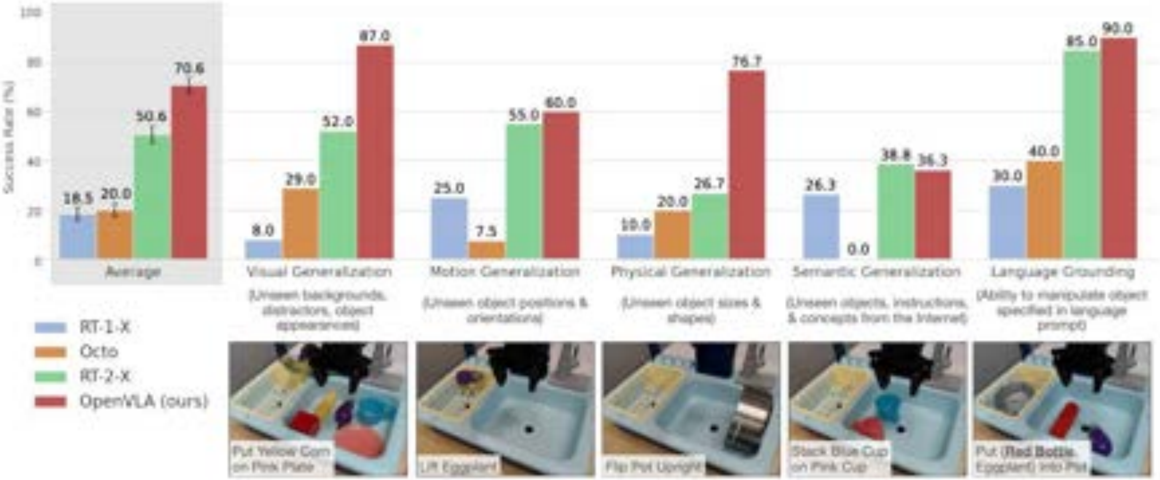
OpenVLA

Action generation as VQA!
Use Prismatic VLA, but finetune to generate action.



Kim, M. J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., ... & Finn, C. (2024). OpenVLA: An Open-Source Vision-Language-Action Model. *arXiv preprint arXiv:2406.09246*.

OpenVLA

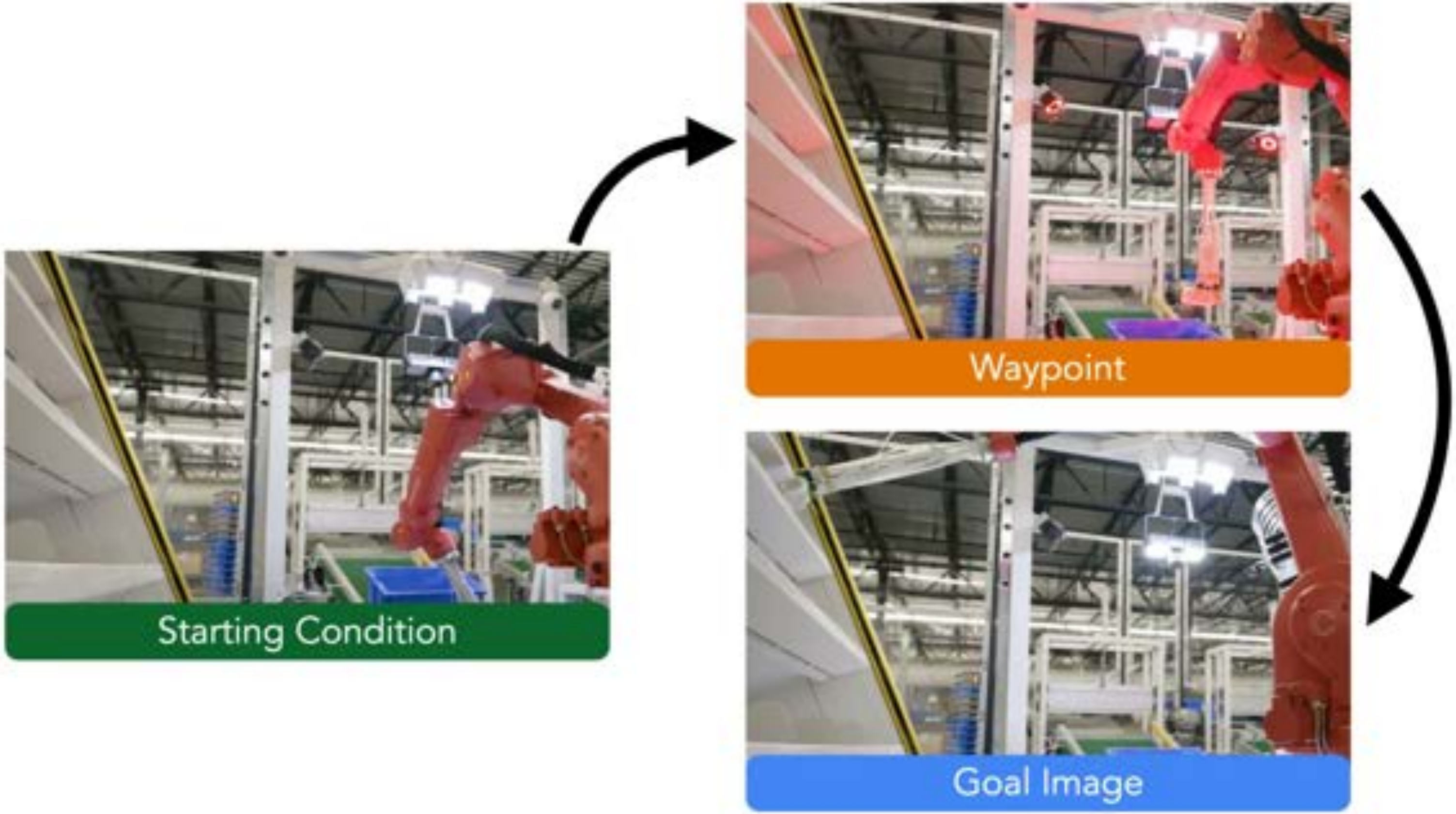


Strategy	Success Rate	Train Params ($\times 10^6$)	VRAM (batch 16)
Full FT	**69.7 ± 7.2 %**	7,188.1	163.3 GB*
Last layer only	30.3 ± 6.1 %	465.1	51.4 GB
Frozen vision	47.0 ± 6.9 %	6,760.4	156.2 GB*
Sandwich	62.1 ± 7.9 %	914.2	64.0 GB
LoRA, rank=32	**68.2 ± 7.5%**	**97.6**	**59.7 GB**
rank=64	**68.2 ± 7.8%**	195.2	60.5 GB

Advantages and disadvantages of the two approaches

1. **Fuse language + vision first, policy learns from the shared latent**
   a. *Advantage*:
      i. if done correctly, you can use the power of pre-trained VLM
      ii. Can be super fast and lightweight
   b. *Disadvantages*:
      i. no multimodal reasoning capability, as it only predicts action -> needs an alternative model to perform planning
2. **Multi-modal sequence modeling / VQA:**
   a. *Advantages*:
      i. Multi-modal reasoning capability, potentially can perform planning along with action generation
   b. *Disadvantages*:
      i. Slow! To have stronger planning capability -> bigger model -> slower inference / cloud compute needed (increase latency)

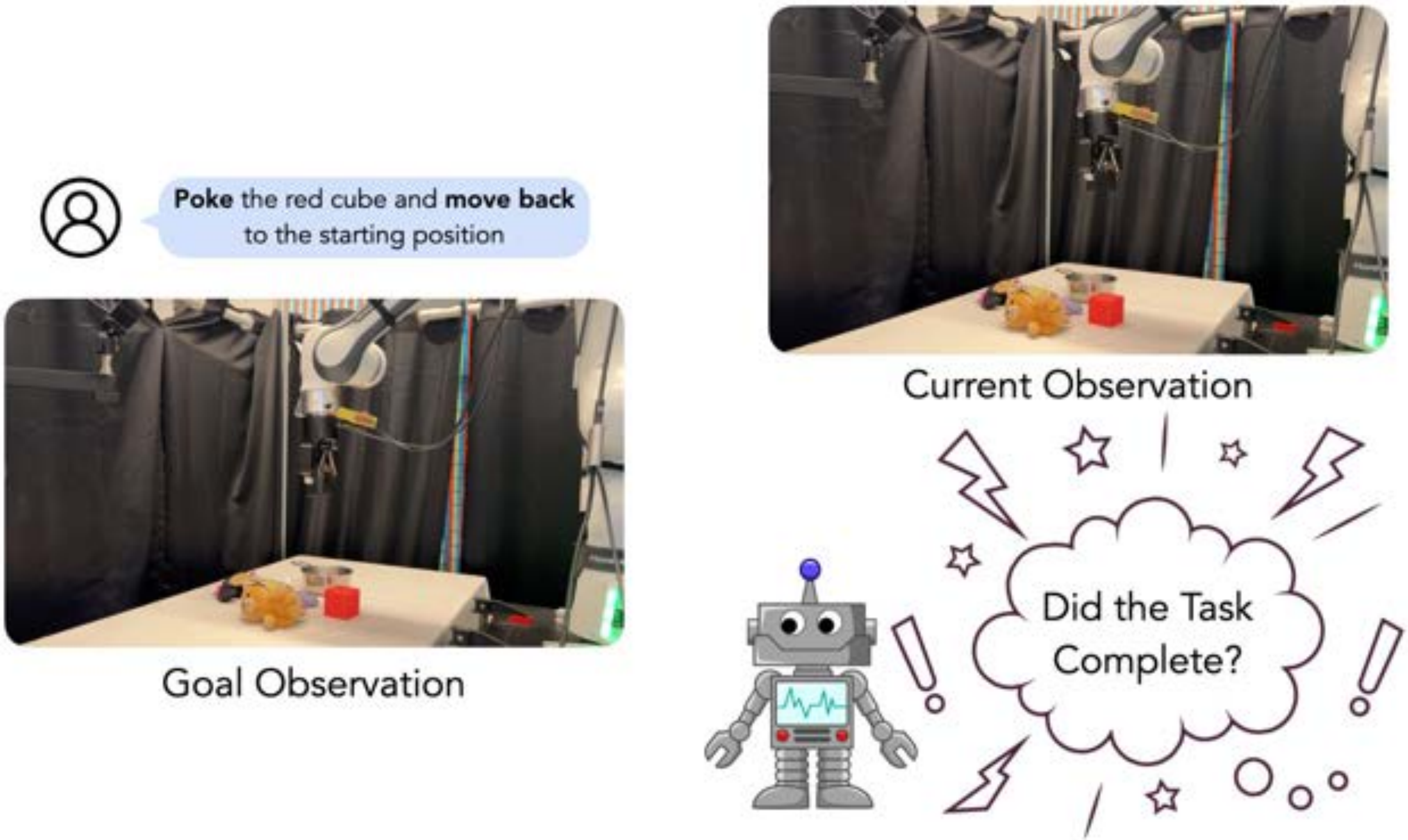
Why do we not want to use language / goal observations?

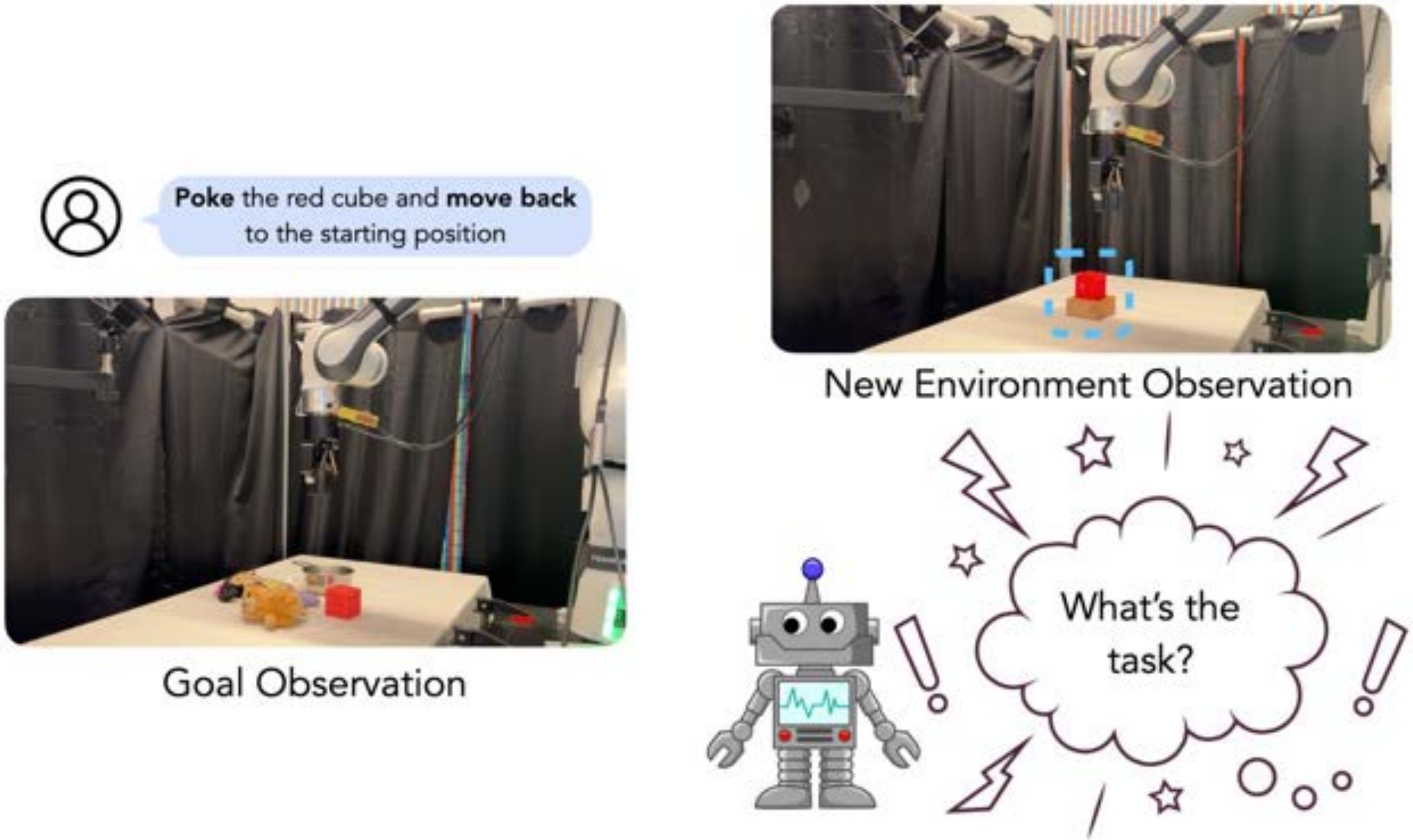


"Pick up parcel, go through the scanner and place it on the shelf"

Language Condition: *"Hand me the steak and **move away** from the wine glasses"*

Generated by DALLE


Poke the red cube and **move back** to the starting position
Goal Observation
Current Observation
Did the Task Complete?


Poke the red cube and **move back** to the starting position
Goal Observation
New Environment Observation
What's the task?

In-context robot transformer

Framework

Robot in-context learning by next-token prediction on sensorimotor trajectories (no fine-tuning)

Input: raw human teleoperation trajectories

Output: real-time continuous control

In-context robot transformer

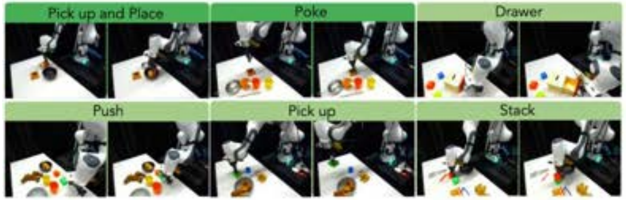
Framework

Robot in-context learning by next-token prediction on sensorimotor trajectories (no fine-tuning)
Input: raw human teleoperation trajectories
Output: real-time continuous control

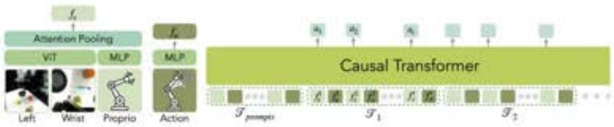
Dataset

ICRT-Multi-Task (ICRT-MT) **Dataset**:
- 1098 trajectories
- 29 tasks
- 6 motion primitives.

+2k trajectories from DROID for pre-training.

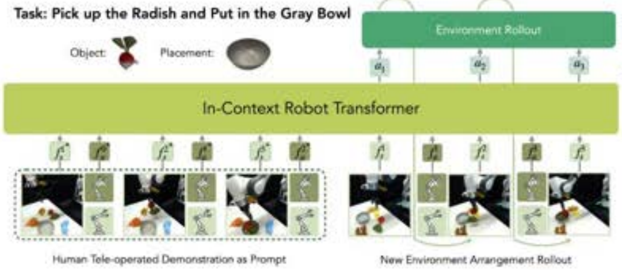


Sensorimotor Prediction



Sequence length: 512 steps from *many* trajectories of the same task
Loss: L1 loss on the *post-prompt* trajectories, no loss on the prompt
Pre-trained on Droid: 4 epochs, **Fine-tuned on ICRT-MT**: 125 epochs

Inference



Results

	Pick and Place	Poke	Average
Goal Condition	33.3 (±6.5)	6.7 (±4.6)	20.0 (±4.3)
Octo [15]	5.0 (±2.7)	13.3 (±6.2)	9.2 (±3.5)
OpenVLA [14]	11.7 (±4.6)	3.3 (±3.3)	7.5 (±2.9)
ICRT	**65.0 (±7.3)**	**93.3 (±4.6)**	**79.2 (±4.6)**

*30 trials per primitive and 60 trials overall. Reported average (standard error)

Ablations

	Pick and Place	Poke	Average
ICRT-Llama2	43.3 (±7.9)	73.3 (±8.2)	58.3 (±6.0)
ICRT (DROID)	0.0 (±0.0)	0.0 (±0.0)	0.0 (±0.0)
ICRT (MT)	**76.7 (±7.1)**	70.0 (±8.5)	73.3 (±5.5)
ICRT +Prompt Loss	21.7 (±6.2)	23.3 (±7.9)	22.5 (±5.0)
ICRT	65.0 (±7.3)	**93.3 (±4.6)**	**79.2 (±4.6)**

Single Stage Training

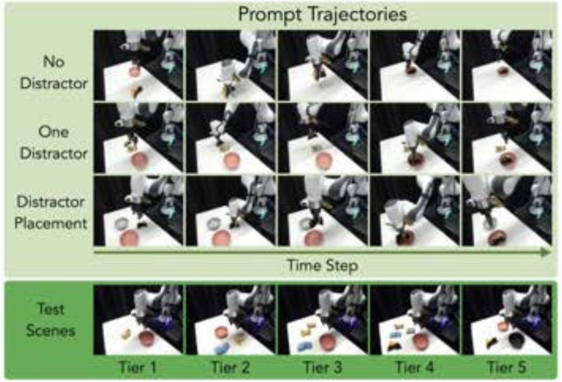
	Pick and Place	Poke	Average
ICRT (Co-train)	13.3 (±5.8)	0.0 (±0.0)	6.7 (±3.0)
ICRT	**65.0 (±7.3)**	**93.3 (±4.6)**	**79.2 (±4.6)**

Control Frequency

	Inference Frequency
ICRT	39.6 Hz
ICRT-Llama2	10.7 Hz

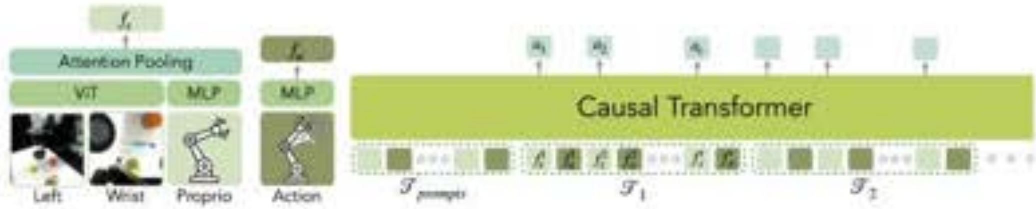
Different Prompts for Inference

Prompt Type	No Distractor	One Distractor	Distractor Placement	Two Prompts	Three Prompts
Success Rate	60%	80%	70%	80%	80%



In-context robot transformer

Sensorimotor Prediction



Sequence length: 512 steps from *many* trajectories of the same task
Loss: L1 loss on the *post-prompt* trajectories, no loss on the prompt
Pre-trained on Droid: 4 epochs, **Fine-tuned on ICRT-MT**: 125 epochs

Results

	Pick and Place	Poke	Average
Goal Condition	33.3 (±6.5)	6.7 (±4.6)	20.0 (±4.3)
Octo [15]	5.0 (±2.7)	13.3 (±6.2)	9.2 (±3.5)
OpenVLA [14]	11.7 (±4.6)	3.3 (±3.3)	7.5 (±2.9)
ICRT	**65.0 (±7.3)**	**93.3 (±4.6)**	**79.2 (±4.6)**

Inference

	Pick and Place	Poke	Average
ICRT-Llama2	43.3 (±7.9)	73.3 (±8.2)	58.3 (±6.0)
ICRT (DROID)	0.0 (±0.0)	0.0 (±0.0)	0.0 (±0.0)
ICRT (MT)	**76.7 (±7.1)**	70.0 (±8.5)	73.3 (±5.5)
ICRT +Prompt Loss	21.7 (±6.2)	23.3 (±7.9)	22.5 (±5.0)
ICRT	65.0 (±7.3)	**93.3 (±4.6)**	**79.2 (±4.6)**

Single Stage Training

	Pick and Place	Poke	Average
ICRT (Co-train)	13.3 (±5.8)	0.0 (±0.0)	6.7 (±3.0)
ICRT	**65.0** (±7.3)	**93.3** (±4.6)	**79.2** (±4.6)

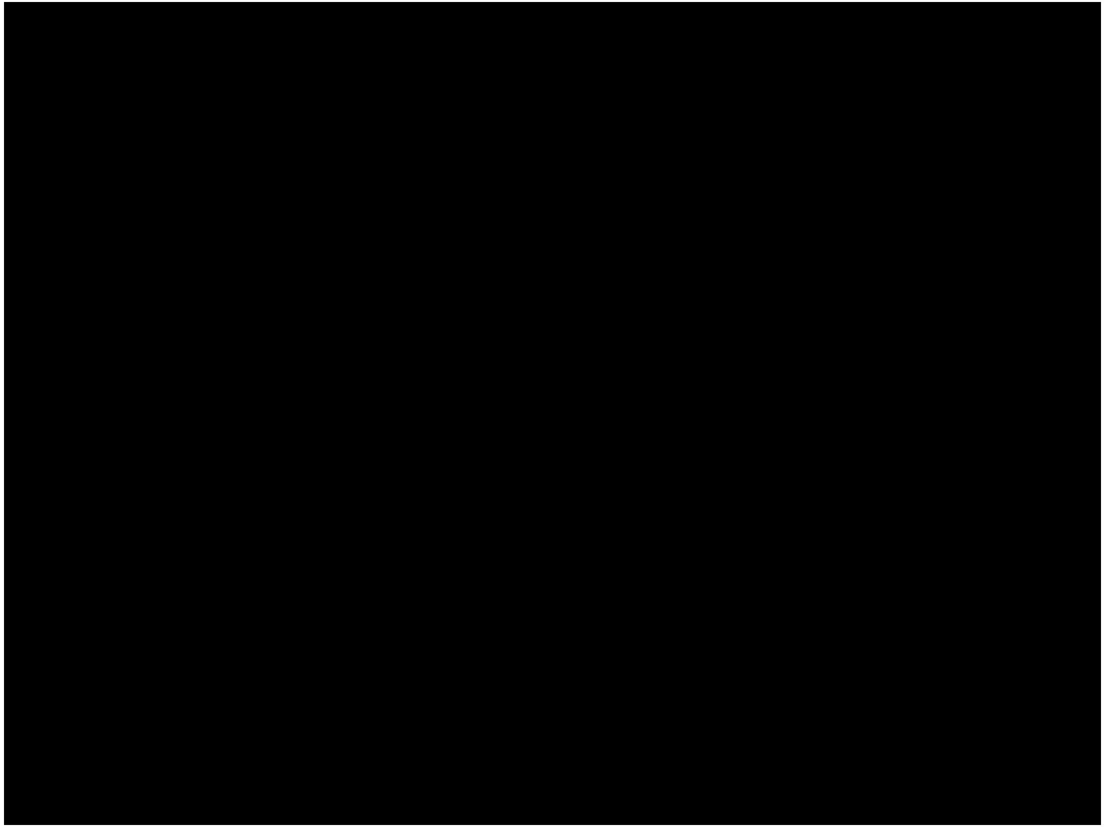
Control Frequency

	Inference Frequency
ICRT	39.6 Hz
ICRT-Llama2	10.7 Hz

Different Prompts for Inference

Prompt Type	No Distractor	One Distractor	Distractor Placement	Two Prompts	Three Prompts
Success Rate	60%	80%	70%	80%	80%

In-context robot transformer

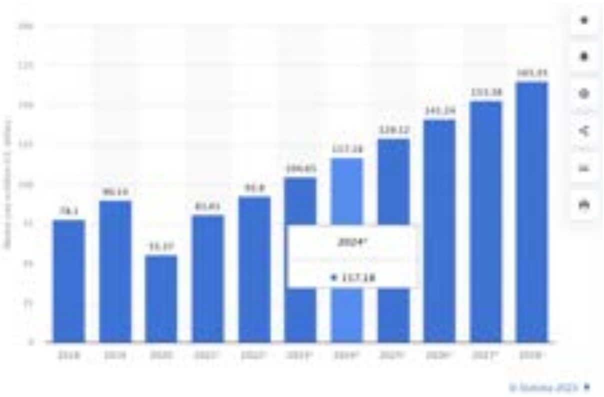


III. Limitations of End2End and Future works

- Advantages
    - Low latency, close loop control
    - Easy to teach a single task
- Disadvantages
    - Task has become a lot simpler (unless we consider single task learning, with a lot of data)
    - If dataset is formulated incorrectly, may suffer from overfitting and doesn't generalize
    - Control is less precise, not interpretable
    - Where is the data?
        - Find VC, pitch them data collection
        - Scaling in simulation
        - Learning from human video data
    - What are the basis vectors of human / robot motion? How to compose them?

