

Visual Reasoning  
will be bigger than language reasoning

Ranjay Krishna  
Assistant Professor, University of Washington  
Research Director, Allen Institute for AI

Average Score on 11 Academic Benchmarks

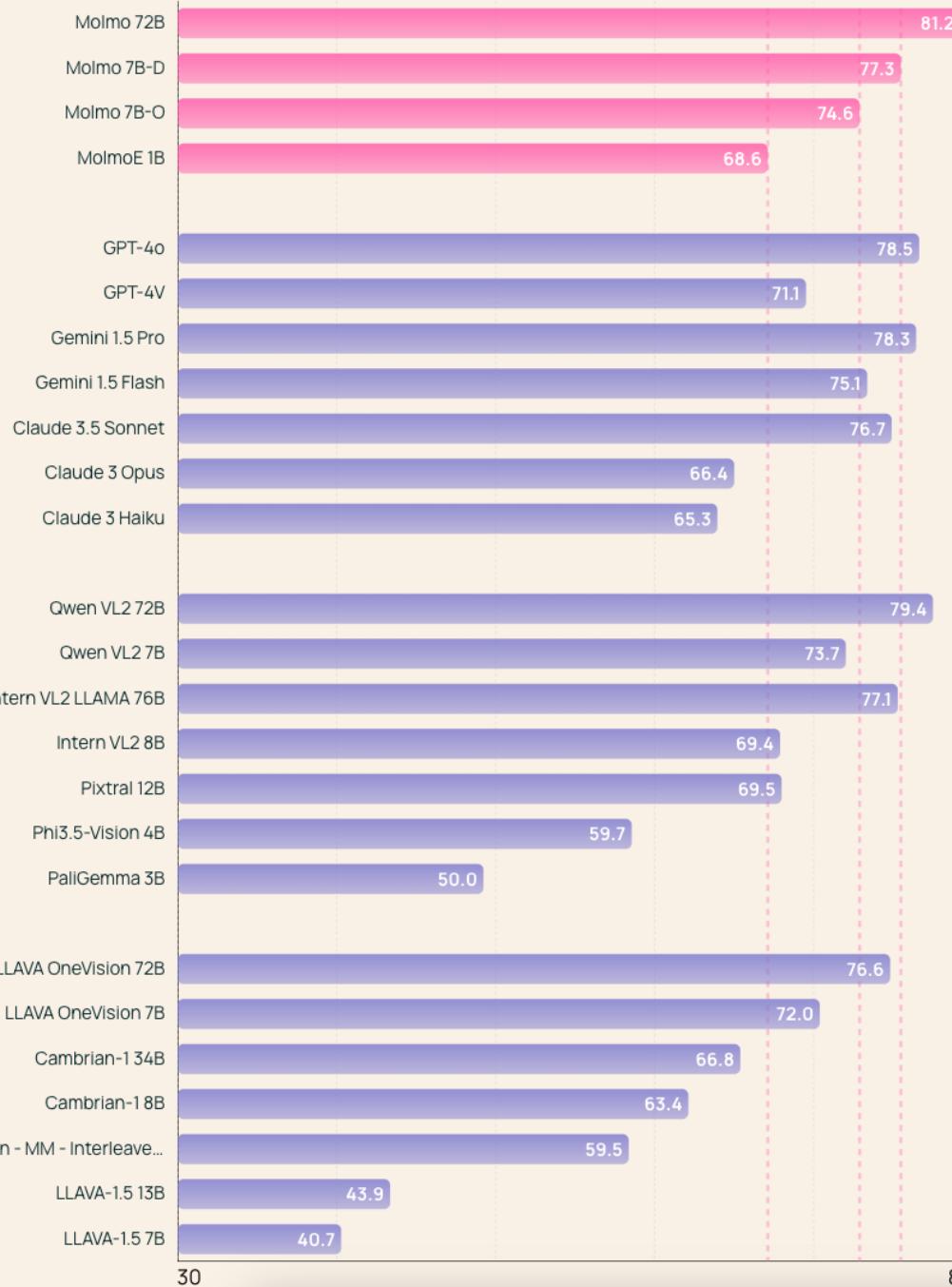
**Open  
Weights  
Data  
Code  
Evals**

**API Only**

**Open  
Weights**

**Distilled**

**Open**



Where I left off at CVPR



**Open weights  
Open training data  
Open training code  
Open Evaluations**

Molmo reasoning directly in the pixels

When it counts,  
it points

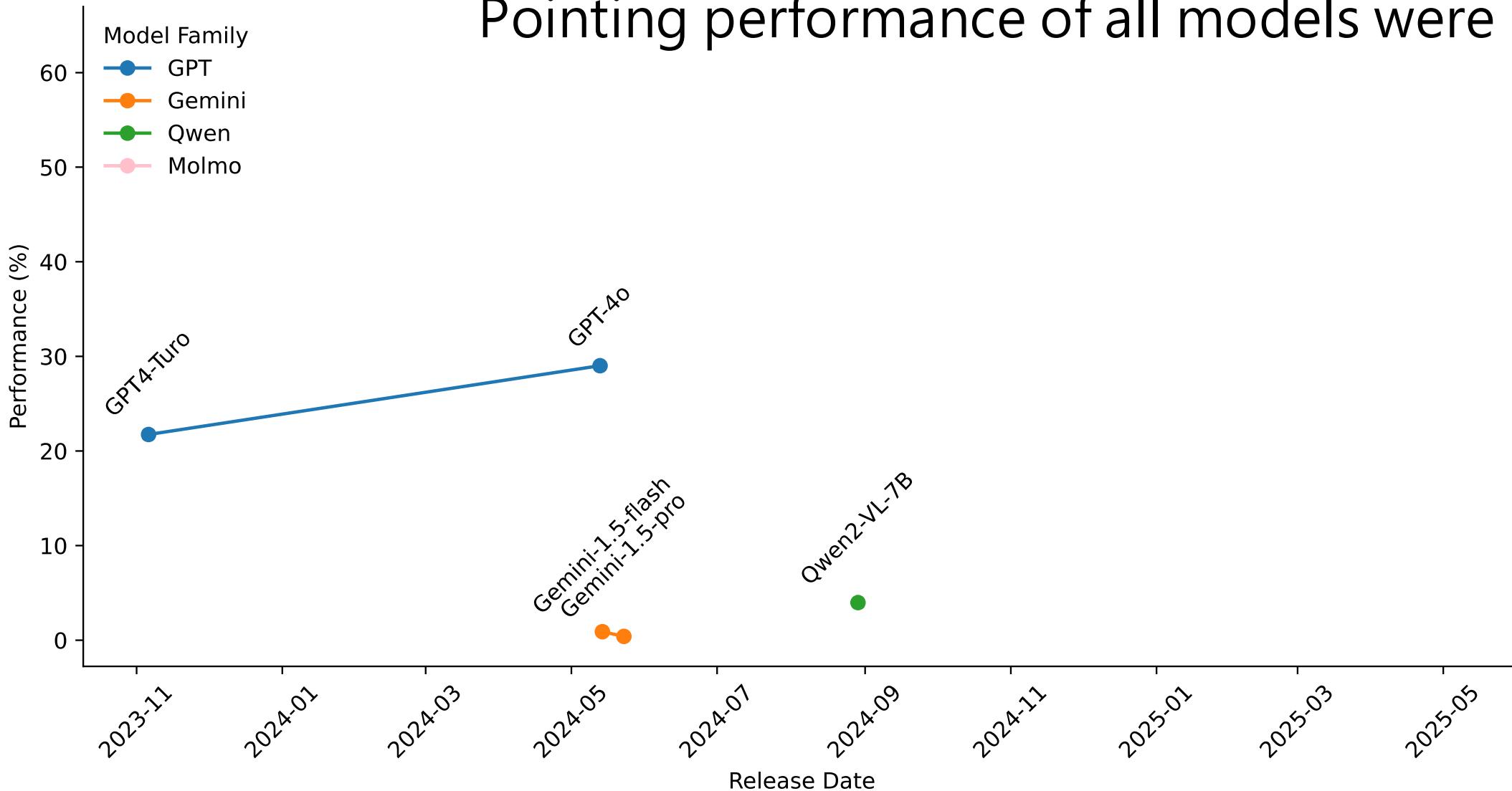
Count the boats



● boats

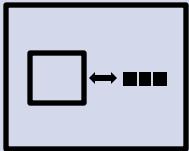
Counting the **boats** shows a total of 35.

# Pointing performance of all models were poor





In today's talk:  
**Visual Reasoning**

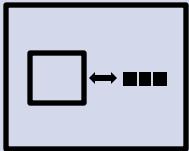


## Prioritizing perception

Perceptual tests for VLMs  
[ECCV 2024]



Most fundamental vision capabilities are still out of reach



## Prioritizing perception

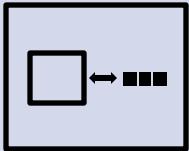
Perceptual tests for VLMs  
[ECCV 2024]

Sketching for perceptual reasoning  
[NeurIPS 2024] [CVPR 2025]



Most fundamental vision capabilities are still out of reach

Enabling sketching:  
visual chain of thought

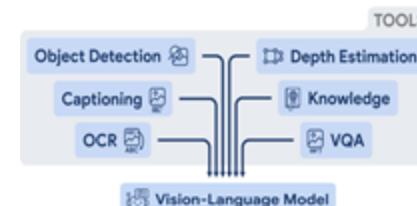


## Prioritizing perception

Perceptual tests for VLMs  
[ECCV 2024]

Sketching for perceptual reasoning  
[NeurIPS 2024] [CVPR 2025]

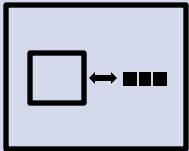
Distilling perceptual capabilities  
[ACL 2023] [CVPR 2024] [CVPR 2025]



Most fundamental vision capabilities are still out of reach

Enabling sketching:  
visual chain of thought

How can we distill from specialist models into generalist VLMs?



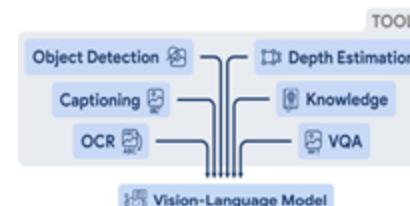
## Prioritizing perception

Perceptual tests for VLMs  
[ECCV 2024]

Sketching for perceptual reasoning  
[NeurIPS 2024] [CVPR 2025]

Distilling perceptual capabilities  
[ACL 2023] [CVPR 2024] [CVPR 2025]

Enabling robots to sketch  
[ArXiv 2025]



Most fundamental vision capabilities are still out of reach

Enabling sketching:  
visual chain of thought

How can we distill from specialist models into generalist VLMs?

Complete open Action Reasoning model for robotics

# Papers we will discuss

## **BLINK: Multimodal Large Language Models Can See but Not Perceive**

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, Ranjay Krishna  
ECCV 2024

## **Visual Sketchpad: Sketching as a Visual Chain of Thought for Multimodal Language Models**

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, Ranjay Krishna  
NeurIPS 2024

## **Visual Program Distillation: Distilling Tools and Programmatic Reasoning into Vision-Language Models**

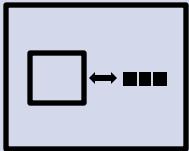
Yushi Hu, Otilia Stretcu, Chun-Ta Lu, Krishnamurthy Viswanathan, Kenji Hata, Enming Luo, Ranjay Krishna, Ariel Fuxman  
CVPR 2025

## **Perception Tokens Enhance Visual Reasoning in Multimodal Language Models**

Mahtab Bigverdi, Zelun Luo, Cheng-Yu Hsieh, Ethan Shen, Dongping Chen, Linda G. Shapiro, Ranjay Krishna  
CVPR 2025

## **MolmoAct: Action Reasoning Models that can Reason in Space**

Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, Winson Han, Wilbert Pumacay, Angelica Wu, Rose Hendrix, Karen Farley, Eli VanderBilt, Ali Farhadi, Dieter Fox, Ranjay Krishna  
ArXiv 2025



## Prioritizing perception

Perceptual tests for VLMs  
[ECCV 2024]

Sketching for perceptual reasoning  
[NeurIPS 2024] [CVPR 2025]

Distilling perceptual capabilities  
[ACL 2023] [CVPR 2024] [CVPR 2025]

Enabling robots to sketch  
[ArXiv 2025]



Most fundamental vision capabilities are still out of reach

Enabling sketching:  
visual chain of thought

How can we distill from specialist models into generalist VLMs?

Complete open Action Reasoning model for robotics

We know that models today can describe images well



What can you see?



In the image, I see a cat with a light brown and white fur coat sitting on a blue chair. The cat appears to be wrapped in or resting on a dark gray fabric, possibly a piece of clothing or a small blanket. The cat's expression seems slightly grumpy or annoyed. The background includes part of the chair and a light-colored cushion.

# But can they understand 3D space?



(A) Left



(B) Right



GPT 4V

(C) I don't know ✨ Gemini

Question: Is the camera moving towards the left or the right?

# Maybe an even easier example



GPT 4V

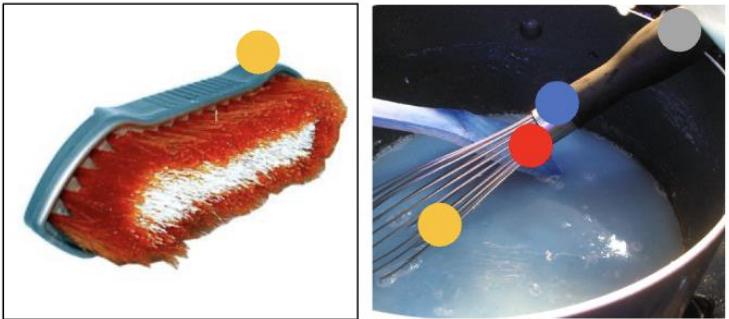


Gemini Pro

Question: Which image fits the missing part?

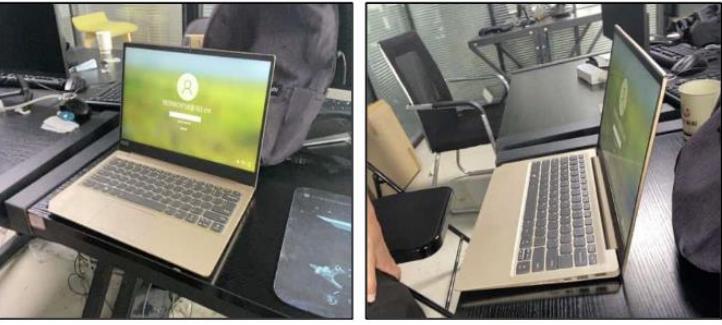
# What fundamental perceptual capabilities do we want VLMs to have

## Semantic affordance



**Q:** Which point has similar affordance?

## Multiview reasoning



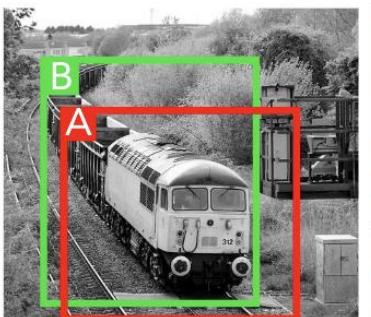
**Q:** Is the camera moving clockwise around the object?

## Visual similarity



**Q:** Which image is more similar to the left one?

## Localization



**Q:** Which box localizes train better?

## Inpainting



**Q:** Which image fits here better?

## Depth estimation



**Q:** Which point is farther?

## Color



**Q:** Which point is darker?

## Image forensics



**Q:** Which image is real?

# What are popular VLM benchmarks measuring?

Forecasting



**Q:** What will happen next?

Social relationships



**Q:** What is their relationship?

Geolocation



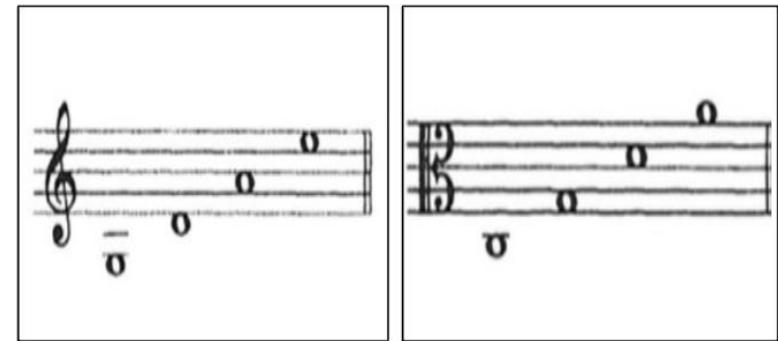
**Q:** Where is this place?

Recognition



**Q:** What kind of animal is this?

Music theory?



**Q:** What is the correct tuning of Violin?

While still valuable, existing benchmarks conflate perception with general knowledge and reasoning

Examples taken from MME – which is a good benchmark

# BLINK : A VLM benchmark aimed at measuring classical notions of perception (at ECCV 2024)

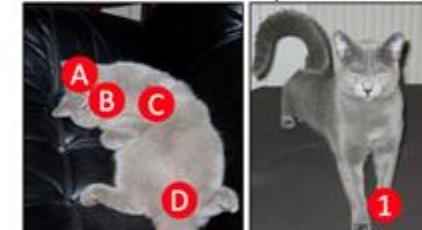
**Relative depth**  
  
Which point is closer?

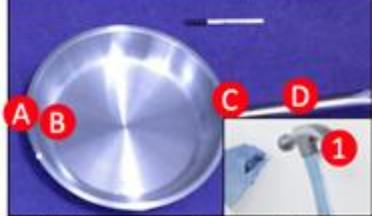
**Jigsaw**  
  
Which image fits here?

**Multi-view reasoning**  
  
Is camera moving right?

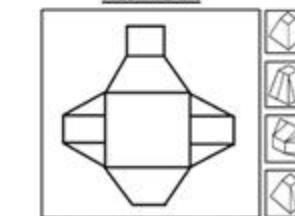
**Visual correspondence**  
  
Which point is the same?

**Relative reflectance**  
  
Which point is darker?

**Semantic correspondence**  
  
Which points have similar semantics?

**Functional correspondence**  
  
Which points have similar affordance when pulling out a nail?

**Visual similarity**  
  
Which image is more similar to the left?

**IQ Test**  
  
Which object does it fold into?

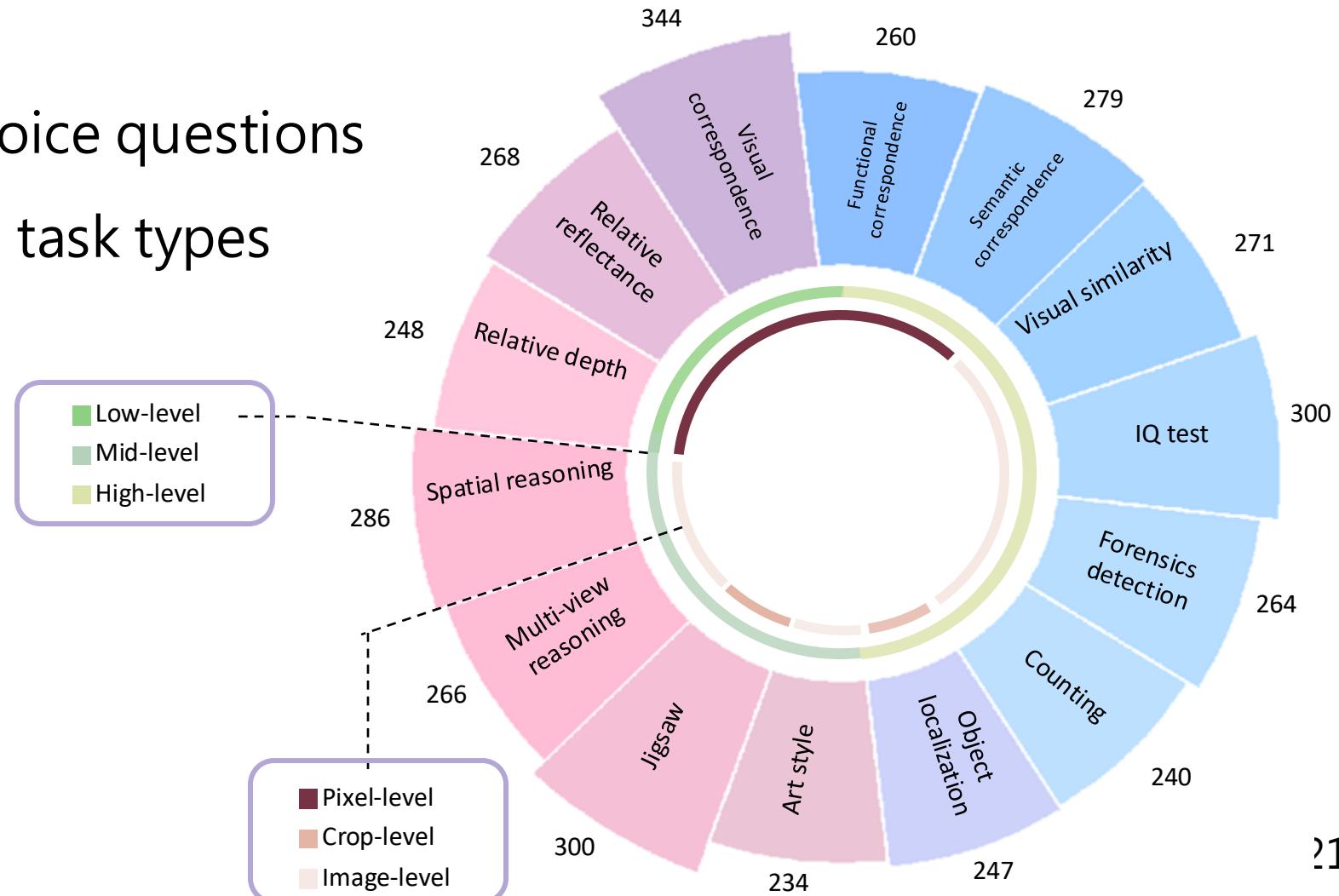
**Forensics detection**  
  
Which image is real?

# Blink has a diverse set of questions

7,358 unique images

3,807 unique multiple-choice questions

Over 14 visual perception task types



# Results on BLINK

- (1) Humans are very good at these tasks (95%)
- (2) Random chance is a reasonable 38%

	Validation (1,901)	Test (1,906)
Random Choice	38.09	38.09
Human	95.67	95.70

# Results on BLINK

- (1) Humans are very good at these tasks (95%)
- (2) Random chance is a reasonable 38%
- (3) **Open-weight models barely perform better than random chance**

	Validation (1,901)	Test (1,906)
Random Choice	38.09	38.09
Human	95.67	95.70
MiniGPT-4-v2 [16]	34.23	34.57
OpenFlamingo-v2 [5]	39.18	38.32
InstructBLIP-7B [24]	39.72	38.65
InstructBLIP-13B [24]	42.24	39.58
LLaVA-internLM2-7B [72]	37.71	36.06
Yi-VL-6B <sup>2</sup>	38.72	41.24
Yi-VL-34B <sup>2</sup>	41.68	42.78
LLaVA-v1.5-7B-xtuner [23]	39.36	40.81
LLaVA-v1.5-13B-xtuner [23]	42.00	41.31
CogVLM [77]	41.54	39.38
LLaVA-v1.5-7B [48]	37.13	38.01
LLaVA-v1.5-13B [48]	42.66	40.55
LLaVA-v1.6-34B [50]	46.80	45.05

# Results on BLINK

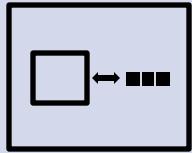
- (1) Humans are very good at these tasks (95%)
- (2) Random chance is a reasonable 38%
- (3) Open-weight models barely perform better than random chance
- (4) Only GPT-4o performs better but not by much (60%).

	Validation (1,901)	Test (1,906)
Random Choice	38.09	38.09
Human	95.67	95.70
MiniGPT-4-v2 [16]	34.23	34.57
OpenFlamingo-v2 [5]	39.18	38.32
InstructBLIP-7B [24]	39.72	38.65
InstructBLIP-13B [24]	42.24	39.58
LLaVA-internLM2-7B [72]	37.71	36.06
Yi-VL-6B <sup>2</sup>	38.72	41.24
Yi-VL-34B <sup>2</sup>	41.68	42.78
LLaVA-v1.5-7B-xtuner [23]	39.36	40.81
LLaVA-v1.5-13B-xtuner [23]	42.00	41.31
CogVLM [77]	41.54	39.38
LLaVA-v1.5-7B [48]	37.13	38.01
LLaVA-v1.5-13B [48]	42.66	40.55
LLaVA-v1.6-34B [50]	46.80	45.05
Qwen-VL-Max [7]	40.28	41.94
Gemini Pro [71]	45.16	45.72
Claude 3 OPUS [1]	44.05	44.11
GPT-4V(ision) [62]	51.14	51.26
GPT-4 Turbo [62]	54.61	53.89
GPT-4o [62]	<b>60.04</b>	<b>59.03</b>

Are the tasks too difficult to solve? **NO!**

Task	Vis.Corr.	Depth	Multi-view	Sem.Corr.	Forensic	Reflect.
Random	25.00	50.00	50.00	25.00	25.00	33.33
Human	99.56	99.59	92.10	94.60	100.00	99.63
Gemini Pro	42.44	40.32	44.36	26.62	50.76	45.52
GPT-4V	33.72	59.68	55.64	28.78	34.09	38.81
Specialist	DIFT [70] DepthAnything [83] LoFTR [68]	96.51	DIFT [70] DIRE [79]	90.22	71.22	68.94
		97.58	Ordinal Shading [14]			77.61

Specialist models can do these classifical perception tasks very well



## Prioritizing perception

Perceptual tests for VLMs  
[ECCV 2024]

Sketching for perceptual reasoning  
[NeurIPS 2024] [CVPR 2025]

Distilling perceptual capabilities  
[ACL 2023] [CVPR 2024] [CVPR 2025]

Enabling robots to sketch  
[ArXiv 2025]

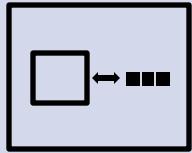


Most fundamental vision capabilities are still out of reach

Enabling sketching:  
visual chain of thought

How can we distill from specialist models into generalist VLMs?

Complete open Action Reasoning model for robotics



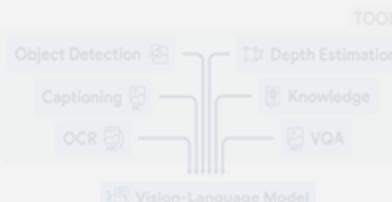
## Prioritizing perception

Perceptual tests for VLMs  
[ECCV 2024]

Sketching for perceptual reasoning  
[NeurIPS 2024] [CVPR 2025]

Distilling perceptual capabilities  
[ACL 2023] [CVPR 2024] [CVPR 2025]

Enabling robots to sketch  
[ArXiv 2025]



Most fundamental vision capabilities are still out of reach

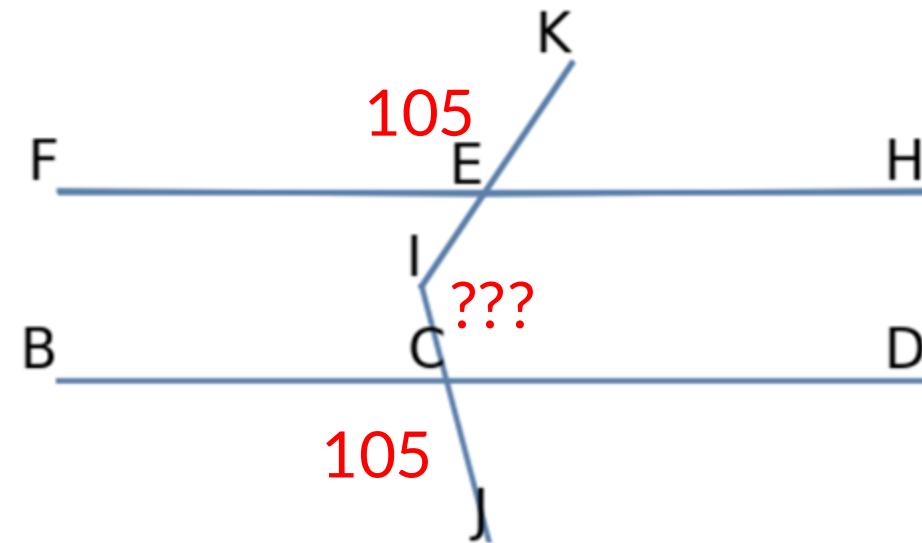
Enabling sketching:  
visual chain of thought

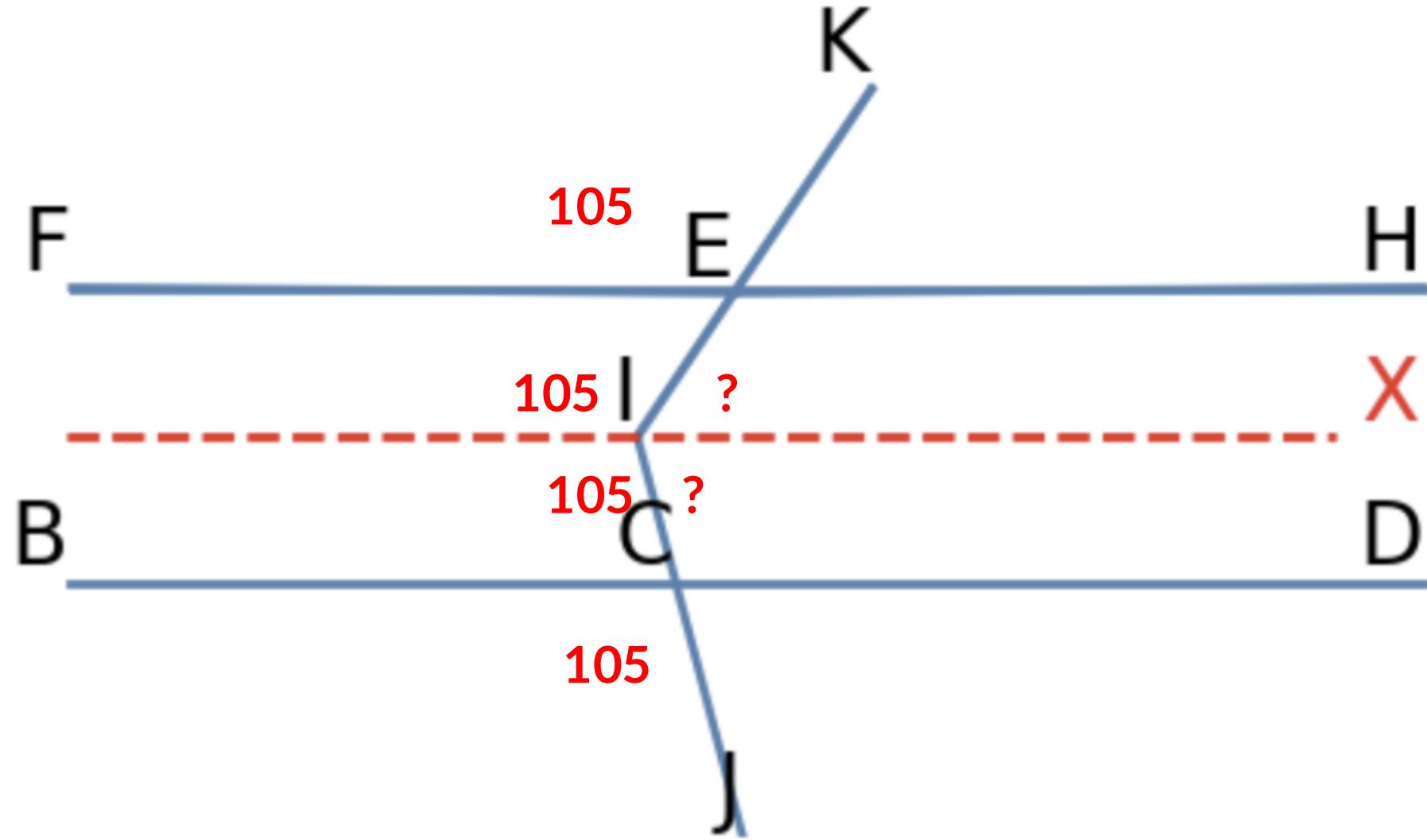
How can we distill from specialist models into generalist VLMs?

Complete open Action Reasoning model for robotics

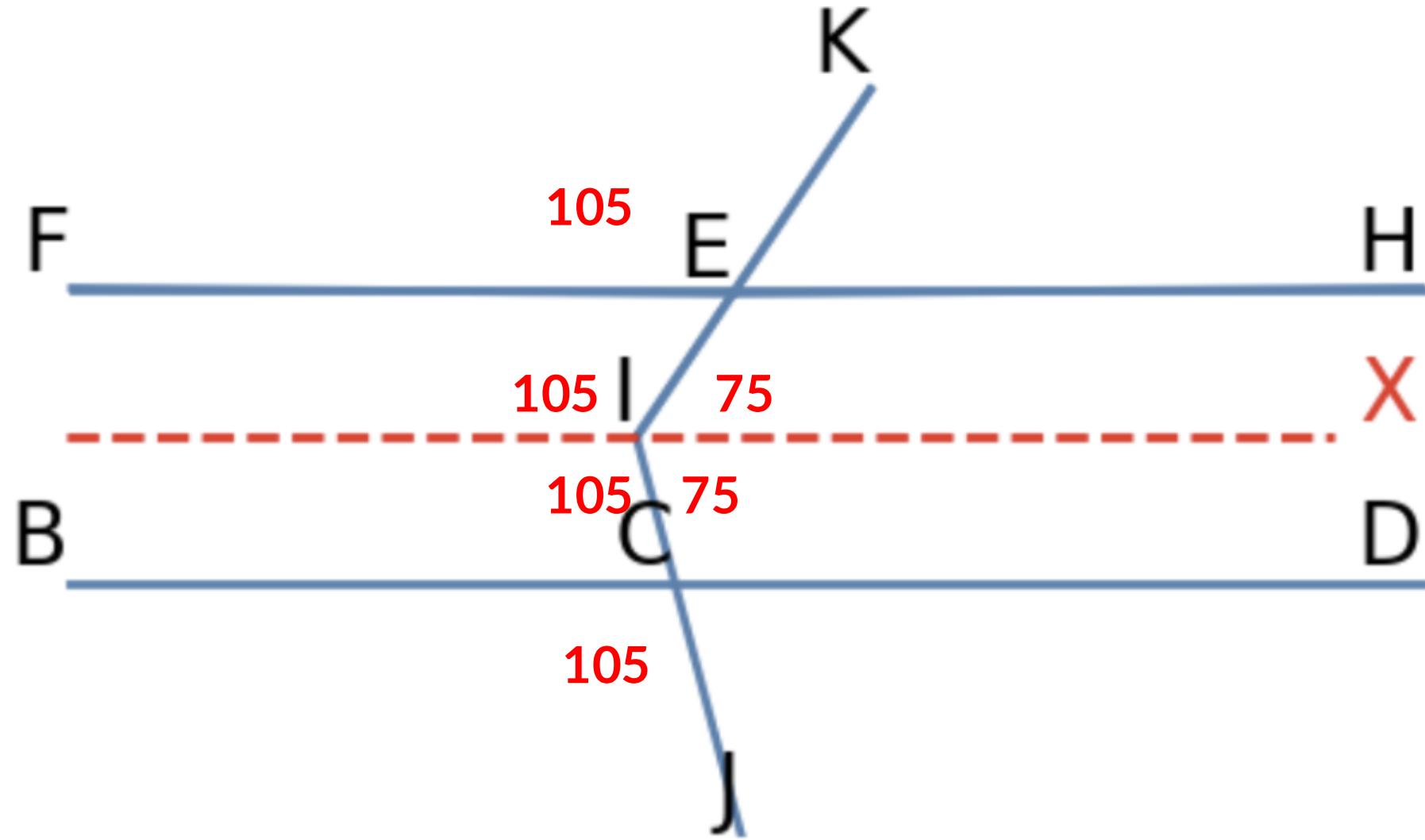
Let's try to solve some BLINK tasks ourselves  
How would you solve this task?

Given  $\angle BCJ = 105^\circ$ ,  
 $\angle KEF = 105^\circ$ . Find  $\angle EIC$





So,  $\angle EIC = ???$



Let's try another one

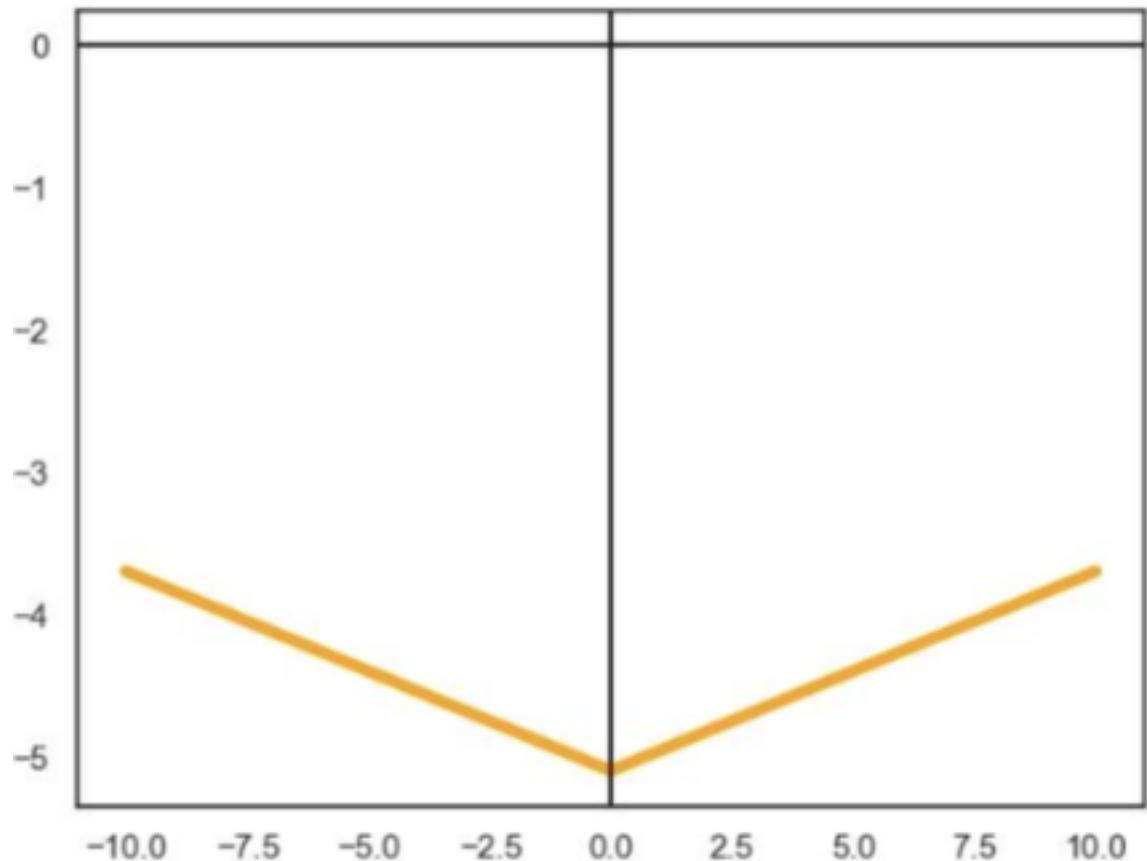
Is  $f(x)$  an convex  
function?

$$f(x) = 0.14 |x| - 5.09$$

Is this easier?

Is  $f(x)$  an convex function?

$$f(x) = 0.14 |x| - 5.09$$



# Sketching

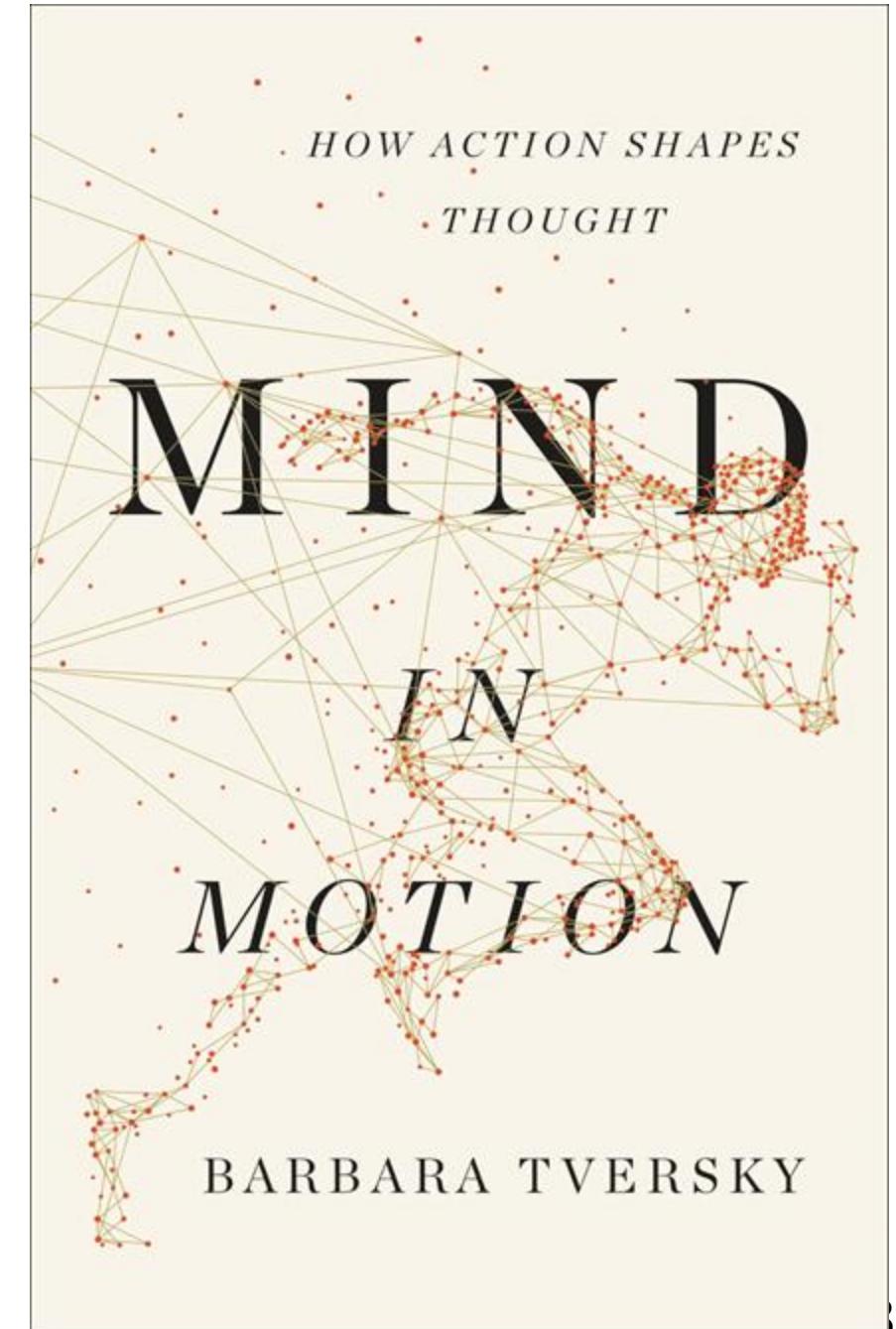
This is how we solve problems

Her work ranges 5 decades and shows culminates to one statement:

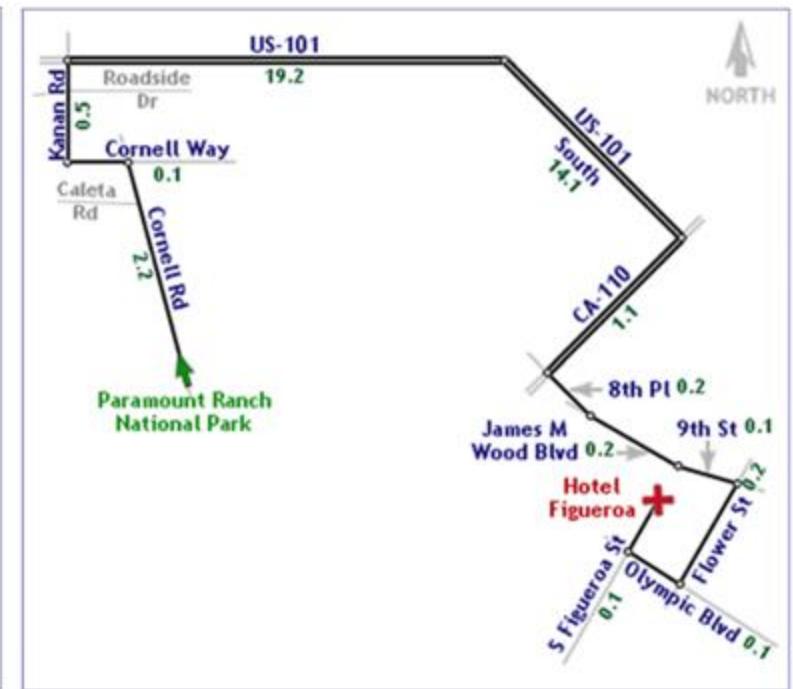
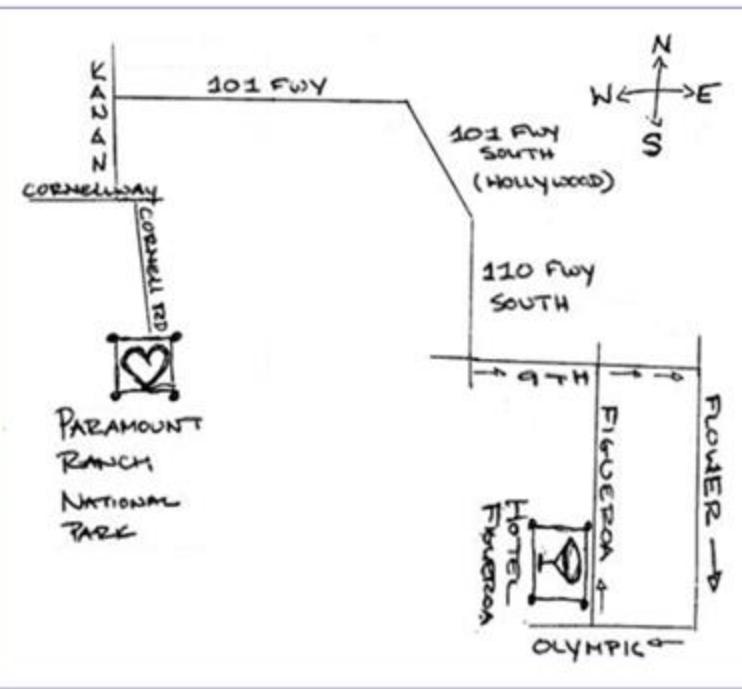
Spatial understanding is the foundation of all intelligence



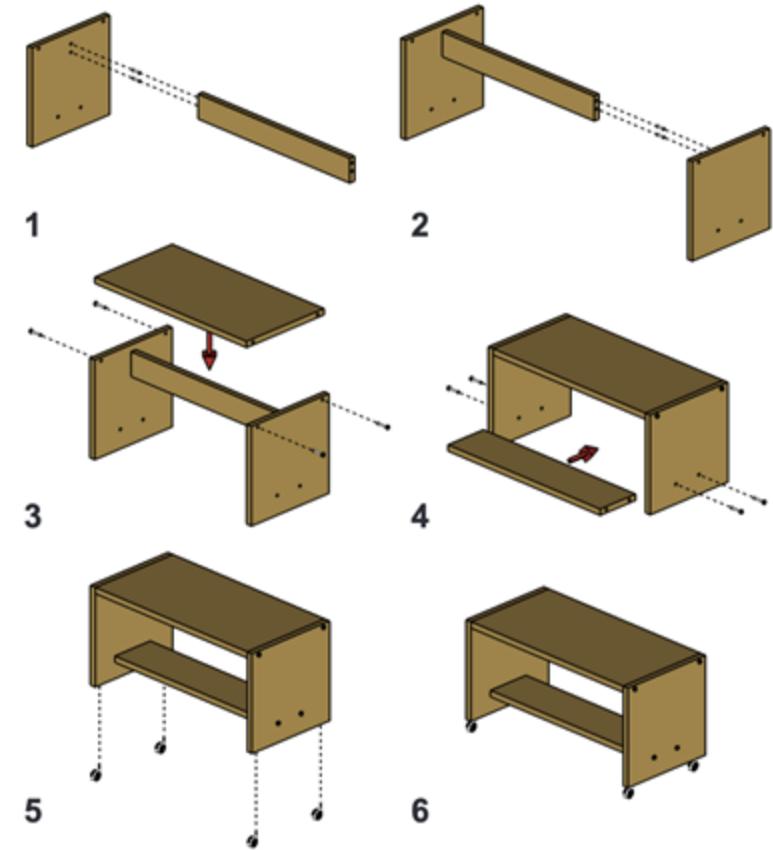
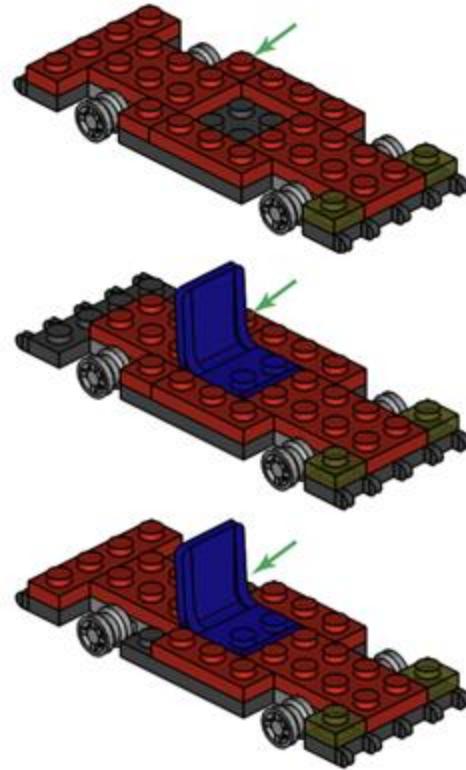
Barbara Tversky