End

Why robotics?



Rapidly growing market

The world is full of autonomous systems!

Why robotics?

L

Why should students in a vision-language seminar care about robotics?

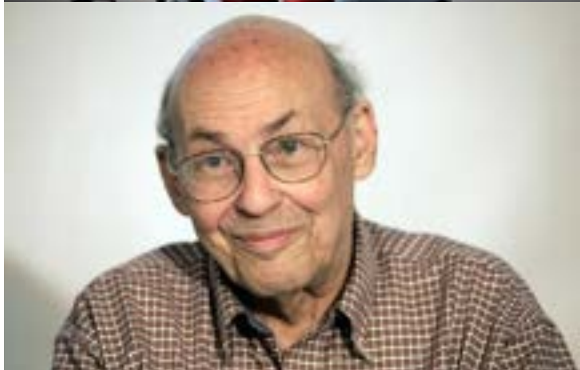
Students in a vision-language seminar should care about robotics because Robotics is an excellent application area for vision and language research. The development of intelligent robots requires the integration of multiple technologies, including computer vision and natural language processing, making it a compelling area of research for students interested in vision and language.

Why robotics?

Moravec's paradox

- Hans Moravec, Rodney Brooks, Marvin Minsky:
- “Reasoning requires very little computation, but sensorimotor and perception skills require enormous computational resources”
- “It is comparatively easy to make computers exhibit adult level performance on intelligence tests or playing checkers, and difficult or impossible to give them the skills of a one-year-old when it comes to perception and mobility"





Why is robotics challenging?
How is robotics different from vision and language?

- How to create a system that perceive and *interacts* with the environment?
- Where's the data and what data is available?
- Correspondence (observation, action, goal?)
- Designing good optimization problems?
- Is the system safe? Easy to use?

Today …

- Focus on robotics manipulation
- Sensing vs Task & Motion Planning
- Three sections
    - Robotics and Vision
    - NLP and Decision Making
    - Unify them all

I. Robotics and Computer Vision

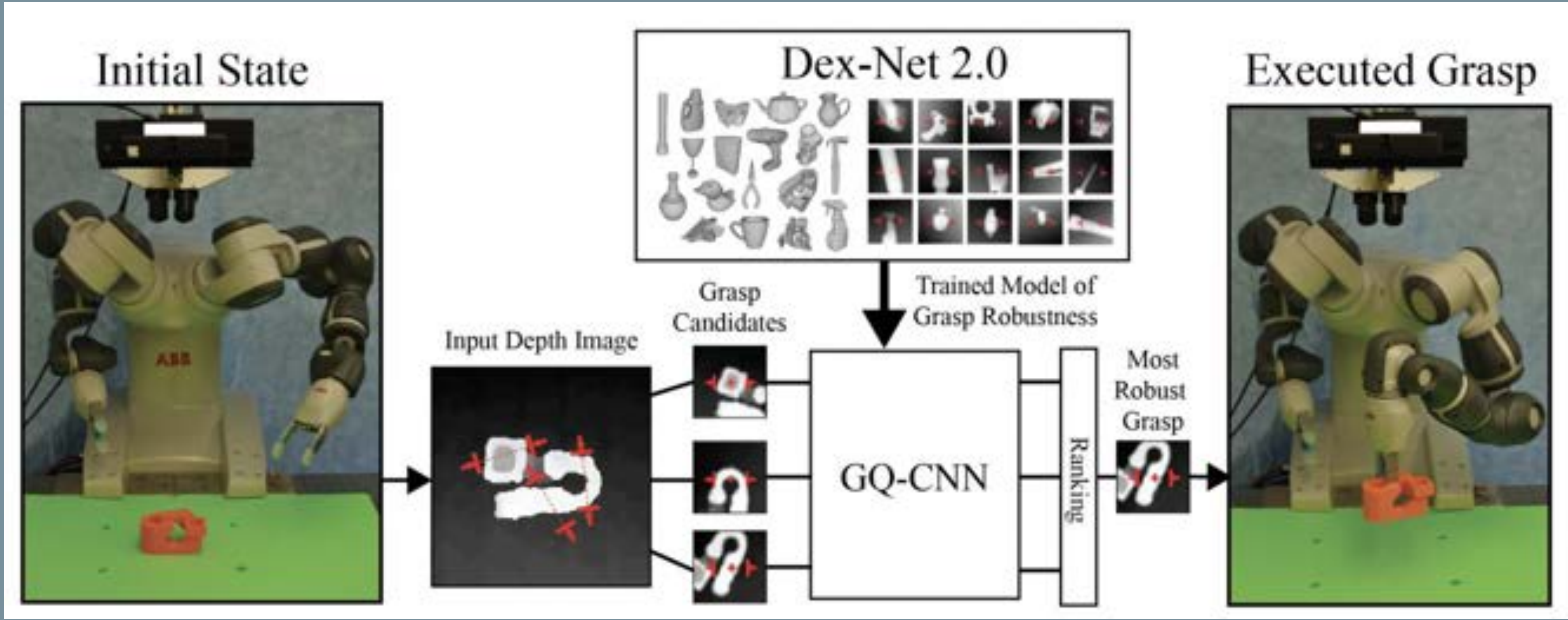
I. Robotics and Computer Vision

- Scaling Data

I.I Scaling Data

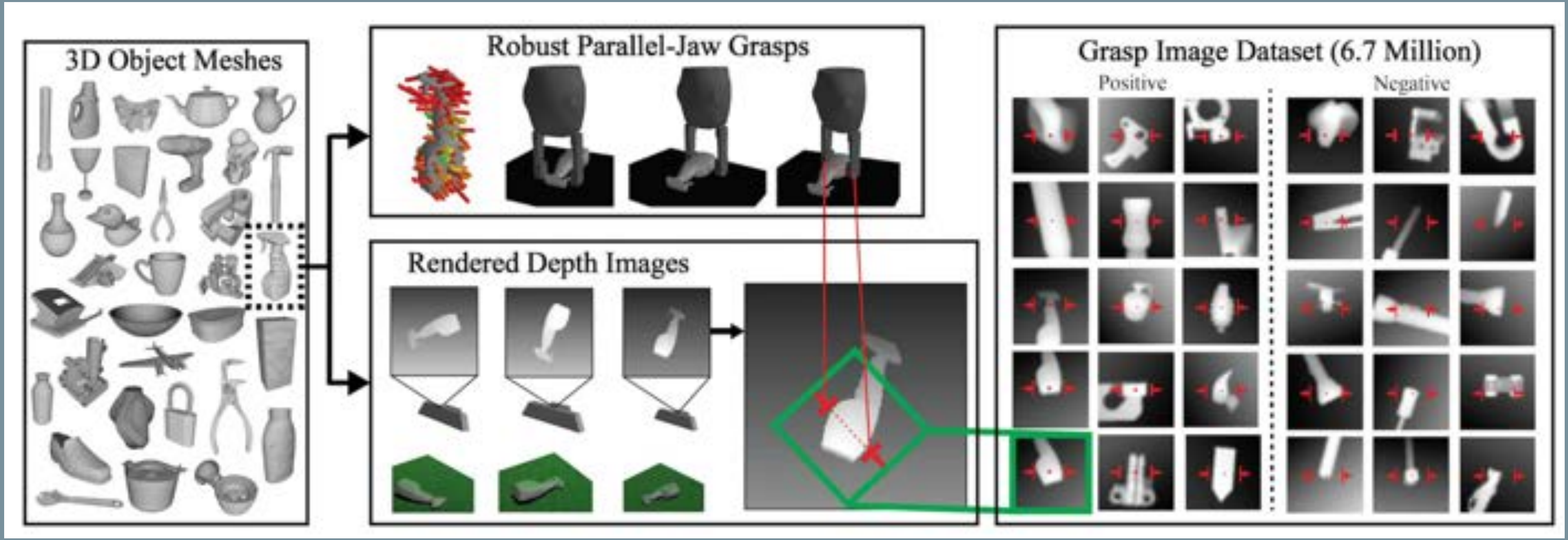
- **Through Simulation**
- Through Real World Data

Case Study 1: DexNet 2.0



Mahler, Jeffrey, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics." arXiv preprint arXiv:1703.09312 (2017).

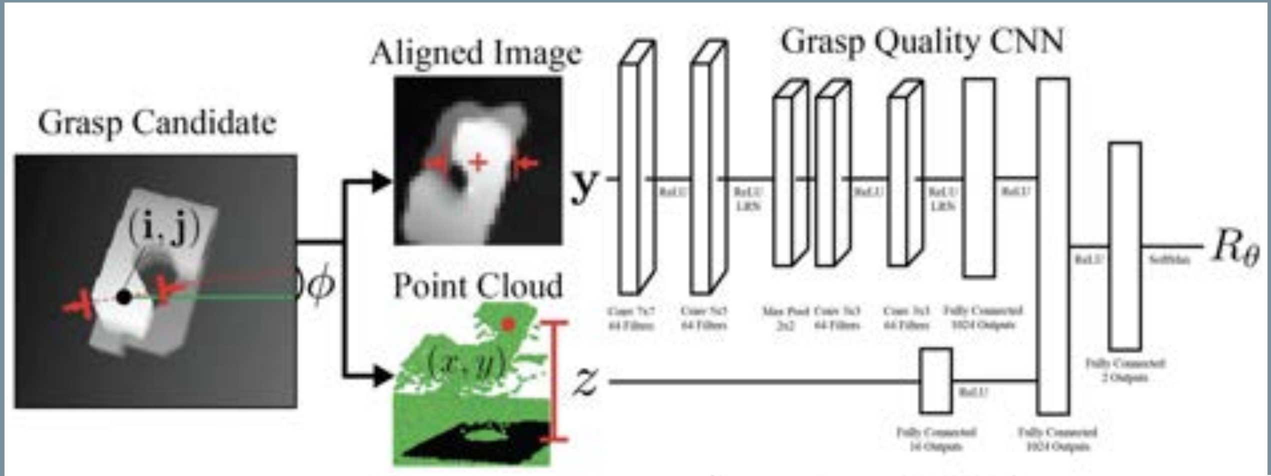
Case Study 1: DexNet 2.0



1,500 3D object mesh models

Mahler, Jeffrey, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics." arXiv preprint arXiv:1703.09312 (2017).

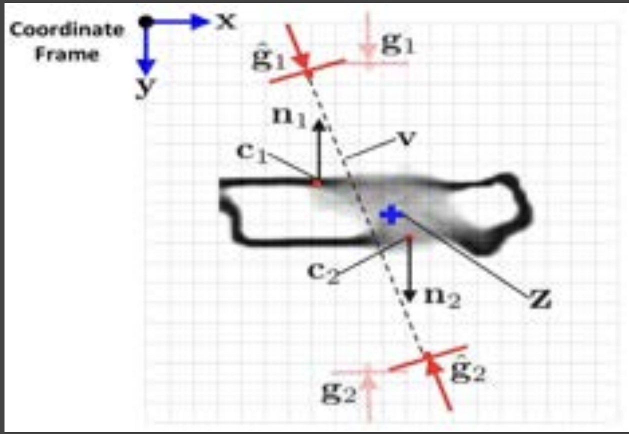
Case Study 1: DexNet 2.0



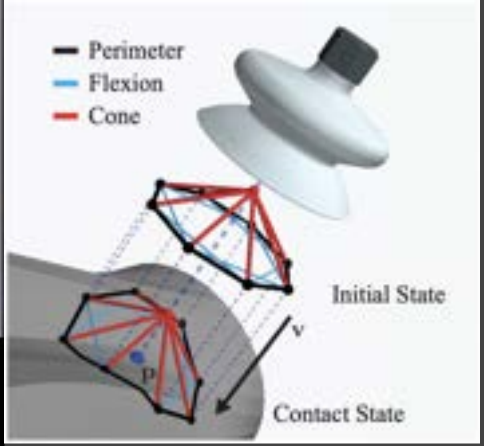
Comparions of Methods						
	Random	**IGQ**	**ML-RF**	**ML-SVM**	**REG**	**GQ-L-Adv**
Success Rate (%)	58±11	70±10	75±9	80±9	95±5	**93±6**
Precision (%)	N/A	N/A	100	100	N/A	**94**
Robust Grasp Rate (%)	N/A	N/A	5	0	N/A	43
Planning Time (sec)	N/A	1.9	0.8	0.9	2.6	**0.8**

Mahler, Jeffrey, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics." arXiv preprint arXiv:1703.09312 (2017).

Case Study 1: DexNet 4.0

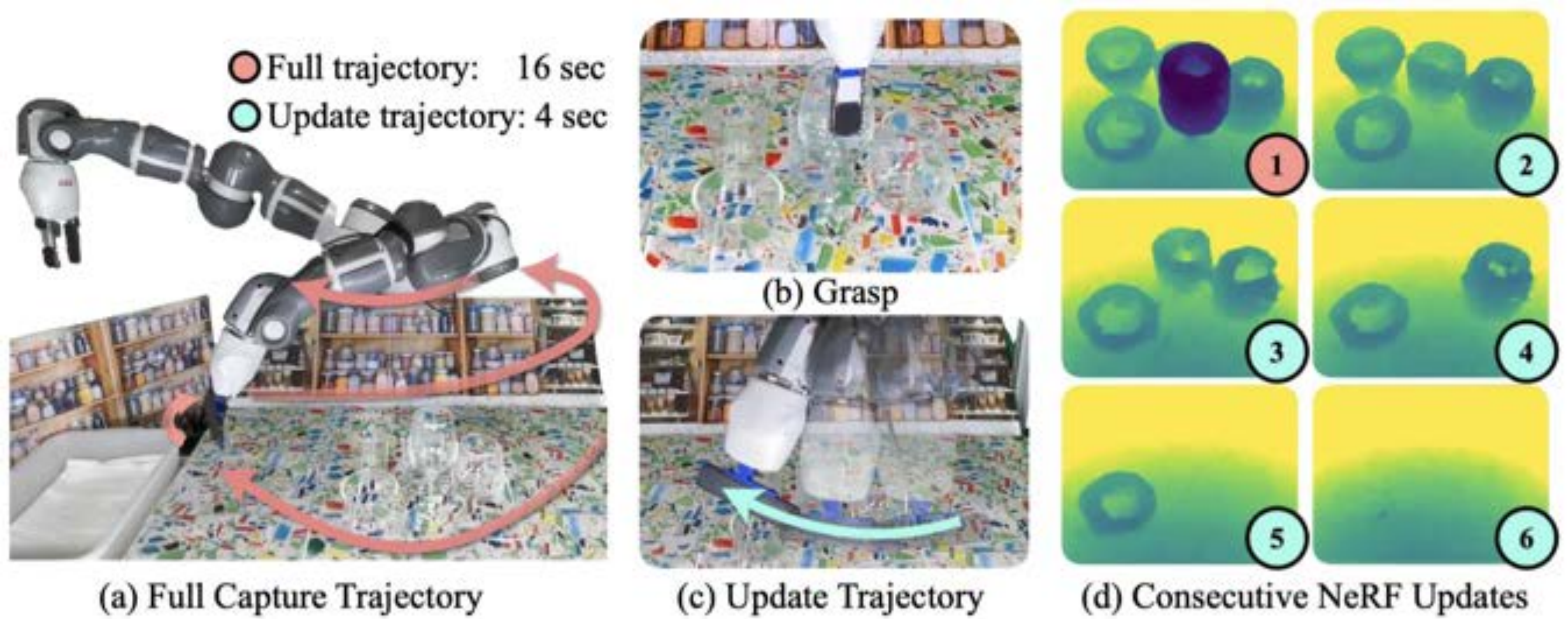


Mahler, J., Patil, S., Kehoe, B., Van Den Berg, J., Ciocarlie, M., Abbeel, P., & Goldberg, K. (2015, May). Gp-gpis-opt: Grasp planning with shape uncertainty using gaussian process implicit surfaces and sequential convex programming. In 2015 IEEE international conference on robotics and automation (ICRA) (pp. 4919-4926). IEEE.



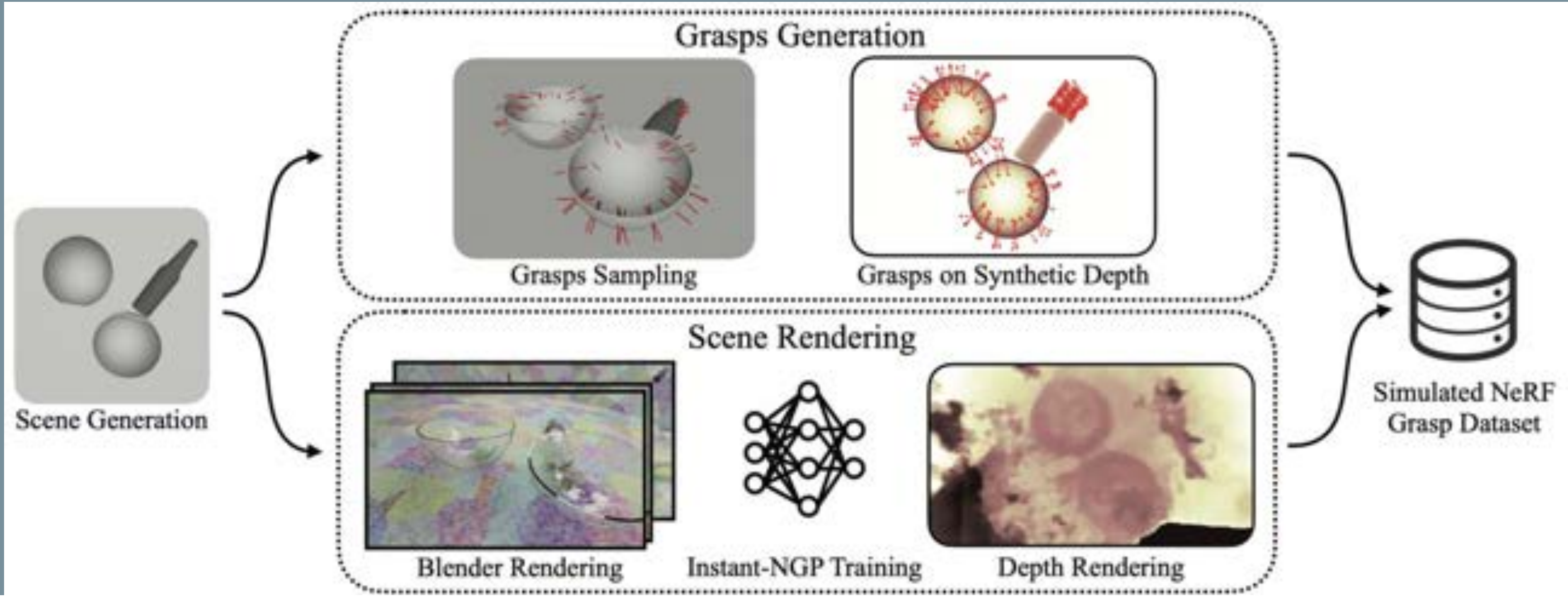
Mahler, Jeffrey, Matthew Matl, Xinyu Liu, Albert Li, David Gealy, and Ken Goldberg. "Dex-net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning." In 2018 IEEE International Conference on robotics and automation (ICRA), pp. 5620-5627. IEEE, 2018.

Case Study 2: EvoNeRF (aka DexNeRF 2)



Kerr, Justin, Letian Fu, Huang Huang, Yahav Avigal, Matthew Tancik, Jeffrey Ichnowski, Angjoo Kanazawa, and Ken Goldberg. "Evo-NeRF: Evolving NeRF for Sequential Robot Grasping of Transparent Objects." In 6th Annual Conference on Robot Learning.

Case Study 2: EvoNeRF (aka DexNeRF 2)



GT-Net achieves 0 %, Rad-Net achieves 42 %, and Dex-Net achieves 0.1%, suggesting that there is a large distribution shift from training on ground-truth depth to testing on NeRF-depth in simulation.

Kerr, Justin, Letian Fu, Huang Huang, Yahav Avigal, Matthew Tancik, Jeffrey Ichnowski, Angjoo Kanazawa, and Ken Goldberg. "Evo-NeRF: Evolving NeRF for Sequential Robot Grasping of Transparent Objects." In 6th Annual Conference on Robot Learning.

Case Study 2: EvoNeRF (aka DexNeRF 2)

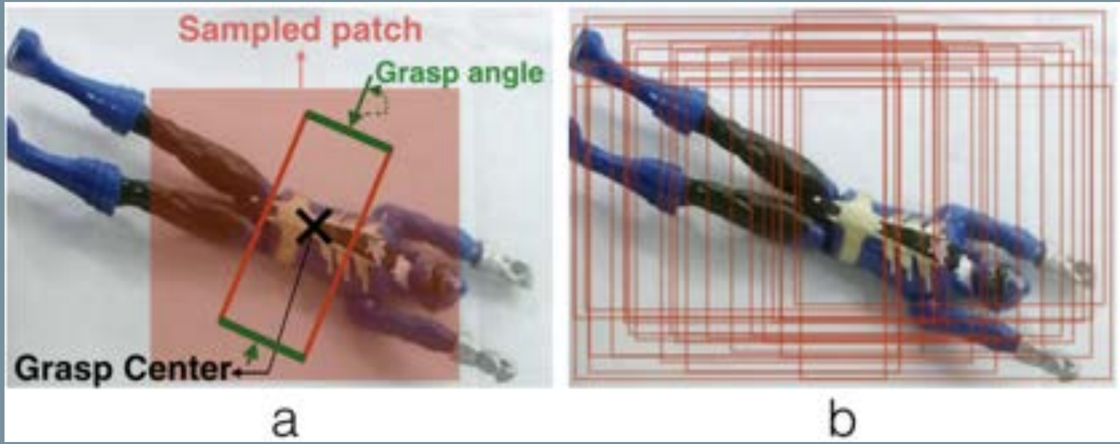
Kerr, Justin, Letian Fu, Huang Huang, Yahav Avigal, Matthew Tancik, Jeffrey Ichnowski, Angjoo Kanazawa, and Ken Goldberg. "Evo-NeRF: Evolving NeRF for Sequential Robot Grasping of Transparent Objects." In 6th Annual Conference on Robot Learning.

I.I Scaling Data

- Through Simulation
- **Through Real World Data**

Case Study 3: Supersizing Self-supervision

50K data points collected over 700 hours
(Before DexNet)



Pinto, Lerrel, and Abhinav Gupta. "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours." In 2016 IEEE international conference on robotics and automation (ICRA), pp. 3406-3413. IEEE, 2016.

Case Study 3: Supersizing Self-supervision

GRASP DATASET STATISTICS

Data Collection Type	Positive	Negative	Total	Grasp Rate
Random Trials	3,245	37,042	40,287	8.05%
Multi-Staged	2,807	4,500	7,307	38.41%
Test Set	214	2,759	2,973	7.19%
	6,266	**44,301**	**50,567**	

COMPARING OUR METHOD WITH BASELINES

	Heuristic			Learning based			
	Min eigenvalue	Eigenvalue limit	Optimistic param. select	kNN	SVM	Deep Net (ours)	Deep Net + Multi-stage (ours)
Accuracy	0.534	0.599	0.621	0.694	0.733	0.769	**0.795**

Pinto, Lerrel, and Abhinav Gupta. "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours." In 2016 IEEE international conference on robotics and automation (ICRA), pp. 3406-3413. IEEE, 2016.

Case Study 4: Equivariant Model

Can we scale more efficiently with online vision data?

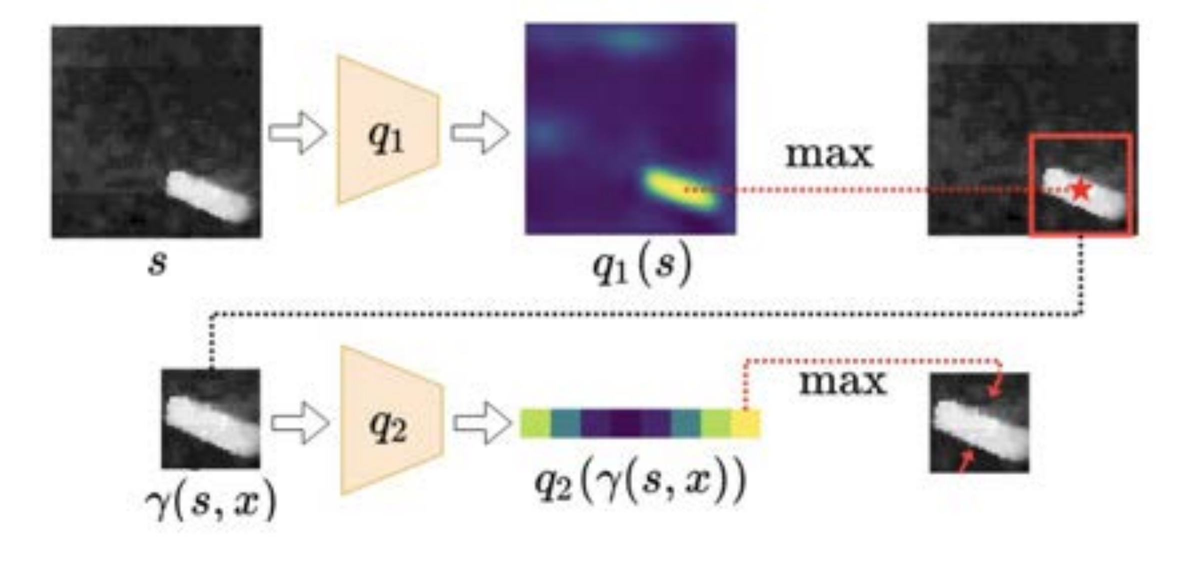
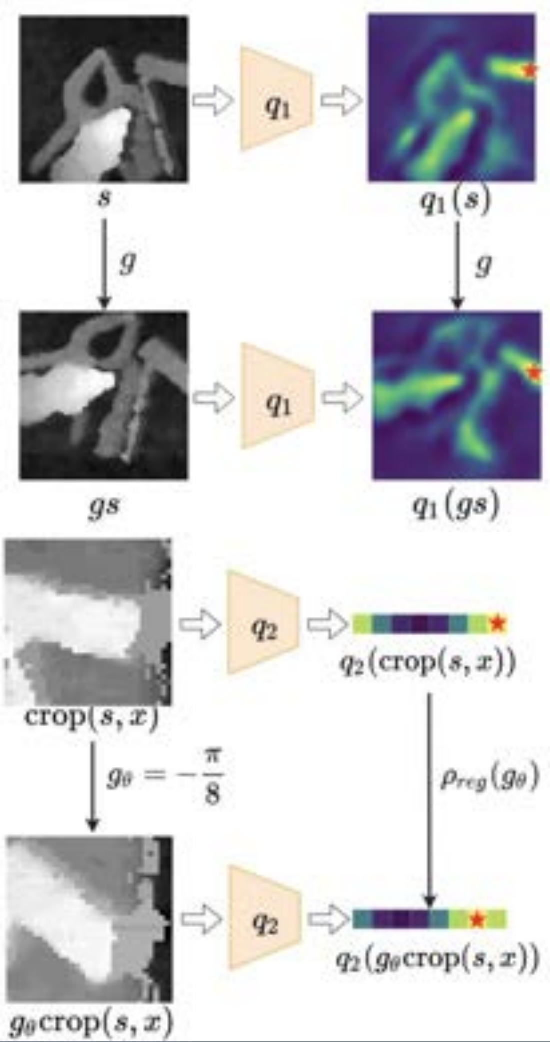


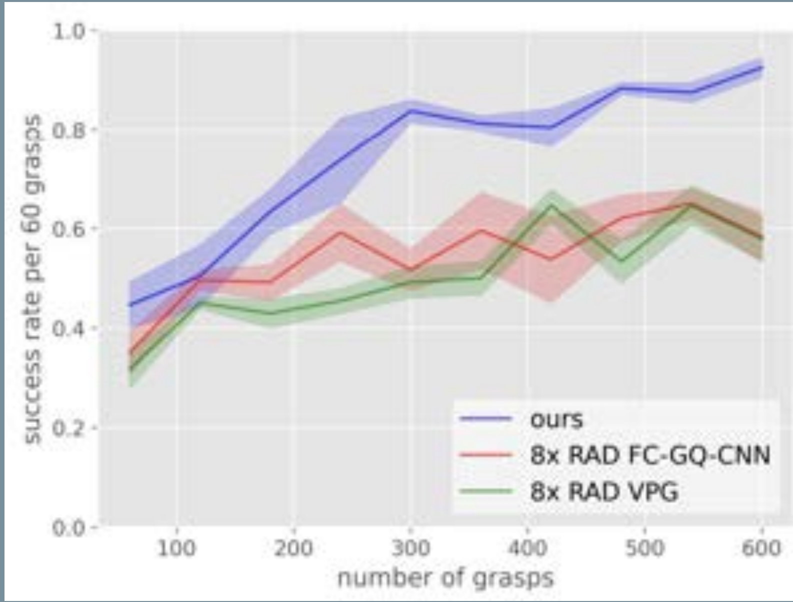
Fig. 1. Illustration of the ASR representation. Q_1 selects the translational component of an action, Q_2 selects the rotational component.



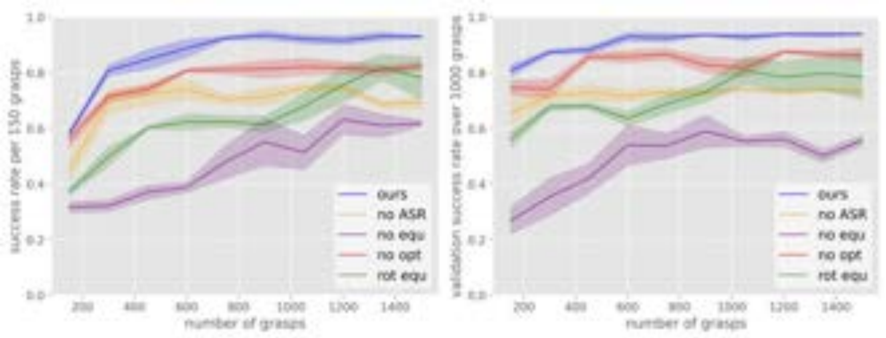
Zhu, Xupeng, Dian Wang, Ondrej Biza, Guanang Su, Robin Walters, and Robert Platt. "Sample efficient grasp learning using equivariant models." arXiv preprint arXiv:2202.09468 (2022).

Case Study 4: Equivariant Model

Can we scale more efficiently with online vision data?



(a) Training set (b) Testing set, easy (c) Testing set, hard

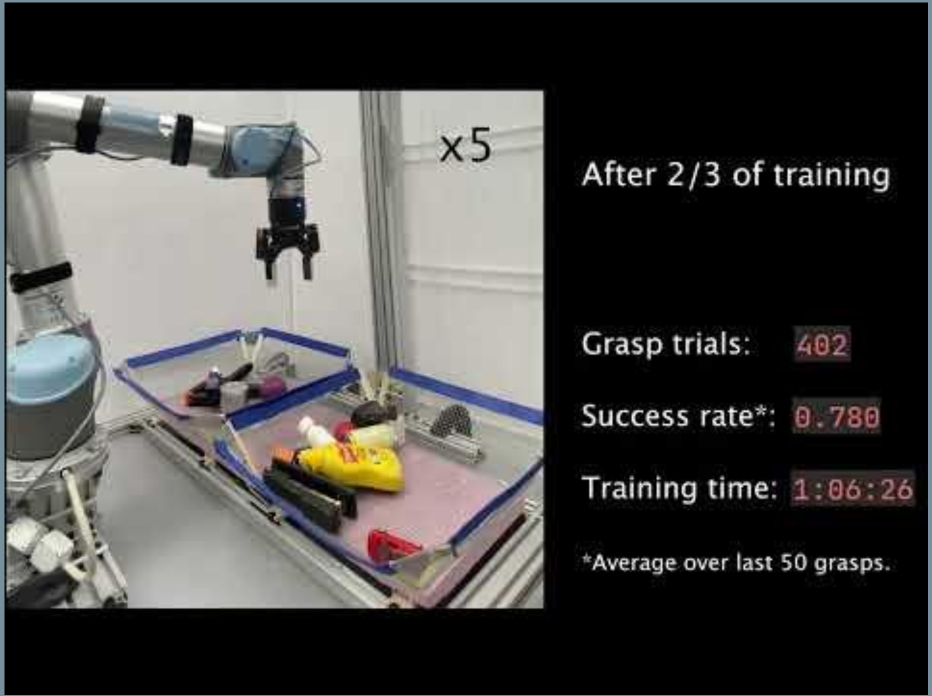


(a) Training (b) Testing

Zhu, Xupeng, Dian Wang, Ondrej Biza, Guanang Su, Robin Walters, and Robert Platt. "Sample efficient grasp learning using equivariant models." arXiv preprint arXiv:2202.09468 (2022).

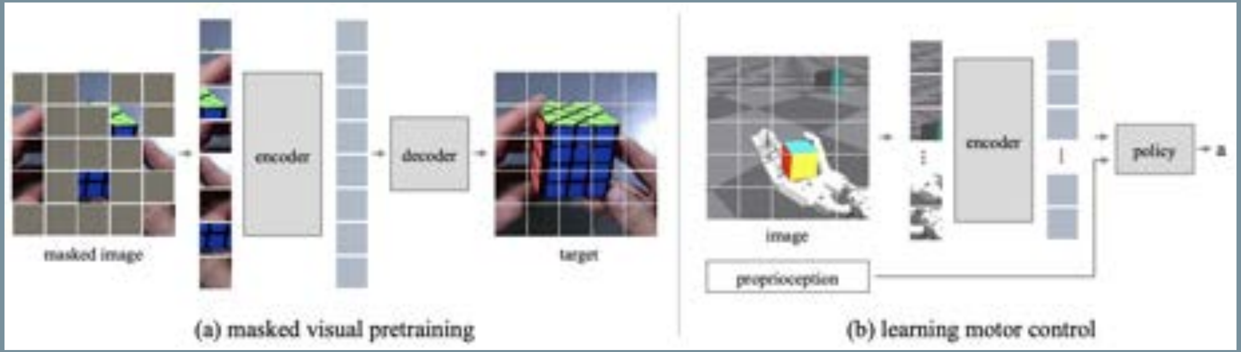
Case Study 4: Equivariant Model

Can we scale more efficiently with online vision data?



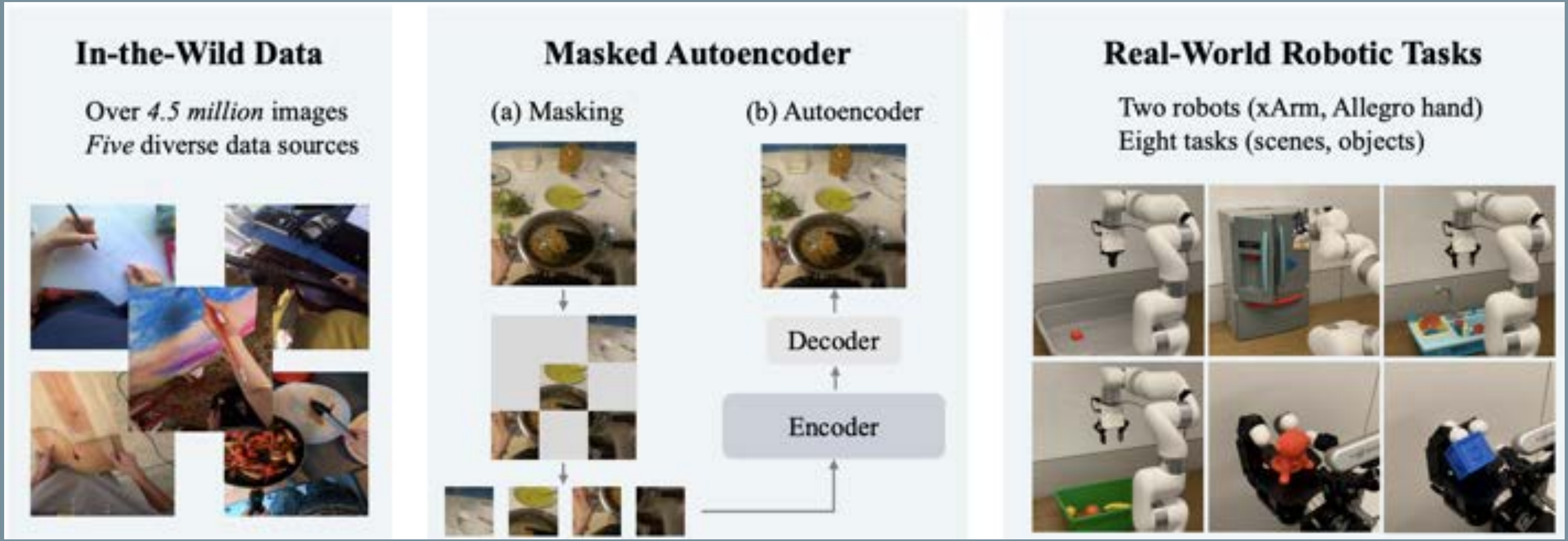
Zhu, Xupeng, Dian Wang, Ondrej Biza, Guanang Su, Robin Walters, and Robert Platt. "Sample efficient grasp learning using equivariant models." arXiv preprint arXiv:2202.09468 (2022).

Case Study 5: MVP



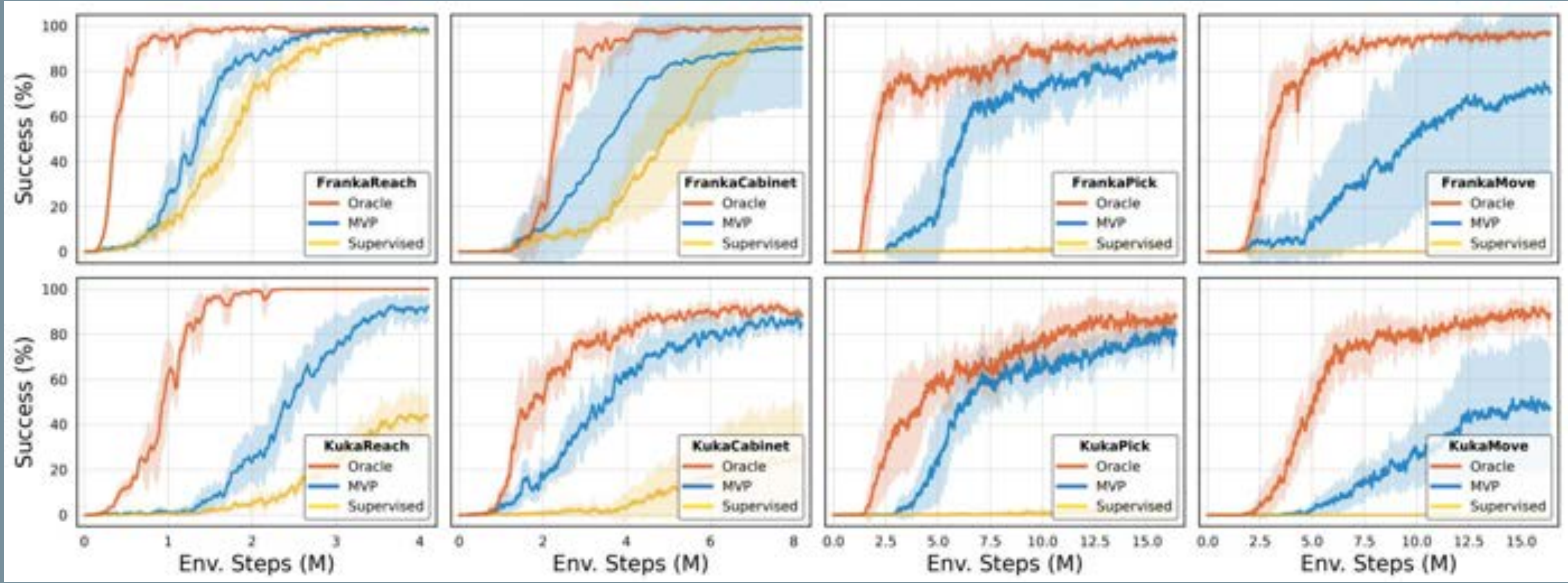
Xiao, Tete, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. "Masked visual pre-training for motor control." arXiv preprint arXiv:2203.06173 (2022).

Case Study 5: MVP



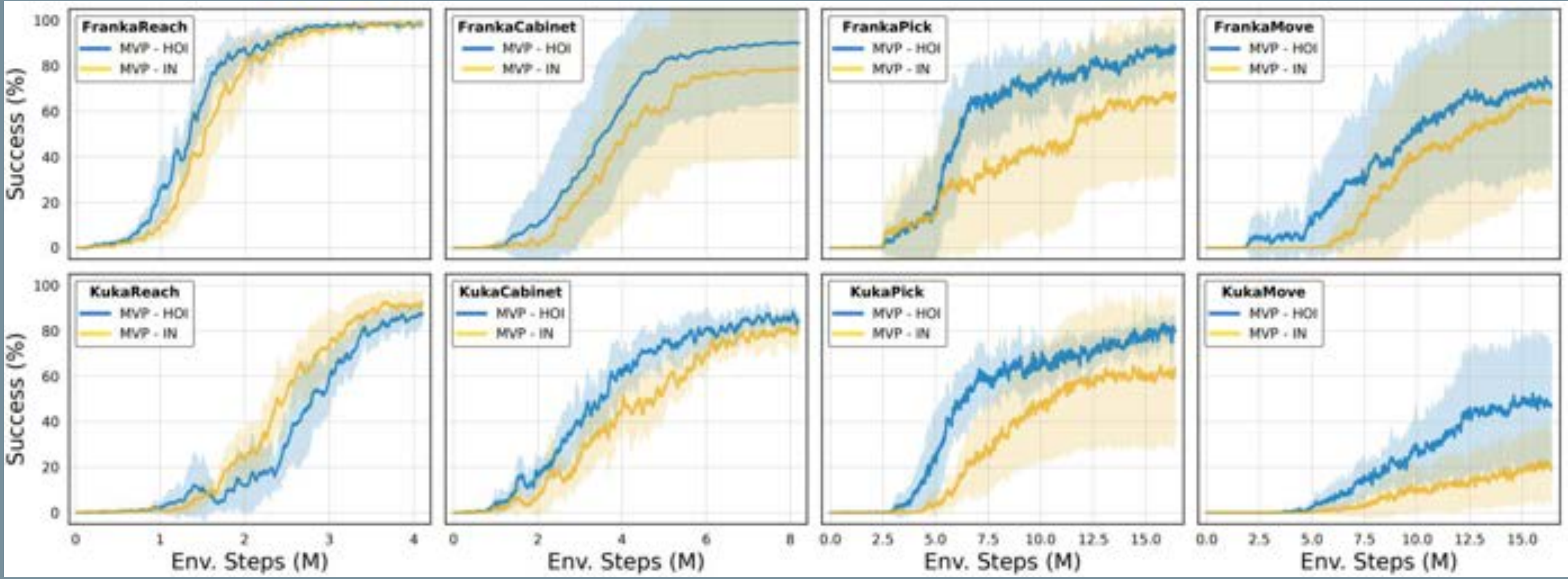
Radosavovic, Ilija, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. "Real-world robot learning with masked visual pre-training." arXiv preprint arXiv:2210.03109 (2022).

Case Study 5: MVP



Radosavovic, Ilija, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. "Real-world robot learning with masked visual pre-training." arXiv preprint arXiv:2210.03109 (2022).

Case Study 5: MVP



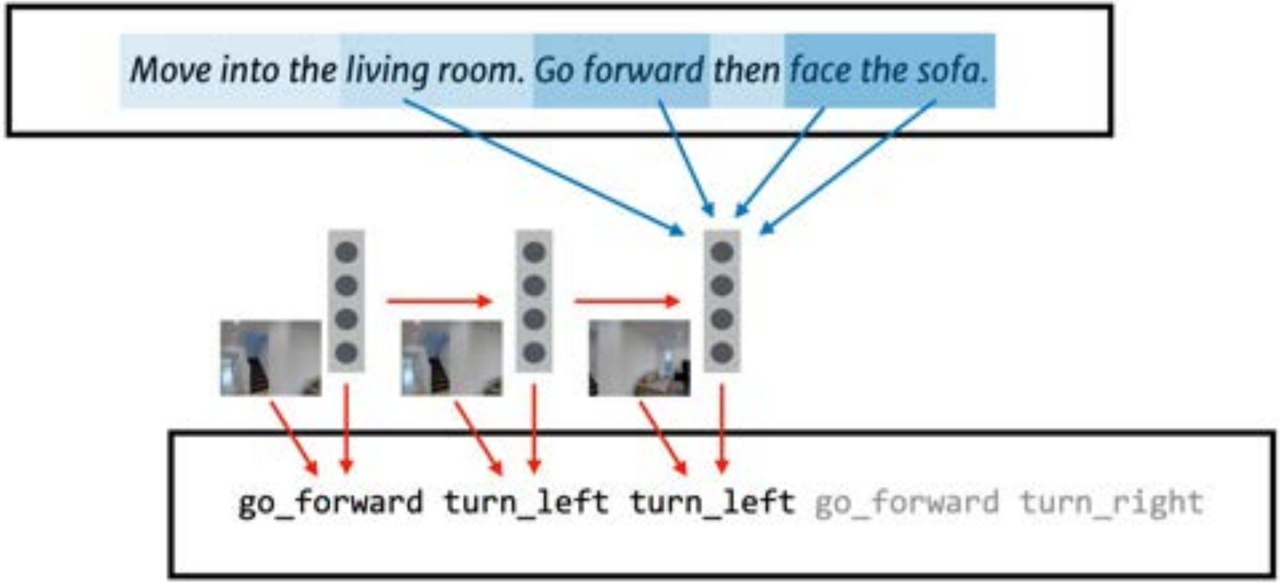
Radosavovic, Ilija, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. "Real-world robot learning with masked visual pre-training." arXiv preprint arXiv:2210.03109 (2022).

II. Robotics and Natural Language Processing

Concepts largely borrowed from Jacob Andrea’s slides [here]

II. Language and Decision Making

Move into the living room. Go forward then face the sofa.


Move into the living room. Go forward then face the sofa.
go_forward turn_left turn_left go_forward turn_right

Why is this hard?