# We use sketches to convey directions

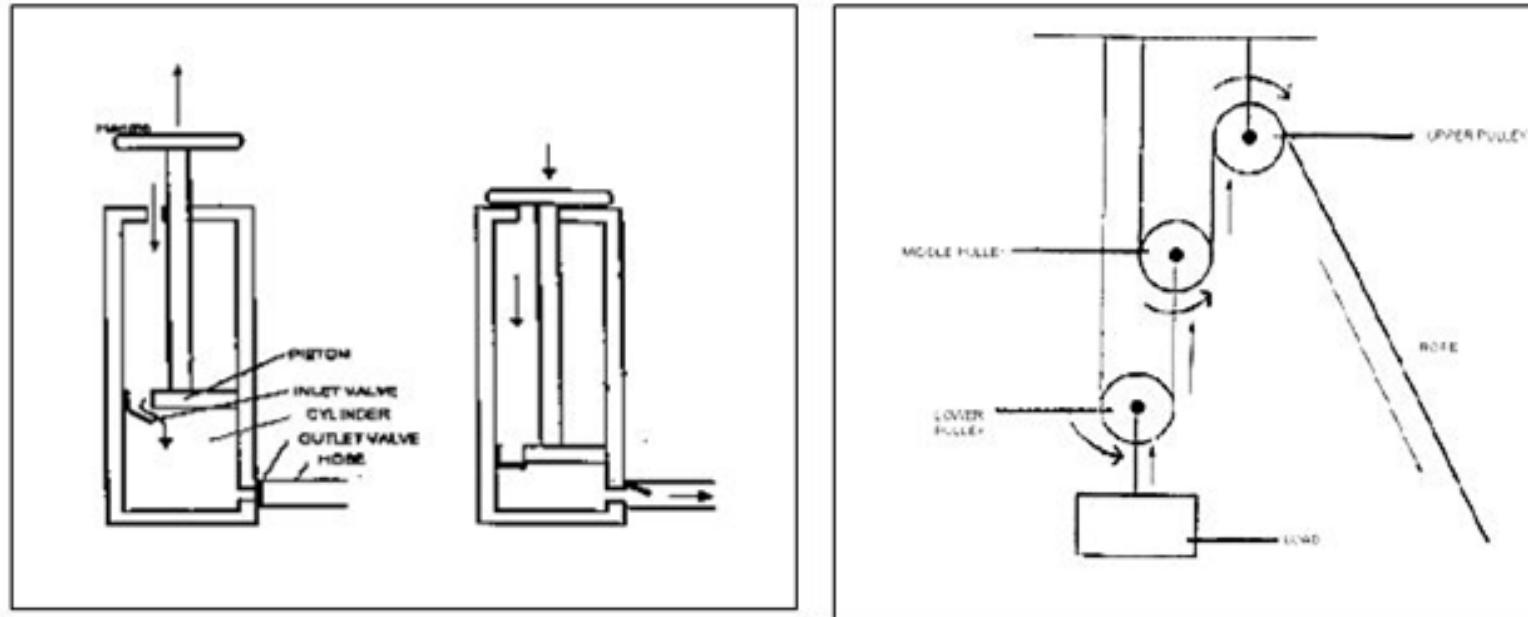


# Sketches provide instructions



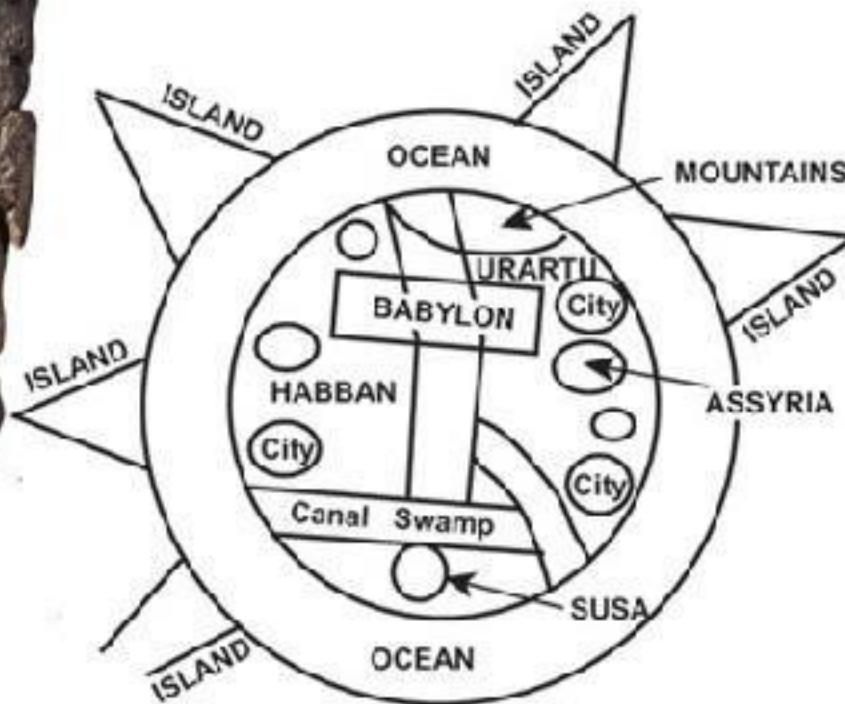
# Sketches explain affordances

People remember more details when they see sketches than when they read

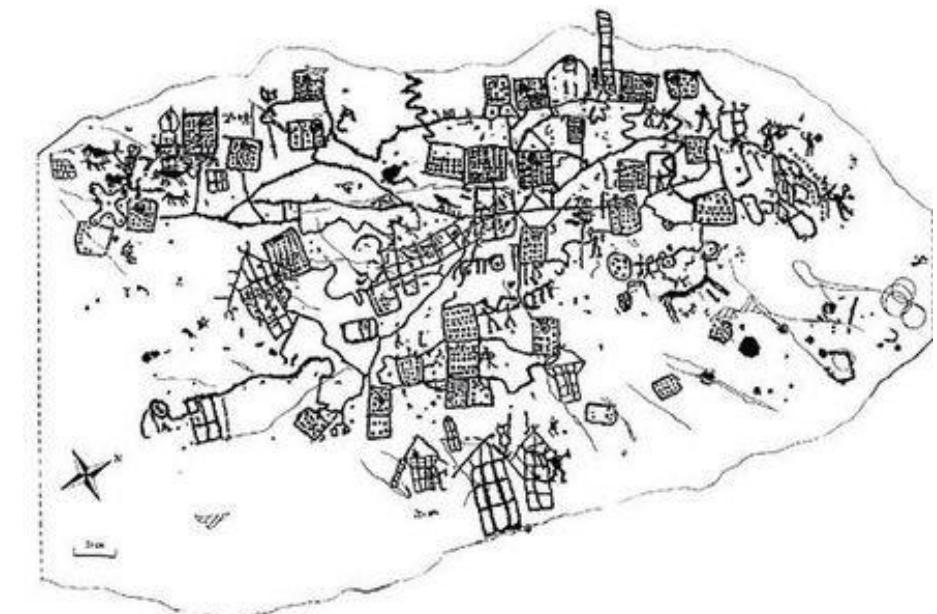


Tversky et al. Animation: can it facilitate?, 2002

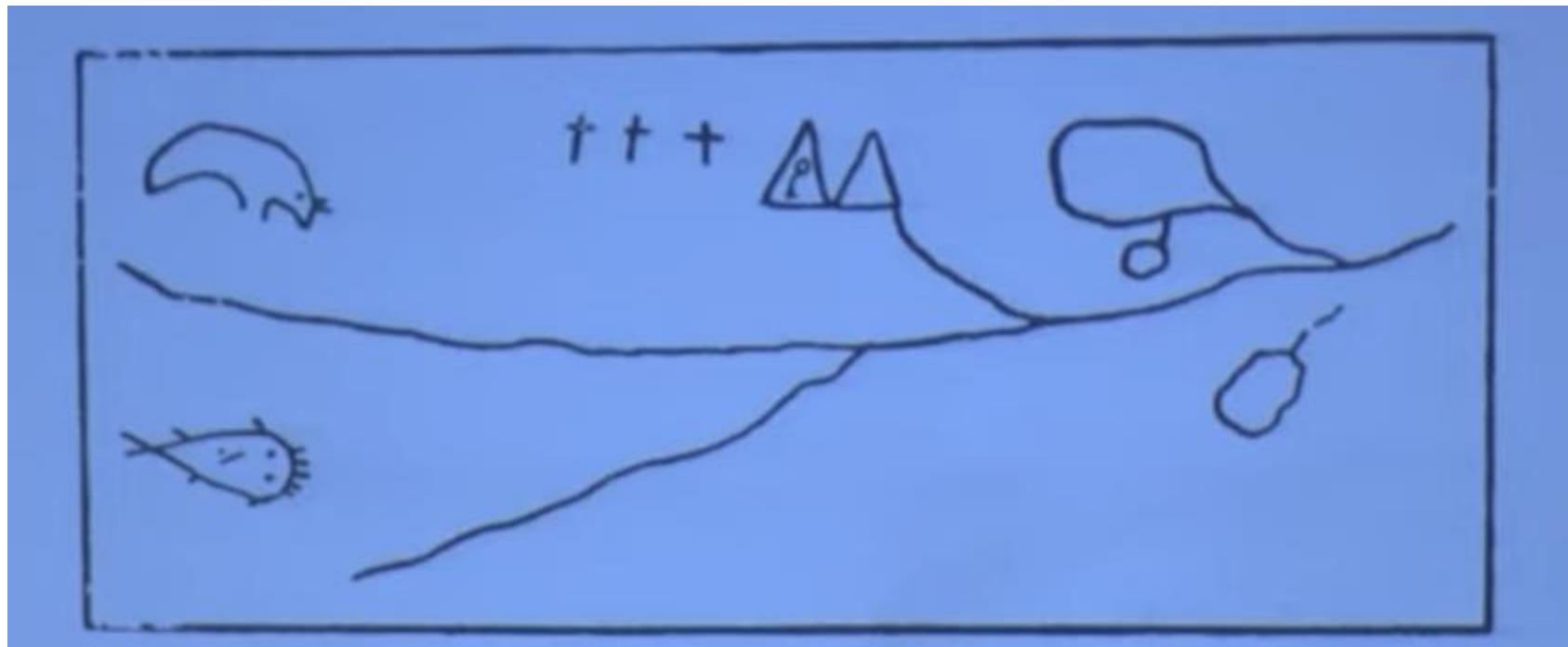
# Sketches as early as ancient babylon



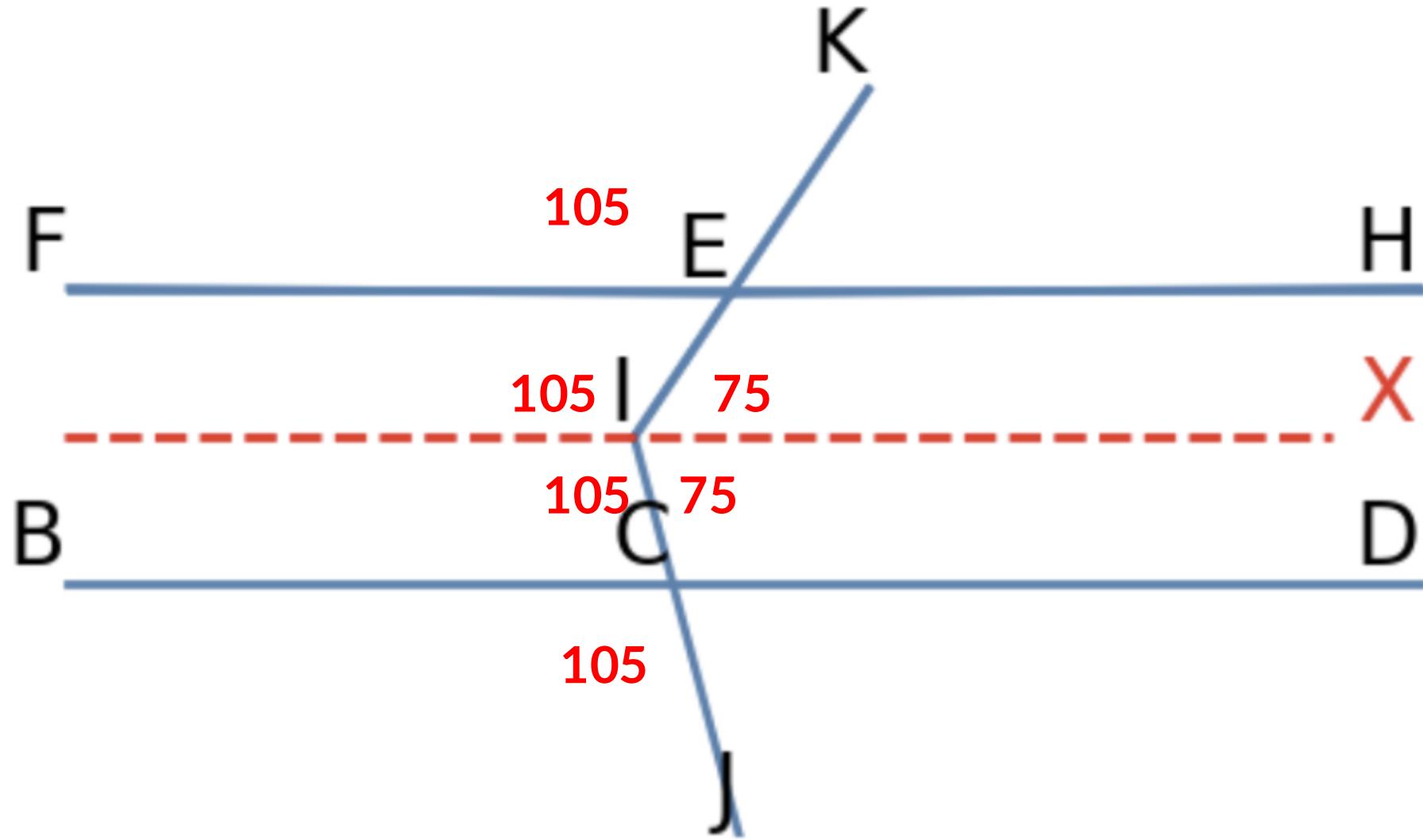
# Bedolina map - first known map from 1000-200BC



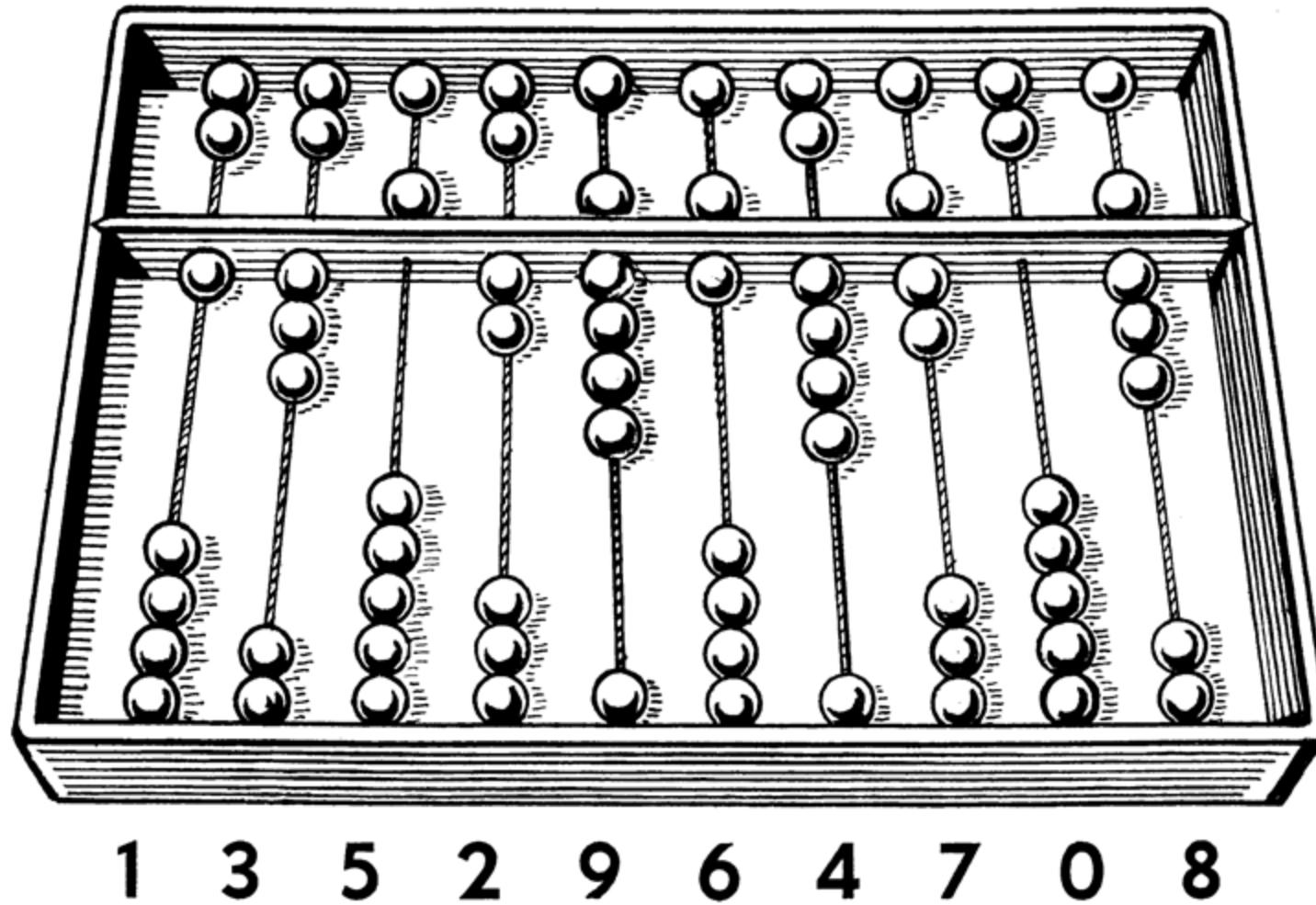
Children learn to sketch very early on



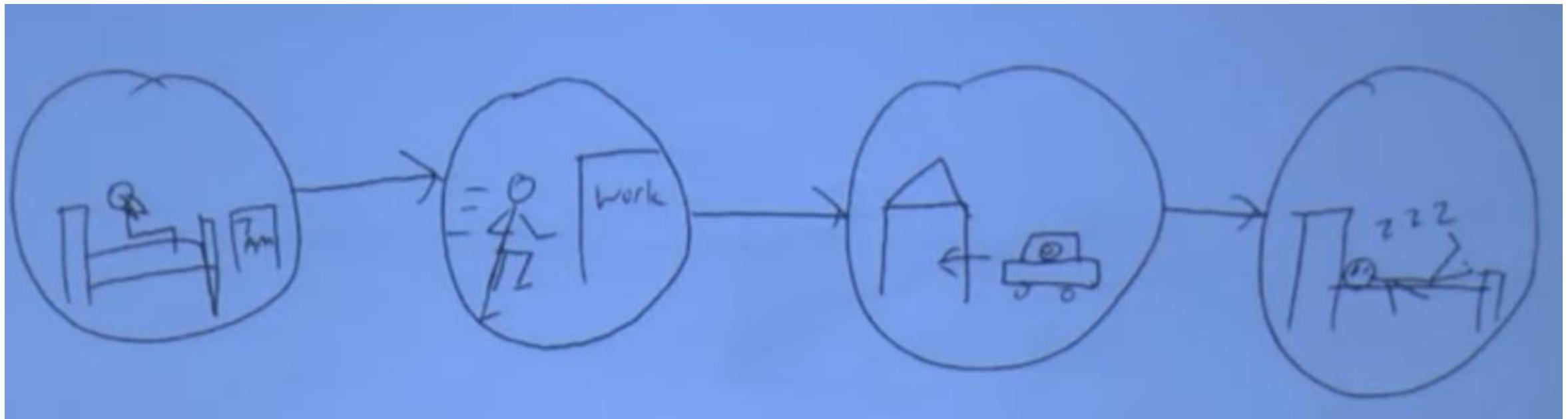
We are taught to sketch to solve math



We are built systems that leverage space to solve arithmetic



We use **sketches** to tell stories and convey time

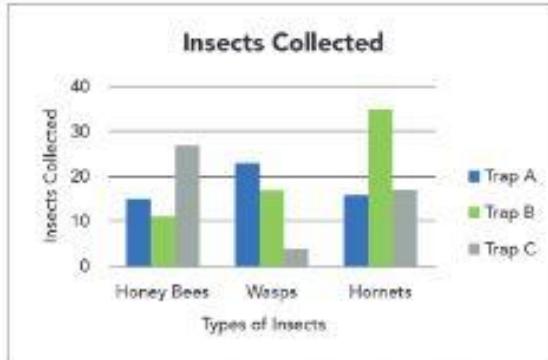


Graphs and visualizations are sketches that are faithful summaries of the underlying data

## Types of Graphs

### Bar Graph

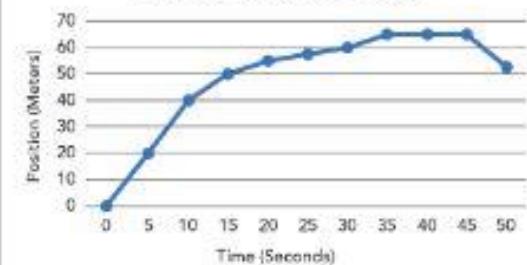
Used for categorical data; good for comparing groups



### Line Graph

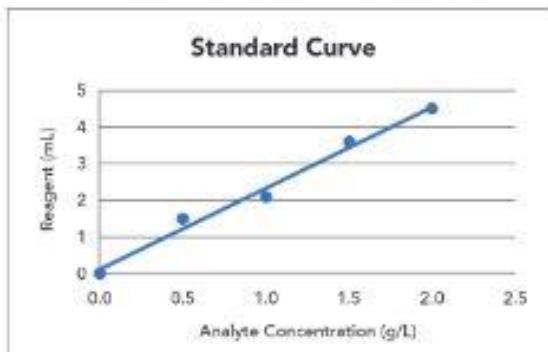
Used for continuous data; good for looking at data over time

#### Distance from the Origin



### Scatterplot (XY Graph)

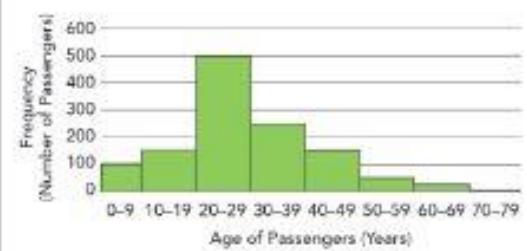
Used to show relationships between 2 variables



### Histogram

Shows frequency data, how often a given variable occurs

#### Passenger Age



Sketches in the form of tables help us arrange items

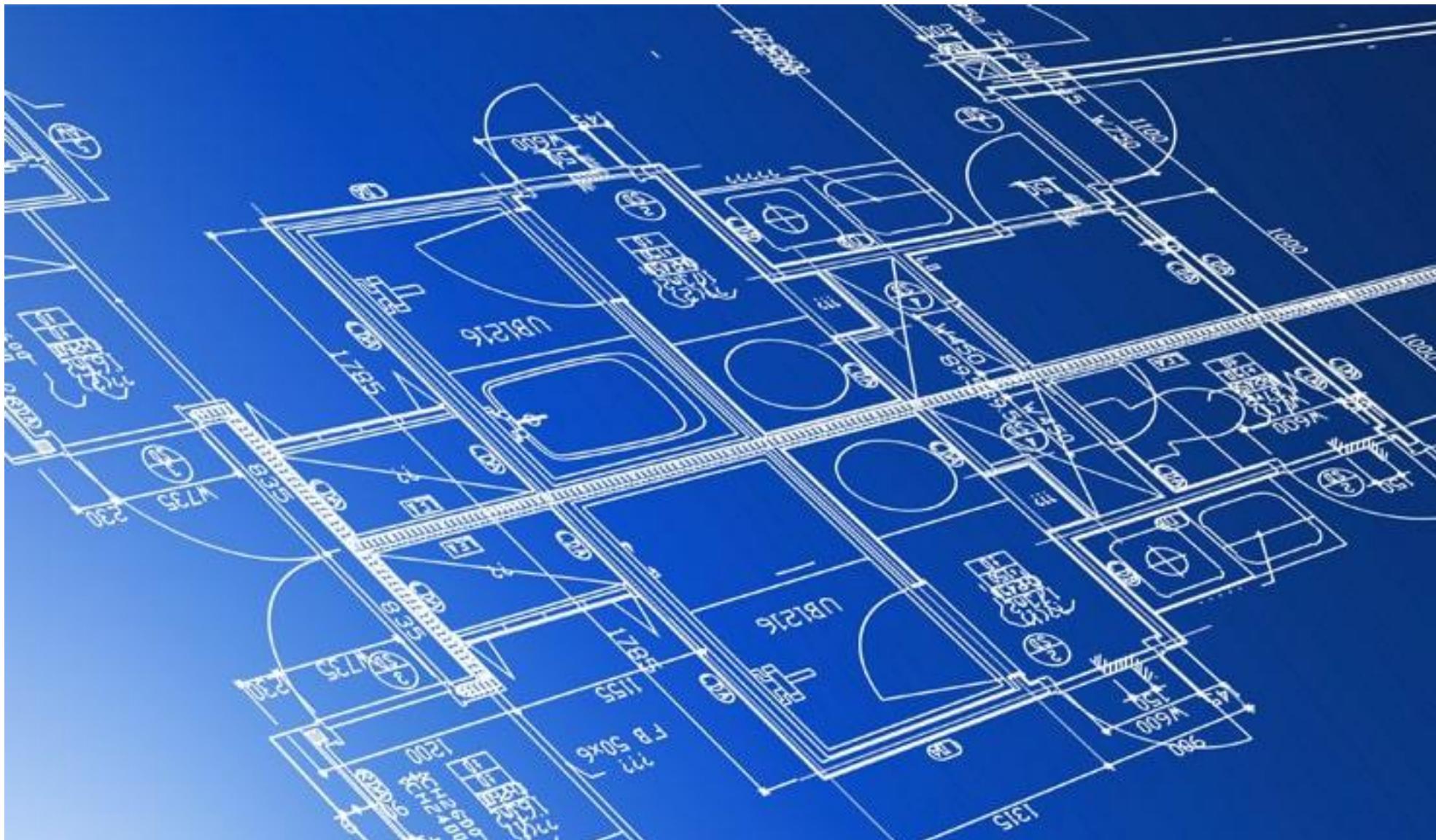
## PERIODIC TABLE OF ELEMENTS

Chemical Group Block

**PubChem**

Atomic Number	Symbol	Atomic Mass, u	Name	Chemical Group Block
1	H	1.0080	Hydrogen Nonmetal	18
2	He	4.00260	Helium Noble Gas	
3	Li	7.0	Lithium Alkali Metal	13
4	Be	9.012183	Beryllium Alkaline Earth Metal	14
11	Na	22.989	Sodium Alkali Metal	15
12	Mg	24.305	Magnesium Alkaline Earth Metal	16
19	K	39.0983	Potassium Alkali Metal	17
20	Ca	40.08	Calcium Alkaline Earth Metal	18
21	Sc	44.95591	Scandium Transition Metal	
22	Ti	47.867	Titanium Transition Metal	
23	V	50.9415	Vanadium Transition Metal	
24	Cr	51.996	Chromium Transition Metal	
25	Mn	54.93804	Manganese Transition Metal	
26	Fe	55.84	Iron Transition Metal	
27	Co	58.93319	Cobalt Transition Metal	
28	Ni	58.693	Nickel Transition Metal	
29	Cu	63.55	Copper Transition Metal	
30	Zn	65.4	Zinc Transition Metal	
31	Ga	69.723	Gallium Post-Transition Metal	
32	Ge	72.63	Germanium Metalloid	
33	As	74.92159	Arsenic Metalloid	
34	Se	78.97	Selenium Metalloid	
35	Br	79.90	Bromine Halogens	
36	Kr	83.80	Krypton Noble Gas	
37	Rb	85.468	Rubidium Alkali Metal	
38	Sr	87.62	Strontrium Alkaline Earth Metal	
39	Y	88.90584	Yttrium Transition Metal	
40	Zr	91.22	Zirconium Transition Metal	
41	Nb	92.90637	Niobium Transition Metal	
42	Mo	95.95	Molybdenum Transition Metal	
43	Tc	96.90636	Technetium Transition Metal	
44	Ru	101.1	Ruthenium Transition Metal	
45	Rh	102.9055	Rhodium Transition Metal	
46	Pd	106.42	Palladium Transition Metal	
47	Ag	107.868	Silver Transition Metal	
48	Cd	112.41	Cadmium Transition Metal	
49	In	114.818	Inium Post-Transition Metal	
50	Sn	118.71	Tin Post-Transition Metal	
51	Sb	121.760	Antimony Metalloid	
52	Te	127.6	Tellurium Metalloid	
53	I	126.9045	Iodine Halogens	
54	Xe	131.29	Xenon Noble Gas	
55	Cs	132.90	Cesium Alkali Metal	
56	Ba	137.33	Barium Alkaline Earth Metal	
72	Hf	178.49	Hafnium Transition Metal	
73	Ta	180.9479	Tantalum Transition Metal	
74	W	183.84	Tungsten Transition Metal	
75	Re	186.207	Rhenium Transition Metal	
76	Os	190.2	Osmium Transition Metal	
77	Ir	192.22	Iridium Transition Metal	
78	Pt	195.08	Platinum Transition Metal	
79	Au	196.96...	Gold Transition Metal	
80	Hg	200.59	Mercury Transition Metal	
81	Tl	204.383	Thallium Post-Transition Metal	
82	Pb	207	Pb Post-Transition Metal	
83	Bi	208.98...	Bismuth Post-Transition Metal	
84	Po	208.98...	Polonium Metalloid	
85	At	209.98...	Astatine Halogens	
86	Rn	222.01	Radon Noble Gas	
104	Rf	247.1	Rutherfordium Transition Metal	
105	Db	248.1	Dubnium Transition Metal	
106	Sg	249.1	Seaborgium Transition Metal	
107	Bh	270.1	Bohrium Transition Metal	
108	Hs	249.1	Hassium Transition Metal	
109	Mt	277.1	Mertensium Transition Metal	
110	Ds	282.1	Darmstadtium Transition Metal	
111	Rg	282.1	Roentgenium Transition Metal	
112	Cn	288.1	Copernicium Post-Transition Metal	
113	Nh	286.1	Nihonium Post-Transition Metal	
114	Fl	290.1	Flerovium Post-Transition Metal	
115	Mc	290.1	Moscovium Post-Transition Metal	
116	Lv	293.2	Livermorium Post-Transition Metal	
117	Ts	294.2	Tennessee Halogens	
118	Og	295.2	Oganesson Noble Gas	
57	La	138.9055	Lanthanum Lanthanide	
58	Ce	140.116	Cerium Lanthanide	
59	Pr	140.90...	Praseodymium Lanthanide	
60	Nd	144.24	Neodymium Lanthanide	
61	Pm	144.91...	Promethium Lanthanide	
62	Sm	150.4	Samarium Lanthanide	
63	Eu	151.964	Europium Lanthanide	
64	Gd	157.2	Gadolinium Lanthanide	
65	Tb	158.92...	Terbium Lanthanide	
66	Dy	162.500	Dysprosium Lanthanide	
67	Ho	164.93...	Holmium Lanthanide	
68	Er	167.26	Erbium Lanthanide	
69	Tm	168.93...	Thulium Lanthanide	
70	Yb	173.05	Ytterbium Lanthanide	
71	Lu	174.9668	Lutetium Lanthanide	
89	Ac	227.02	Actinium Actinide	
90	Th	232.038	Thorium Actinide	
91	Pa	231.03...	Protactinium Actinide	
92	U	238.0289	Uranium Actinide	
93	Np	237.04...	Neptunium Actinide	
94	Pu	244.06...	Plutonium Actinide	
95	Am	243.06...	Americium Actinide	
96	Cm	247.07...	Curium Actinide	
97	Bk	247.07...	Berkelium Actinide	
98	Cf	251.07...	Californium Actinide	
99	Es	252.0830	Einsteinium Actinide	
100	Fm	257.0	Fermium Actinide	
101	Md	258.0...	Mendelevium Actinide	
102	No	259.1...	Nobelium Actinide	
103	Lr	266.1	Lawrencium Actinide	