Move into the living room. Go forward then face the sofa.

People don't talk about low-level actions!

Trying to learn π directly means simultaneously learning how words relate to goals and how goals specify actions.

Can we separate the two?

Why might mapping words to goals be easy?

Instruction givers are *cooperative*, instructions optimized to be *understandable* (subject to constraints).

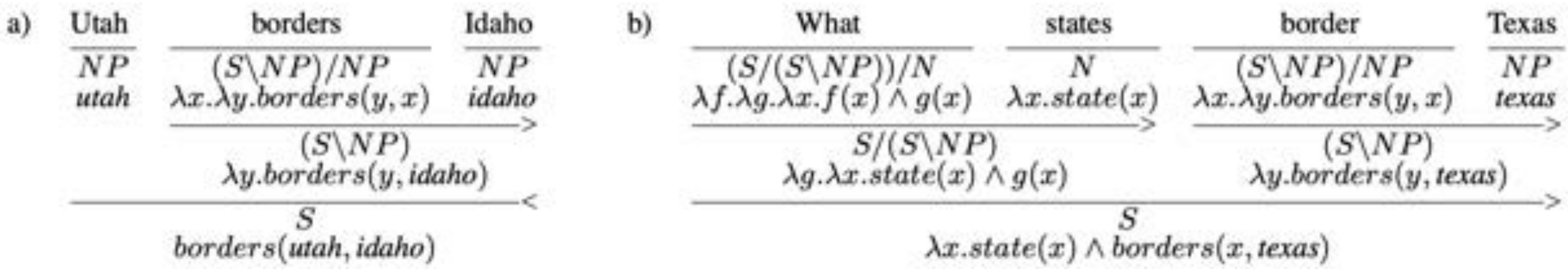
Language is well-suited (optimized?) for communicating the kinds of *goals* that *people care about*.

Insights from linguistic semantics:

1. The meaning of a sentence is a function from possible worlds to *truth values*.

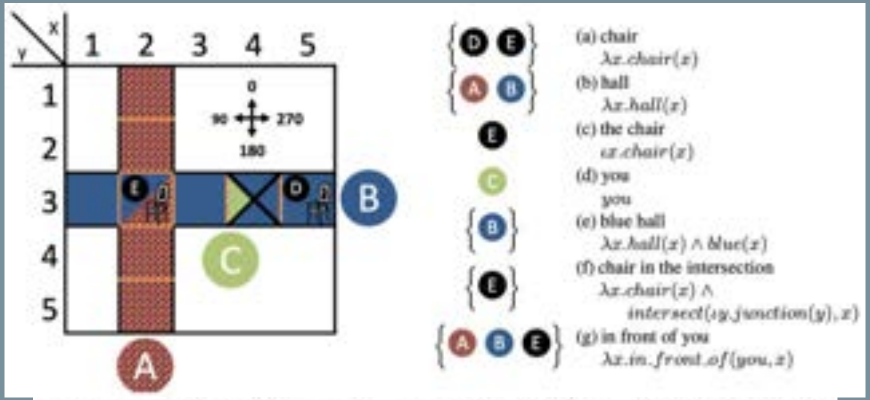
i.e "the blue ball is next to the open red door" is a boolean function that inspects the current state of the world…

2. Sentence meanings / functions are built compositionally from sentences themselves.

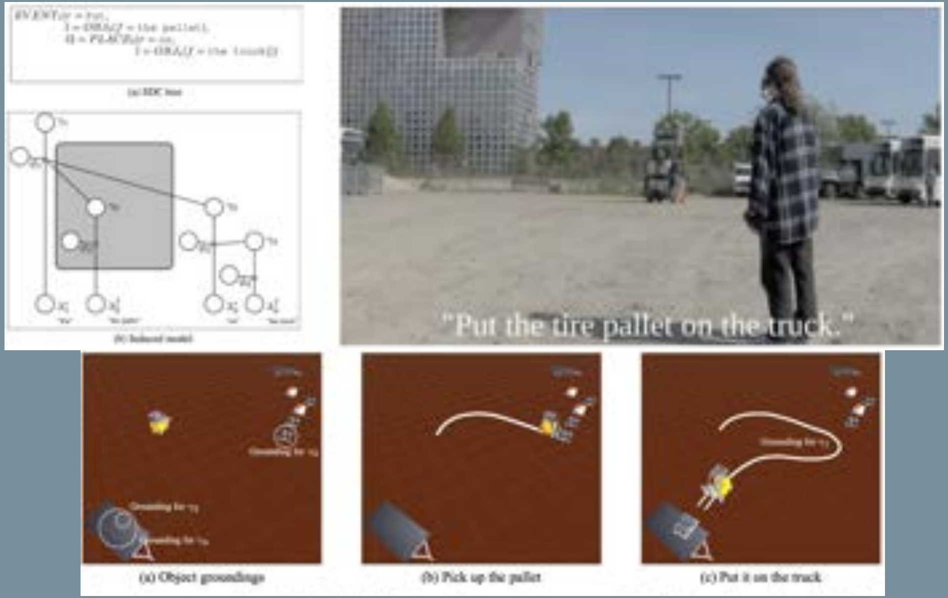


Tarski, Alfred. "Der Wahrheitsbegriff in den formalisierten Sprachen." Studia philosophica 1 (1936).
Montague, Richard. "The proper treatment of quantification in ordinary English." In Approaches to natural language: Proceedings of the 1970 Stanford workshop on grammar and semantics, pp. 221-242. Springer Netherlands, 1973.

Mapping Instructions to Actions



$$\max_{C,\theta} p(C,\theta|\tau,s) \propto p(\tau|C)p_\theta(C|s)p(\theta)$$



Learning a function from strings to goal predicates using task completion as feedback

Jointly Optimize for the Cost function + LM

C: Example specific cost function

θ: language understanding model

τ: dataset of trajectories

s: instruction

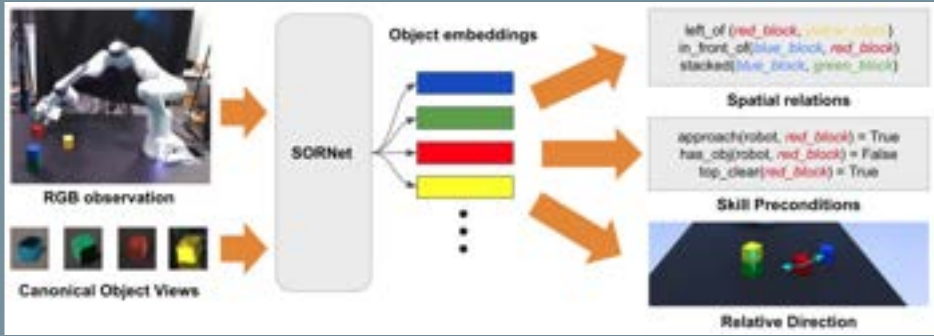
Inferring plans from natural language commands

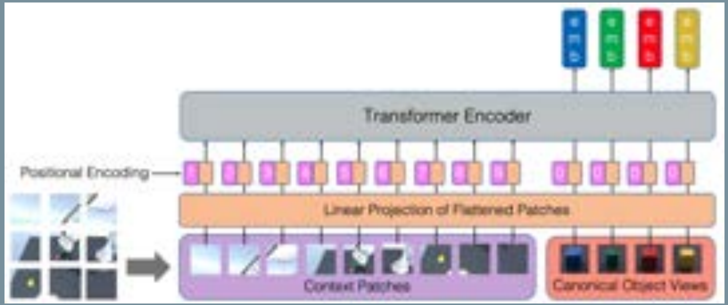
Artzi, Y. and Zettlemoyer, L. (2013). Weakly supervised learning of semantic parsers for mapping instructions to actions. Transactions of the Association for Computational Linguistics, 1:49–62.
Tellex, S., Kollar, T., Dickerson, S., Walter, M. R., Banerjee, A. G., Teller, S., and Roy, N. (2011). Understanding natural language commands for robotic navigation and mobile manipulation. In Proceedings of the Twenty-Fifth AAAI Conference on Articial Intelligence, AAAI'11, page 1507–1514. AAAI Press.

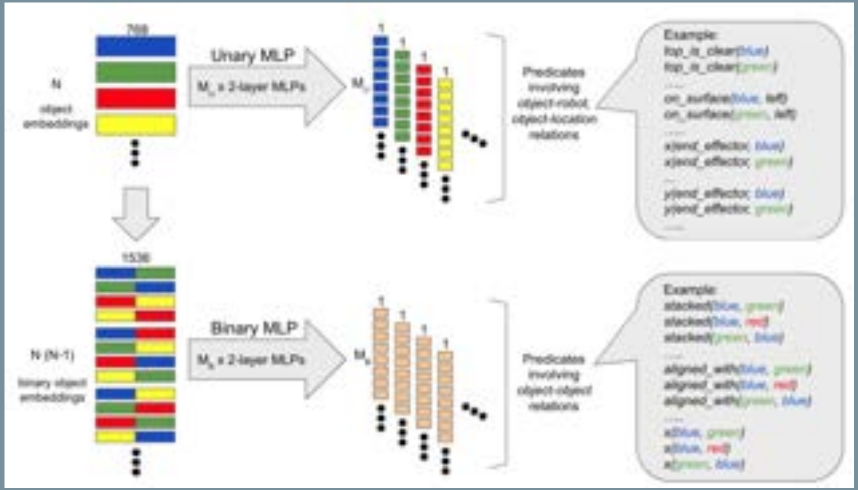
II. Language and Decision Making

- Language as a medium for task specification -> learn vision models to check constraints!

Case Study 6: Spatial Object-Centric Representations for Sequential Manipulation







Only unary and binary relations

Yuan, Wentao, Chris Paxton, Karthik Desingh, and Dieter Fox. "Sornet: Spatial object-centric representations for sequential manipulation." In Conference on Robot Learning, pp. 148-157. PMLR, 2022.

Case Study 6: Spatial Object-Centric Representations for Sequential Manipulation

How to use this?

1. Generate a state vector based on SORNet
2. A task and motion planner takes the state vector and desired goal (formulated as a list of predicate values to be satisfied, i.e. goto, grasp, etc.)
   a. Outputs a sequence of primitive skills

Yuan, Wentao, Chris Paxton, Karthik Desingh, and Dieter Fox. "Sornet: Spatial object-centric representations for sequential manipulation." In Conference on Robot Learning, pp. 148-157. PMLR, 2022.

Case Study 6: Spatial Object-Centric Representations for Sequential Manipulation

Method	ResNet18	ResNet18 (MV)	ResNet18 (P)	CLIP-ViT	CLIP-ViT (P)	SORNet (P)	SORNet (P MV)
Obj-Obj	0.4308	0.6068	0.9876	0.9875	0.6145	0.1679	**0.1458**
EE-Obj	0.3251	0.3464	0.5929	0.6544	0.4960	0.1962	**0.1777**

Table 4: Euclidean error on regression of continuous 3D unit vector between entities in the scene. The regressors are trained on 1000 examples with unseen objects. Methods labeled with P are pretrained on the Leonardo dataset. Methods labeled with MV use 3 views.

*Can also do viso-servoing:

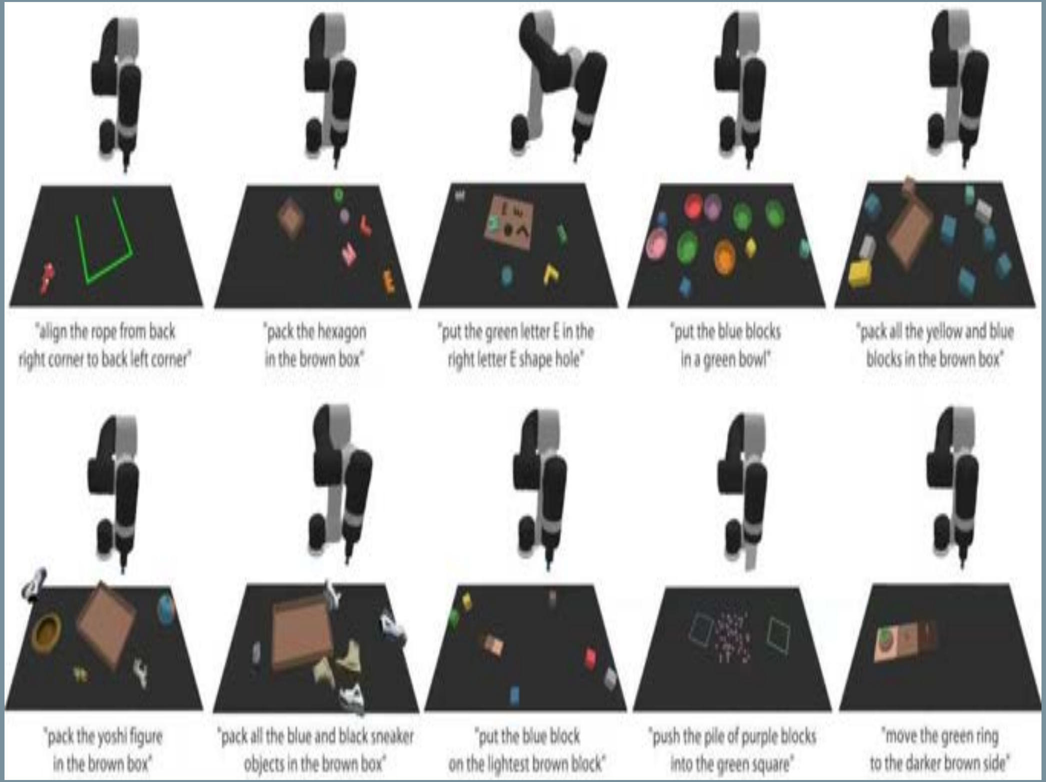
1. Trained a regressor on top of frozen SORNet embeddings to predict the continuous direction between two objects (Obj-Obj) or the direction the end effector should move to reach a certain object (EE-Obj)
2. Use the distance as the objective to guide the robot to reach a target object

Yuan, Wentao, Chris Paxton, Karthik Desingh, and Dieter Fox. "Sornet: Spatial object-centric representations for sequential manipulation." In Conference on Robot Learning, pp. 148-157. PMLR, 2022.

III. Unify Robotics, Vision, and NLP

- **Understanding Semantics with CLIP**
- Language models as planners
- Unify it all?

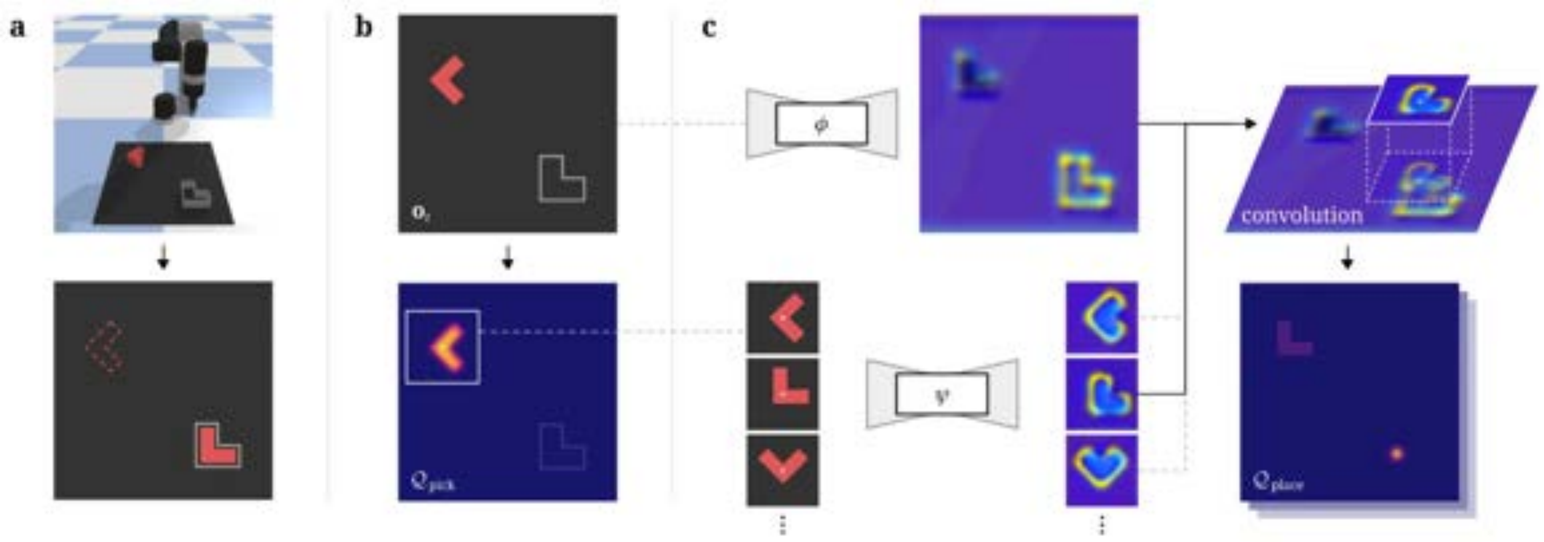
Case Study 7: CLIPORT



Trained from demonstration

Shridhar, Mohit, Lucas Manuelli, and Dieter Fox. "Cliport: What and where pathways for robotic manipulation." In Conference on Robot Learning, pp. 894-906. PMLR, 2022.
Zeng, Andy, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong et al. "Transporter networks: Rearranging the visual world for robotic manipulation." In Conference on Robot Learning, pp. 726-747. PMLR, 2021.

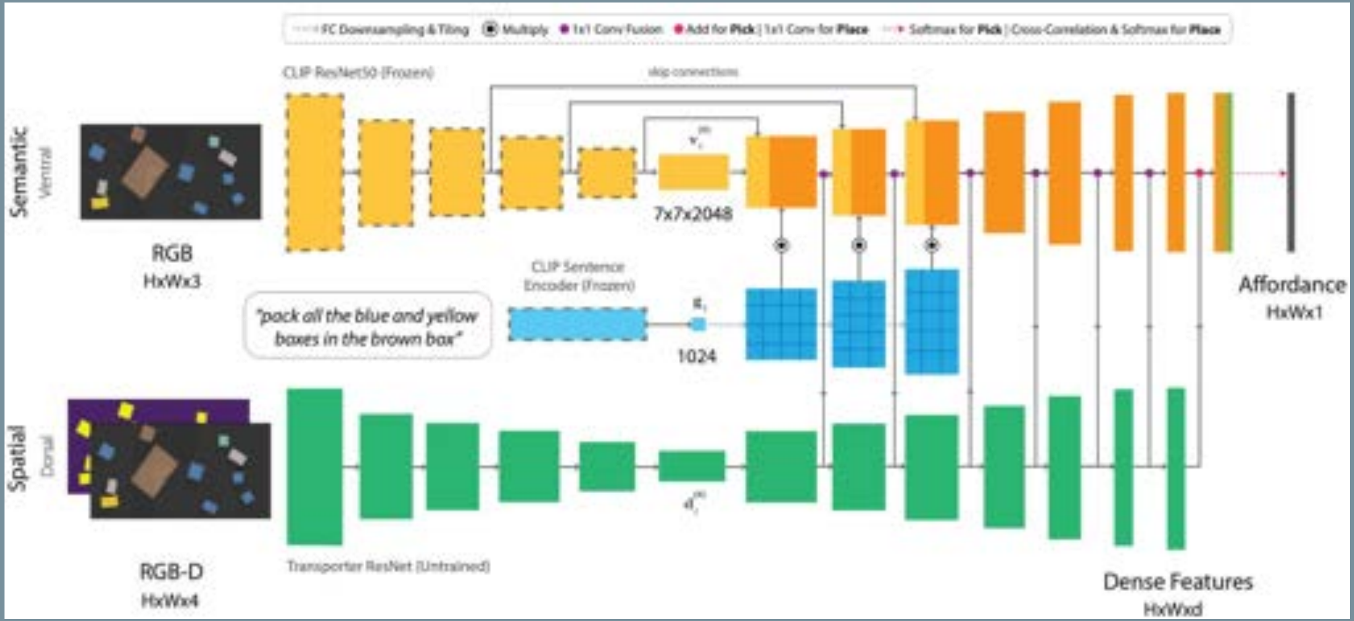
Case Study 7: CLIPORT



For pick-conditioned placing (c), deep feature template matching occurs with a local crop around the sampled pick as the exemplar. Rotations of the crop around the pick are used to decode the best placing rotation.

Shridhar, Mohit, Lucas Manuelli, and Dieter Fox. "Cliport: What and where pathways for robotic manipulation." In Conference on Robot Learning, pp. 894-906. PMLR, 2022.
Zeng, Andy, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong et al. "Transporter networks: Rearranging the visual world for robotic manipulation." In Conference on Robot Learning, pp. 726-747. PMLR, 2021.

Case Study 7: CLIPORT



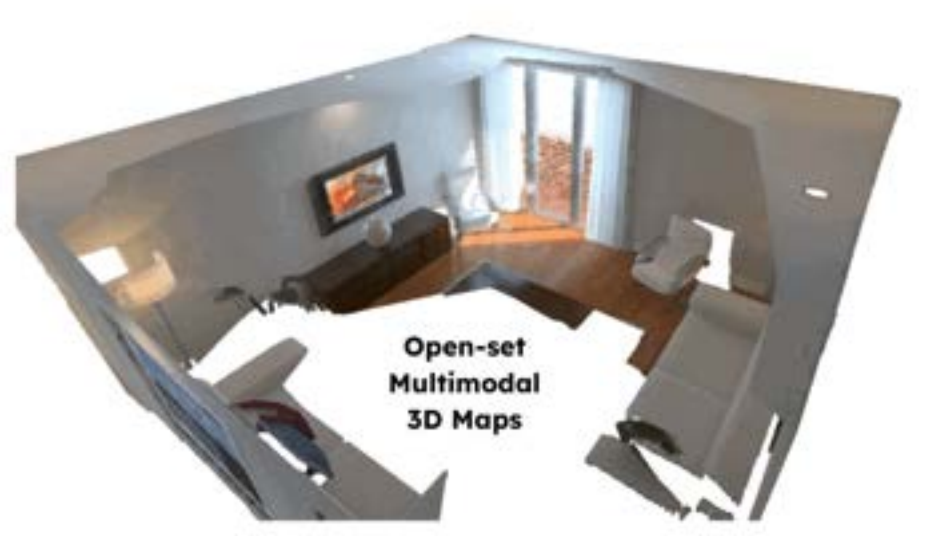
We consider the problem of learning a goal-conditioned policy π that outputs actions $\mathbf{a}_t$ given input $\gamma_t = (\mathbf{o}_t, \mathbf{l}_t)$ consisting of a visual observation $\mathbf{o}_t$ and an English language instruction $\mathbf{l}_t$:

$$\pi(\gamma_t) = \pi(\mathbf{o}_t, \mathbf{l}_t) \rightarrow \mathbf{a}_t = (\mathcal{T}_{\text{pick}}, \mathcal{T}_{\text{place}}) \in \mathcal{A} \quad (1)$$

Shridhar, Mohit, Lucas Manuelli, and Dieter Fox. "Cliport: What and where pathways for robotic manipulation." In Conference on Robot Learning, pp. 894-906. PMLR, 2022.
Zeng, Andy, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong et al. "Transporter networks: Rearranging the visual world for robotic manipulation." In Conference on Robot Learning, pp. 726-747. PMLR, 2021.