# Sketches help us build



# Sketches appear in all our research papers

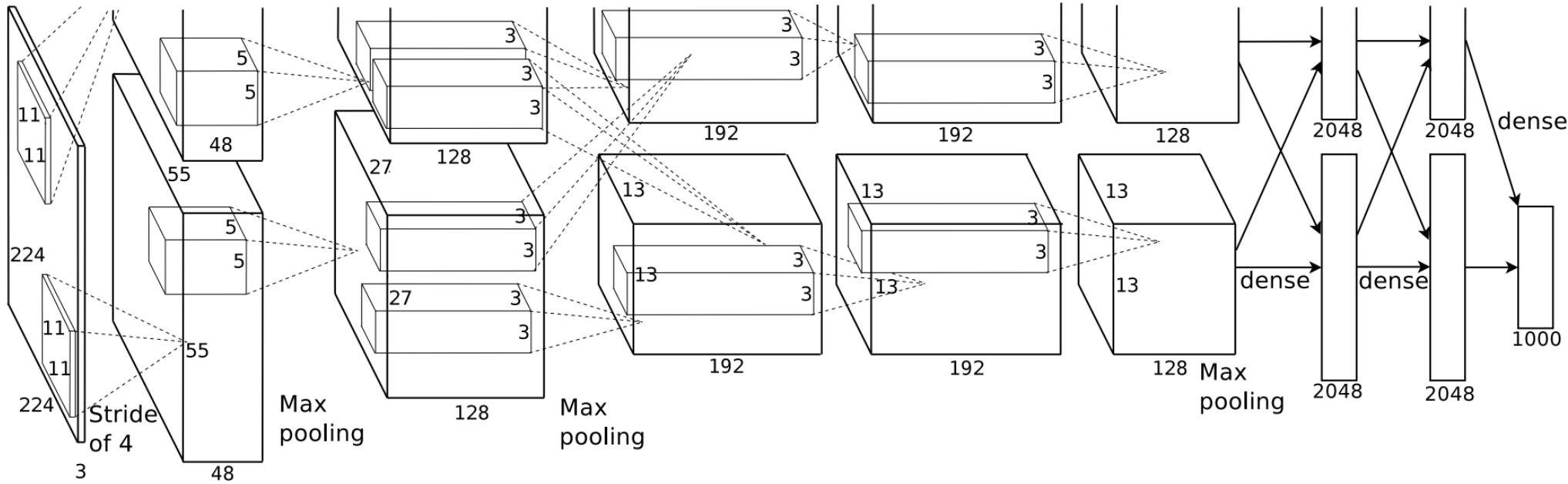


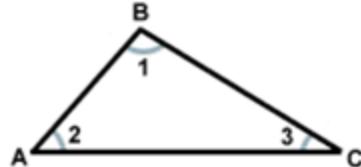
Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

# Sketchpad for VLMs: Geometry

## Geometry



Prove the angles of  $\triangle ABC$  sum to  $180^\circ$



GPT-4o

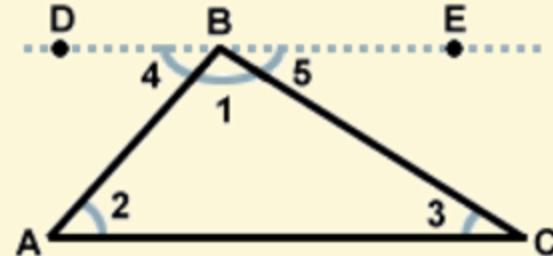
Start by assuming, for contradiction, that the sum is not  $180^\circ$ . We'll introduce a new angle,  $\angle 4$  ...



SketchPad + GPT-4o



Draw line DE parallel to AC



$\angle 4$  is congruent to  $\angle 2$ , and  $\angle 5$  is congruent to  $\angle 3$ .  $\angle 1$ ,  $\angle 4$ , and  $\angle 5$  form a straight angle =  $180^\circ$ . So  $\angle 1$ ,  $\angle 2$ ,  $\angle 3$  add up to  $180^\circ$ .



# Sketchpad for VLM: Math

## Math Function



Is  $f(x)$  an convex function?

$$f(x) = 0.14 |x| - 5.09$$



GPT-4o

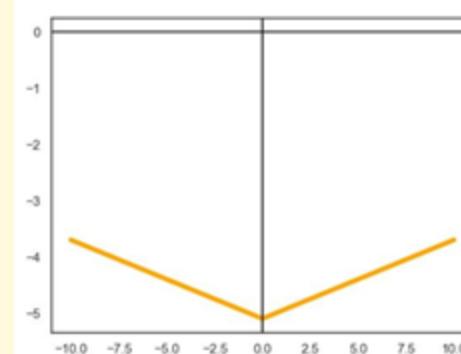
It is convex for  $x > 0$   
but concave for  $x < 0$



SketchPad + GPT-4o



Plot the function



convex because line between any two points lie above the graph.



# Sketchpad for LLM: Visual Search

## Visual Search

What is the type of the shop?



The sign in the image reads “ceremonial matcha,.”. it's a cafe.



SketchPad + GPT-4o

Use **sliding\_window** to find the board



I see a board in patch 2 and 3 and will **zoom in**



The zoomed-in image shows the text **“corepower YOGA”**. So it's a yoga studio



# Sketchpad for LLM: Spatial Reasoning

## Spatial Reasoning



Are there cookies stacked on other cookies?



No



SketchPad + GPT-4o



Use **segmentation & marking**



Cookies 2 and 5 seem to overlap. Need to analyze their depth



Use **depth estimation**



Warmer colors indicate closer proximity, suggesting they are stacked on cooler-colored ones.



# Sketching to Solve Math Problems

## Mathematical Functions

### Task: Count Breakpoints

Text Representation

$$f(x) = \begin{cases} -26.25x - 220.97, & -10.31 < x \leq -4.61 \\ 75.84x + 249.44, & -4.61 < x \leq -1.97 \\ 61.92 - 19.33x, & -1.97 < x \leq 8.36 \\ 32.12x - 368.48, & 8.36 < x \leq 14.58 \end{cases}$$

Image Representation



Answer: 3

### Task: Function Parity

Text Representation

$$f(x) = -\frac{18x}{2x^{10} + 16x^4 - 10x^2 + 29.34}$$

Image Representation



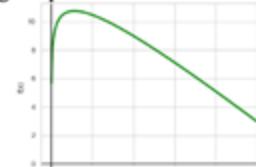
Answer: Odd Function

### Task: Function Convexity

Text Representation

$$f(x) = 1.36 \cdot \log(x) - 1.21 \cdot |x| + 11.95$$

Image Representation



Answer: Concave Function

## Science Questions

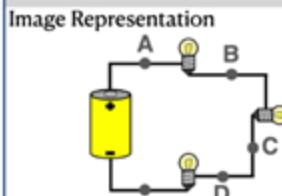
### Task: Physics QA

Text Representation

Question: Consider the given circuit. What is the current in amperes if 9.9 Coulombs of charge flow past point A in 1.1 seconds?

- (A) 6 (B) 7 (C) 8 (D) 9

Description: The circuit consists of a single battery connected to three light bulbs arranged in a parallel configuration. A, B, C, D, and E are points in the circuit. The battery provides the electrical energy that powers the light bulbs.



Answer: (D) 9 Amps

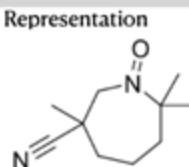
### Task: Chemistry QA

Text Representation

Question: How many nitrogens are in the following organic molecule?

- (A) 0 (B) 1 (C) 2 (D) 3

Description: The SMILES notation of the organic molecule is CC#C#N(C(=O)C#N)C#O.



Answer: (C) 2 nitrogens

## Graph Algorithms

### Task: Maximum Flow

Text Representation

0	6	6	1	7
0	0	0	0	6
0	5	0	0	9
0	3	2	0	7
0	0	0	0	0

Adjacency Matrix:

$$\begin{bmatrix} 0 & 6 & 6 & 1 & 7 \\ 0 & 0 & 0 & 0 & 6 \\ 0 & 5 & 0 & 0 & 9 \\ 0 & 3 & 2 & 0 & 7 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Source: 0; Sink: 4

Image Representation



Answer: 20

### Task: Connectivity

Text Representation

Adjacency Matrix:

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Query Nodes: 4 and 8.

Image Representation



Answer: Not Connected

### Task: Graph Isomorphism

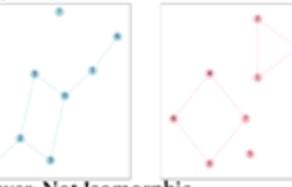
Text Representation

Adjacency Matrices:

G =  $\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$

H =  $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$

Image Representation



Answer: Not Isomorphic

## Chess Games

### Task: Winner Identification

Text Representation

p R . . r . . k .  
Q . . . . p P P P  
P . . . . p . . .  
P . . . . K q . .  
. . . B p . . P  
. . . R . . .

Image Representation



Answer: Black Wins

### Task: Chess Puzzles

Text Representation

FEN: r2qk2r/ppp2ppp/  
2n2n2/6B1/b2Nb1/5N2/PPP1PPP/  
R2QKB1R w KQkq - 6 8

Image Representation



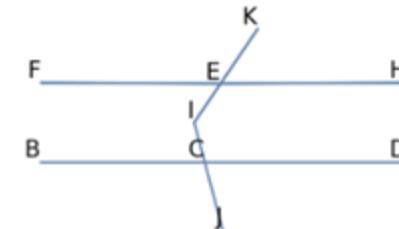
Answer: c2c3

## Geometry

# How do we enable VLMs to sketch?



Given  $\angle BCJ = 105^\circ$ ,  
 $\angle KEF = 105^\circ$ . Find  $\angle EIC$

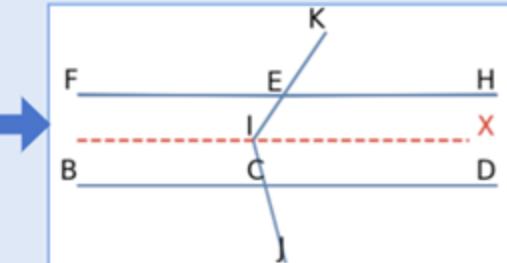


**THOUGHT 1:** I need to draw an auxiliary Line IX parallel to Line BD

**ACTION 1:** Draw the aux. line using Python Code

```
# Define auxiliary point X for the line IX  
x_I = points["I"][0]  
y_I = points["I"][1]  
x_X = x_I + 100  
y_X = y_I + slope_bd * (x_X - x_I)  
# Plot the auxiliary line IX in red color  
ax.plot([x_I, x_X], [y_I, y_X], 'red') ...
```

**OBSERVATION 1**



**THOUGHT 2:** Given  $\angle BCJ = \angle KEF = 105^\circ$ , their supplementary angles,  $\angle KEH = \angle DCJ = 180^\circ - 105^\circ = 75^\circ$ . Given BD parallel to IX, it follows that  $\angle EIX = \angle KEH$ , and  $\angle XIC = \angle DCJ$ , both measuring  $75^\circ$ . Therefore,  $\angle EIC = \angle EIX + \angle XIC = 150^\circ$

# How do we enable GPT-4o to sketch?

## React-style use of tools to sketch

### Graph Connectivity



Is there a path between node 0 and node 4 in the graph?

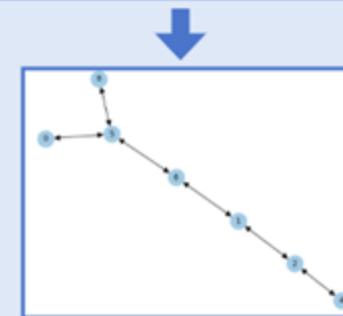
```
adjacency_matrix = [[0, 0, 0, 0, 0, 1, 0, 0, 0],  
[0, 0, 1, 0, 0, 0, 1, 0, 0],  
[0, 1, 0, 0, 1, 0, 0, 0, 0],  
[0, 0, 0, 0, 0, 0, 0, 0, 0],  
[0, 0, 1, 0, 0, 0, 0, 0, 0],  
[1, 0, 0, 0, 0, 0, 1, 0, 1],  
[0, 1, 0, 0, 0, 1, 0, 0, 0],  
[0, 0, 0, 0, 0, 0, 0, 0, 0],  
[0, 0, 0, 0, 0, 1, 0, 0, 0]]
```

**THOUGHT 1:** We can use the networkx library to plot the graph

**ACTION 1:** Call Python Ploting tool

```
G = nx.from_numpy_matrix(np.array(adjacency_matrix),  
create_using=nx.DiGraph)  
pos = nx.spring_layout(G)  
nx.draw(G, pos, with_labels=True)
```

**OBSERVATION 1**



**THOUGHT 2:** From the graph, I saw there is a path between node 0 and node 4.

# Sketching can solve math problems

Model	Geometry		Graph			Math		Game
	Geometry		Maxflow	Isomorphism	Connectivity	Convexity	Parity	Winner ID
<i>Prior LLMs without visual inputs</i>								
Gemini-Pro	\	15.6	47.7	50.0	87.9	48.2	8.1	
Claude 3 OPUS	\	56.3	50.0	82.0	93.0	77.6	74.4	
Mixtral 8x7B [19]	\	8.6	50.0	62.5	69.1	41.7	7.4	
LLaMA-2-70B [45]	\	18.0	50.0	50.0	74.2	33.3	12.4	
<i>Latest multimodal LLMs + Visual Sketchpad</i>								
GPT-4 Turbo	37.5	32.8	62.5	66.0	57.0	80.5	50.4	
+ Sketchpad	45.8	63.3	64.2	95.1	<b>93.1</b>	<b>93.1</b>	74.3	
	+8.3	+30.5	+1.7	+29.1	+25.4	+12.6	+23.9	
GPT-4o	62.5	25.0	50.8	96.1	87.2	84.4	61.1	
+ Sketchpad	<b>66.7</b>	<b>66.3</b>	<b>65.3</b>	<b>98.1</b>	90.1	88.1	<b>81.2</b>	
	+4.2	+41.3	+14.5	+2.0	+2.9	+3.7	+20.1	

Table 1: Accuracy scores on geometry problems, graph algorithms, mathematical functions, and game. **SKETCHPAD yields large performance gains across all tasks and outperform all baselines.**

# Large improvements across multiple benchmarks

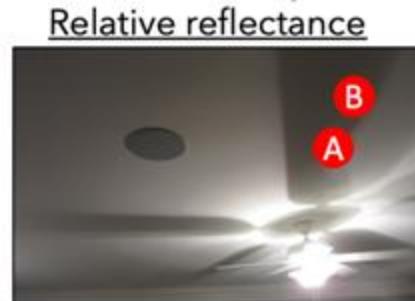
Model	Geometry		Graph			Math		Game
	Geometry		Maxflow	Isomorphism	Connectivity	Convexity	Parity	Winner ID
<i>Prior LLMs without visual inputs</i>								
Gemini-Pro	\	15.6	47.7	50.0	87.9	48.2	8.1	
Claude 3 OPUS	\	56.3	50.0	82.0	93.0	77.6	74.4	
Mixtral 8x7B [19]	\	8.6	50.0	62.5	69.1	41.7	7.4	
LLaMA-2-70B [45]	\	18.0	50.0	50.0	74.2	33.3	12.4	
<i>Latest multimodal LLMs + Visual Sketchpad</i>								
GPT-4 Turbo	37.5	32.8	62.5	66.0	57.0	80.5	50.4	
+ Sketchpad	45.8	63.3	64.2	95.1	<b>93.1</b>	<b>93.1</b>	74.3	
	<b>+8.3</b>	<b>+30.5</b>	<b>+1.7</b>	<b>+29.1</b>	<b>+25.4</b>	<b>+12.6</b>	<b>+23.9</b>	
GPT-4o	62.5	25.0	50.8	96.1	87.2	84.4	61.1	
+ Sketchpad	<b>66.7</b>	<b>66.3</b>	<b>65.3</b>	<b>98.1</b>	90.1	88.1	<b>81.2</b>	
	<b>+4.2</b>	<b>+41.3</b>	<b>+14.5</b>	<b>+2.0</b>	<b>+2.9</b>	<b>+3.7</b>	<b>+20.1</b>	

Table 1: Accuracy scores on geometry problems, graph algorithms, mathematical functions, and game. **SKETCHPAD yields large performance gains across all tasks and outperform all baselines.**

# Sketching to solve BLINK

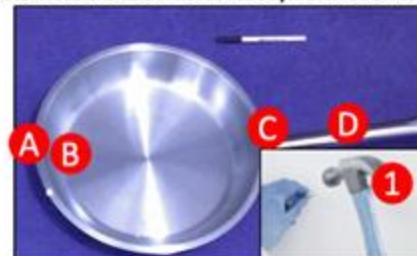


Which point is closer?



Which point is darker?

## Functional correspondence



Which points have similar affordance when pulling out a nail?

## Jigsaw



Which image fits here?

## Multi-view reasoning



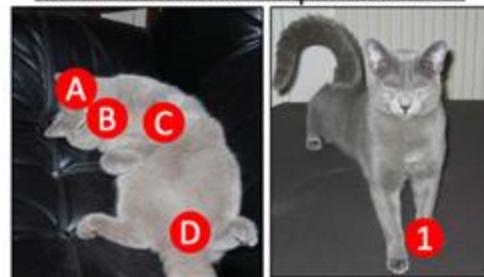
Is camera moving right?

## Visual correspondence



Which point is the same?

## Semantic correspondence



Which points have similar semantics?

## Forensics detection



Which image is real?



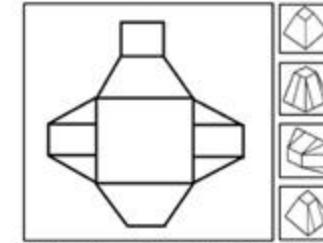
visual tasks beyond language descriptions

## Visual similarity



Which image is more similar to the left?

## IQ Test



Which object does it fold into?

# Example sketching on BLINK

## Relative depth



Which point is closer to the camera, A or B?



**THOUGHT 1:** I can use a depth estimation tool that provides a depth map with colors indicating distances.

**ACTION 1:** Call depth estimation

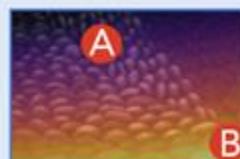
```
depth_map = depth(image)  
display(depth_map)
```



**THOUGHT 2:** The depth map uses colors from purple (farther) to yellow (closer). I need to overlay the original image with the depth map

**ACTION 2:** Call overlay

```
overlay = overlay_images(depth_map, image)  
display(overlay)
```



**THOUGHT 3:** It is evident that point B, in the yellow region, is closer to the camera compared to point A, which is in the purple region

## Visual Search

### Example sketching on BLINK



Tell me the number of that player who is shooting



**THOUGHT 1:** I need to detect the players in the image and zoom in on the player who is shooting to see the number clearly

**ACTION 1:** 🖌 Call object detection

```
new_img, boxes = detection(image_1, ["player"])
display(new_img.annnotated_image)
```



**THOUGHT 2:** From the image, Player2 seems to be shooting. I need to zoom in on this player to see the number on their jersey clearly

**ACTION 2:** 🖌 Call overlay

```
shooter = boxes[1]
zoomed = zoom_in_image_by_bbox(image_1, shooter)
display(zoomed)
```



**THOUGHT 3:** From the zoomed-in image, I can clearly see that the number of the player who is shooting is 8

# Sketching to solve MMVP (Eyes wide shut)



Q: Is the dog facing left or right from the camera's perspective?

The dog is **facing to the right** from the camera's perspective. You can tell by the position of its tail and the orientation of its body.



Q: Do you see any window in this image?

No, there are **no windows visible** in this image. The image shows a traditional red pillar box, which is a type of mailbox found in the United Kingdom.



Q: Is the door of the truck cab open?

No, the door of the truck is **not open** in the image provided.



Q: In this image, how many eyes can you see on the animal?

The image shows **one eye** of the animal. It's a close-up of a bald eagle facing slightly to its right, which presents only one side of its face to the viewer.



Q: How many wheels can you see in the image?

In the image provided, I can see **two wheels** on the visible side of the car.



Q: Does the keyboard have a backlight?

Based on the image, it appears that the keyboard **does have a backlight**. This is indicated by the illumination seen around the keys, which suggests that there is a light source beneath them to enhance visibility in low-light conditions.



Q: From the camera's perspective, is the piano's back panel on the left or right side?

From the camera's perspective, the piano's back panel is **on the left side**. The keys visible in the foreground suggest that we are looking at the piano from front-right side, thus placing the back panel on the left from this angle.



Q: Is the edge of the heart in the picture white or dark-colored?

The image you've provided is small and of low resolution, but it appears that the hearts have a **dark-colored edge** or outline. This outline varies in visibility, likely due to the quality of the image.



Q: Is the school bus parked facing the camera or away from the camera?

The school bus is parked **facing away from the camera**. You can tell because the rear lights and the back of the exterior rearview mirror are visible. The text "SCHOOL BUS" is also reversed, which is typically seen on the front of a bus, designed so that it reads correctly in a rearview mirror.



Q: Are the butterfly's feet visible?

The butterfly's feet are **not visible** in this image. The wings and part of the body of the butterfly are in view, but the feet are either hidden beneath the body or out of the frame of the picture. Butterflies have delicate, slender feet which are often not the most noticeable feature when they are perched, especially from the angle shown in this image.

# Sketching can solve perception tasks

Model	<i>V</i> *Bench	MMVP	Depth	Spatial	Jigsaw	Vis. Corr.	Sem. Corr.
<i>Prior multimodal LLMs</i>							
LLaVA-1.5-7B [26]	48.7	-	52.4	61.5	11.3	25.6	23.0
LLaVA-1.5-13B [26]	-	24.7	53.2	67.8	58.0	29.1	32.4
LLaVA-NeXT-34B [27]	-	-	67.7	74.8	54.7	30.8	23.7
Claude 3 OPUS [1]	-	-	47.6	58.0	32.7	36.6	25.2
Gemini-Pro [41]	48.2	40.7	40.3	74.8	57.3	42.4	26.6
GPT-4V-preview [35]	55.0	38.7	59.7	72.7	70.0	33.7	28.8
Previous state of the art	75.4 [50]	49.3 [10]	67.7 [27]	76.2 [42]	70.0 [35]	42.4 [41]	33.1 [48]
<i>Latest multimodal LLMs + Visual Sketchpad</i>							
GPT-4 Turbo	52.5	71.0	66.1	68.5	64.7	48.8	30.9
+ Sketchpad	71.0	73.3	68.5	80.4	68.5	52.3	42.4
	+18.5	+2.3	+2.4	+11.9	+3.8	+3.5	+11.5
GPT-4o	66.0	85.3	71.8	72.0	64.0	73.3	48.6
+ Sketchpad	80.3	86.3	83.9	81.1	70.7	80.8	58.3
	+14.3	+1.0	+12.1	+9.1	+6.7	+7.5	+9.7

Table 2: Accuracy on complex visual reasoning tasks. **SKETCHPAD enhances both GPT-4 Turbo and GPT-4o performance, establishing new SOTA performance levels on all the tasks.**

# Large improvements across multiple benchmarks

Model	<i>V</i> *Bench	MMVP	Depth	Spatial	Jigsaw	Vis. Corr.	Sem. Corr.
<i>Prior multimodal LLMs</i>							
LLaVA-1.5-7B [26]	48.7	-	52.4	61.5	11.3	25.6	23.0
LLaVA-1.5-13B [26]	-	24.7	53.2	67.8	58.0	29.1	32.4
LLaVA-NeXT-34B [27]	-	-	67.7	74.8	54.7	30.8	23.7
Claude 3 OPUS [1]	-	-	47.6	58.0	32.7	36.6	25.2
Gemini-Pro [41]	48.2	40.7	40.3	74.8	57.3	42.4	26.6
GPT-4V-preview [35]	55.0	38.7	59.7	72.7	70.0	33.7	28.8
Previous state of the art	75.4 [50]	49.3 [10]	67.7 [27]	76.2 [42]	70.0 [35]	42.4 [41]	33.1 [48]
<i>Latest multimodal LLMs + Visual Sketchpad</i>							
GPT-4 Turbo	52.5	71.0	66.1	68.5	64.7	48.8	30.9
+ Sketchpad	71.0	73.3	68.5	80.4	68.5	52.3	42.4
	+18.5	+2.3	+2.4	+11.9	+3.8	+3.5	+11.5
GPT-4o	66.0	85.3	71.8	72.0	64.0	73.3	48.6
+ Sketchpad	80.3	86.3	83.9	81.1	70.7	80.8	58.3
	+14.3	+1.0	+12.1	+9.1	+6.7	+7.5	+9.7

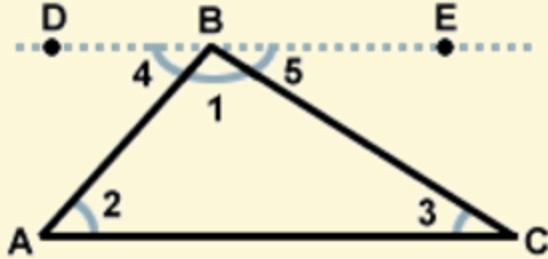
Table 2: Accuracy on complex visual reasoning tasks. **SKETCHPAD enhances both GPT-4 Turbo and GPT-4o performance, establishing new SOTA performance levels on all the tasks.**

# Humans draw the same auxiliary lines for Geometry questions

(N=2) human participants, when asked to solve a geometry problem, draw the same auxiliary lines as GPT-4o **80% of the time.**

 SketchPad + GPT-4o

 Draw line DE parallel to AC



$\angle 4$  is congruent to  $\angle 2$ , and  $\angle 5$  is congruent to  $\angle 3$ .  $\angle 1$ ,  $\angle 4$ , and  $\angle 5$  form a straight angle =  $180^\circ$ . So  $\angle 1$ ,  $\angle 2$ ,  $\angle 3$  add up to  $180^\circ$ .



# GPT-4o correctly uses vision tools to sketch on images

Human evaluation of GPT-4o plans finds that the tool usage is valid in 92.8% of instances.

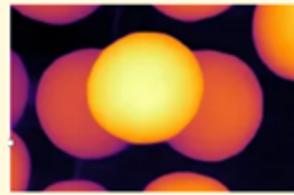
Most of the remaining errors on these benchmarks are because of failures of specialized models, not planning

 **SketchPad + GPT-4o**

 **Use segmentation & marking**

 Cookies 2 and 5 seem to overlap. Need to analyze their depth

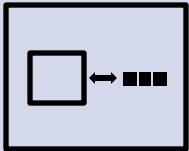
 **Use depth estimation**

 Warmer colors indicate closer proximity, suggesting they are stacked on cooler-colored ones. 