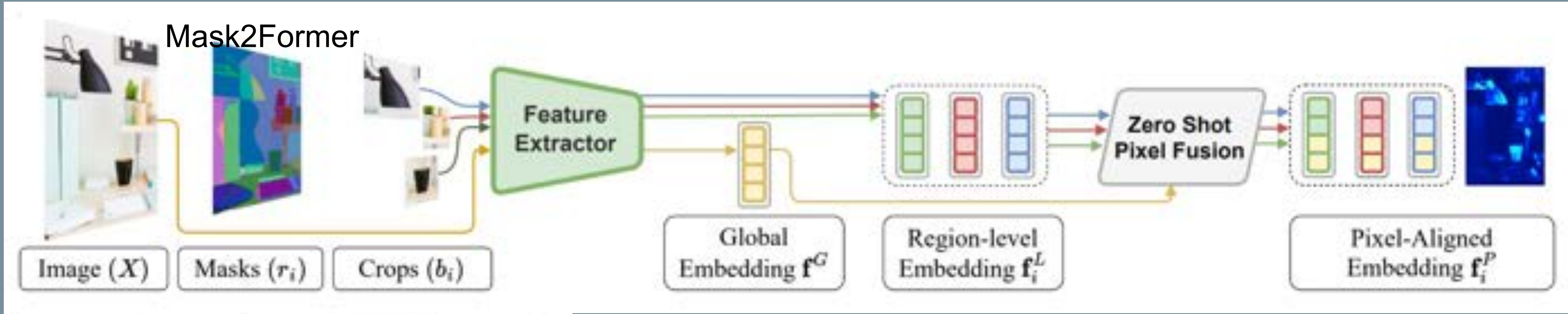
Case Study 8: ConceptFusion

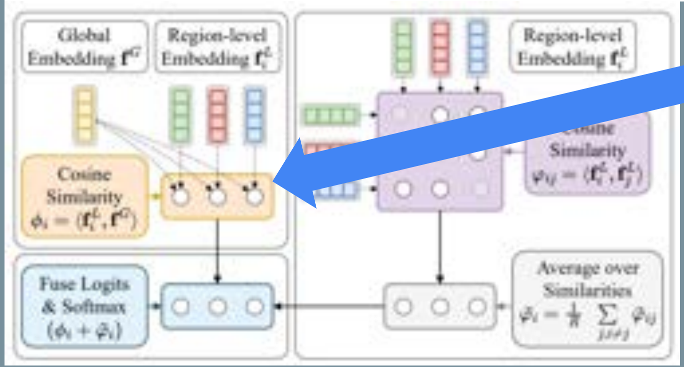
Multi-modal concepts mapped from 2D into 3D (i.e. point clouds)



Jatavallabhula, Krishna Murthy, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Shuang Li, Ganesh Iyer et al. "ConceptFusion: Open-set Multimodal 3D Mapping." arXiv preprint arXiv:2302.07241 (2023).

Case Study 8: ConceptFusion





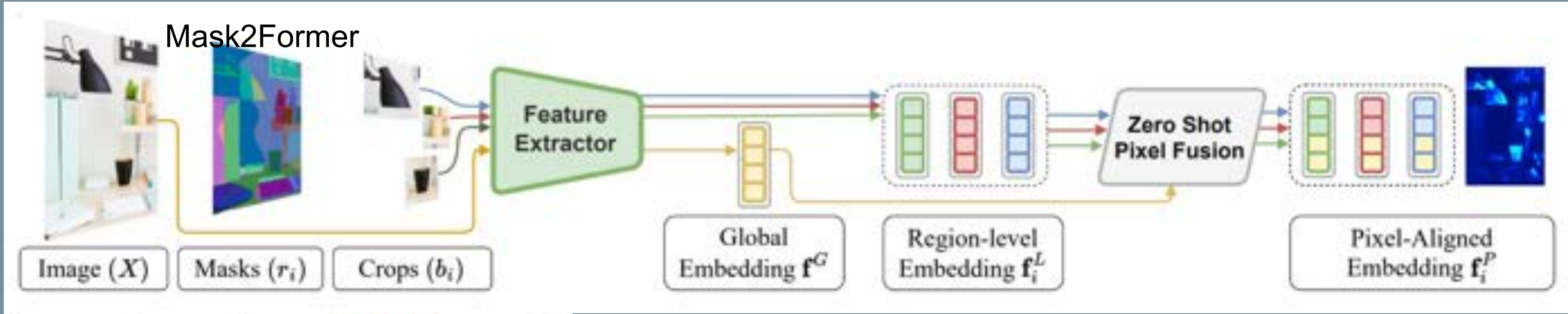
$$\phi_i = \left\langle \mathbf{f}_i^L, \mathbf{f}^G \right\rangle = \frac{(\mathbf{f}_i^L)^T \mathbf{f}^G}{\|\mathbf{f}_i^L\| \|\mathbf{f}^G\| + \epsilon}$$

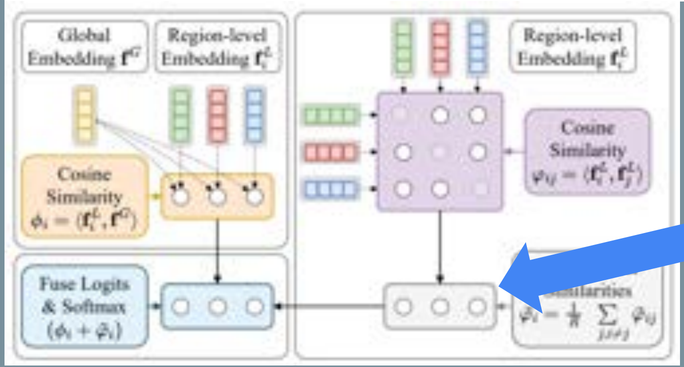
$$w_i = \frac{\exp\left(\frac{\phi_i + \bar{\varphi}_i}{\tau}\right)}{\sum_{i=1}^{R} \exp\left(\frac{\phi_i + \bar{\varphi}_i}{\tau}\right)}$$

$$\mathbf{f}_i^P = w_i \mathbf{f}^G + (1 - w_i) \mathbf{f}^L$$

Jatavallabhula, Krishna Murthy, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Shuang Li, Ganesh Iyer et al. "ConceptFusion: Open-set Multimodal 3D Mapping." arXiv preprint arXiv:2302.07241 (2023).

Case Study 8: ConceptFusion





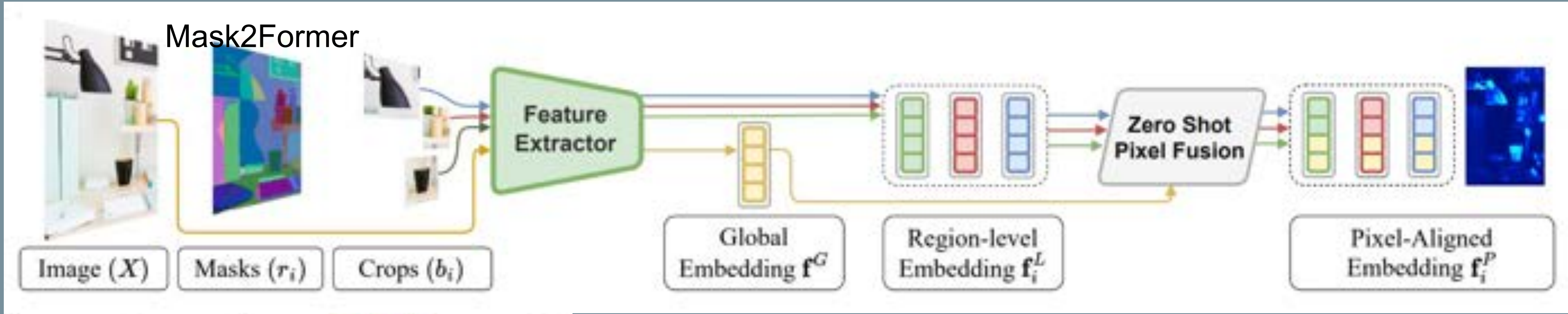
$$\phi_i = \left\langle \mathbf{f}_i^L, \mathbf{f}^G \right\rangle = \frac{(\mathbf{f}_i^L)^T \mathbf{f}^G}{\|\mathbf{f}_i^L\| \|\mathbf{f}^G\| + \epsilon}$$

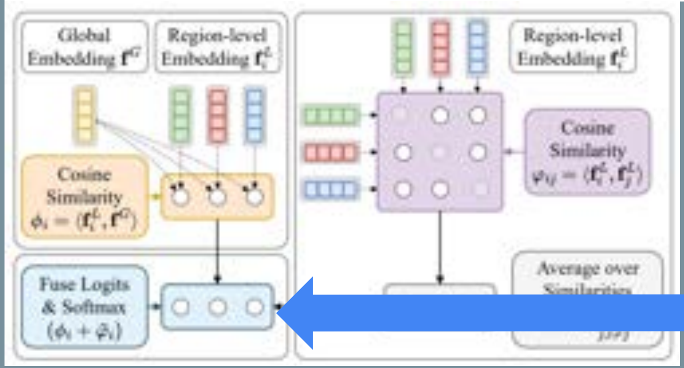
$$w_i = \frac{\exp\left(\frac{\phi_i + \bar\varphi_i}{\tau}\right)}{\sum_{i=1}^{R} \exp\left(\frac{\phi_i + \bar\varphi_i}{\tau}\right)}$$

$$\mathbf{f}_i^P = w_i \mathbf{f}^G + (1 - w_i) \mathbf{f}^L$$

Jatavallabhula, Krishna Murthy, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Shuang Li, Ganesh Iyer et al. "ConceptFusion: Open-set Multimodal 3D Mapping." arXiv preprint arXiv:2302.07241 (2023).

Case Study 8: ConceptFusion





$$\phi_i = \left\langle \mathbf{f}_i^L, \mathbf{f}^G \right\rangle = \frac{(\mathbf{f}_i^L)^T \mathbf{f}^G}{\|\mathbf{f}_i^L\| \|\mathbf{f}^G\| + \epsilon}$$

$$w_i = \frac{\exp\left(\frac{\phi_i + \bar{\varphi}_i}{\tau}\right)}{\sum_{i=1}^{R} \exp\left(\frac{\phi_i + \bar{\varphi}_i}{\tau}\right)}$$

$$\mathbf{f}_i^P = w_i \mathbf{f}^G + (1 - w_i) \mathbf{f}^L$$

Jatavallabhula, Krishna Murthy, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Shuang Li, Ganesh Iyer et al. "ConceptFusion: Open-set Multimodal 3D Mapping." arXiv preprint arXiv:2302.07241 (2023).

Case Study 8: ConceptFusion

perception and/or manipulation planning. Two sides of the workspace (see Fig. 8) are tagged *left* and *right* respectively (areas on either side of the table, as indicated by the green and yellow lines). For each set of objects, a goal instruction

Jatavallabhula, Krishna Murthy, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Shuang Li, Ganesh Iyer et al. "ConceptFusion: Open-set Multimodal 3D Mapping." arXiv preprint arXiv:2302.07241 (2023).

III. Unify Robotics, Vision, and NLP

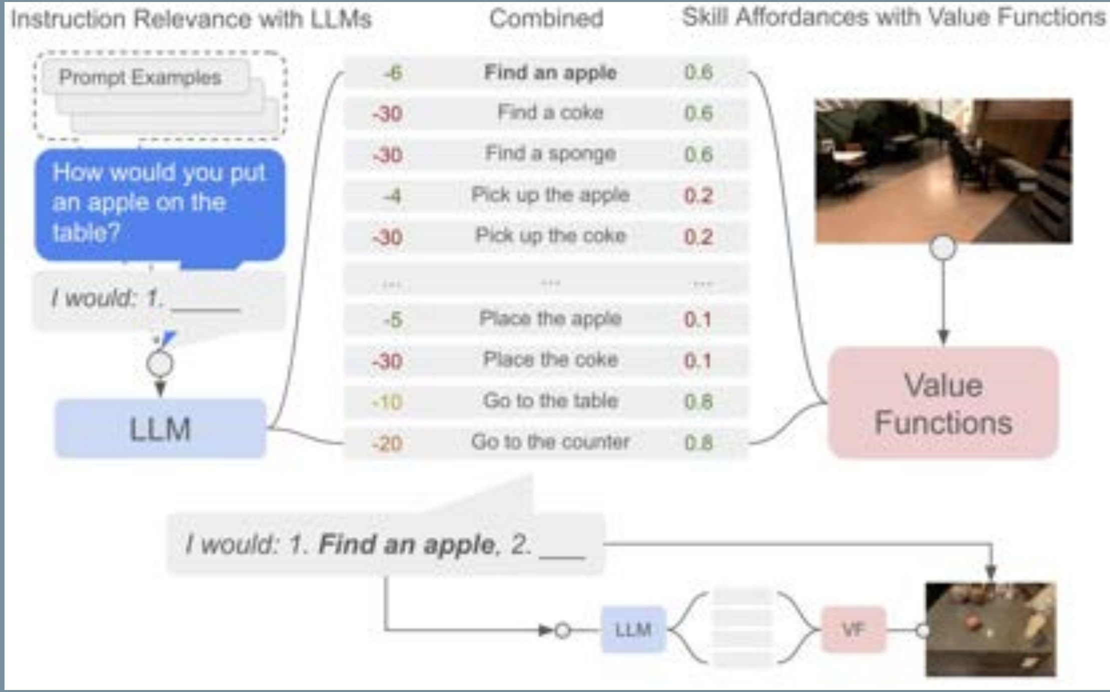
- Understanding Semantics with CLIP
- **Language models as planners**
- Unify it all?

Case Study 9: SayCan

Problems: (1) task-grounding (i.e., a skill language description) and (2) world-grounding (i.e., skill feasibility in the current state).
Solution: (1) Image based behavior cloning for each skill
(2) train affordance functions: a particular skill -> success probability in the current state

Ahn, Michael, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn et al. "Do as i can, not as i say: Grounding language in robotic affordances." arXiv preprint arXiv:2204.01691 (2022).

Case Study 9: SayCan



Probability of a language description of a skill π making progress towards executing the instruction i

$$\pi = \arg\max_{\pi \in \Pi} p(c_\pi | s, \ell_\pi) p(\ell_\pi | i)$$

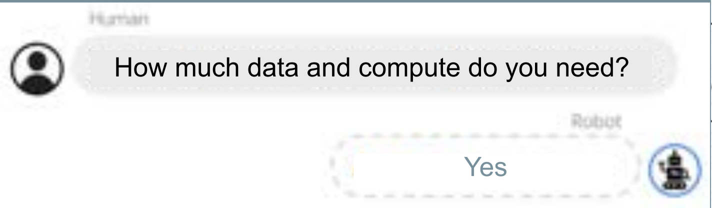
Probability of completing a subtask conditioned on the current state (aka affordance)

Ahn, Michael, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn et al. "Do as i can, not as i say: Grounding language in robotic affordances." arXiv preprint arXiv:2204.01691 (2022).

Case Study 9: SayCan

Ahn, Michael, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn et al. "Do as i can, not as i say: Grounding language in robotic affordances." arXiv preprint arXiv:2204.01691 (2022).

Case Study 9: SayCan

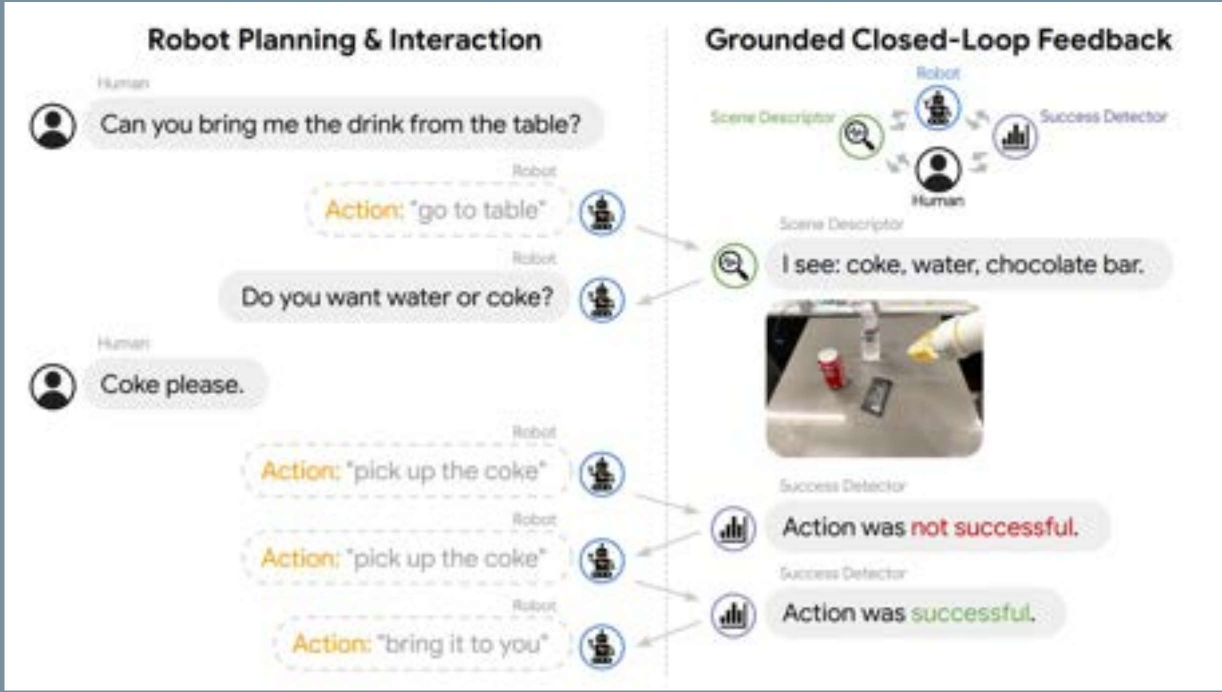


BC training. We use 68000 teleoperated demonstrations that were collected over the course of 11 months using a fleet of 10 robots. The operators use VR headset controllers to track the motion of their hand, which is then mapped onto the robot's end-effector pose. The operators can also use a joystick to move the robot's base. We expand the demonstration dataset with 276000 autonomous episodes of learned policies which are later success-filtered and included in BC training, resulting in an additional 12000 successful episodes. To additionally process the data, we also ask the raters to mark the episodes as unsafe (i.e., if the robot collided with the environment), undesirable (i.e., if the robot perturbed objects that were not relevant to the skill) or infeasible (i.e., if the skill cannot be done or is already accomplished). If any of these conditions are met, the episode is excluded from training.

The RL model is trained using 16 TPUv3 chips and for about 100 hours, as well as a pool of 3000 CPU workers to collect episodes and another 3000 CPU workers to compute target Q-values.

Ahn, Michael, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn et al. "Do as i can, not as i say: Grounding language in robotic affordances." arXiv preprint arXiv:2204.01691 (2022).

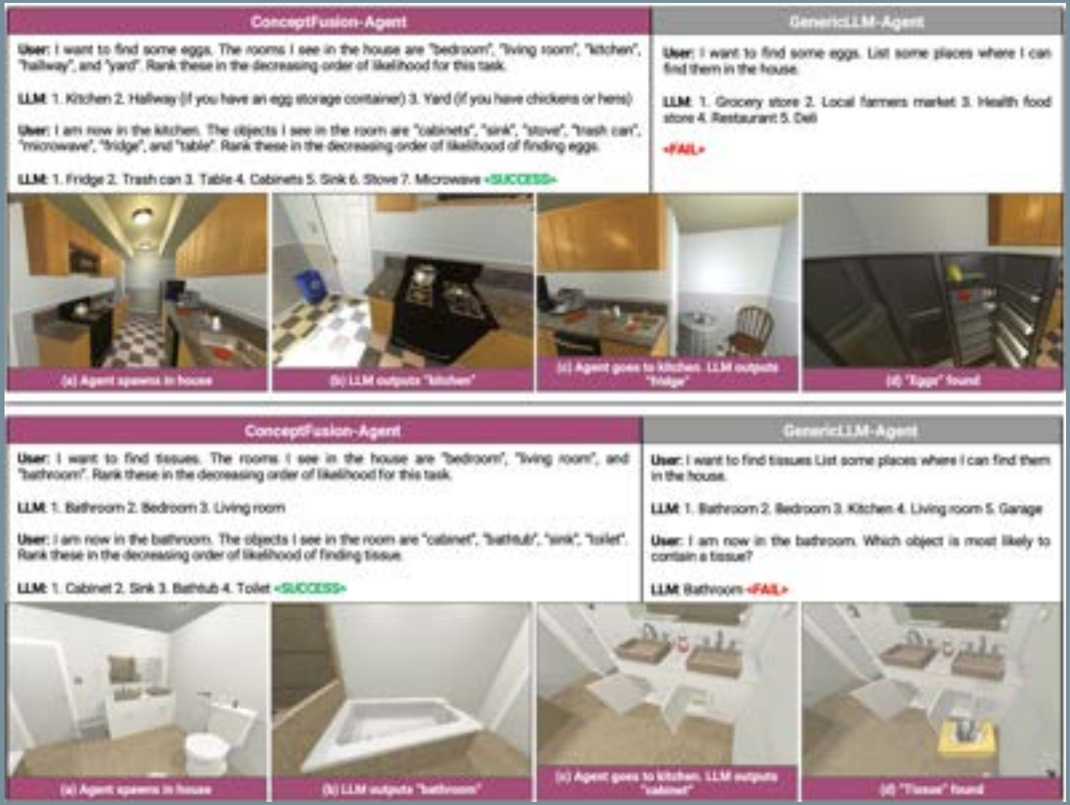
Case Study 10: Inner Monologue



Introduce feedback!

Huang, Wenlong, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng et al. "Inner monologue: Embodied reasoning through planning with language models." arXiv preprint arXiv:2207.05608 (2022).

Case Study 11: ConceptFusion revisited



Can we improve grounding of general LLM?
Solution: provide more options and context!
(prompt tuning)

Jatavallabhula, Krishna Murthy, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Shuang Li, Ganesh Iyer et al. "ConceptFusion: Open-set Multimodal 3D Mapping." arXiv preprint arXiv:2302.07241 (2023).

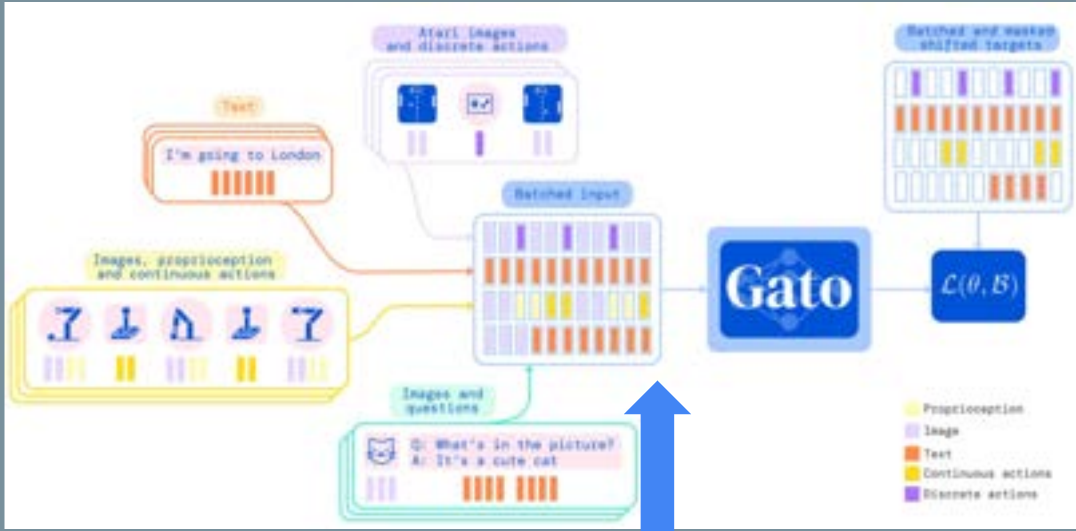
III. Unify Robotics, Vision, and NLP

- Language models as planners
- **Unify it all?**

Case Study 13: A Generalist Agent

Can one model rule all? *Yes

Model outputs a distribution over the next discrete token.



$$\mathcal{L}(\theta, \mathcal{B}) = -\sum_{b=1}^{|\mathcal{B}|} \sum_{l=1}^{L} m(b,l) \log p_\theta \left(s_l^{(b)} | s_1^{(b)}, \ldots, s_{l-1}^{(b)} \right)$$

Tokenize everything

Invert the tokenization

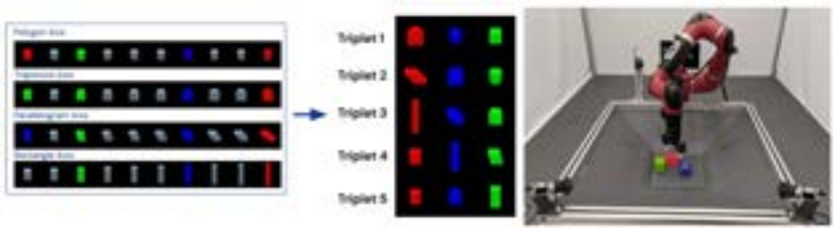
*May not be the best performing agent in each subcategories yet.