Open-source models also improve if they have GPT-4o generated sketches

Model	Geometry	Maxflow	Convexity	Winner ID
LLaVA-NeXT-13B	11.1	7.8	50.39	5.8
+ oracle Sketchpad	22.2	10.2	50.0	36.7
LLaVA-NeXT-34B	26.1	0.8	81.6	49.0
+ oracle Sketchpad	28.3	14.1	87.1	49.4

But open sourced models aren't able to effectively sketch



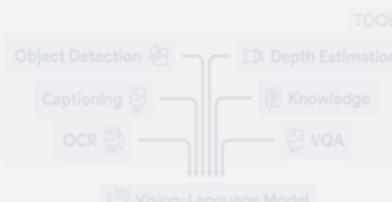
## Prioritizing perception

Perceptual tests for VLMs  
[ECCV 2024]

Sketching for perceptual reasoning  
[NeurIPS 2024] [CVPR 2025]

Distilling perceptual capabilities  
[ACL 2023] [CVPR 2024] [CVPR 2025]

Enabling robots to sketch  
[ArXiv 2025]

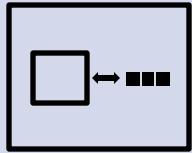


Most fundamental vision capabilities are still out of reach

Enabling sketching:  
visual chain of thought

How can we distill from specialist models into generalist VLMs?

Complete open Action Reasoning model for robotics



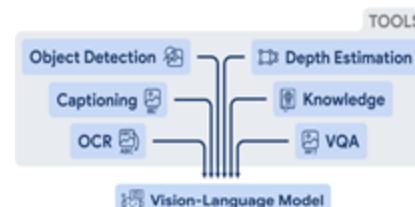
## Prioritizing perception

Perceptual tests for VLMs  
[ECCV 2024]

Sketching for perceptual reasoning  
[NeurIPS 2024] [CVPR 2025]

Distilling perceptual capabilities  
[ACL 2023] [CVPR 2024] [CVPR 2025]

Enabling robots to sketch  
[ArXiv 2025]



Most fundamental vision capabilities are still out of reach

Enabling sketching:  
visual chain of thought

How can we distill from specialist models into generalist VLMs?

Complete open Action Reasoning model for robotics

# Strengths of sketching

- LLMs are good at **generating code**.
- We have good **specialized models** for tasks such as:
  - low level vision tasks (e.g., object detectors, segmentation, depth)
  - search & retrieval
  - encyclopedia knowledge
  - domain-specific models

# Limitations of sketching

- **Error accumulation:**
  - Programs can be wrong
  - Tools can be wrong
- **Latency:**
  - Inference for each sample requires generating code and calling multiple tools.

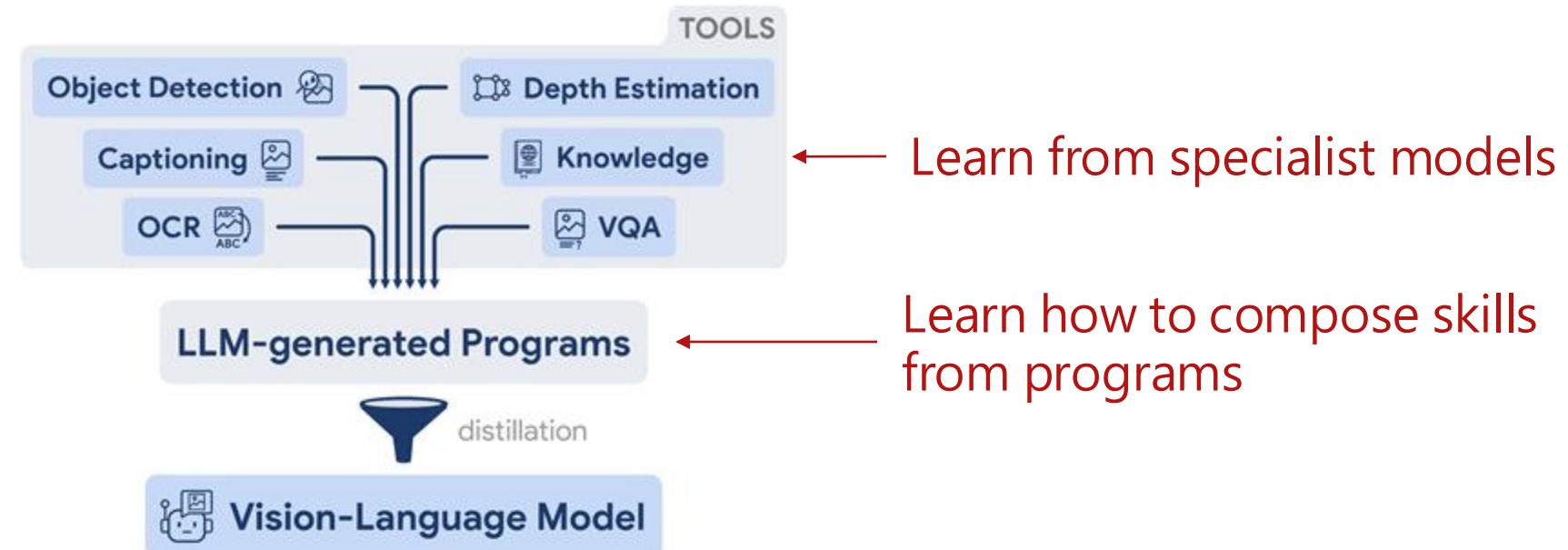


Q: How many red balloons are there?

GPT-4o + sketching: 1

# Can we distil perceptual capabilities from specialist models to VLMs?

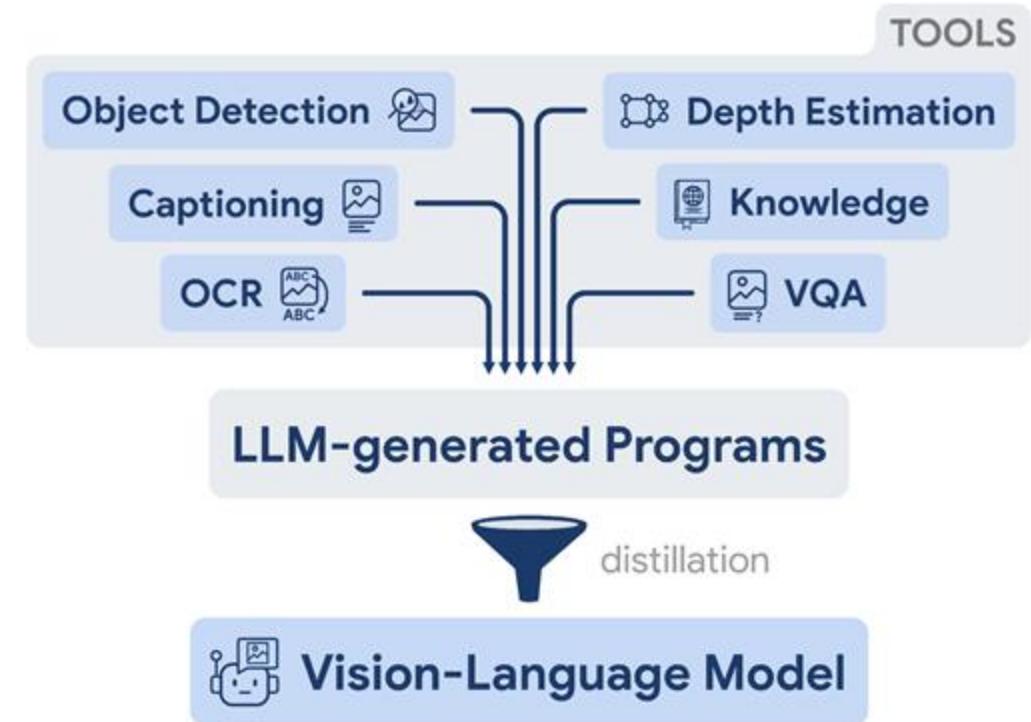
Our solution: **visual program distillation (VPD)**



# Visual Program Distillation (VPD): Training phase

Step 1: Generate training data using programs.

Step 2: Fine-tune VLMs on the generated data.



# Visual Program Distillation (VPD)

## Input Data

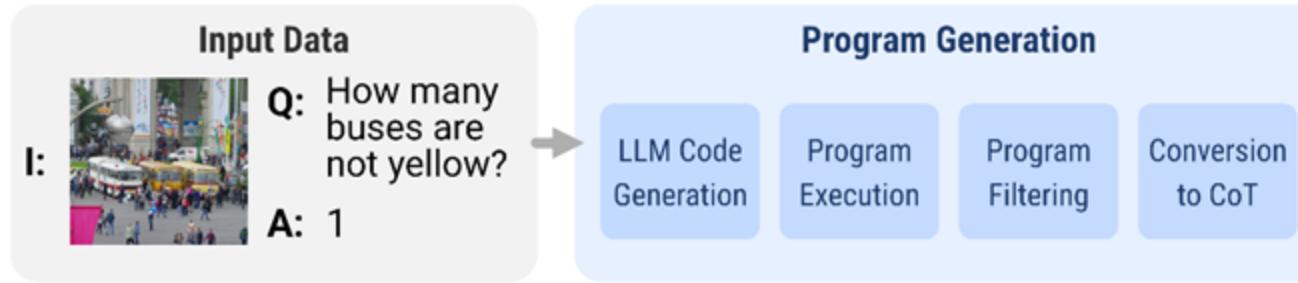
I:



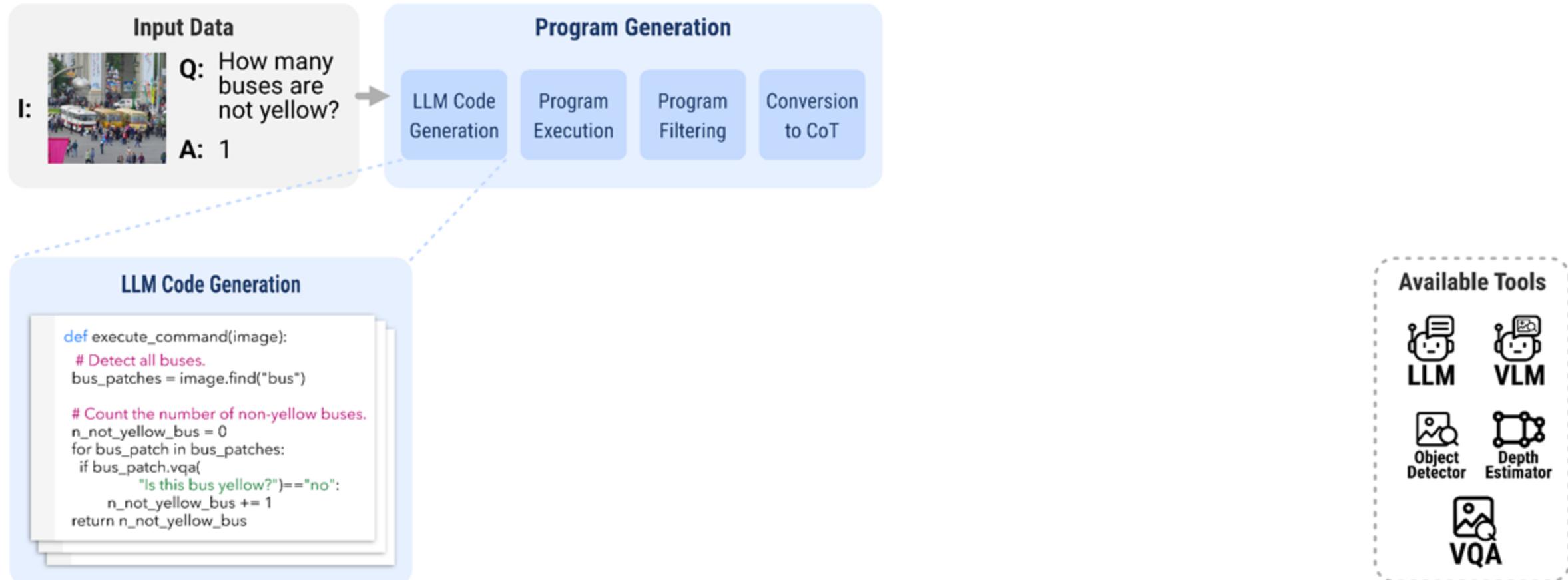
Q: How many  
buses are  
not yellow?

A: 1

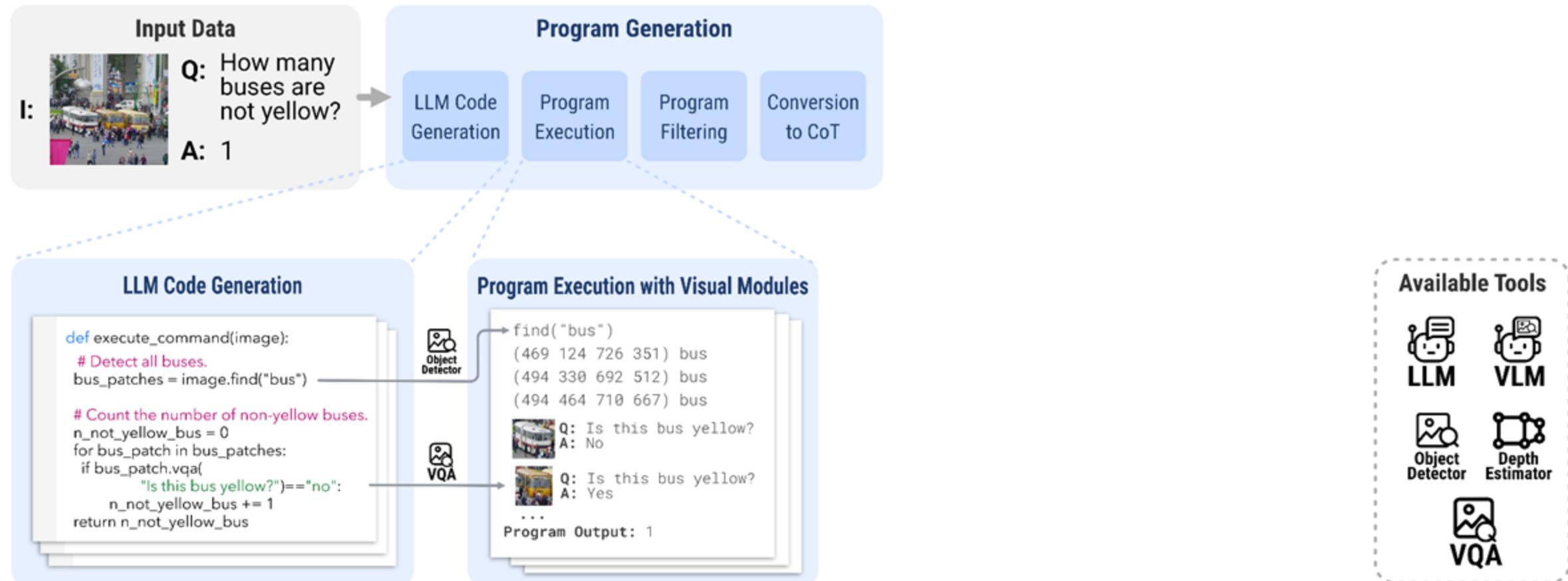
# Visual Program Distillation (VPD): Training phase



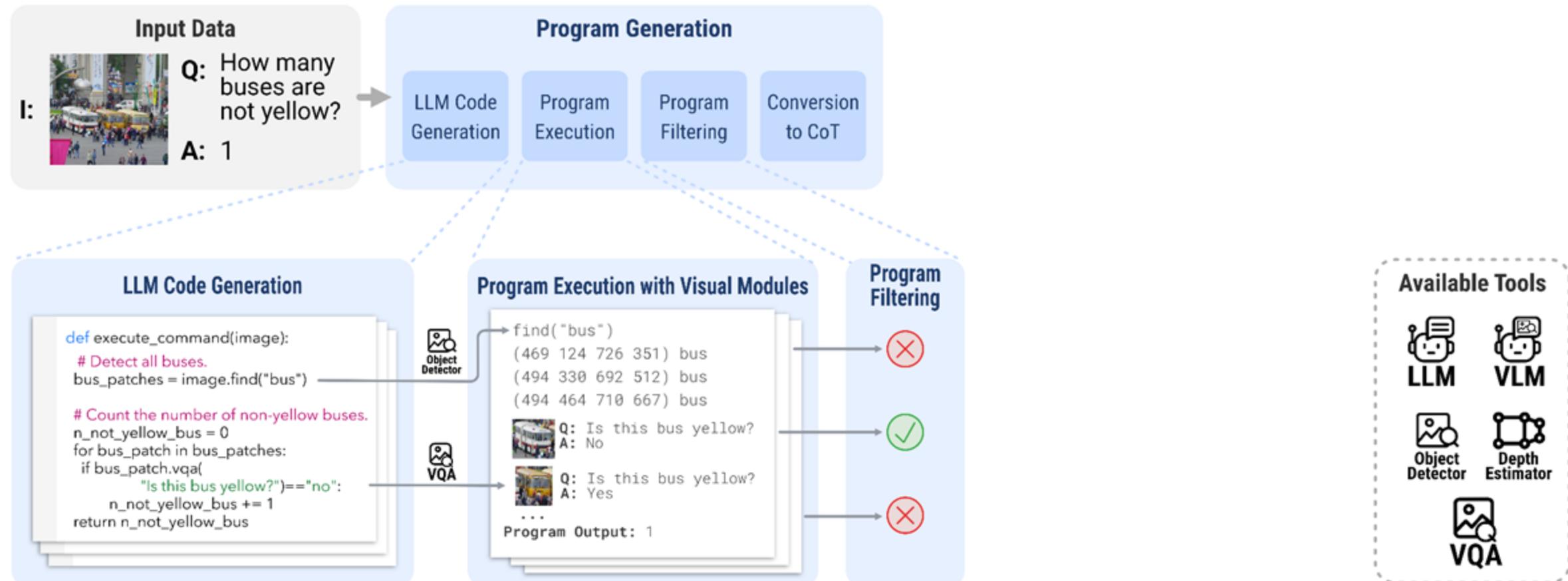
# Visual Program Distillation (VPD): Training phase



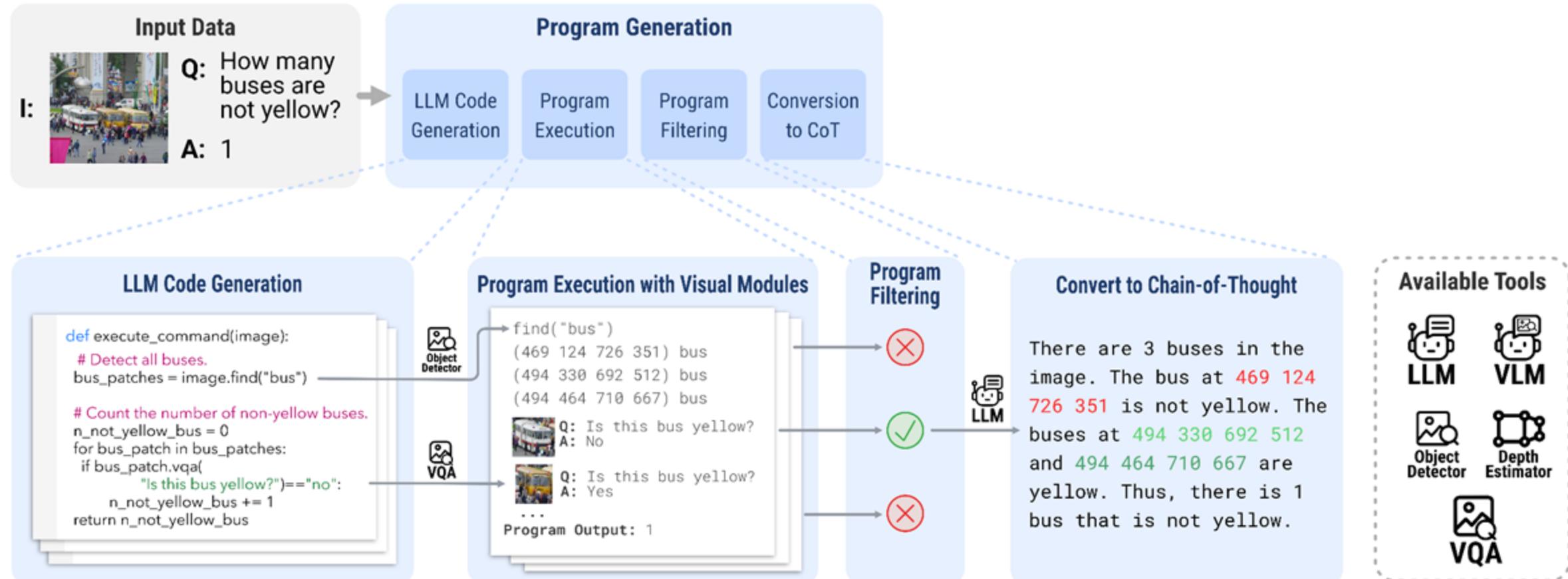
# Visual Program Distillation (VPD): Training phase



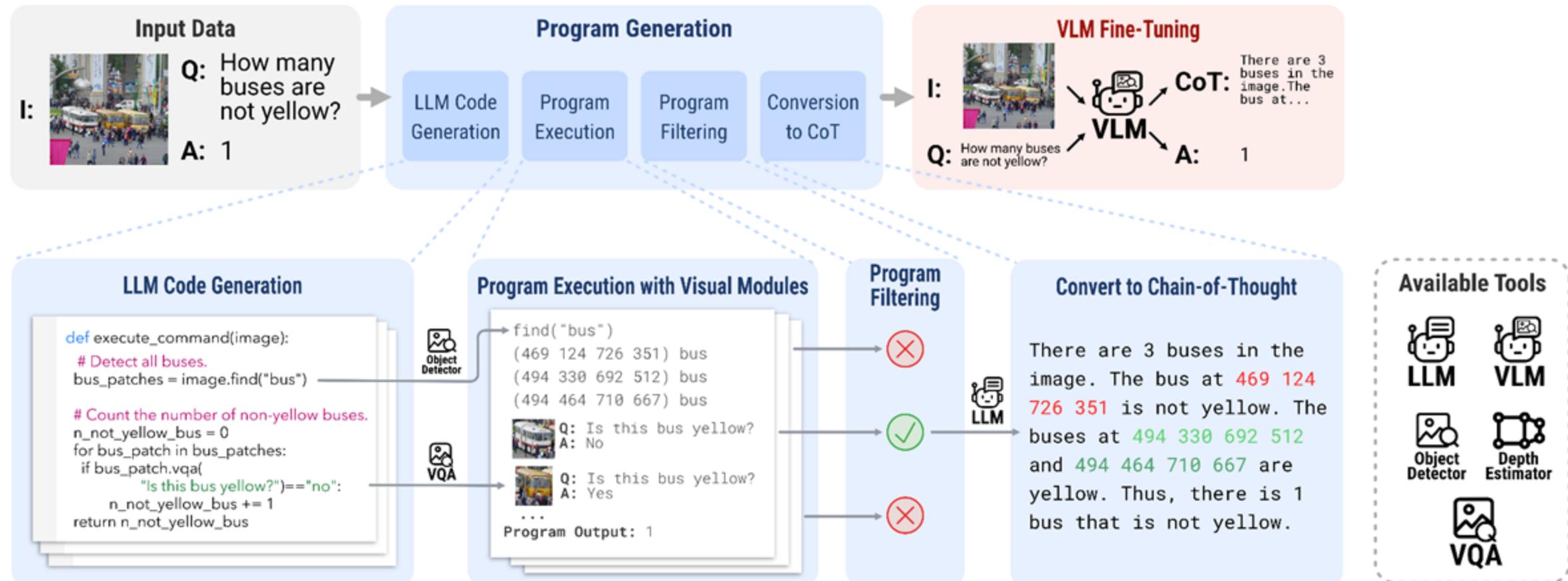
# Visual Program Distillation (VPD): Training phase



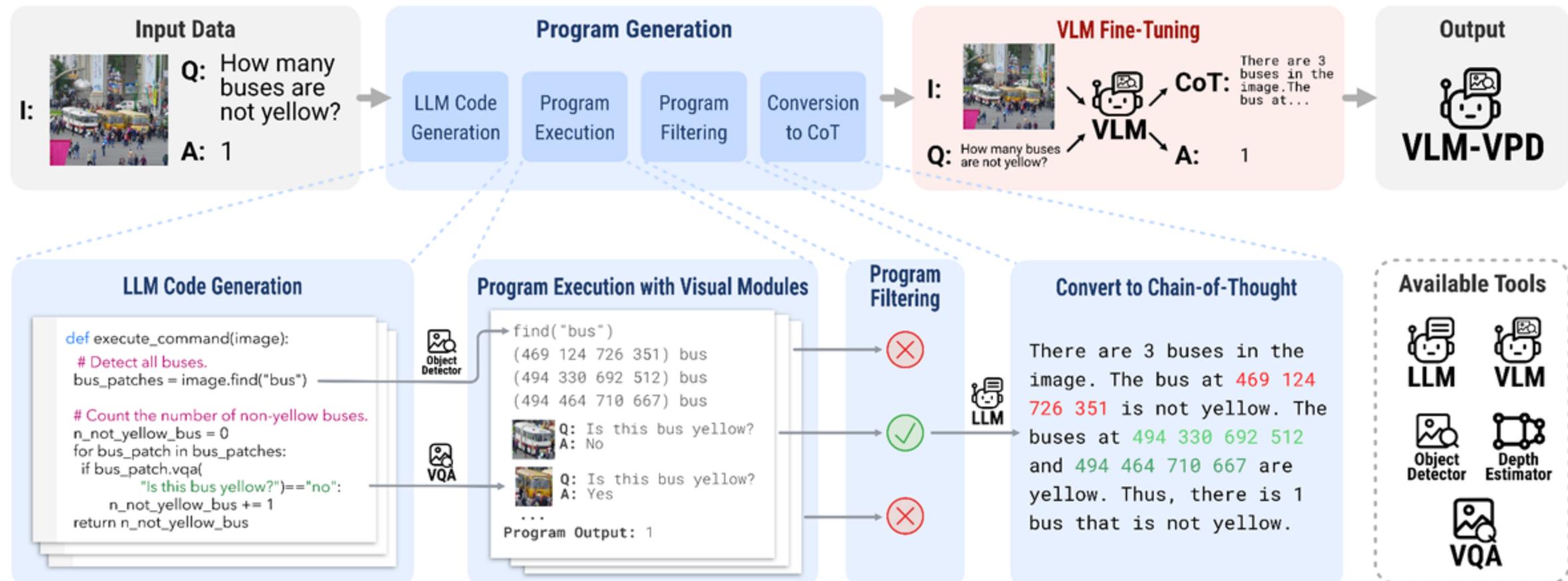
# Visual Program Distillation (VPD): Training phase



# Visual Program Distillation (VPD): Training phase



# Visual Program Distillation (VPD): Training phase



# Visual Program Distillation (VPD): Inference phase

- One forward pass!
- No code generation, tool usage, etc.
- VLMs produce interpretable reasoning steps.

*How many green vases are there?*



There are 5 green vases.



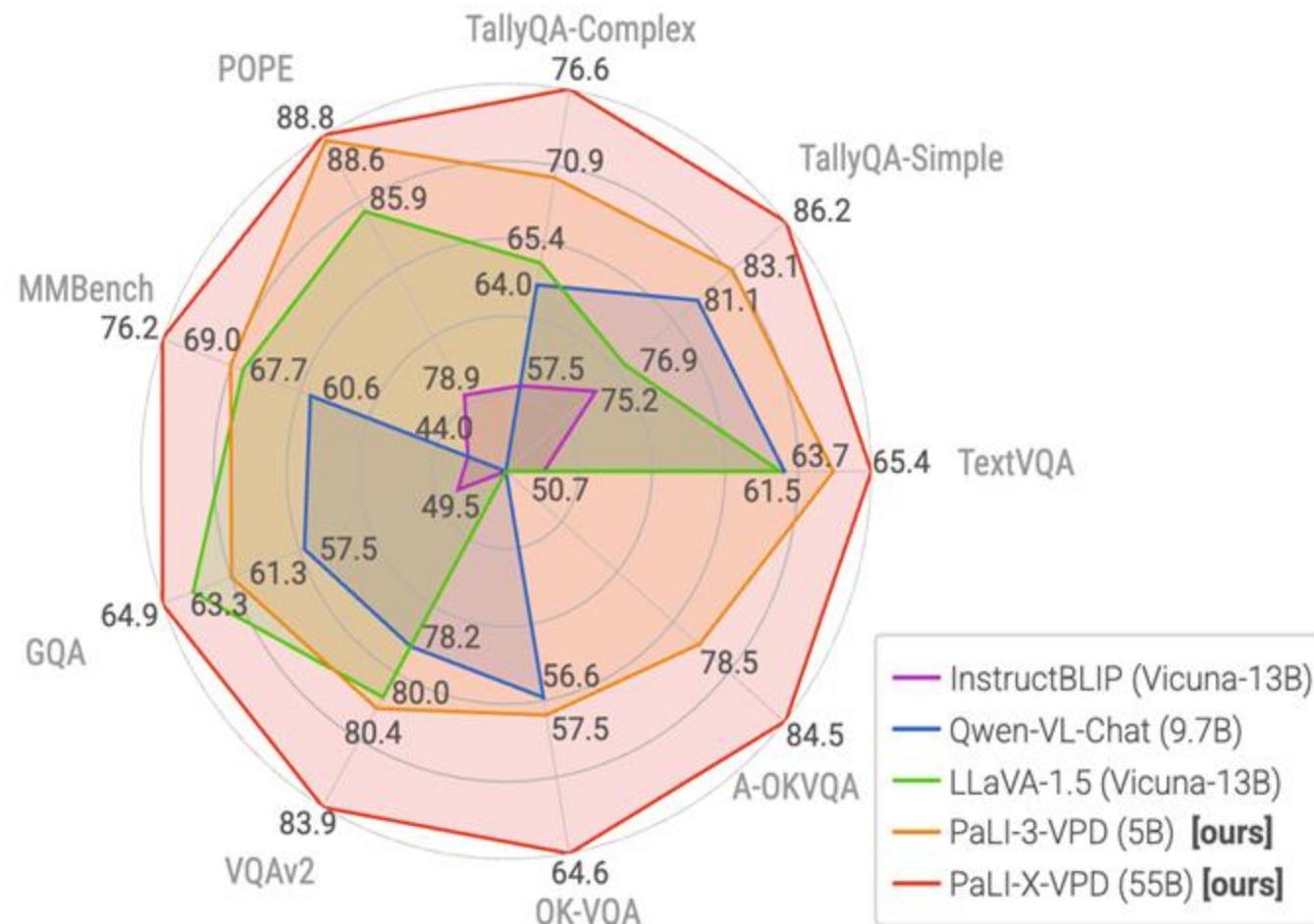
The vases at 348 139 771 371, 207 409 410 549, and 286 613 659 792 are green. Thus, there are 3 green vases.



## Key results

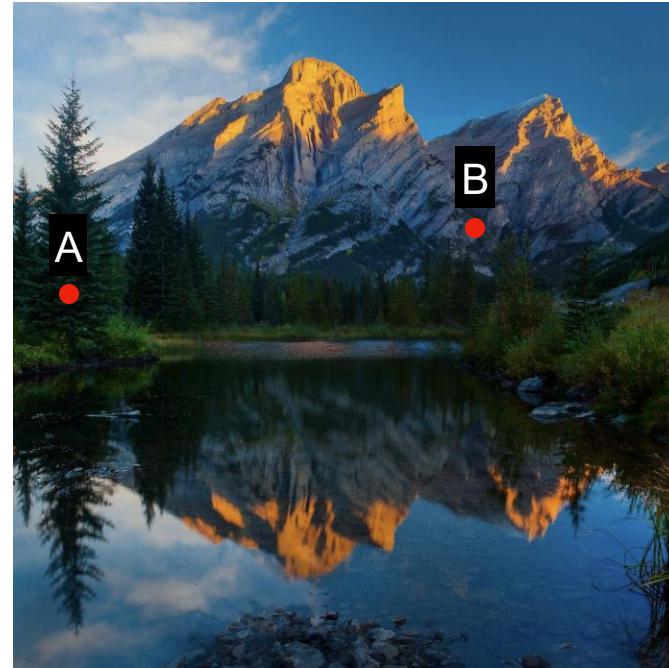
- VPD improves VLM's **accuracy, consistency, and factuality**.
- During inference, VPD model produces interpretable and **faithful visual reasoning in one forward pass**.
- Sets a new SotA on a wide range of VQA benchmarks.

# VPD held SOTA results for most benchmarks until GPT-4o1



Let's go back to our BLINK task for a second

What chain-of-thought  
would help solve this task?



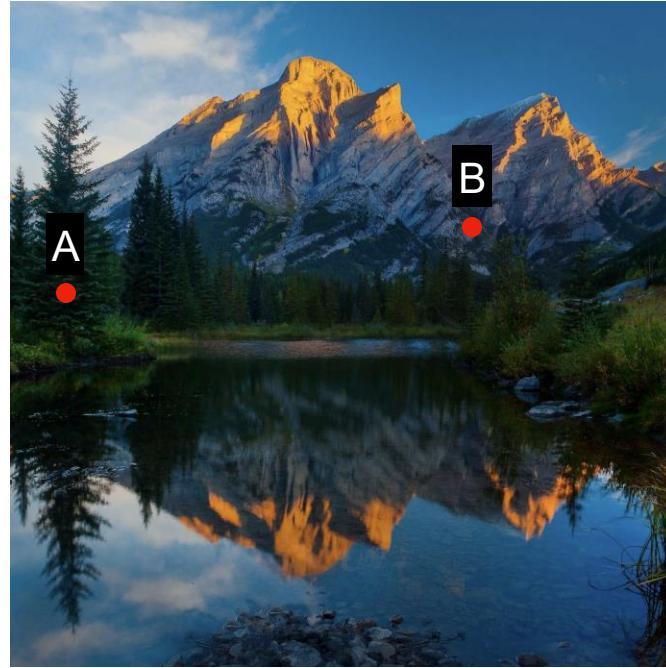
Which point is closer to the camera?

Let's go back to our BLINK task for a second

What chain-of-thought  
would help solve this task?

Depth estimation at:

- point A
- point B



Which point is closer to the camera?

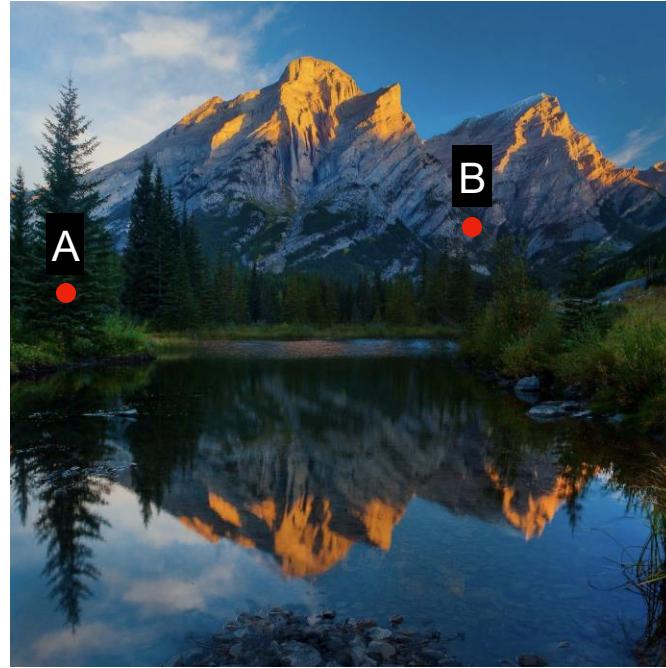
Let's go back to our BLINK task for a second

What chain-of-thought  
would help solve this task?

Depth estimation at:

- point A
- point B

but expressing depth estimations in language leads to hallucinations



Which point is closer to the camera?

# We introduce perception tokens



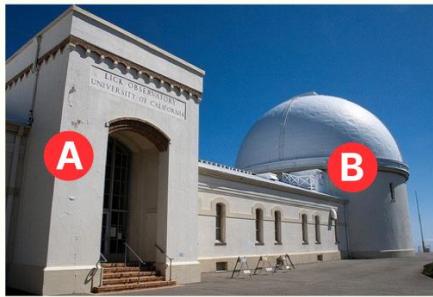
Text tokens

Perception tokens

We enable models to generate tokens that produce tokens  
that can be decoded into implicit images

# We enable models to use perceptual tokens to reason

❓❓: Which point is the closest to the camera?



With CoT



: The depth map is [1 12 ... 8 68].

Perception Tokens

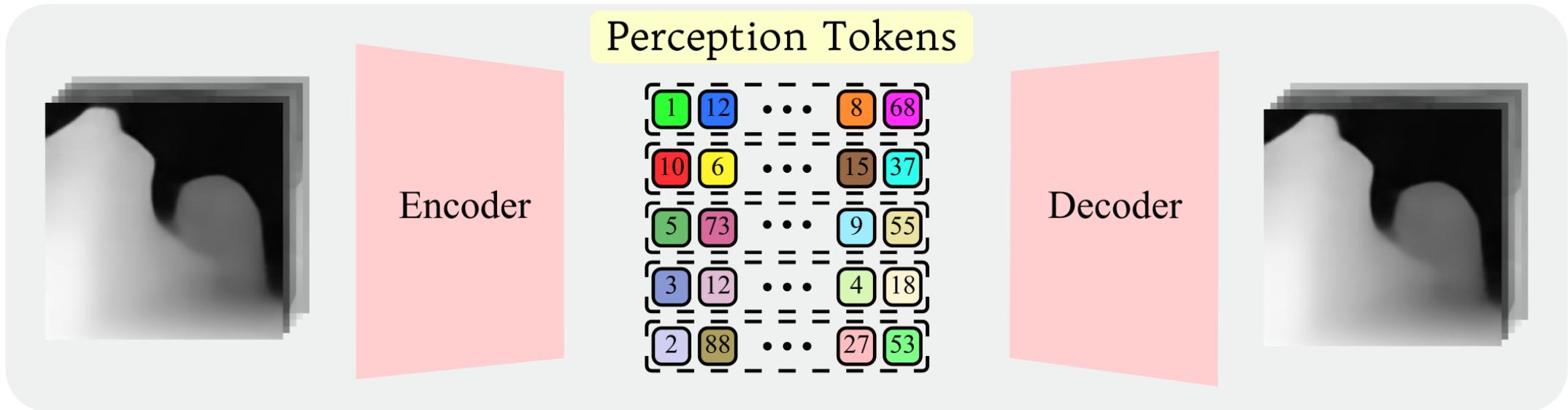
Thus, point A is closer than B.

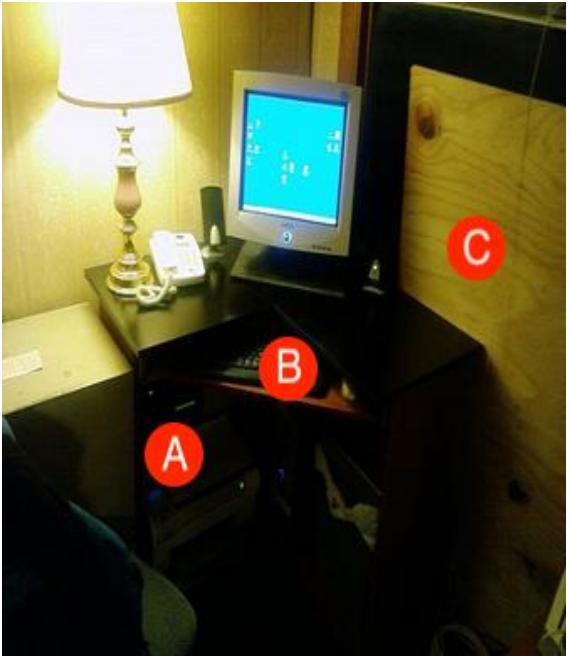
Without CoT



: Point A is closer than B.

The model doesn't produce **explicitly** depth maps  
Instead, it produces **implicit** depth maps





Which point is the closest to the camera?

Baseline

Answer: C



Ours:

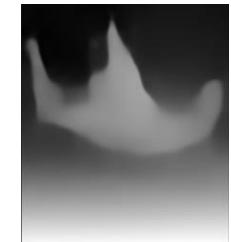
Answer: B



A: (100,230), B:(170,190), C:(280,125) and depth map:  
Point B has higher pixel value on the depth map.



Decoder



# Visual reasoning outperforms GPT-4o on BLINK

Model	Training			BLINK [9] 2 Points	HardBLINK 3 Points	HardBLINK 4 Points	HardBLINK 5 Points	Average
	Direct La- beling Data	Depth Gener- ation Data	CoT Data					
LLaVA OneVision	✗	✗	✗	51.6	33.1	22.6	18.5	31.4
LLaVA 1.5 13B	✗	✗	✗	54.0	35.5	37.9	29	39.1
Fine-tuned LLaVA	✓	✗	✗	68.5	58.9	52.4	41.1	55.2
AURORA (Ours)	✓	✓	✓	64.5	<b>66.9</b>	<b>60.5</b>	<b>54.8</b>	<b>61.6</b>
GPT-4o	✗	✗	✗	53.2	58.9	50	36.3	49.6
GPT-4 Turbo	✗	✗	✗	58.1	54.8	41.9	32.2	46.7
GPT-4 Turbo + Tool	✗	✗	✗	70.2	57.2	44.3	26.6	49.6



How many people  
are in the iamge?

Ours:

Bounding Box 1: (

Bounding Box 3: (

130	30	225	300
250	130	335	335

Answer: 4 

Bounding Box 2: (

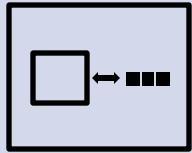
Bounding Box 4: (

240	115	300	250
250	150	335	230



# Our training algorithm

- Auto-encode perception tokens
- Curate data such that chain of thought prompting produces perception tokens
- Multi tasking data (with and without CoT)
- Curriculum learning



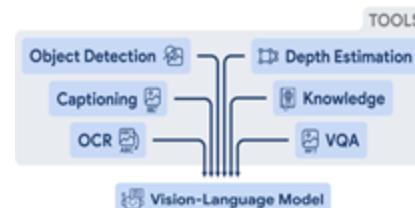
## Prioritizing perception

Perceptual tests for VLMs  
[ECCV 2024]

Sketching for perceptual reasoning  
[NeurIPS 2024] [CVPR 2025]

Distilling perceptual capabilities  
[ACL 2023] [CVPR 2024] [CVPR 2025]

Enabling robots to sketch  
[ArXiv 2025]

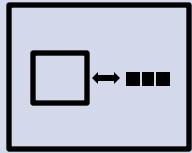


Most fundamental vision capabilities are still out of reach

Enabling sketching:  
visual chain of thought

How can we distill from specialist models into generalist VLMs?

Complete open Action Reasoning model for robotics



## Prioritizing perception

Perceptual tests for VLMs  
[ECCV 2024]

Sketching for perceptual reasoning  
[NeurIPS 2024] [CVPR 2025]

Distilling perceptual capabilities  
[ACL 2023] [CVPR 2024] [CVPR 2025]

Enabling robots to sketch  
[ArXiv 2025]



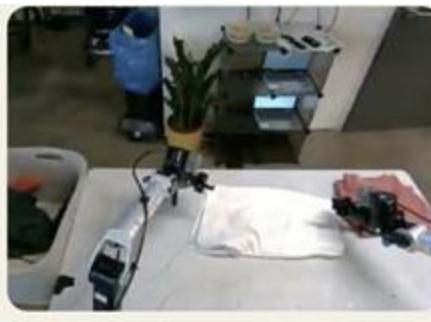
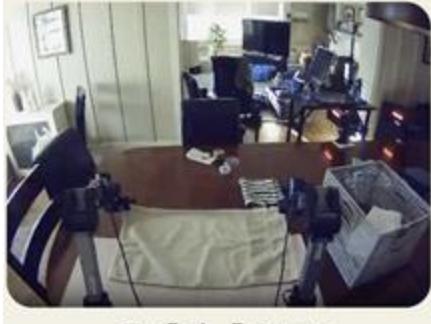
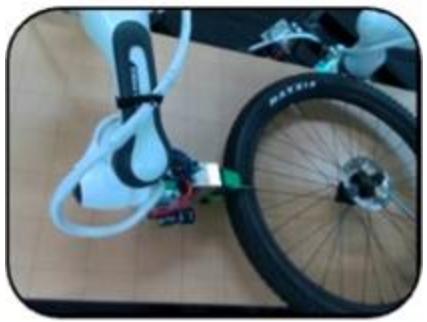
Most fundamental vision capabilities are still out of reach

Enabling sketching: visual chain of thought

How can we distill from specialist models into generalist VLMs?

Complete open Action Reasoning model for robotics

# We want the models to work in our homes

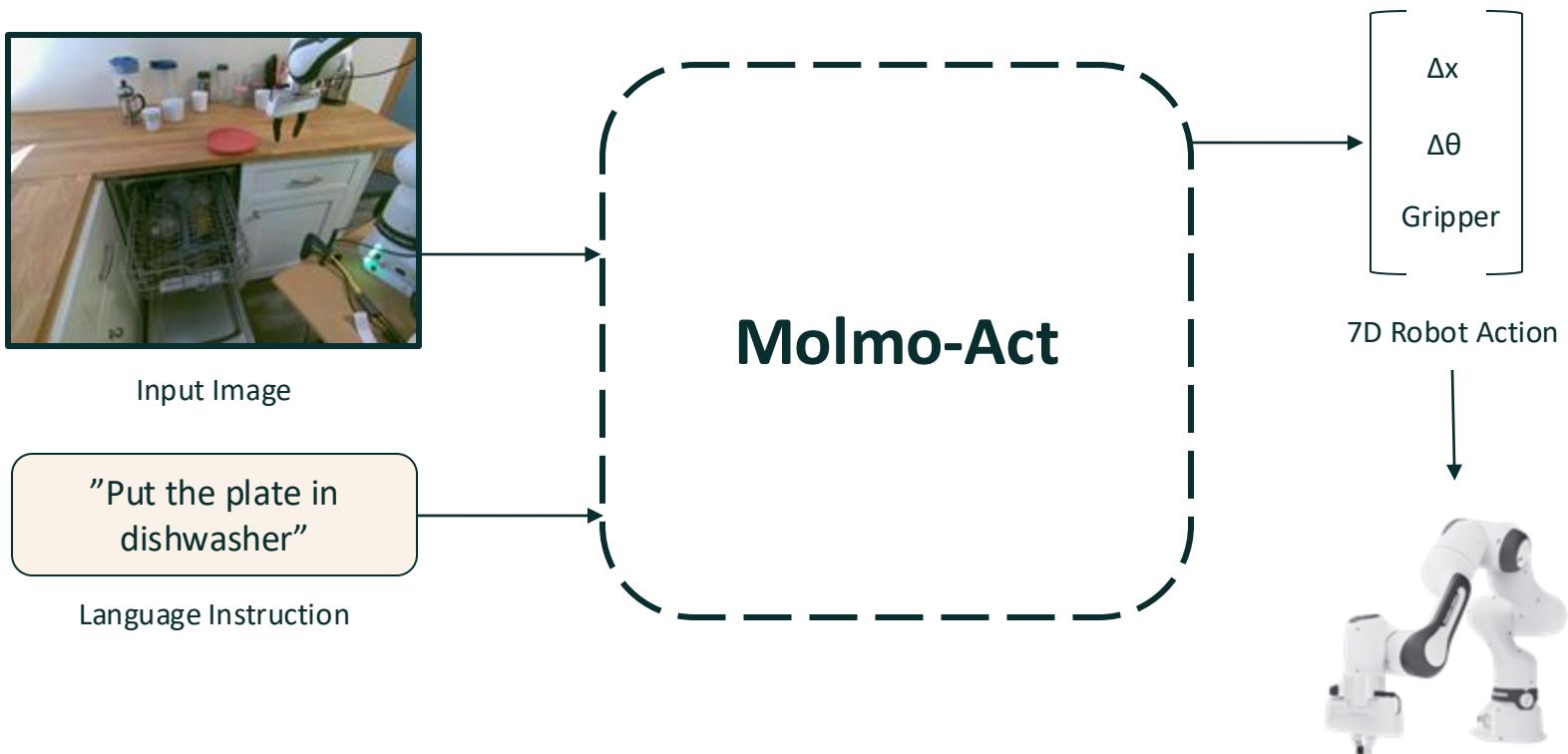




A Family of Completely Open  
Vision-Language-Action Model



# How does Molmo-Act work?



Many manipulation tasks are long horizon

### 3. Clean after wild party



# Language models usually use intermediate reasoning to solve long horizon tasks

## Standard Prompting

### Model Input

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

A: The answer is 11.

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

### Model Output

A: The answer is 27. 

## Chain-of-Thought Prompting

### Model Input

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

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

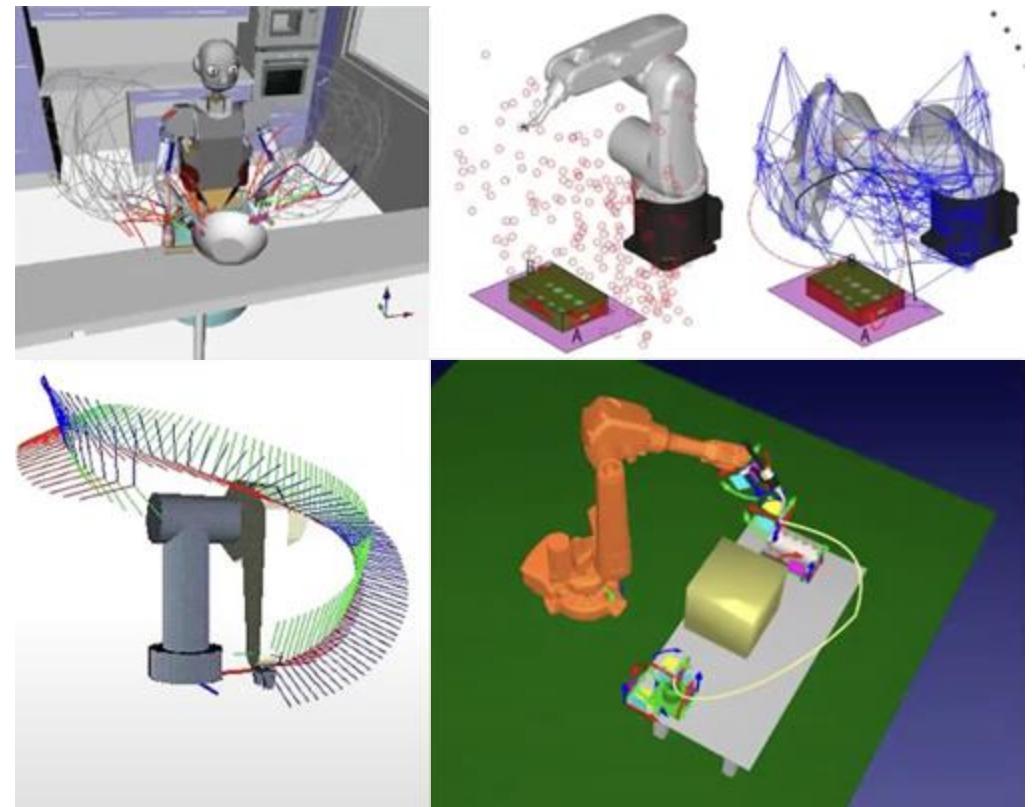
Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

### Model Output

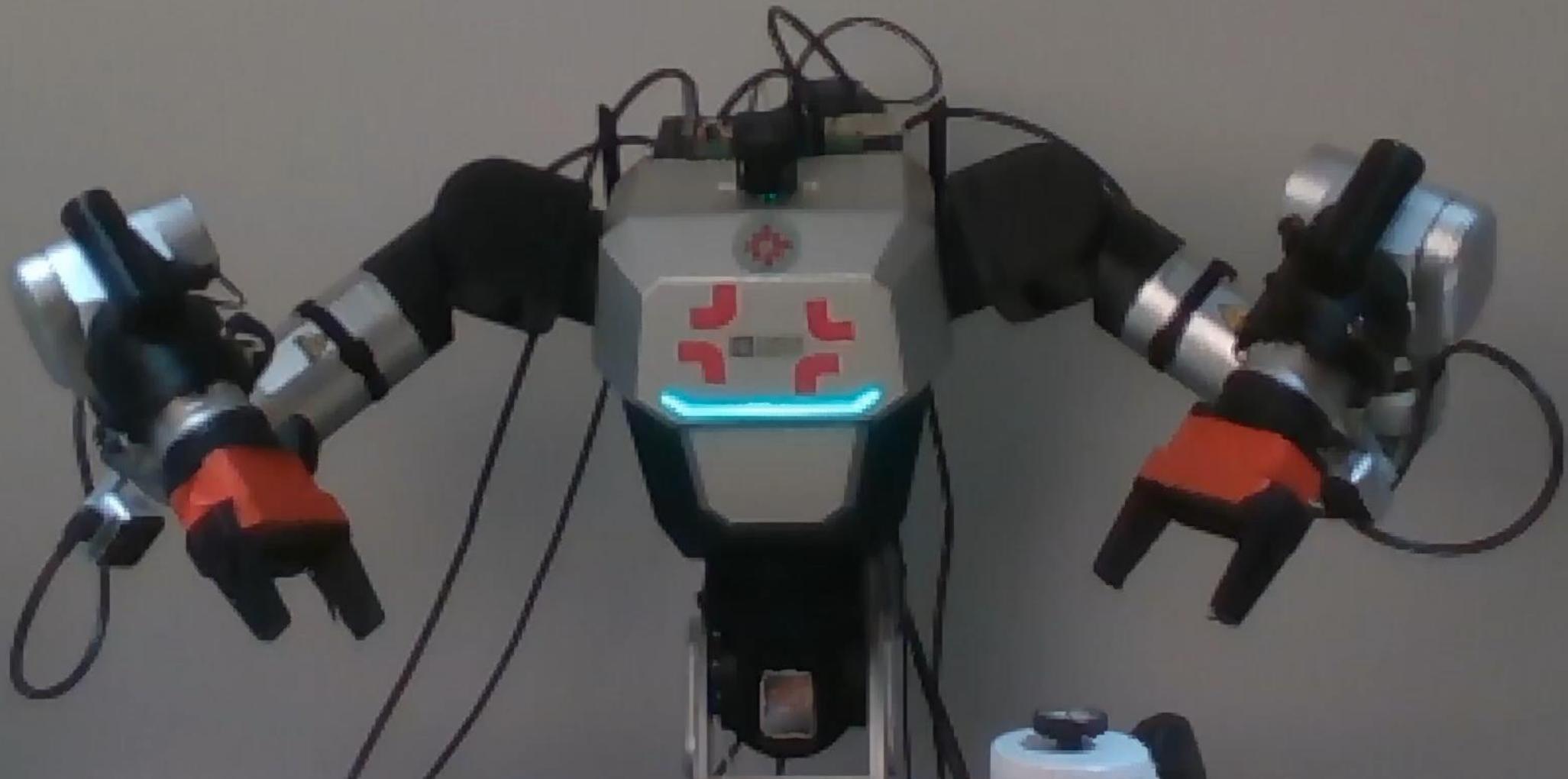
A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had  $23 - 20 = 3$ . They bought 6 more apples, so they have  $3 + 6 = 9$ . The answer is 9. 