Reed, Scott, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez et al. "A generalist agent." arXiv preprint arXiv:2205.06175 (2022).

Case Study 13: A Generalist Agent

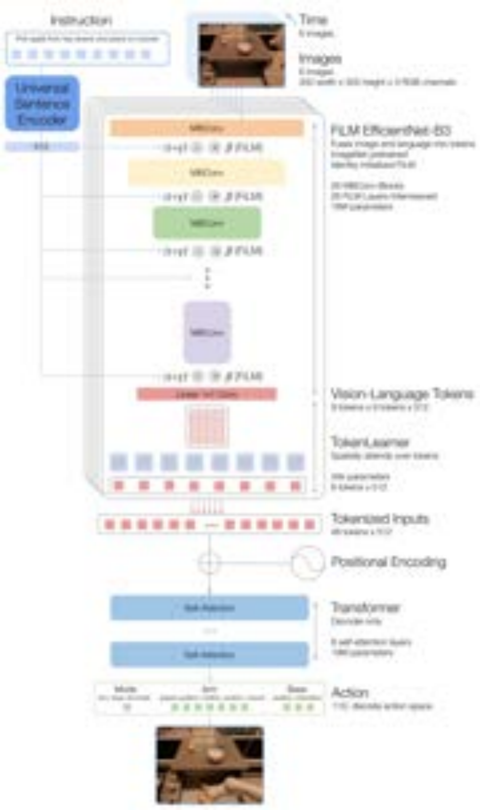
Control environment	Tasks	Episodes	Approx. Tokens	Sample Weight
DM Lab	254	16.4M	194B	9.35%
ALE Atari	51	63.4K	1.26B	9.5%
ALE Atari Extended	28	28.4K	565M	10.0%
Sokoban	1	27.2K	298M	1.33%
BabyAI	46	4.61M	22.8B	9.06%
DM Control Suite	30	395K	22.5B	4.62%
DM Control Suite Pixels	28	485K	35.5B	7.07%
DM Control Suite Random Small	26	10.6M	313B	3.04%
DM Control Suite Random Large	26	26.1M	791B	3.04%
Meta-World	45	94.6K	3.39B	8.96%
Procgen Benchmark	16	1.6M	4.46B	5.34%
RGB Stacking simulator	1	387K	24.4B	1.33%
RGB Stacking real robot	1	15.7K	980M	1.33%
Modular RL	38	843K	69.6B	8.23%
DM Manipulation Playground	4	286K	6.58B	1.68%
Playroom	1	829K	118B	1.33%
Total	596	63M	1.5T	85.3%

Vision / language dataset	Sample Weight
MassiveText	6.7%
M3W	4%
ALIGN	0.67%
MS-COCO Captions	0.67%
Conceptual Captions	0.67%
LTIP	0.67%
OKVQA	0.67%
VQAV2	0.67%
Total	14.7%



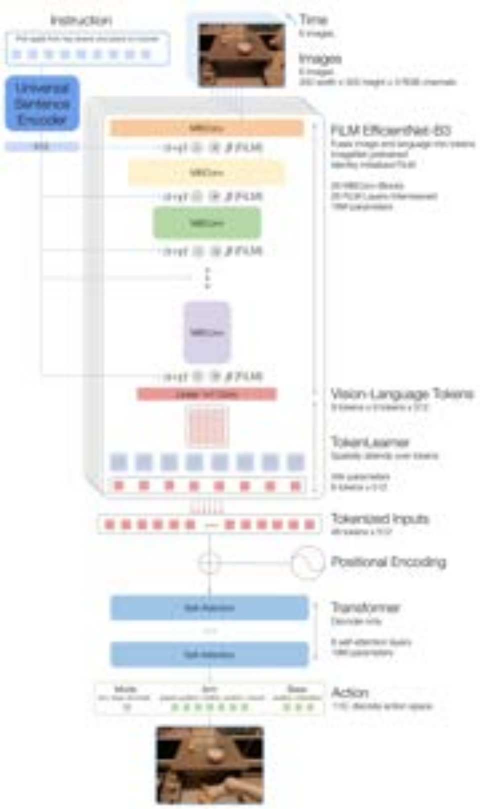
Reed, Scott, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez et al. "A generalist agent." arXiv preprint arXiv:2205.06175 (2022).

Case Study 14: RT1



Brohan, Anthony, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan et al. "Rt-1: Robotics transformer for real-world control at scale." arXiv preprint arXiv:2212.06817 (2022).

Case Study 14: RT1



Not tokenized images (Extracted by EfficientNetB3)

Training Data

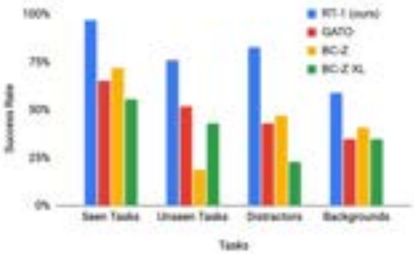
- Trained on 130k tele-operation demonstrations over 13 robots and 744 tasks.

Skill	Count	Description	Example Instruction
Pick `Object`	130	Lift the object off the surface	pick iced tea can
Move `Object` Near `Object`	337	Move the first object near the second	move pepsi can near rxbar blueberry
Place `Object` Upright	8	Place an elongated object upright	place water bottle upright
Knock `Object` Over	8	Knock an elongated object over	knock redbull can over
Open / Close `Drawer`	6	Open or close any of the cabinet drawers	open the top drawer
Place `Object` into `Receptacle`	84	Place an object into a receptacle	place brown chip bag into white bowl
Pick `Object` from `Receptacle` and Place on the Counter	162	Pick an object up from a location and then place it on the counter	pick green jalapeno chip bag from paper bowl and place on counter
Additional tasks	9	Skills trained for realistic, long instructions	pull napkin out of dispenser
Total	744		

Evaluation Data

- Evaluated on real-world randomized scenes and over 3000 total rollouts in the environment it was trained on as well as two new office kitchen environments.

Model	Seen Tasks	Unseen Tasks	Distractors	Backgrounds
Gato (Reed et al., 2022)	65	52	43	35
BC-Z (Jang et al., 2021)	72	19	47	41
BC-Z XL	56	43	23	35
RT-1 (ours)	**97**	**76**	**83**	**59**

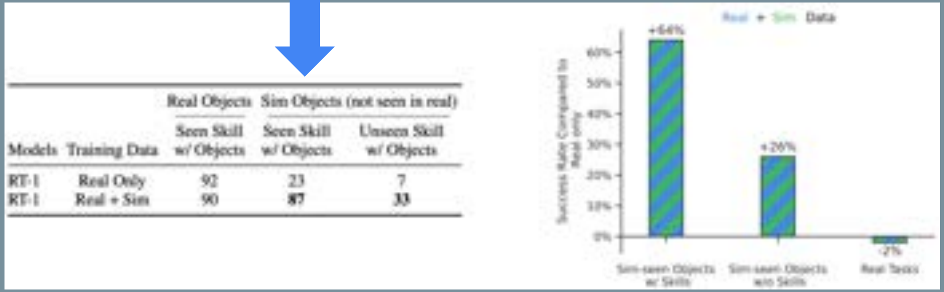


Brohan, Anthony, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan et al. "Rt-1: Robotics transformer for real-world control at scale." arXiv preprint arXiv:2212.06817 (2022).

Case Study 14: RT1

Use RetinaGan

From MT-Opt



Models	Training Data	Real Objects Seen Skill w/ Objects	Sim Objects (not seen in real) Seen Skill w/ Objects	Sim Objects (not seen in real) Unseen Skill w/ Objects
RT-1	Real Only	92	23	7
RT-1	Real + Sim	90	**87**	**33**

Results on Sim2Real Transfer

Models	ng Data	Classroom eval	Bin-picking eval
RT-1	Kuka bin-picking data + EDR data	90	**39**
RT-1	EDR only data	92	22
RT-1	Kuka bin-picking only data	0	0

Results on Mixed Dataset

“Classroom eval: “pick” and “move to” skills

Little degradation in performance on mixed dataset training

Brohan, Anthony, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan et al. "Rt-1: Robotics transformer for real-world control at scale." arXiv preprint arXiv:2212.06817 (2022).

Case Study 14: RT1 with Saycan

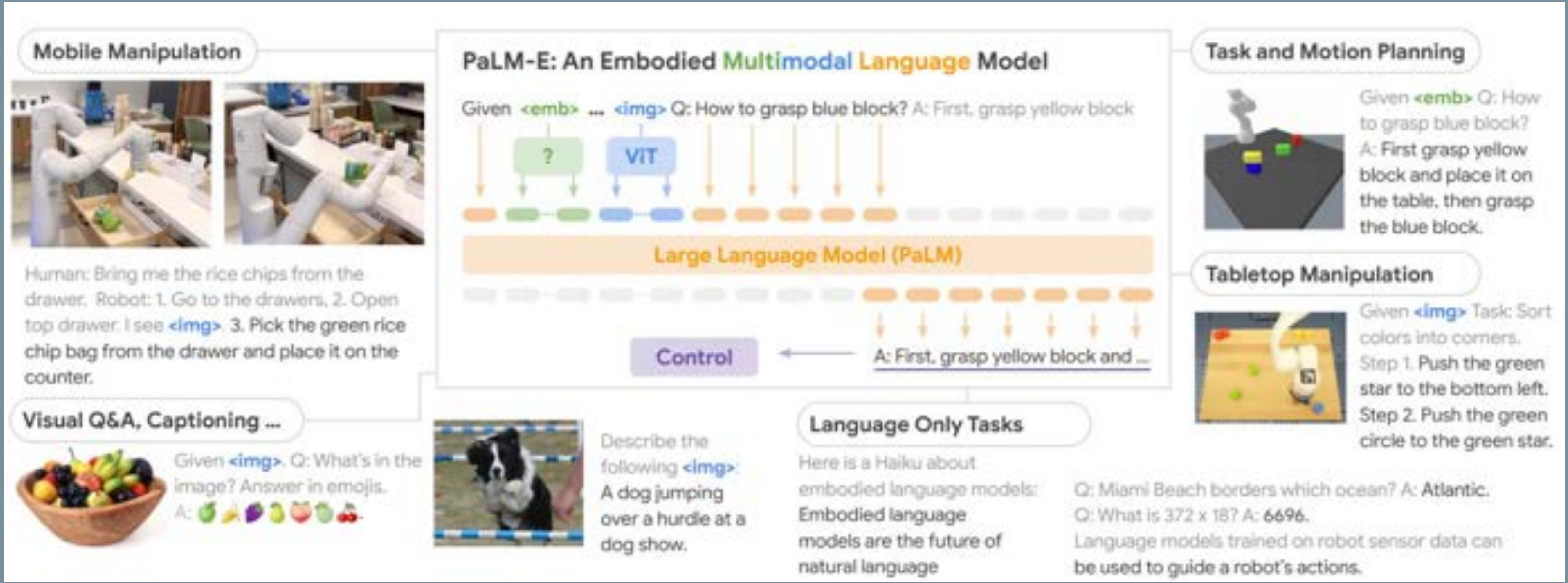
	SayCan tasks in Kitchen1		SayCan tasks in Kitchen2	
	Planning	Execution	Planning	Execution
Original SayCan (Ahn et al., 2022)*	73	47	-	-
SayCan w/ Gato (Reed et al., 2022)	87	33	87	0
SayCan w/ BC-Z (Jang et al., 2021)	87	53	87	13
SayCan w/ RT-1 (ours)	87	**67**	87	**67**

We see that RT-1 achieves a 67% execution success rate in Kitchen1, and is better than other baselines. Due to the generalization difficulty presented by the new unseen kitchen, the performance of SayCan with Gato and SayCan with BCZ shapely falls, while RT-1 does not show a visible drop.

Brohan, Anthony, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan et al. "Rt-1: Robotics transformer for real-world control at scale." arXiv preprint arXiv:2212.06817 (2022).

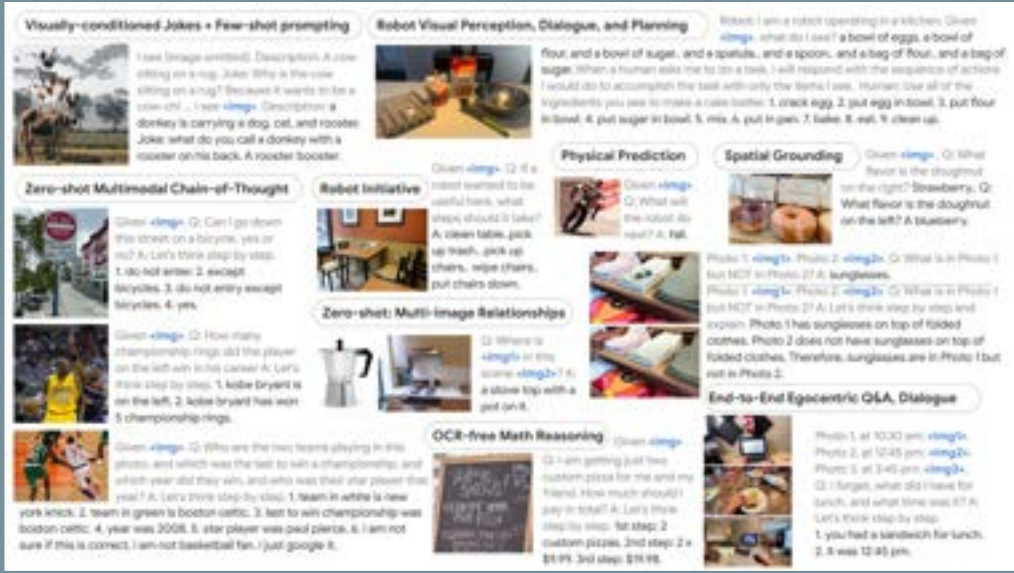
Case Study 15: PaLM-E

A (540B PaLM+22B ViT) multimodal PaLM that do the following

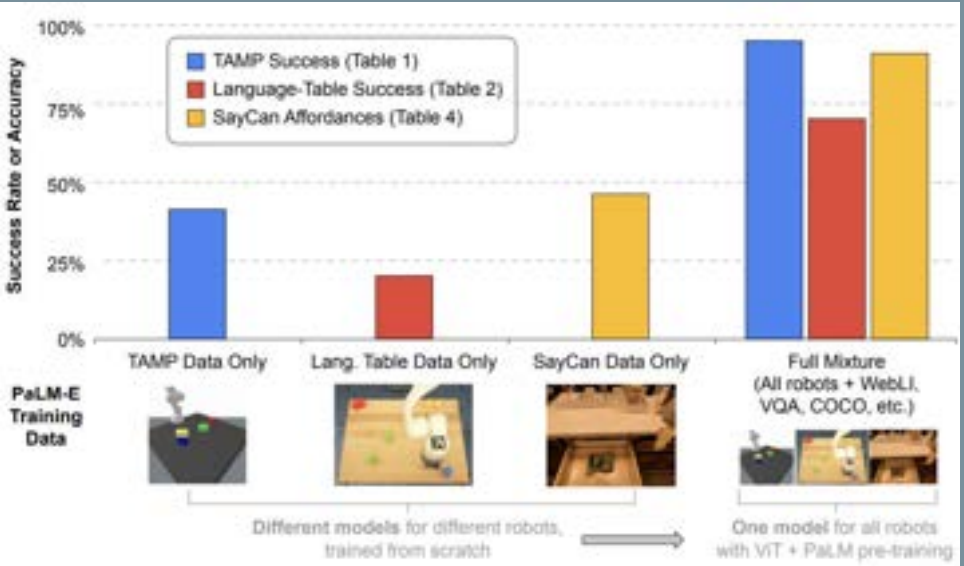


Case Study 15: PaLM-E

The *inputs* to PaLM-E consist of text and (multiple) continuous observations. The multimodal tokens corresponding to these observations are interleaved with the text to form *multi-modal sentences*. An example of such a multi-modal sentence is Q: What happened between <img_1> and <img_2>? where <img_i> represents an embedding of an image. The *output* of PaLM-E is text generated auto-regressively by the model, which could be an answer to a question, or a sequence of decisions produced by PaLM-E in textual form that should be executed by a robot. When PaLM-E is tasked with producing decisions or plans, we assume that there exists a low-level policy or planner that can translate these decisions into low-level actions. Prior work has discussed a variety of ways to train such low-level policies (Lynch & Sermanet, 2020; Brohan et al., 2022), and we use these prior methods directly without modification. In the following, we describe our approach more formally.



Case Study 15: PaLM-E



Our three robot environments (Fig. 1) include a Task and Motion Planning (TAMP) domain where a robot has to manipulate (grasp and stack) objects, a table-top pushing environment, and a mobile manipulation domain. In each

Downstream

1. TAMP Simulation Controller
2. Interactive language: Talking to robots in real time.
3. RT1

Case Study 15: PaLM-E

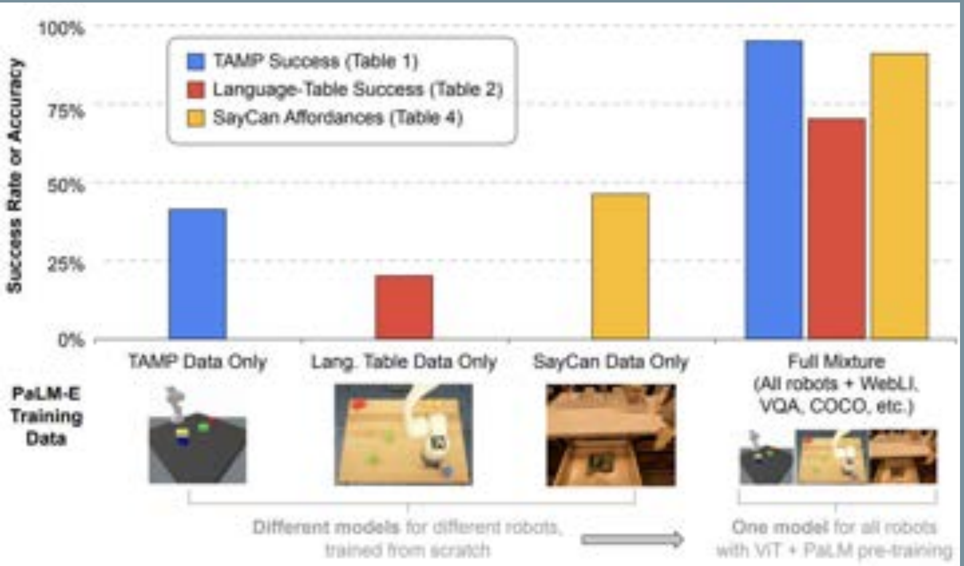
- p_1: grasping. Example prompt: Given <img>. Q: How to grasp the green object?. Target: A: First grasp the orange object and place it on the table, then grasp the green object.
- p_2: stacking. Example prompt: Given <img>. Q: How to stack the white object on top of the red object?. Target: A: First grasp the green object and place it on the table, then grasp the white object and place it on the red object.

TAMP Task

Table top pushing environment

RT1 Environments

Case Study 15: PaLM-E



Our three robot environments (Fig. 1) include a Task and Motion Planning (TAMP) domain where a robot has to manipulate (grasp and stack) objects, a table-top pushing environment, and a mobile manipulation domain. In each

Downstream

1. TAMP Simulation Controller
2. Interactive language: Talking to robots in real time.
3. RT1

Case Study 15: PaLM-E

	Object-centric	LLM pre-train	Embodied VQA q_1	q_2	q_3	q_4	Planning p_1	p_2
SayCan (oracle afford.) (Ahn et al., 2022)		✓	-	-	-	-	38.7	33.3
PaLI (zero-shot) (Chen et al., 2022)		✓	-	0.0	0.0	-	-	-
PaLM-E (ours) w/ input enc:								
State	✓(GT)	✗	99.4	89.8	90.3	88.3	45.0	46.1
State	✓(GT)	✓	**100.0**	96.3	95.1	93.1	55.9	49.7
ViT + TL	✓(GT)	✓	34.7	54.6	74.6	91.6	24.0	14.7
ViT-4B single robot	✗	✓	-	45.9	78.4	92.2	30.6	32.9
ViT-4B full mixture	✗	✓	-	70.7	93.4	92.1	74.1	74.6
OSRT (no VQA)	✓	✓	-	-	-	-	71.9	75.1
OSRT	✓	✓	99.7	**98.2**	**100.0**	**93.7**	**82.5**	**76.2**

TAMP Task

Zero-shot Baselines						Task 1			Task 2			Task 3		
SayCan (oracle afford.) (Ahn et al., 2022)							0.0			-			-	
PaLI (Chen et al., 2022)							0.0			-			-	
PaLM-E-	trained on	from scratch	LLM+ViT pretrain	LLM frozen	Task finetune	*# Demos* *10*	*20*	*40*	*10*	*20*	*40*	*10*	*20*	*80*
12B	Single robot	✓	✗	n/a	✓	20.0	30.0	50.0	2.5	6.3	2.5	11.3	16.9	28.3
12B	Full mixture	✗	✓	✓	✗	-	-	20.0	-	-	36.3	-	-	29.4
12B	Full mixture	✗	✓	✗	✗	-	-	80.0	-	-	57.5	-	-	50.0
12B	Full mixture	✗	✓	✗	✓	**70.0**	**80.0**	80.0	**31.3**	**58.8**	**58.8**	**57.5**	**54.4**	56.3
84B	Full mixture	✗	✓	✗	✗	-	-	**90.0**	-	-	53.8	-	-	**64.4**

Table 2: Results on planning tasks in the simulated environment from Lynch et al. (2022).

Table top pushing environment

Task 1. Q: There is a block that is closest to {*i.e., top right corner*}. Push that block to the other block of the same color.

Task 2. Q: How to sort the blocks by colors into corners?

Task 3. Q: How to push all the blocks that are on the {*left/right*} side together, without bringing over any of the blocks that are on the {*right/left*} side?

Case Study 15: PaLM-E

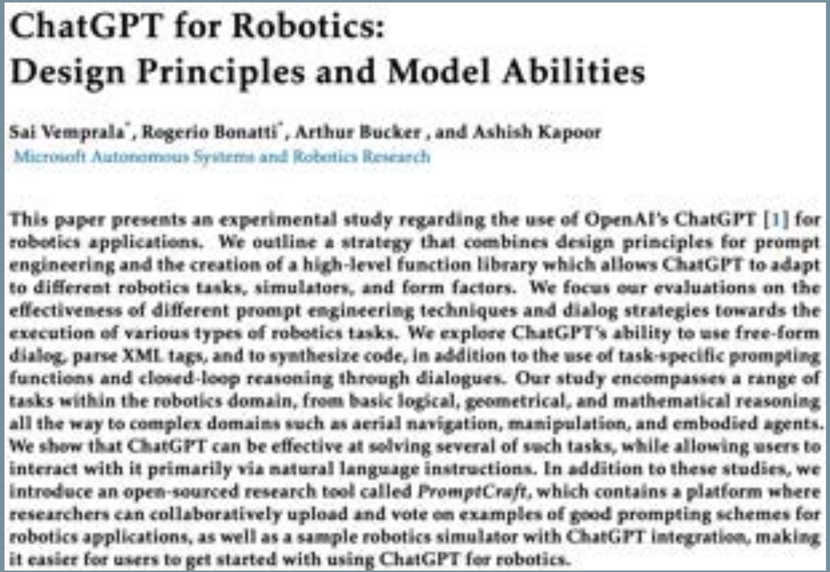
Baselines				Failure det.	Affordance
PaLI (Zero-shot) (Chen et al., 2022)				0.73	0.62
CLIP-FT (Xiao et al., 2022)				0.65	-
CLIP-FT-hindsight (Xiao et al., 2022)				0.89	-
QT-OPT (Kalashnikov et al., 2018)				-	0.63
PaLM-E-12B trained on	from scratch	LLM+ViT pretrain	LLM frozen		
Single robot	✓	✗	n/a	0.54	0.46
Single robot	✗	✓	✓	**0.91**	0.78
Full mixture	✗	✓	✓	**0.91**	0.87
Full mixture	✗	✓	✗	0.77	**0.91**

RT1 Environments

I. Robotics and Vision
II. Robotics and NLP
III. Unify it all?
IV. Where does this leave us?

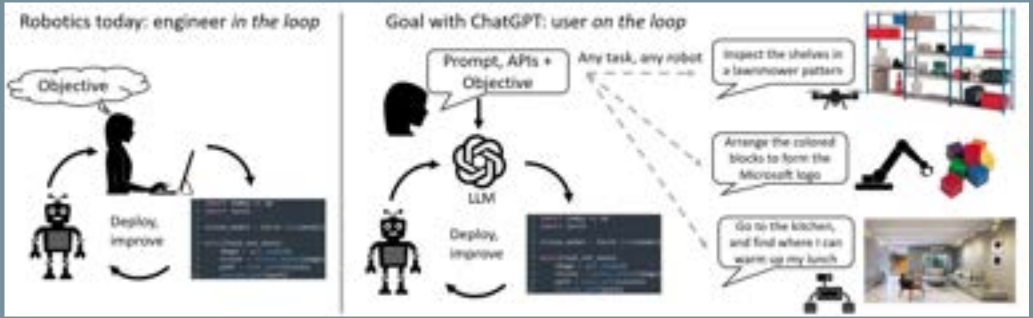
Generalization in robotics vs Specialist Systems: Can we unify them?