# But language reasoning doesn't allow models to reason in space

Standard Prompting	Chain-of-Thought Prompting
<p>Model Input</p> <p>Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?</p> <p>A: The answer is 11.</p> <p>Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?</p>	<p>Model Input</p> <p>Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?</p> <p>A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. <math>5 + 6 = 11</math>. The answer is 11.</p> <p>Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?</p>
<p>Model Output</p> <p>A: The answer is 27. ❌</p>	<p>Model Output</p> <p>A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had <math>23 - 20 = 3</math>. They bought 6 more apples, so they have <math>3 + 6 = 9</math>. The answer is 9. ✓</p>



Need to reason  
about space



Molmo-Acts reasons in **space** – it sketches a plan in 2.5D



“Move pepsi can near Red bull”

# MolmoAct Reasoning

 MolmoAct



Depth Perception Token



Visual Reasoning Trace

$[[202,15], [153,178], [130,145], [52,145]]$



Robot Actions

$[\Delta x, \Delta \theta, \text{Gripper}]$



 Ai2



Q: To figure out the action that the robot should take to **put plate into dishwasher**, let's think through it **step by step**.

First, what is the **depth map** for this image?

Second, what is the **trajectory** of the end effector?

Based on these, what is the **action** that the robot should take?



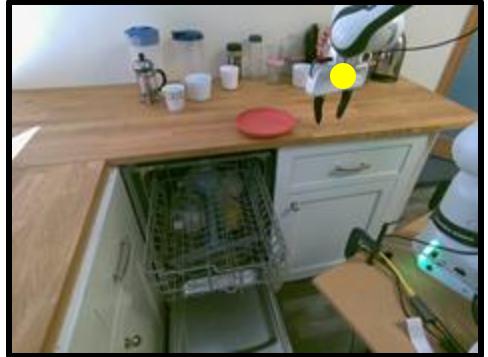
# How do we curate ground truth trajectory



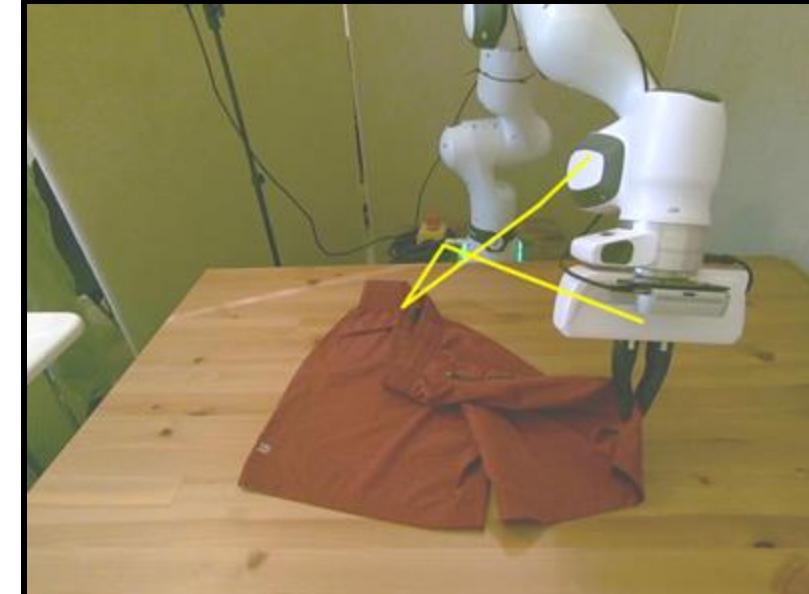
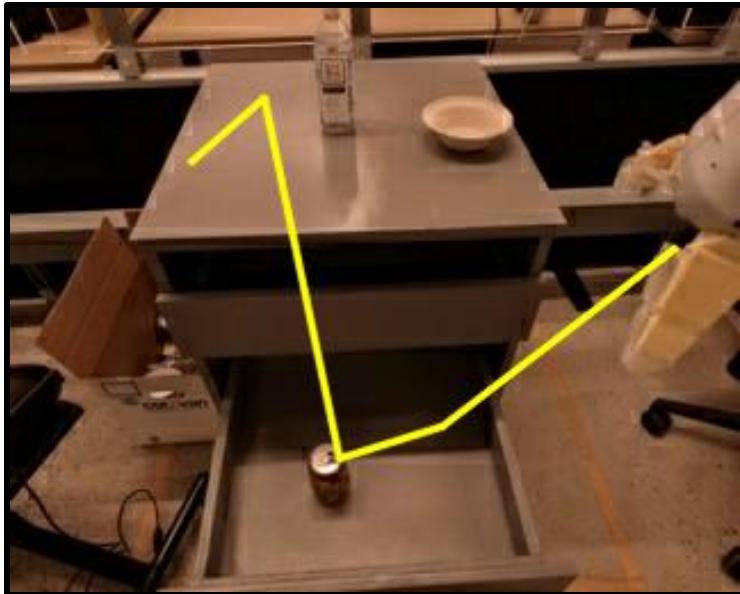
Molmo

“Point to the gripper”

# How do we curate ground truth trajectory



# How do we curate ground truth trajectory



8M Image, Trajectory pairs

~170k Robot Trajectories

# Reasoning in 2D is not enough!

So, we also reason using **depth**!



“Put the plate in  
dishwasher”

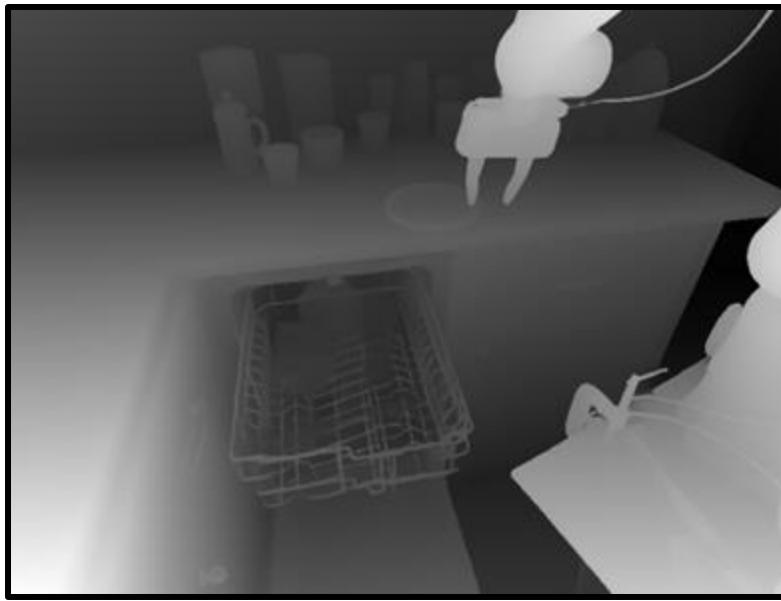
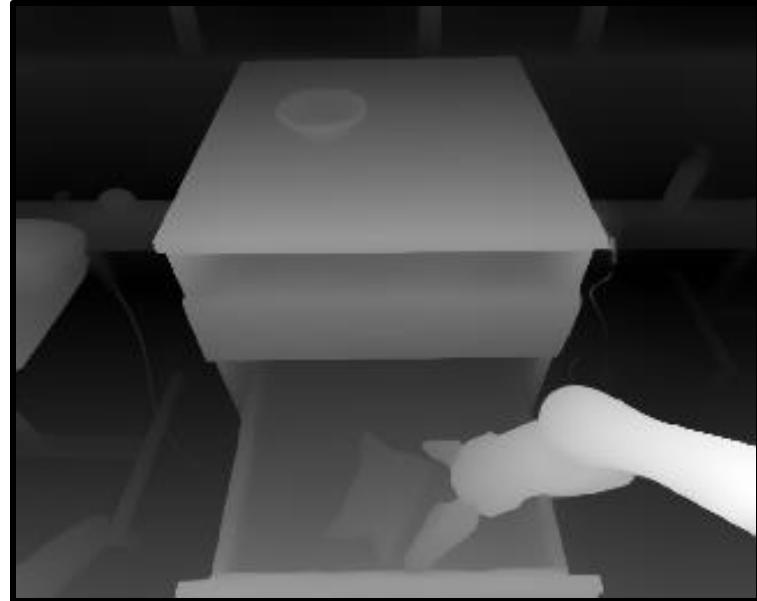


Molmo-ACT

The trajectory of the end effector is  
[234, 68], [245, 71]...

The action that the robot should take is  
 $[\Delta x, \Delta \theta, \text{Gripper}]$ .

# We automatically generate depth data

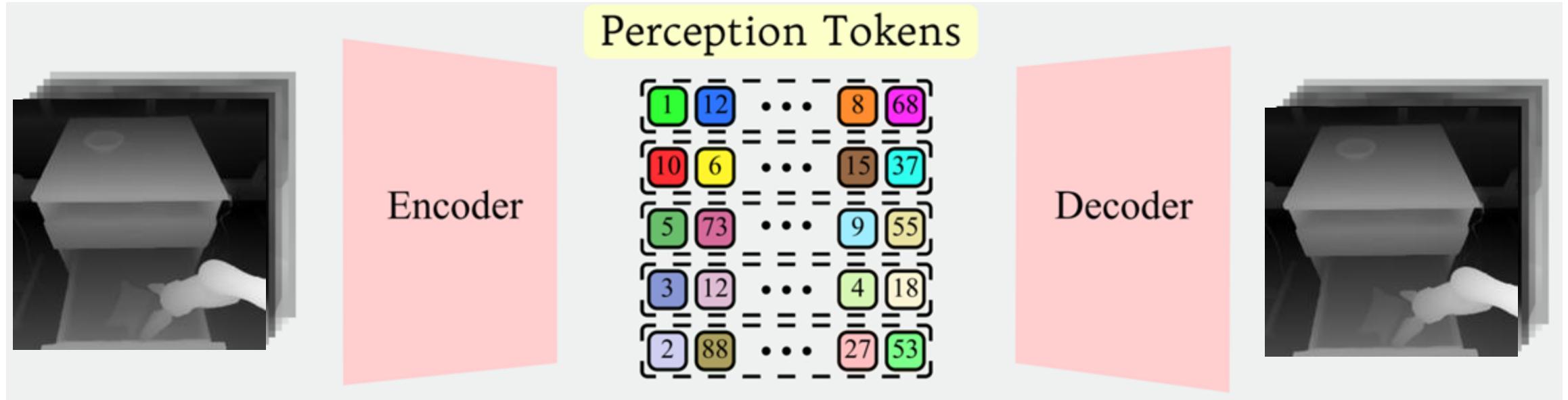


8M Image, Depth pairs

~170k Robot Trajectories

# Reasoning in depth requires new innovations

-> Perception Tokens



# Grounding Action with Depth



“Put the plate in  
the dishwasher”



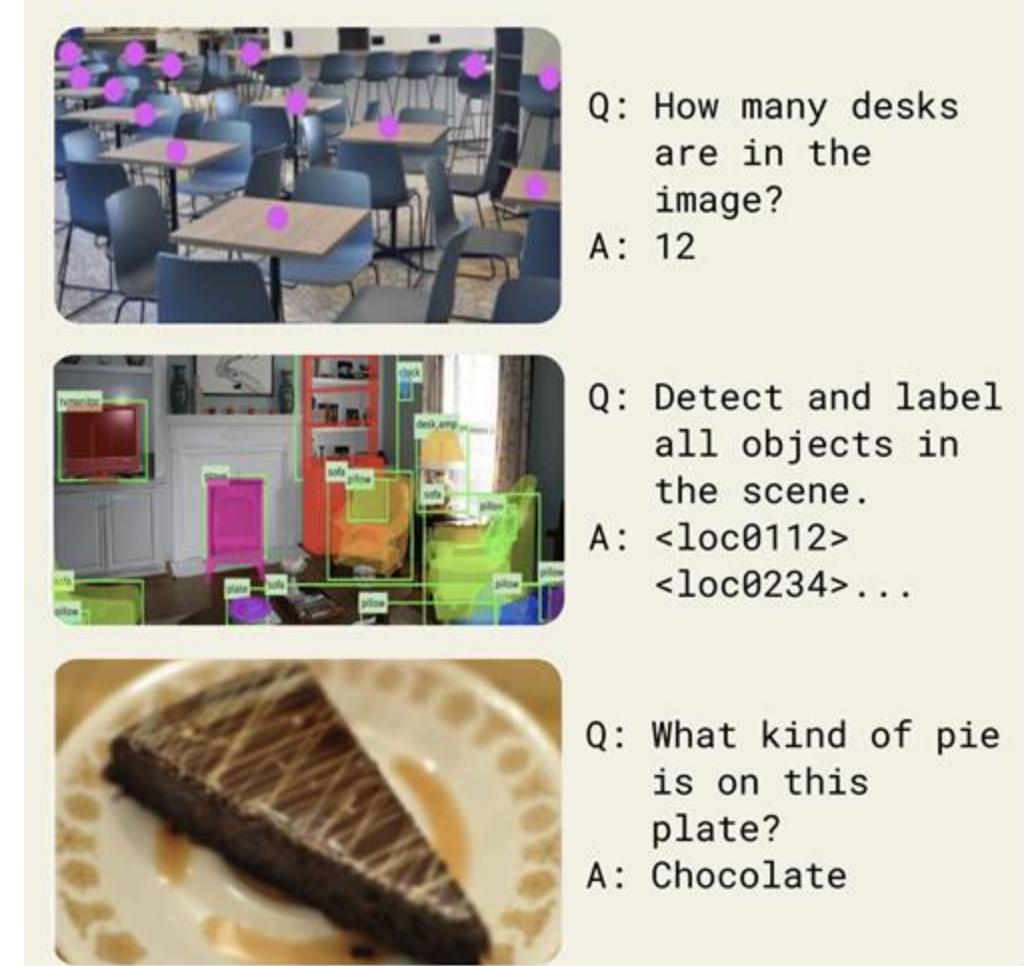
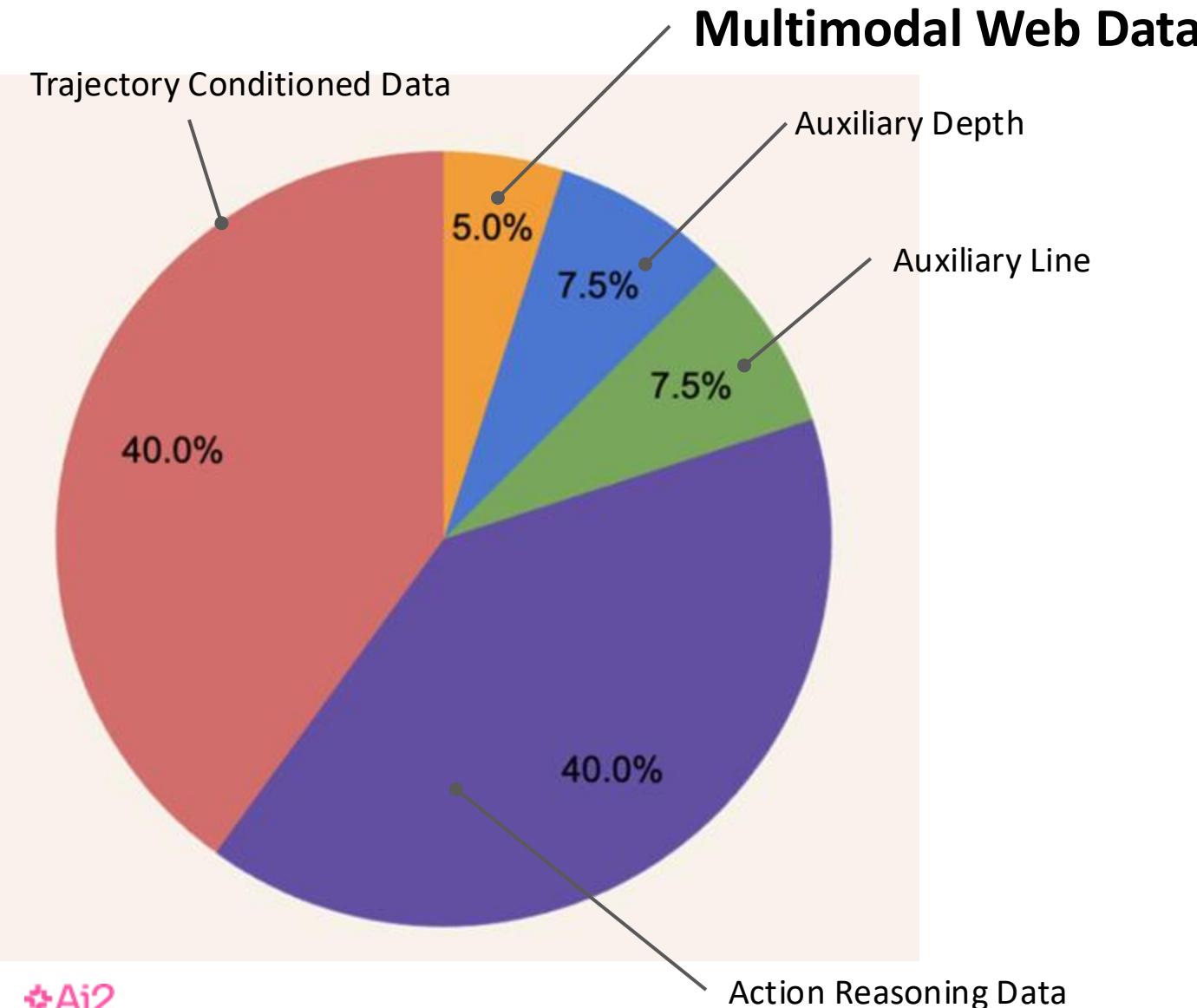
Molmo-ACT

The depth map is

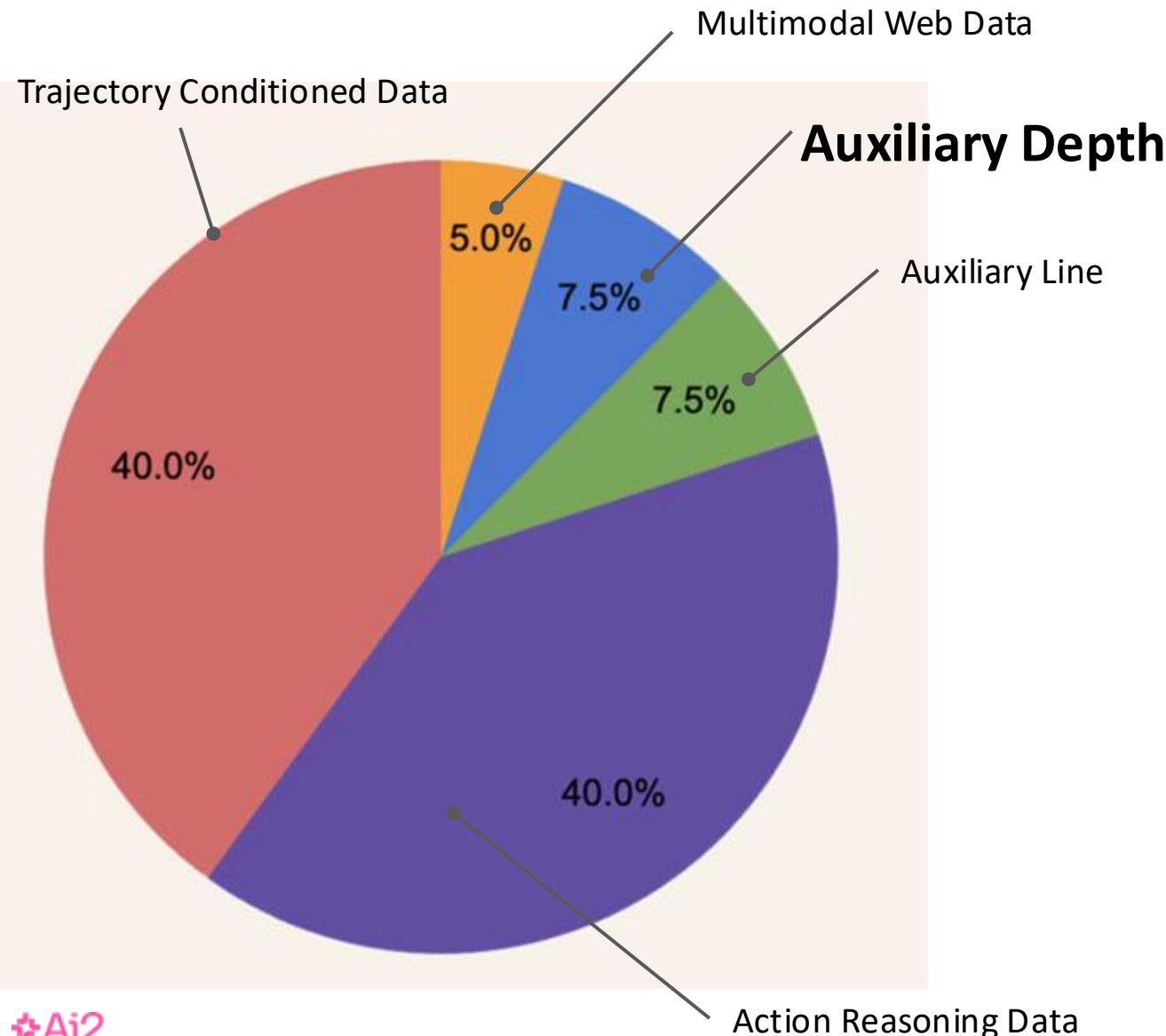
[1 12 ... 8 68]

The action that the robot  
should take is  $[\Delta x, \Delta \theta, \text{Gripper}]$ .

# Our pretraining datamix has general multimodal



# Our pretraining datamix has produce depth estimation

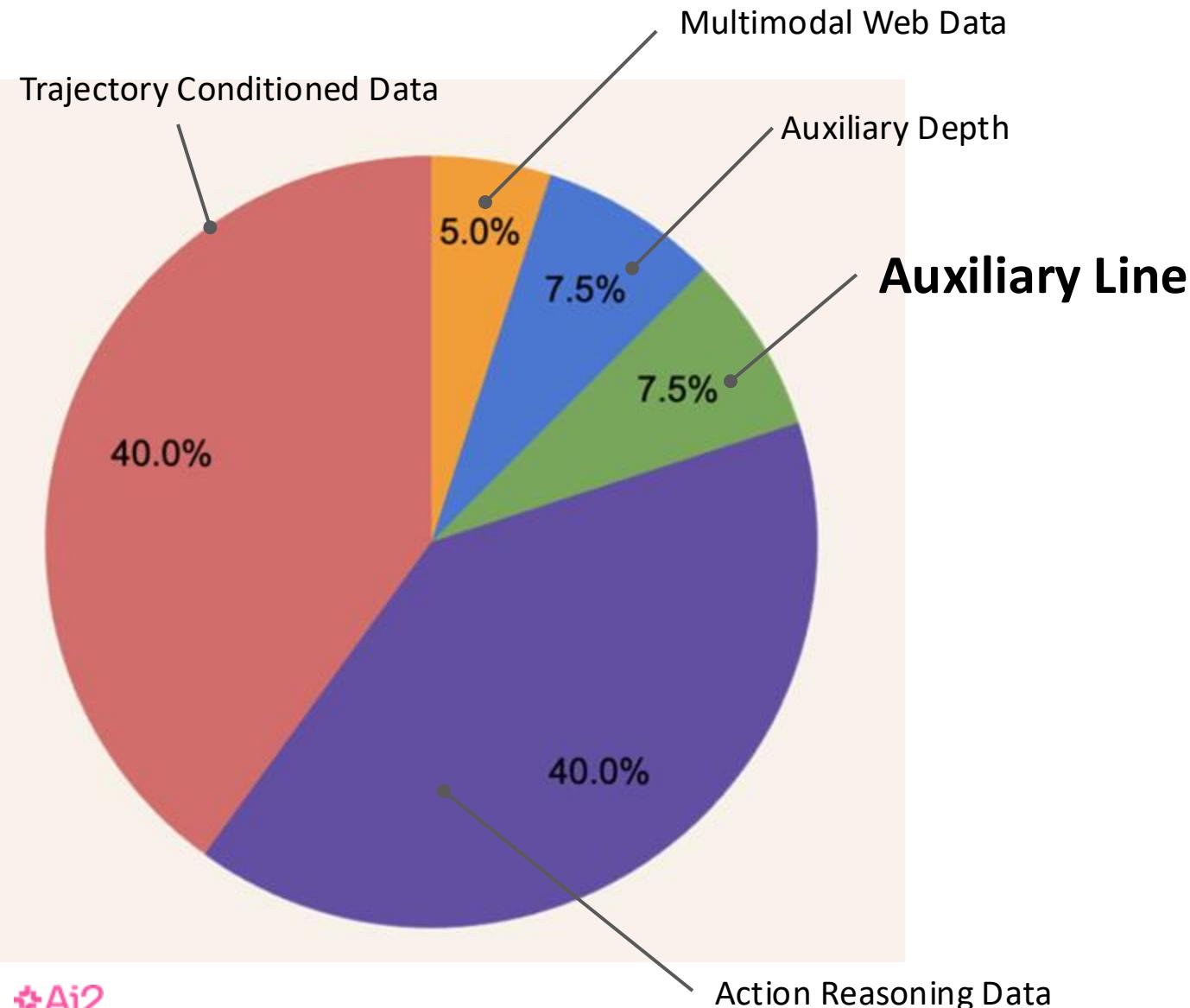


Q: The task is put the plate in the dishwasher, what is the depth map?