Low level stuff?


ChatGPT for Robotics: Design Principles and Model Abilities

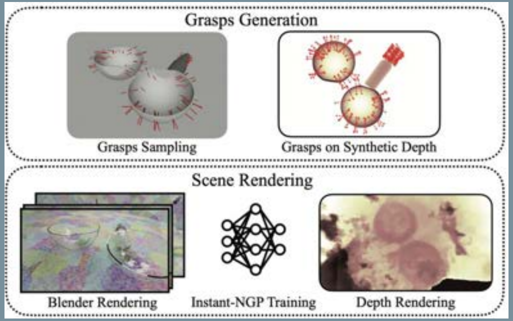
Sai Vemprala*, Rogerio Bonatti*, Arthur Bucker, and Ashish Kapoor
Microsoft Autonomous Systems and Robotics Research

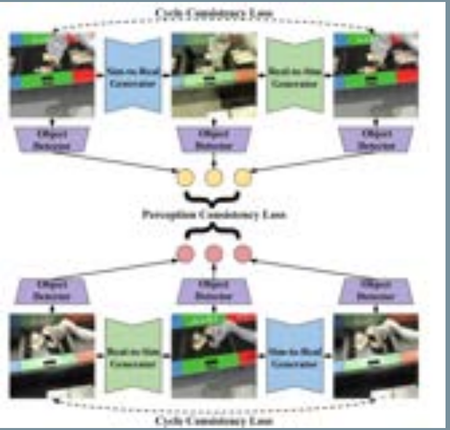
This paper presents an experimental study regarding the use of OpenAI's ChatGPT [1] for robotics applications. We outline a strategy that combines design principles for prompt engineering and the creation of a high-level function library which allows ChatGPT to adapt to different robotics tasks, simulators, and form factors. We focus our evaluations on the effectiveness of different prompt engineering techniques and dialog strategies towards the execution of various types of robotics tasks. We explore ChatGPT's ability to use free-form dialog, parse XML tags, and to synthesize code, in addition to the use of task-specific prompting functions and closed-loop reasoning through dialogues. Our study encompasses a range of tasks within the robotics domain, from basic logical, geometrical, and mathematical reasoning all the way to complex domains such as aerial navigation, manipulation, and embodied agents. We show that ChatGPT can be effective at solving several of such tasks, while allowing users to interact with it primarily via natural language instructions. In addition to these studies, we introduce an open-sourced research tool called *PromptCraft*, which contains a platform where researchers can collaboratively upload and vote on examples of good prompting schemes for robotics applications, as well as a sample robotics simulator with ChatGPT integration, making it easier for users to get started with using ChatGPT for robotics.

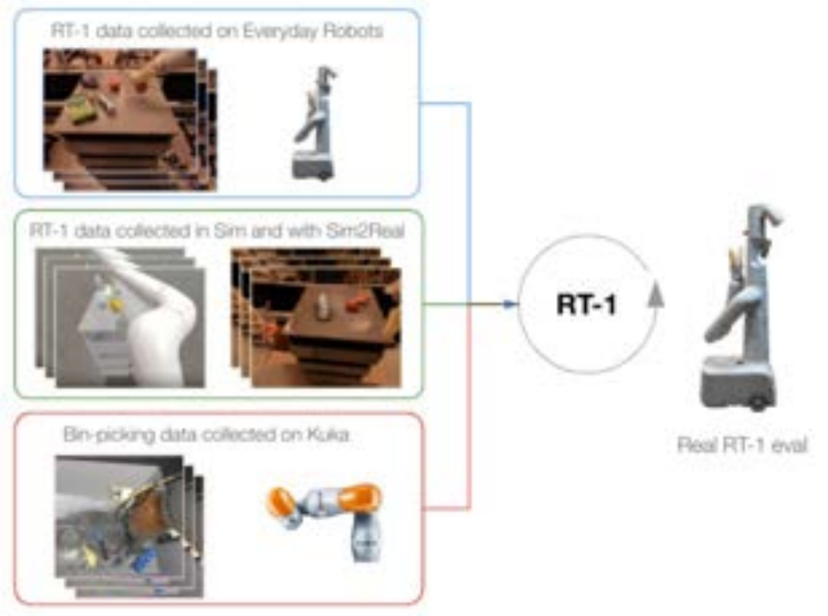


Sai Vemprala, Rogerio Bonatti, Arthur Bucker, Ashish Kapoor. ChatGPT for Robotics: Design Principles and Model Abilities. MSR-TR-2023-8, Microsoft (2023).

Can we scale efficiently?
And how much data?







Sim2Real

Real2Real

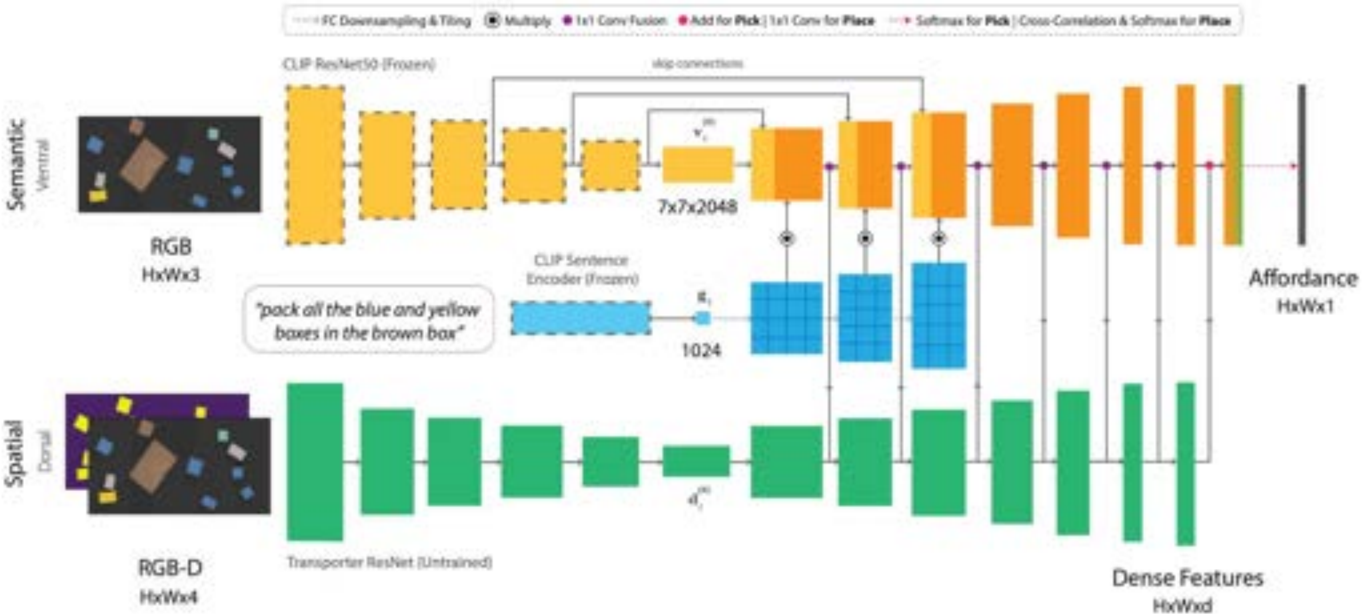
Kerr, Justin, Letian Fu, Huang Huang, Yahav Avigal, Matthew Tancik, Jeffrey Ichnowski, Angjoo Kanazawa, and Ken Goldberg. "Evo-NeRF: Evolving NeRF for Sequential Robot Grasping of Transparent Objects." In 6th Annual Conference on Robot Learning.
Ho, Daniel, Kanishka Rao, Zhuo Xu, Eric Jang, Mohi Khansari, and Yunfei Bai. "Retinagan: An object-aware approach to sim-to-real transfer." In 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 10920-10926. IEEE, 2021.
Brohan, Anthony, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan et al. "Rt-1: Robotics transformer for real-world control at scale." arXiv preprint arXiv:2212.06817 (2022).

Thanks!

CLIPort

What and Where Pathways for Robotic Manipulation

- **Conference on Robot Learning, 2022.**
- **Authors:** Mohit Shridhar, Lucas Manuelli, Dieter Fox
- **Presenter: Yatong Bai**



CLIPort

Task: Language-conditioned object manipulation.

- An imitation learning task.
- Idea: try to mimic an expert via supervised learning.

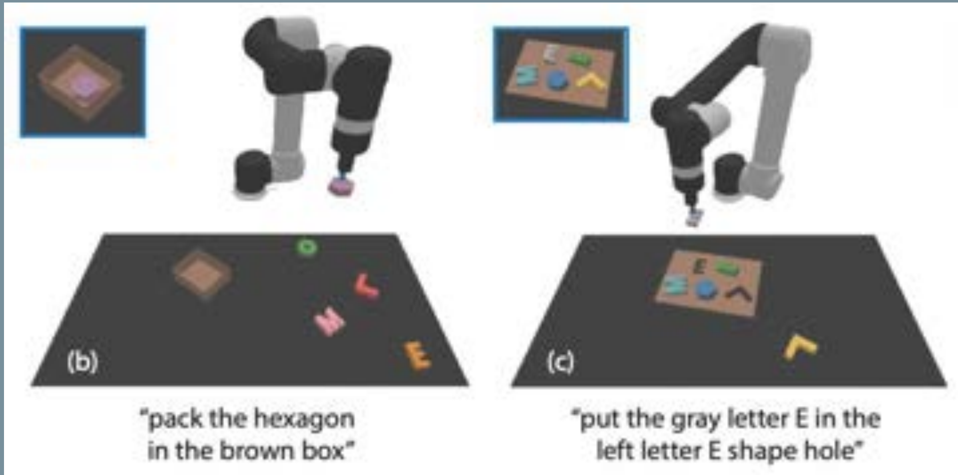
Inputs:

- A visual observation (RGB and depth, shape $H \times W \times 4$) $\mathbf{o_t}$;
- A language instruction $\mathbf{l_t}$.

  e. g. pack all blue and yellow boxes in the brown box.

Outputs:

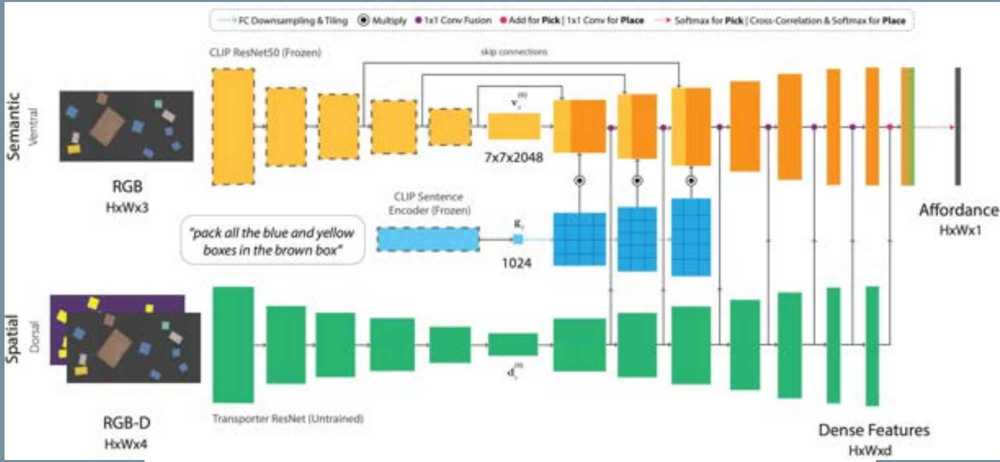
- Picking location $\mathcal{T}_{\text{pick}}$;
- Placing location $\mathcal{T}_{\text{place}}$.



Architecture of CLIPort

Three Fully-Convolutional-Networks (FCNs):

- **Pick FCN:** $f_{\text{pick}}: (\mathbf{o}_t, \mathbf{l}_t) \rightarrow Q_{\text{pick}}$.
  - Q_{pick} is $H \times W \times 1$. $T_{\text{pick}} = \text{argmax}_{\text{location}} Q_{\text{pick}}$.
- **Query FCN:** $\Phi_{\text{query}}: (\mathbf{o}_t[T_{\text{pick}}], \mathbf{l}_t) \rightarrow Q_{\text{query}}$.
  - $\mathbf{o}_t[T_{\text{pick}}]$ is a $c \times c$ crop of $\mathbf{o}_t$ centered at T_{pick}.
  - Q_{query} is $c \times c \times 3$.
- **Key FCN:** $\Phi_{\text{key}}: (\mathbf{o}_t, \mathbf{l}_t) \rightarrow Q_{\text{key}}$.
  - Q_{key} is $H \times W \times 3$.
  - $Q_{\text{place}} = Q_{\text{query}} * Q_{\text{key}}$, where $*$ is the correlation operation
  - Do this for a number of rotations.
  - Q_{place} is $H \times W \times 1$. $T_{\text{place}} = \text{argmax}_{\text{location}} Q_{\text{place}}$.



Architecture of each FCN (more on this later)

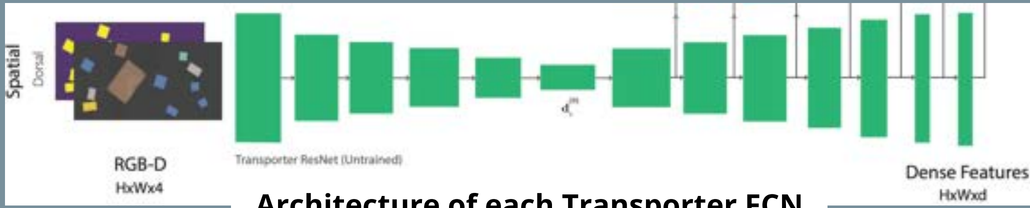
CLIP + Transporter = CLIPort

CLIP:

- Vision-language pre-training matching images to descriptions.
- Jointly learns a vision encoder and a language encoder.
- Match the direction of the visual and language embeddings.

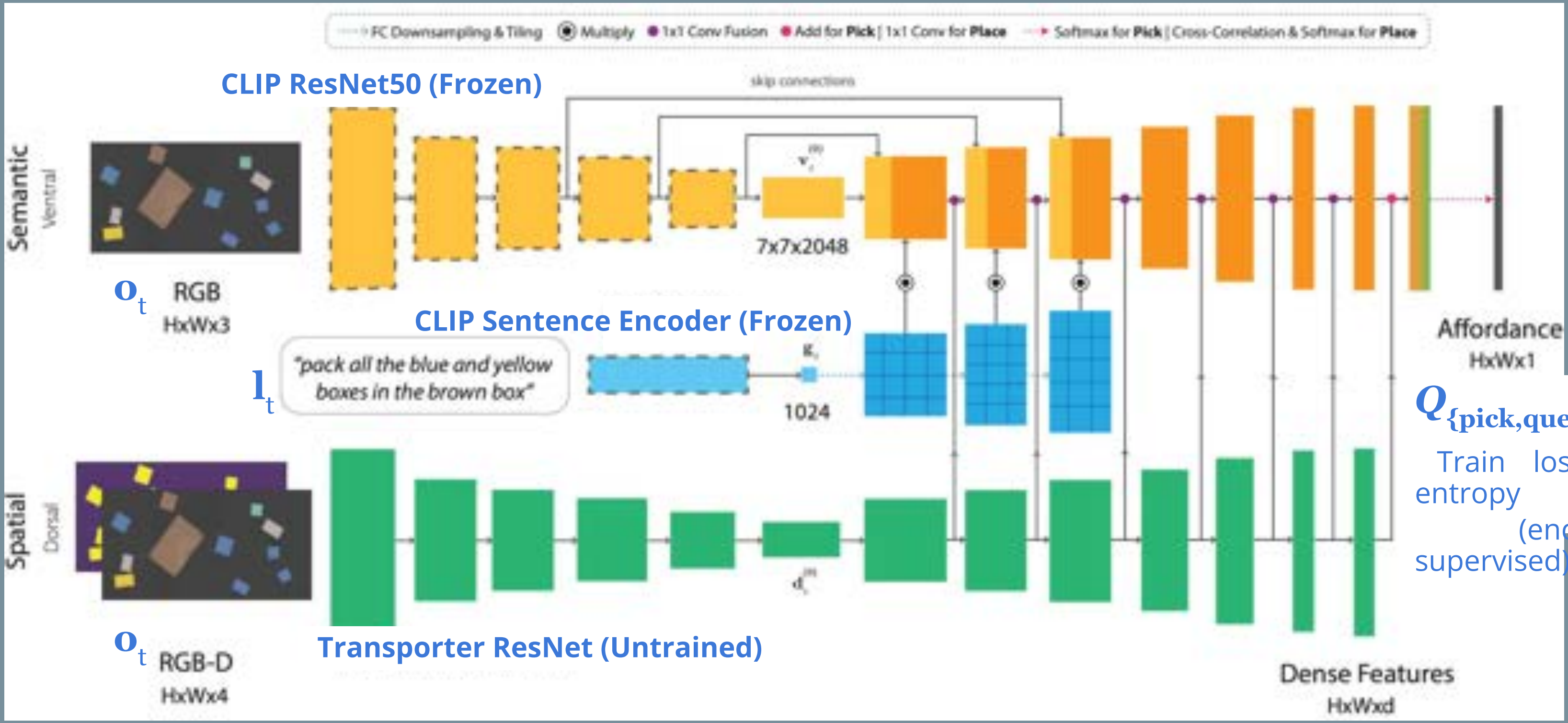
Transporter:

- Also for object manipulation imitation learning.
- Same tri-FCN architecture.
- No language conditioning.



Architecture of each Transporter FCN

Architecture of CLIPort – Details



$Q_{\{pick,query,key\}}$

Train loss: cross entropy (end-to-end supervised)

Despite its name, CLIPort's training isn't CLIP-like. It uses CLIP-pretrained modules.

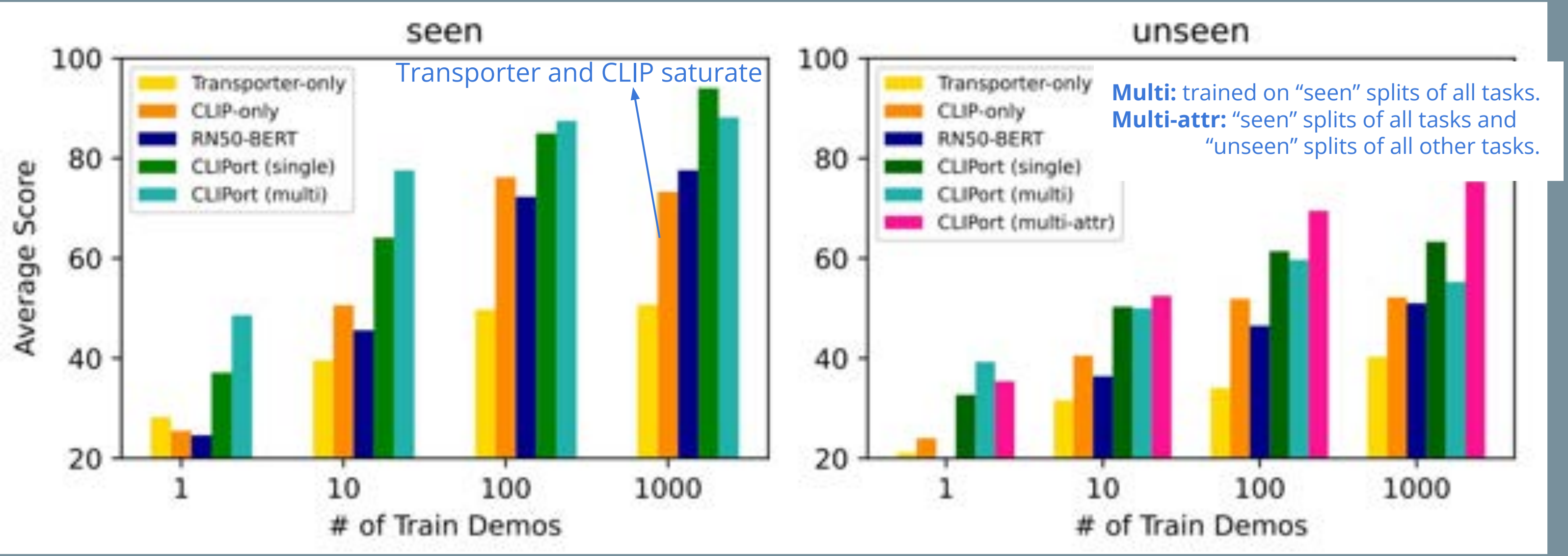
Experiment Results

Baselines:

- **Transporter-only**: no language grounding.
- **CLIP-only**: only the CLIP branch of CLIPort.
  - No depth information.

Task examples:

- Separating piles (seen colors yellow, brown, gray, cyan).
- Separating piles (unseen colors orange, purple, pink, white).
- Packing seen Google objects.
- Packing unseen Google objects,
- Etc... **10 tasks in total, 8 has seen and unseen**.



More Experiment Results

Main claim:

- The spatial and semantic streams enable accurate language-grounded object manipulation.

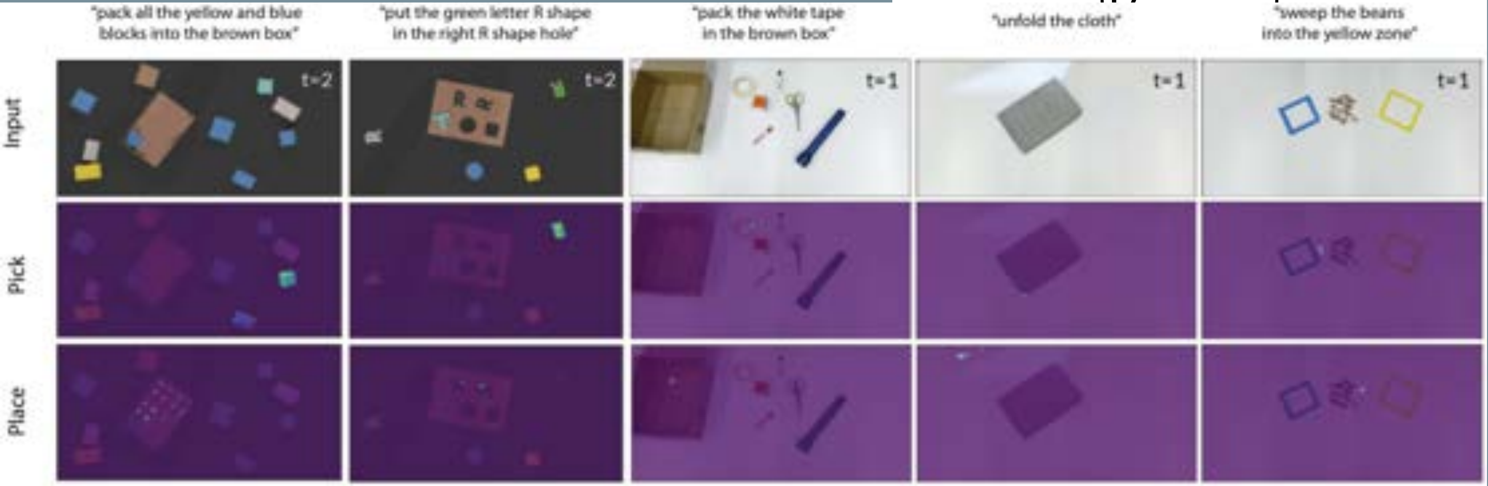
Question:

- CLIP-only does reasonably well.
- Is CLIPort's improvement due to the spatial stream or simply the depth map?

Task	# Train (Samples)	# Test	Succ. %
Stack Blocks	5 (13)	10	70.0
Put Blocks in Bowl	5 (10)	10	65.0
Pack Objects	10 (31)	10	60.0
Move Rook	4 (29)	10	70.0
Fold Cloth	9 (9)	10	57.0
Read Text	2 (26)	10	55.0
Loop Rope	4 (12)	10	60.0
Sweep Beans	5 (23)	5	60.6
Pick Cherries	4 (26)	5	75.0

Real-world tasks

Affordance predictions

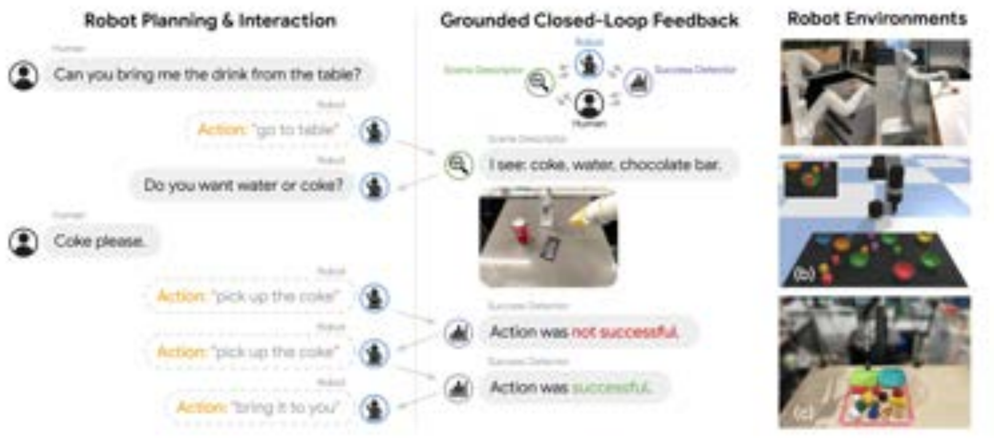


Thank you.

Questions?

Inner Monologue: Embodied Reasoning through Planning with Language Models

- **6th Annual Conference on Robot Learning**
- **Authors:** Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Noah Brown, Tomas Jackson, Linda Luu, Sergey Levine, Karol Hausman, Brian Ichter



What happens in the robot planner

What happens in the human planner

Inner Monolog

Human: Bring me a drink from the table

Robot: Go to the table

Scene: I see a coke and lime soda

Robot: What kind of drink would you like?

Human: Something with caffeine

Robot: Pick up coke

Success Detector: False

Robot: Pick up coke

Robot: Bring it to you

Robot: Put down the coke

Robot: Done

Problem statement

- A library of short horizon skills from a **library of previously trained policies πi** and their language descriptions
- **Planner which is LLM** - attempts to find the sequence of of skill to accomplish a task
- Environment - success detection, object detection, scene description, visual-question answering, and even human feedback.

```
---------------
Scene: There is a cyan, yellow, brown block.
Human: Move all the blocks to the top left corner.
Robot thought: Goal state is ["Cyan block is
       on the top left corner.", "Yellow block is on the top left corner.", "Brown block is on the top left corner."]

Robot action: Pick the cyan block and place it on the top left corner.
Scene: You have completed ["Cyan block is on the top left corner."]
Robot thought: Yellow and brown blocks are not on the top left corner.

Robot action: Pick the yellow block and place it on the top left corner.
Scene: You have completed ["Cyan block is on the top left corner."]
Robot thought: Yellow and brown blocks are not on the top left corner.

Robot action: Pick the yellow block and place it on the top left corner.
Scene: You have completed ["Cyan block is on the top left corner.", "Yellow block is on the top left corner."]
Robot thought: Brown block is not on the top left corner."

Robot action: Pick the brown block and place it on the top left corner.
Scene: You have completed ["Yellow block is on the top left corner.", "Brown block is on the top left corner."]
Robot thought: Cyan block is not on the top left corner

Robot action: Pick the cyan block and place it on the top left corner.
Scene: You have completed ["Yellow block
       is on the top left corner.", "Brown block is on the top left corner.", "Cyan block is on the top left corner."]
Robot thought: None.

---------------
Scene: There is a pink, gray, orange block.
Human: Move all the blocks to the bottom side.
Robot thought: Goal state is ["
      Pink block is on the bottom side.", "Gray block is on the bottom side.", "Orange block is on the bottom side."].

Robot action: Pick the pink block and place it on the bottom side.
Scene: You have completed ["Pink block is on the bottom side."]
Robot thought: Gray and orange blocks are not on the bottom side.

Robot action: Pick the gray block and place it on the bottom side.
Scene: You have completed ["Pink block is on the bottom side.", "Gray block is on the bottom side."]
Robot thought: Orange block is not on the bottom side.

Robot action: Pick the orange block and place it on the bottom side.
Scene: You have completed ["Orange block is on the bottom side."]
Robot thought: Pink and gray blocks are not on the bottom side.

Robot action: Pick the gray block and place it on the bottom side.
Scene: You have completed ["Pink block is on the bottom side.", "Gray block is on the bottom side."]
Robot thought: Orange block is not on the bottom side.

Robot action: Pick the orange block and place it on the bottom side.
Scene: You have completed ["
      Pink block is on the bottom side.", "Gray block is on the bottom side.", "Orange block is on the bottom side."]
Robot thought: None.
```

Planner - LLM

One of the remarkable observations in recent machine learning research is that large language models (LLMs) can not only generate fluent textual descriptions, but also appear to have rich internalized knowledge about the world.

Sources of feedback

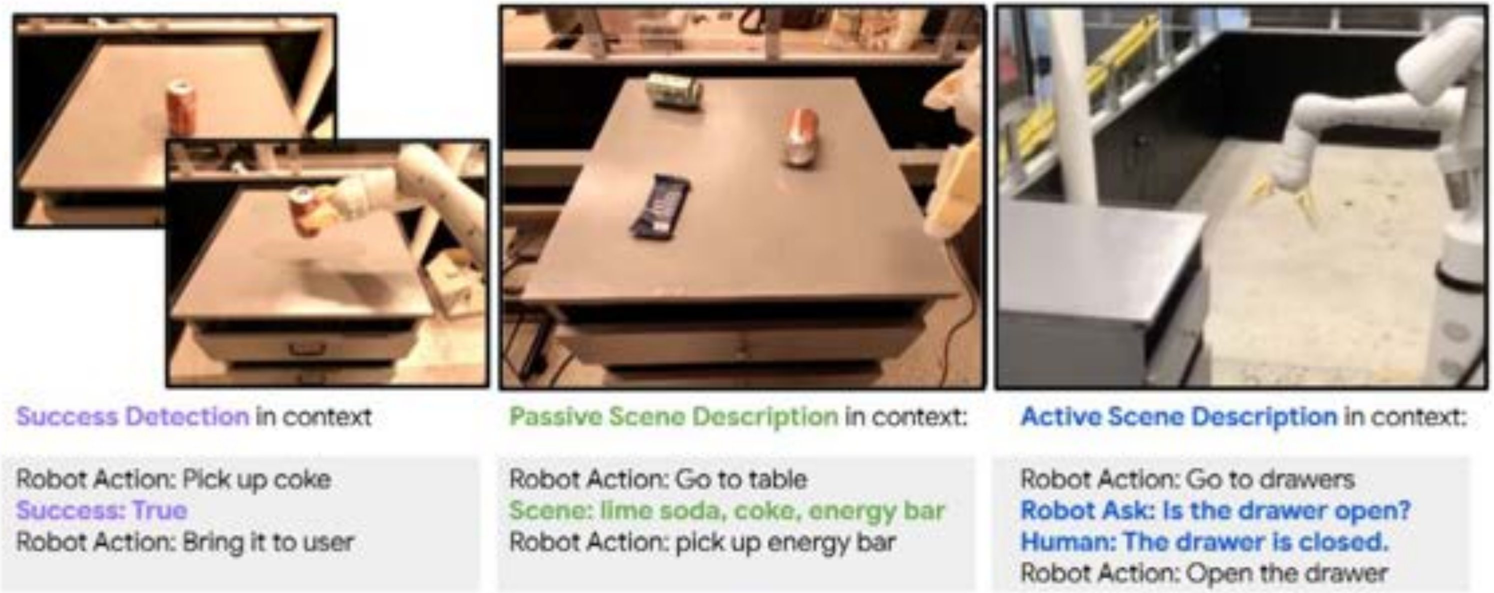
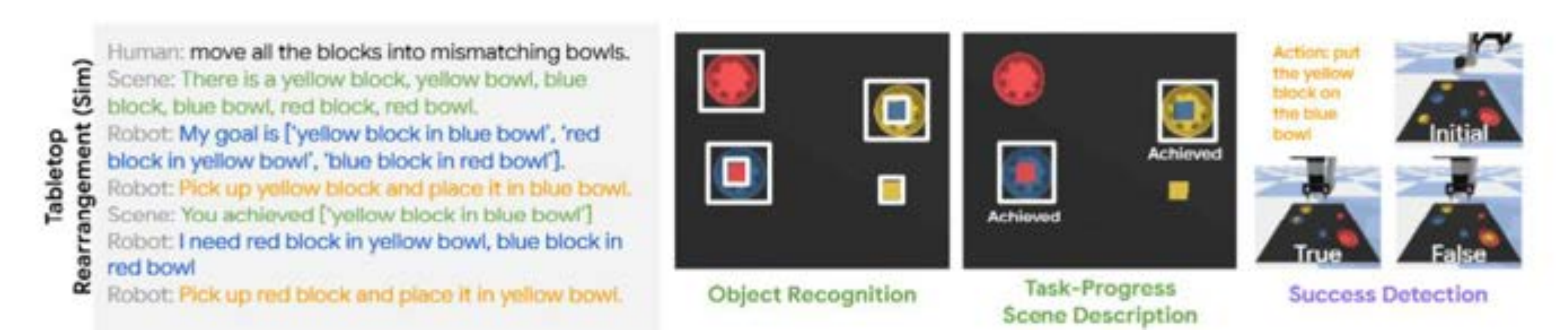


Figure 2: Various types of textual feedback. Success Detection gives task-specific task completion information, Passive Scene Description gives structured semantic scene information at every planning step, and Active Scene Description gives unstructured semantic information only when queried by the LLM planner.

Simulated Table top Rearrangement



- We use InstructGPT [91], a 1.3B parameter language model fine-tuned from GPT-3 [9] with human feedback, accessed through OpenAI API.
- Scripted modules to provide language feedback in the form of object recognition (Object), success detection (Success), and task-orogress scene description (Scene), and (iii) a pre-trained language-conditioned pick-and-place primitive
- **For Object + Success method,** we provide textual feedback of low-level policy success detection results after each policy execution.
- **For Object + Scene method,** we provide task-progress scene description as a list of achieved sub-goals after each pick-and-place execution

	Tasks	CLIPort	+oracle	+LLM	+Inner Monologue	
				Object	*Object + Success*	*Object + Scene*
Seen Tasks	"Pick and place"	24.0%	74.0%	80.0%	90.0%	**94.0%**
	"Stack all the blocks"	2.0%	32.0%	4.0%	10.0%	**26.0%**
	"Put all the blocks on the [x] corner/side"	2.0%	32.0%	**30.0%**	28.0%	**30.0%**
	"Put all the blocks in the [x] bowl"	32.0%	94.0%	52.0%	46.0%	**56.0%**
Unseen Tasks	"Put all the blocks in different corners"	0.0%	0.0%	20.0%	20.0%	**26.0%**
	"Put the blocks in their matching bowls"	0.0%	0.0%	56.0%	70.0%	**82.0%**
	"Put the blocks on mismatched bowls"	0.0%	0.0%	62.0%	76.0%	**86.0%**
	"Stack all the blocks on the [x] corner/side"	0.0%	0.0%	0.0%	4.0%	**6.0%**

Table 1: Success rates for various methods, averaged across 50 episodes in Ravens-based environment with test-time disturbances. CLIPort + oracle indicates that CLIPort was provided a "termination" oracle. Although CLIPort can receive visual feedback from the environment, we show that LLM-informed feedback can effectively enable the planner to retry/replan in the presence of failures, while enjoying the generalization benefits of LLMs to unseen tasks.

Scene: There is a cyan, yellow, brown block.
Human: Move all the blocks to the top left corner.
Robot thought: Goal state is ["Cyan block is
on the top left corner.", "Yellow block is on the top left corner.", "Brown block is on the top left corner."]

Robot action: Pick the cyan block and place it on the top left corner.
Scene: You have completed ["Cyan block is on the top left corner."]
Robot thought: Yellow and brown blocks are not on the top left corner.

Robot action: Pick the yellow block and place it on the top left corner.
Scene: You have completed ["Cyan block is on the top left corner."]
Robot thought: Yellow and brown blocks are not on the top left corner.

Robot action: Pick the yellow block and place it on the top left corner.
Scene: You have completed ["Cyan block is on the top left corner.", "Yellow block is on the top left corner."]
Robot thought: Brown block is not on the top left corner."

Robot action: Pick the brown block and place it on the top left corner.
Scene: You have completed ["Yellow block is on the top left corner.", "Brown block is on the top left corner."]
Robot thought: Cyan block is not on the top left corner

Robot action: Pick the cyan block and place it on the top left corner.
Scene: You have completed ["Yellow block
is on the top left corner.", "Brown block is on the top left corner.", "Cyan block is on the top left corner."]
Robot thought: None.

Scene: There is a pink, gray, orange block.
Human: Move all the blocks to the bottom side.
Robot thought: Goal state is ["
Pink block is on the bottom side.", "Gray block is on the bottom side.", "Orange block is on the bottom side."].

Robot action: Pick the pink block and place it on the bottom side.
Scene: You have completed ["Pink block is on the bottom side."]
Robot thought: Gray and orange blocks are not on the bottom side.

Robot action: Pick the gray block and place it on the bottom side.
Scene: You have completed ["Pink block is on the bottom side.", "Gray block is on the bottom side."]
Robot thought: Orange block is not on the bottom side.

Robot action: Pick the orange block and place it on the bottom side.
Scene: You have completed ["Orange block is on the bottom side."]
Robot thought: Pink and gray blocks are not on the bottom side.

Robot action: Pick the gray block and place it on the bottom side.
Scene: You have completed ["Pink block is on the bottom side.", "Gray block is on the bottom side."]
Robot thought: Orange block is not on the bottom side.

Robot action: Pick the orange block and place it on the bottom side.
Scene: You have completed ["
Pink block is on the bottom side.", "Gray block is on the bottom side.", "Orange block is on the bottom side."]
Robot thought: None.

Real World Table top Rearrangement

- We use InstructGPT [91], a 1.3B parameter language model fine-tuned from GPT-3 [9] with human feedback, accessed through OpenAI API.
- Two tasks: (i) a simple 3-block stacking task where 2 blocks are already pre-stacked, and (ii) a more complex long-horizon sorting task to place food in one plate and condiments in another (where categorizing food versus condiments is autonomously done by the LLM planner).
- For the block stacking task, the scene description contains a list of currently visible objects and a list of previously visible objects that are no longer visible.
- For the object sorting task, the scene description contains a list of currently visible objects and a list of objects that the robot has successfully moved into a plate.

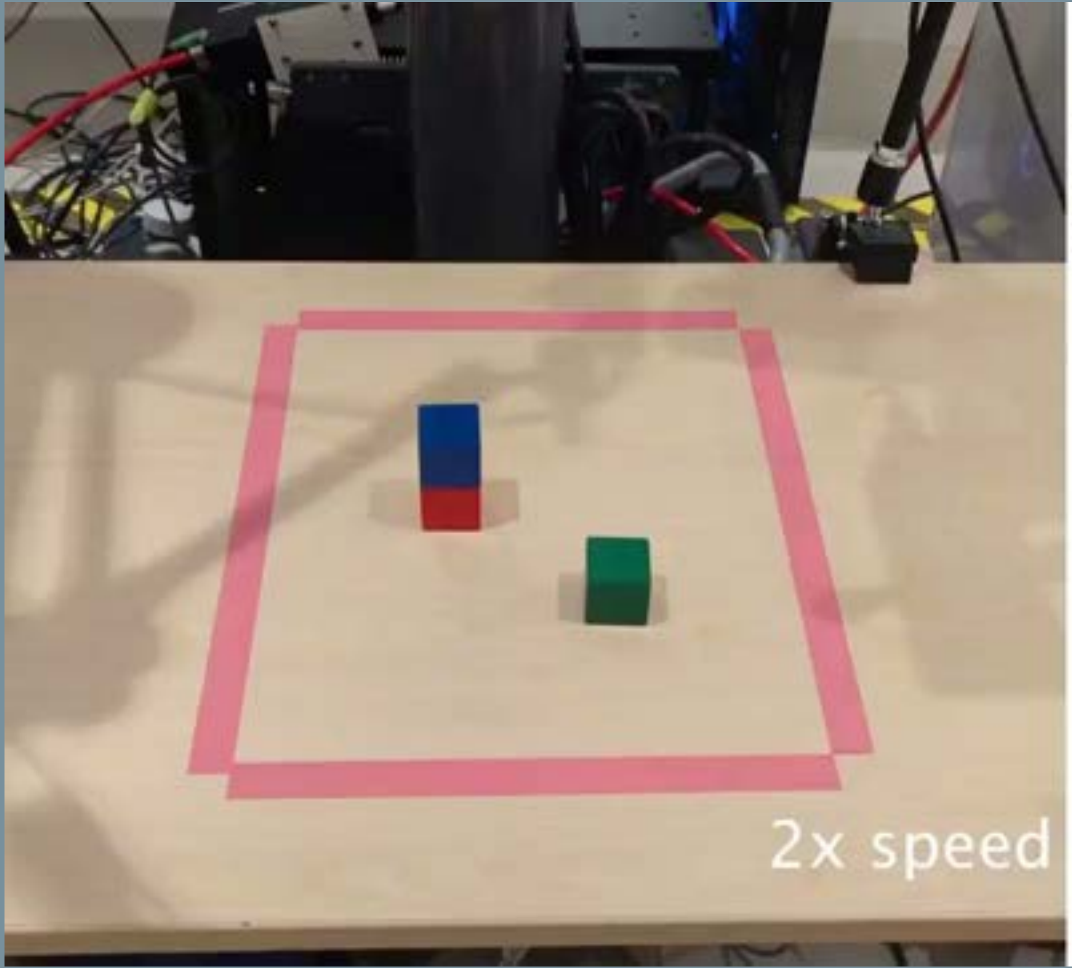
Task Family	LLM	+Inner Monologue		
	Object	*Object*	*Success*	*Object + Success*
Finish 3-block stacking	20%	40%	40%	**100%**
Sort fruits from bottles	20%	50%	40%	**80%**
Total	20%	45%	40%	**90%**

Table 2: Inner Monologue (with object recognition and success detection feedback) on a real pick and place robot exceeds the performance of baseline alternatives, as measured by average task success rates over 10 runs. For both tasks, we observe significant improvement in Inner Monologue compared to the baseline, with the two embodied feedback being complementary to each other.

Parts of the prompt

```
=================
Task: Move all blocks to the top left corner

Scene: Visible objects are ["cyan block", "yellow block", "brown block"]
Scene: Occluded objects are []
Robot action: robot.pick_place("cyan block", "top left corner")
Successful action: True

Scene: Visible objects are ["cyan block", "yellow block", "brown block"]
Scene: Occluded objects are []
Robot action: robot.pick_place("yellow block", "top left corner")
Successful action: False

Scene: Visible objects are ["cyan block", "yellow block", "brown block"]
Scene: Occluded objects are []
Robot action: robot.pick_place("yellow block", "top left corner")
Successful action: True

Scene: Visible objects are ["yellow block", "brown block"]
Scene: Occluded objects are ["cyan block"]
Robot action: robot.pick_place("brown block", "top left corner")
Successful action: True
```

```
=================
Task: Stack all the blocks.

Scene: Visible objects are ["brown block", "purple block"]
Scene: Occluded objects are []
Robot action: robot.pick_place("brown block", "purple block")
Successful action: False

Scene: Visible objects are ["brown block", "purple block"]
Scene: Occluded objects are []
Robot action: robot.pick_place("brown block", "purple block")
Successful action: True

Scene: Visible objects are ["brown block", "orange block"]
Scene: Occluded objects are ["purple block"]
Robot action: robot.pick_place("orange block", "brown block")
Successful action: True

Scene: Visible objects are ["orange block"]
Scene: Occluded objects are ["purple block", "brown block"]
Robot action: robot.stop()
STOP

=================
Task: Stack all the blocks.

Scene: Visible objects are ["brown block", "purple block"]
Scene: Occluded objects are []
Robot action: robot.pick_place("brown block", "purple block")
Successful action: True

Scene: Visible objects are ["brown block"]
Scene: Occluded objects are ["purple block"]
Robot action: robot.stop()
STOP

=================
```



Task: Stack all the blocks

Scene: Visible objects are ["green block", "blue block"]
Scene: Occluded objects are []
Robot action: robot.pick_place("blue block", "green block")
Successful action: False

Real World Mobile Manipulation

- Three task families: four manipulation tasks, two dexterous manipulation tasks utilizing drawers, and two long-horizon combined manipulation and navigation tasks.
- We use PALM, a 540B parameter language model trained on a large datasets that include high-quality web documents, books, Wikipedia, conversations, and GitHub code.
- We use human-provided object recognition to provide feedback about the presence of objects visible to the robot camera. For example, if there were only a coke can and an apple on top of the kitchen counter, then the human-provided object recognition feedback would appear as “[scene: coke can, apple]”

		+Inner Monologue	
Task Family	**SayCan**	*Success*	*Object + Success*
No Disturbances			
Manipulation	50.0%	62.5%	**75.0%**
Mobile Manipulation	50.0%	50.0%	**75.0%**
Drawers	83.3%	83.3%	**100.0%**
With Disturbances			
Manipulation	12.5%	25.0%	**33.3%**
Mobile Manipulation	0.0%	25.0%	**75.0%**
Drawers	0.0%	**44.4%**	**44.4%**
Total	30.8%	48.7%	**60.4%**

Table 3: Averaged success rate across 120 evaluations on several task families in our real-world mobile manipulation environment. We consider a standard setting and adversarial setting with external human disturbances. In all cases, LLM-informed embodied feedback is shown to be effective in improving robustness of the system, especially when low-level policies are prone to failures.

The baseline, SayCan [21], is a method that plans and acts in diverse real world scenarios by combining an LLM with value functions of underlying control policies. While SayCan creates **plans that are grounded by the affordances of value functions, the LLM predictions in isolation are never given any closed-loop feedback.**

```
Robot: Hi there, I'm a robot operating in an office kitchen.
You can ask me to do various tasks and I'll tell you the sequence of actions I would do to accomplish your task.
Human: hold the snickers
Robot: 1. pick up the snickers[success: no][scene: snickers]
2. pick up the snickers
2. done.
Human: put the trailmix here
Robot: 1. put down the trailmix
2. done.
Human: put a water bottle and an oatmeal next to the microwave
Robot: 1. find a water
2. pick up the water[success: no]
3. pick up the water[success: no]
4. pick up the water
5. go to microwave
6. put down the water[scene: water, microwave]
7. find an oatmeal
8. pick up the oatmeal[scene: oatmeal]
9. go to the microwave
10. put down the oatmeal[scene: microwave, oatmeal, water]
11. done.
Human: put a grapefruit from the table into the bowl
Robot: 1. find a grapefruit
2. pick up the grapefruit
3. go to the bowl
4. put down the grapefruit
5. done.
Human: get a sponge from the counter and put it in the sink
Robot: 1. find a sponge
2. pick up the sponge[success: no][scene: sponge]
3. pick up the sponge[success: no]
4. pick up the sponge[scene: sponge]
5. go to the sink
6. put down the sponge
7. done.
```

Part of the prompt