A: The depth map is [1 12 ... 8 68]

# Our pretraining datamix has grounded reasoning



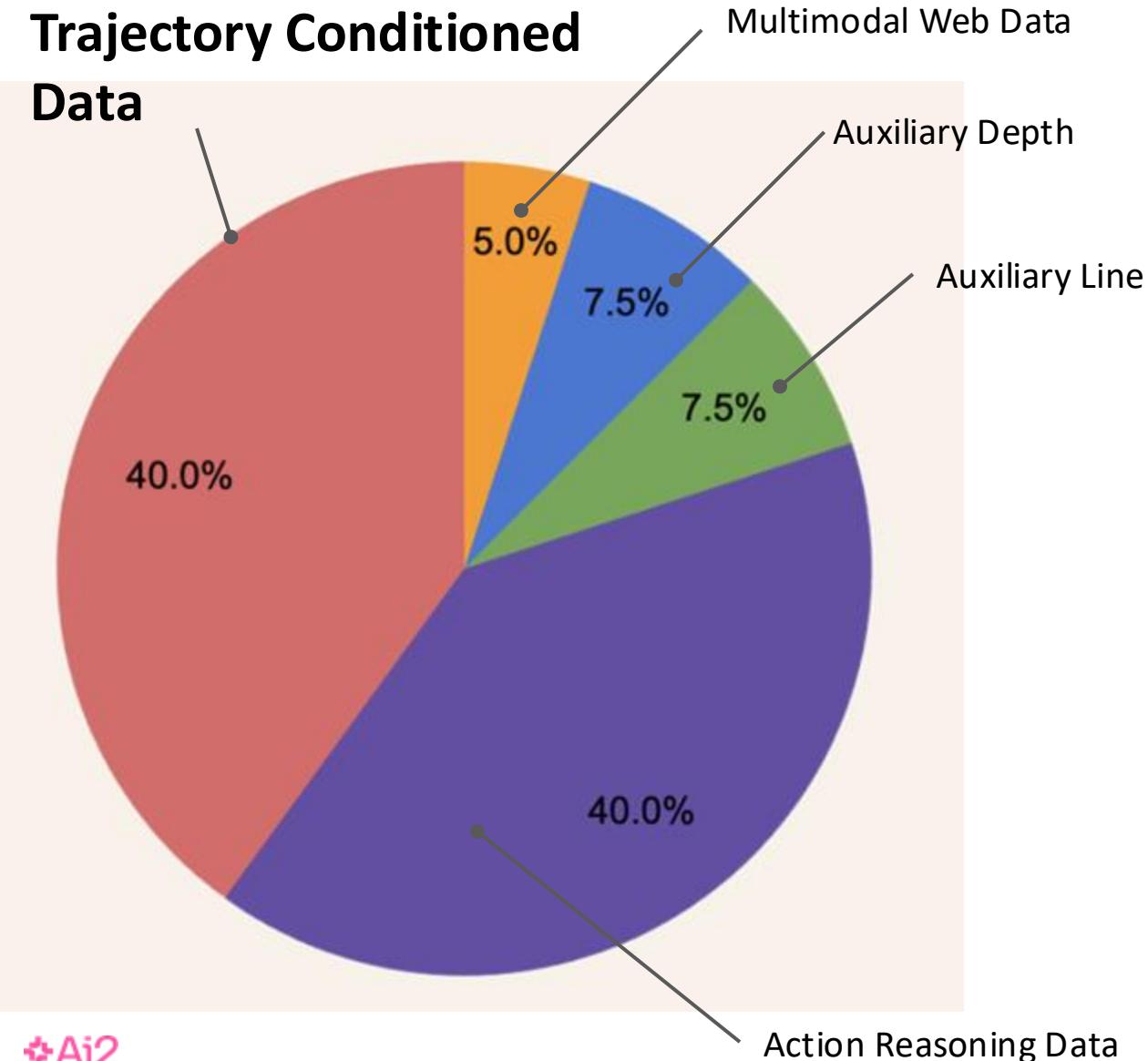
Q: The task is put the plate in the dishwasher, what is the trajectory of the end effector?



A: The trajectory that the robot should take is [194, 24],[203,44]...

# Our pretraining datamix has trajectory conditioned

## Trajectory Conditioned Data



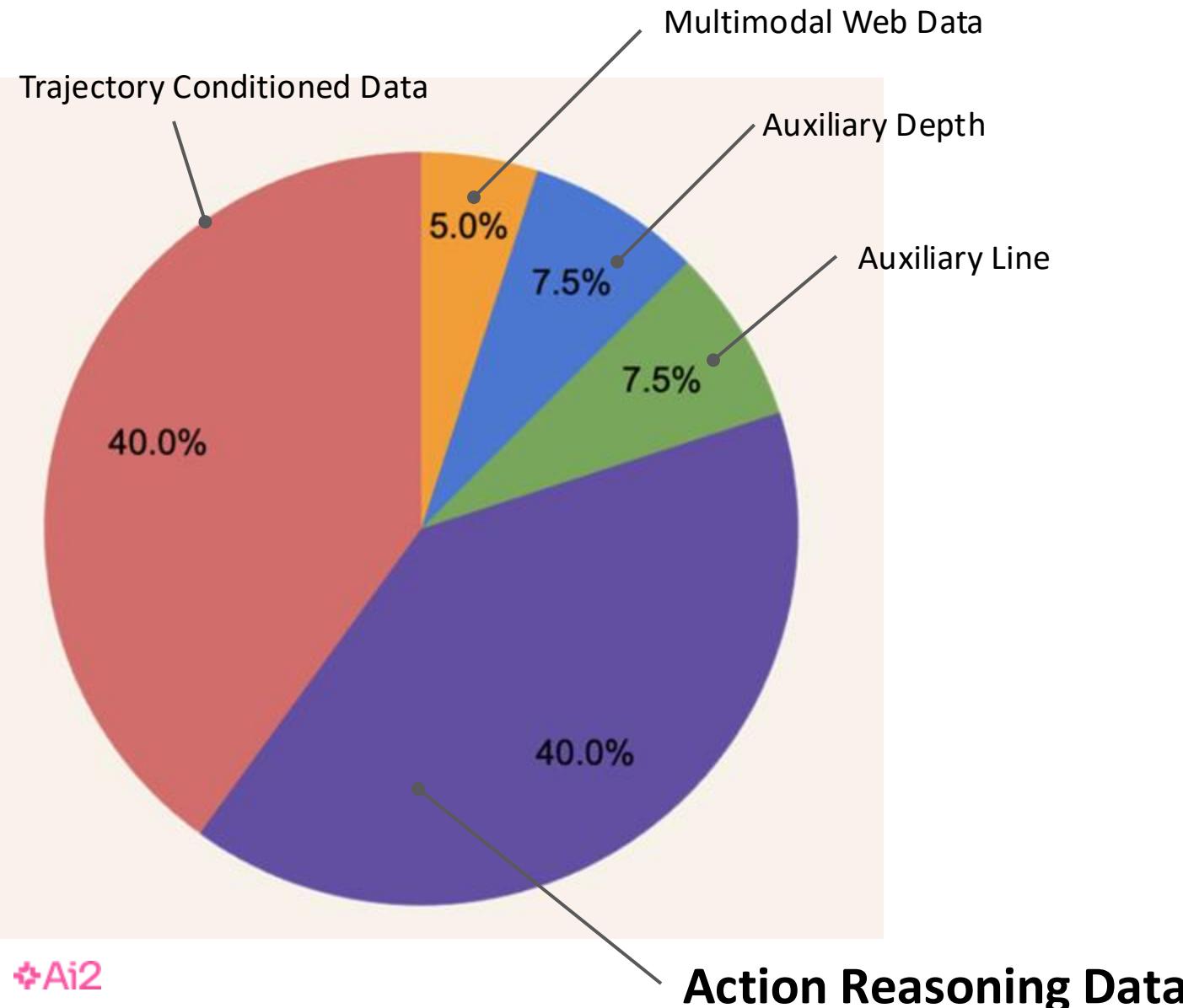
Q: The task is put the plate in the dishwasher, what action should the robot take based on the traj?



A:

The action that the robot should take is  $[\Delta x, \Delta \theta, \text{Gripper}]$ .

# Our pretraining datamix has large robot behavior



Q: The task is put the plate in the dishwasher, what action should the robot take?



A: The depth map is   
The trajectory that the robot should take is  $[194, 24], [203, 44] \dots$   
The action that the robot should take is  $[\Delta x, \Delta \theta, \text{Gripper}]$ .

# MolmoAct dataset contains different household environments



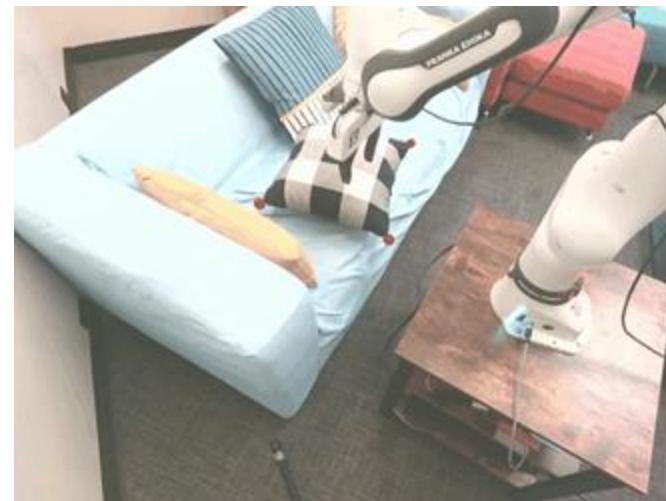
Kitchen



Bedroom



Bathroom



Living Room

# Breaking subtask into atomic motions



Load the dishwasher

- Open the dishwasher
- Pick up the dishes/utensils
- Flip the plates/bowls upright
- Place it in the dishwasher
- Close the tray and the dishwasher

Place the plate on the stand



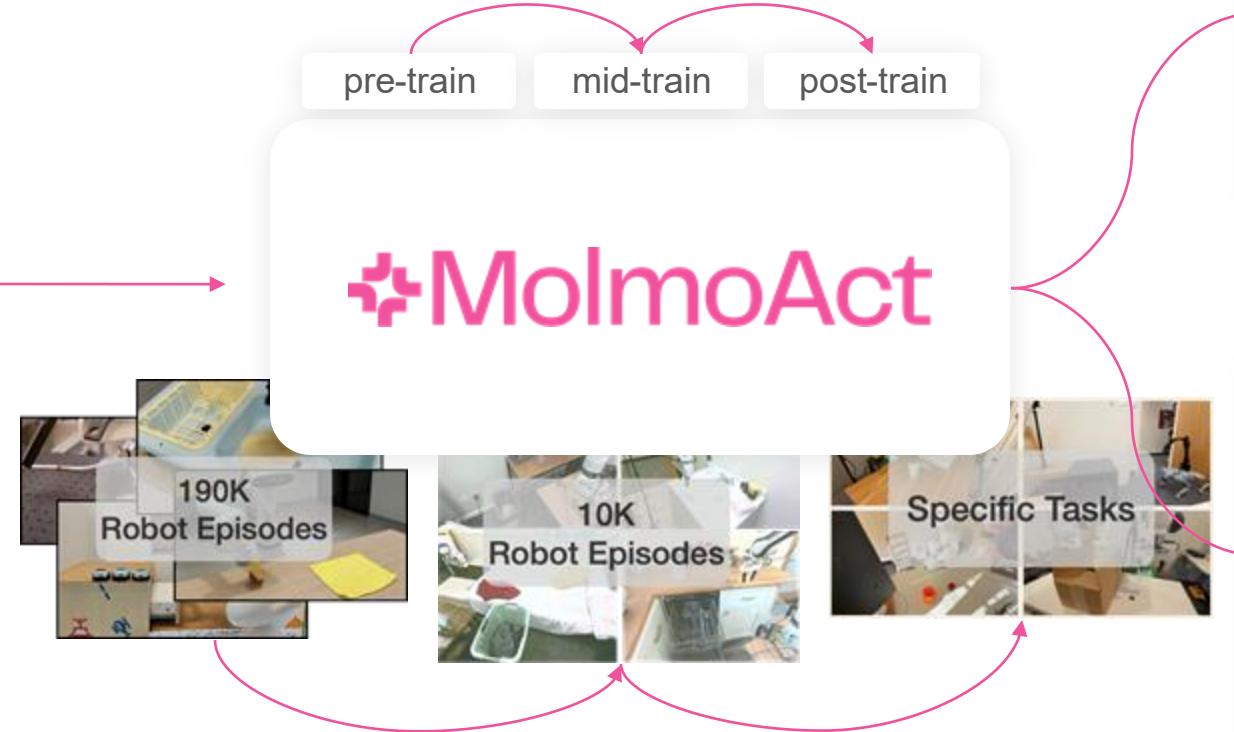
Flip the mug



Close the drawer



# MolmoAct Overview

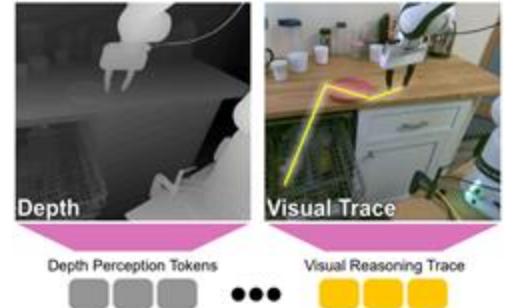


Molmo Initiation

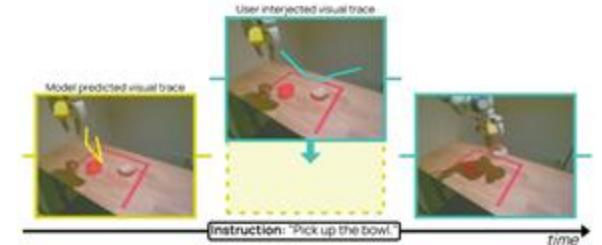
MolmoAct Training

MolmoAct Inference

Reasoning in



Steerability



# Pre-training Evaluations (SimplerEnv)

Visual Matching (In-distribution)



move pepsi can near blue plastic bottle (real)



move pepsi can near blue plastic bottle (sim)

Variant Aggregation (Out-of-distribution)

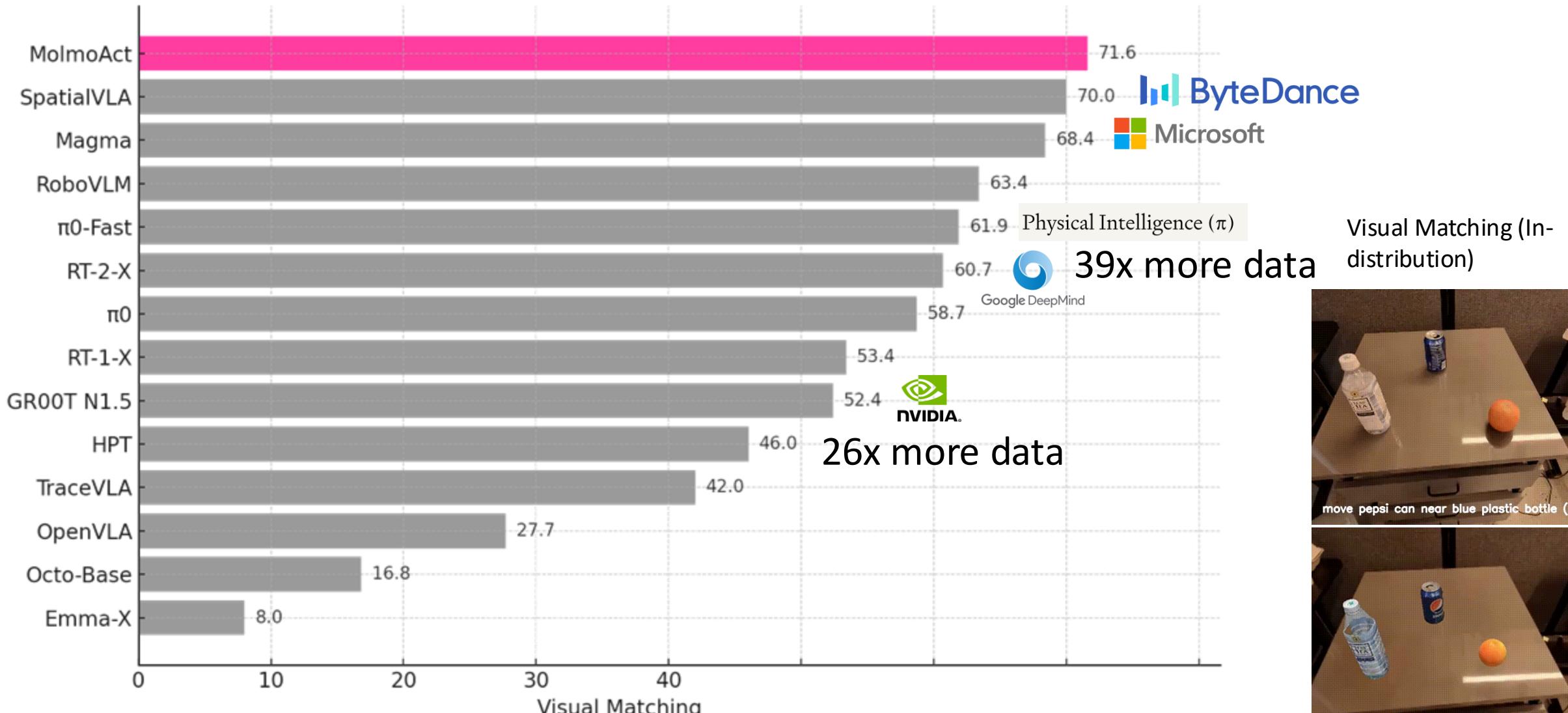


open middle drawer (real)



open middle drawer (sim)

# Pre-training evaluation with MolmoAct



# Post-training Evaluations (LIBERO)

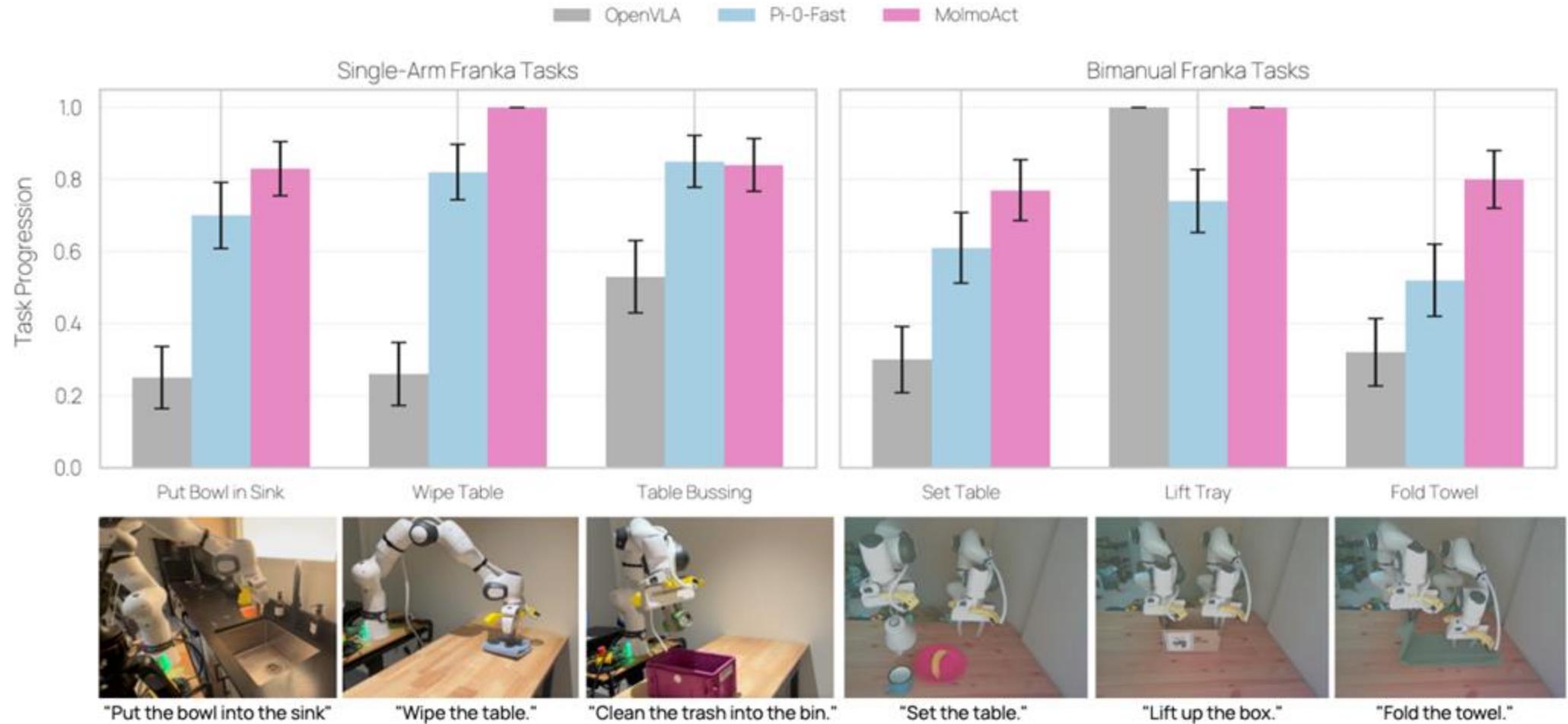


# SOTA in Post-training Evaluations (LIBERO)\*

\*compared to other autoregressive models

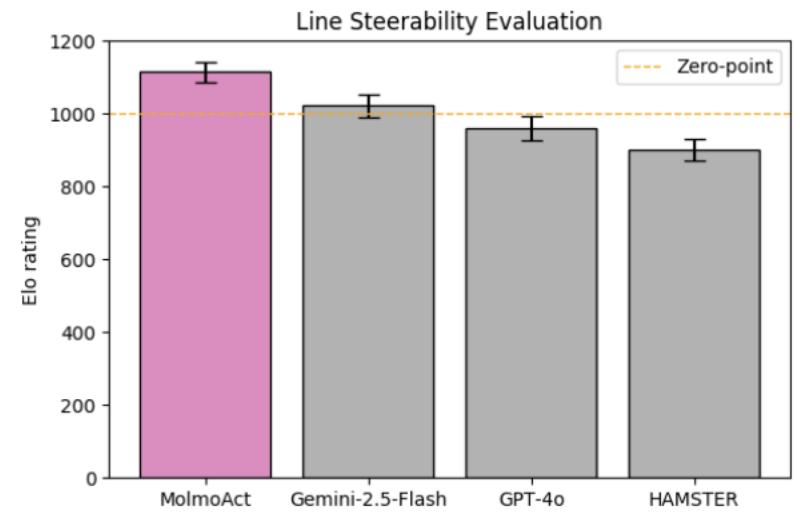
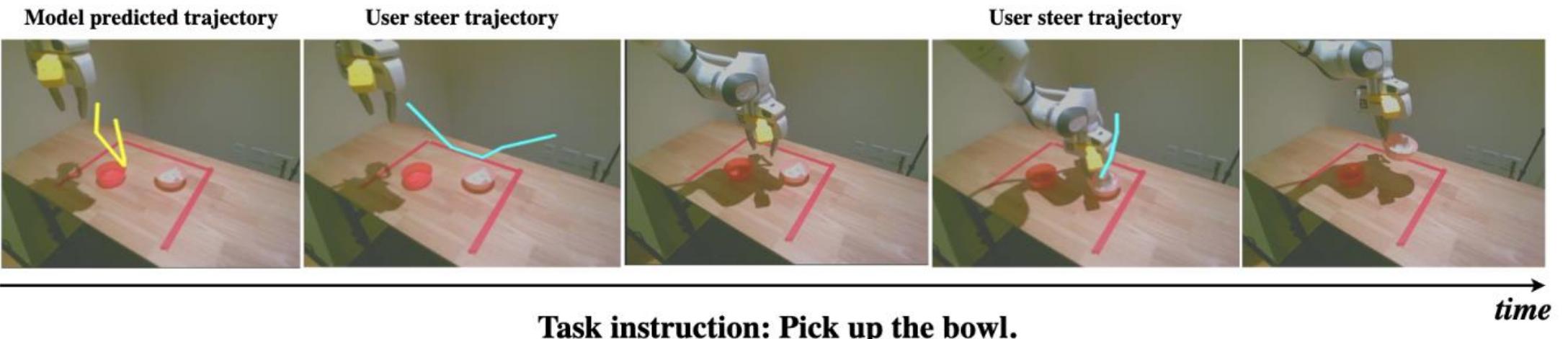
Baseline	Spatial	Object	Goal	Long	Avg
TraceVLA (Zheng et al., 2024)	84.6%	85.2%	75.1%	54.1%	74.8%
Octo-Base (Team et al., 2024b)	78.9%	85.7%	84.6%	51.1%	75.1%
OpenVLA (Kim et al., 2024)	84.7%	88.4%	79.2%	53.7%	76.5%
SpatialVLA (Qu et al., 2025)	88.2%	89.9%	78.6%	55.5%	78.1%
CoT-VLA (Zhao et al., 2025)	87.5%	91.6%	87.6%	69.0%	83.9%
NORA-AC (Hung et al., 2025)	85.6%	89.4%	80.0%	63.0%	79.5%
WorldVLA (Cen et al., 2025)	87.6%	96.2%	83.4%	60.0%	79.1%
$\pi_0$ -FAST (Black et al.)	96.4%	96.8%	88.6%	60.2%	85.5%
ThinkAct (Huang et al., 2025)	88.3%	91.4%	87.1%	70.9%	84.4%
MOLMOACT-7B-D	87.0%	95.4%	87.6%	77.2%	<b>86.6%</b>

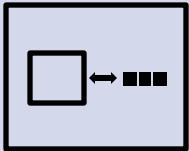
# Post-training Evaluations (Real-world)



MolmoAct outperforms  $\pi$ 0-FAST by an average of 10% in task progression on single-arm tasks and by 22.7% on bimanual tasks.

Steerability -> allow users to interpret and guide robot behavior





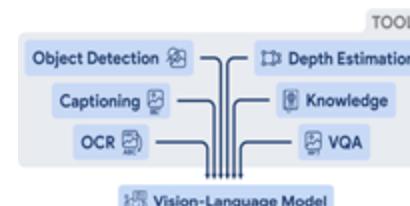
## Prioritizing perception

Perceptual tests for VLMs  
[ECCV 2024]

Sketching for perceptual reasoning  
[NeurIPS 2024] [CVPR 2025]

Distilling perceptual capabilities  
[ACL 2023] [CVPR 2024] [CVPR 2025]

Enabling robots to sketch  
[ArXiv 2025]



Most fundamental vision capabilities are still out of reach

Enabling sketching:  
visual chain of thought

How can we distill from specialist models into generalist VLMs?

Complete open Action Reasoning model for robotics

# Thank you

Ranjay Krishna

## **BLINK: Multimodal Large Language Models Can See but Not Perceive**

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, Ranjay Krishna  
ECCV 2024

## **Visual Sketchpad: Sketching as a Visual Chain of Thought for Multimodal Language Models**

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, Ranjay Krishna  
NeurIPS 2024

## **Visual Program Distillation: Distilling Tools and Programmatic Reasoning into Vision-Language Models**

Yushi Hu, Otilia Stretcu, Chun-Ta Lu, Krishnamurthy Viswanathan, Kenji Hata, Enming Luo, Ranjay Krishna, Ariel Fuxman  
CVPR 2025

## **Perception Tokens Enhance Visual Reasoning in Multimodal Language Models**

Mahtab Bigverdi, Zelun Luo, Cheng-Yu Hsieh, Ethan Shen, Dongping Chen, Linda G. Shapiro, Ranjay Krishna  
CVPR 2025

## **MolmoAct: Action Reasoning Models that can Reason in Space**

Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, Winson Han, Wilbert Pumacay, Angelica Wu, Rose Hendrix, Karen Farley, Eli VanderBilt, Ali Farhadi, Dieter Fox, Ranjay Krishna  
ArXiv 2025