ROB 498/599 · 3D Robot Perception · Fall 2026

L15 Spatial memory and language grounding

Scene graphs and memories as a language model's context, and what a language model can and cannot work out about space on its own.

Wed Oct 28, 2026 • WRITE-IN build

Spatial memory and language grounding

L15 · Unit 2

L13 built the graph and Monday built it online. Today a language model reads it, plans with it, and answers questions from it.

Then the question under all of that: what can a language model work out about space by itself, and what has to be computed for it?

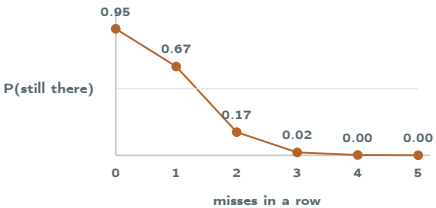
Today

1. **Where L13 stopped:** one slide on change detection.
2. **Planning on a graph:** SayPlan's search, path planner and plan check.
3. **Memory over time:** ReMEmbR's caption log and retrieval.
4. **Asking questions:** OpenEQA, and the blind baseline.
5. **What a language model can infer about space**, from text, images and video; what improved in 2025 and 2026, GPT-6 Astra included; and three ways to compute the rest for it.

How many misses before we delete it?

where L13 stopped · illustrative detector

Prior 0.95 that the mug is there. The detector misses a present mug 10% of the time and reports an absent one 5% of the time. Belief after one miss? After k?



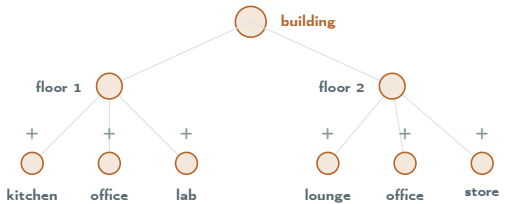
From answering to planning

ConceptGraphs answers which object. A robot told to refrigerate the orange on the kitchen bench needs a sequence of actions, across rooms, that it can actually execute.

- Hand the graph to an LLM as its context, and ask for a **plan**, not a node id.
- Two problems from L13 come back: the graph is too big for the context, and the LLM writes plans that cannot run.

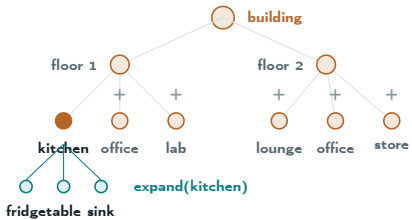
SayPlan

Rana et al., CoRL 2023



1. **Search:** the LLM starts from a collapsed graph and opens only the rooms it needs.
2. **Paths:** a classical planner, not the LLM, finds the route between rooms.
3. **Check:** a scene-graph simulator runs the plan and reports what fails; the LLM tries again.

1 of 3: start from a collapsed graph



- The LLM first sees only the top of the hierarchy: floors and rooms. Every object is hidden.
- `expand_node(room)` reveals what is in a room; `contract_node(room)` hides it again.
- The LLM calls these until it decides it has the **subgraph** the task needs, then stops.

```
Task: refrigerate the orange left on the kitchen bench.
LLM:  expand_node(kitchen)      # reveals bench, fridge, sink
LLM:  expand_node(lab)          # nothing relevant
LLM:  contract_node(lab)
LLM:  terminate                 # subgraph: kitchen and what is in it
```

What the search saves

an illustrative building

The building from L13's *How much smaller?*: 500 objects, 3,000 places, 40 rooms. Say 30 tokens per object, 8 per place, 10 per room. Tokens for the full graph, without the places, collapsed, and after the search opens 2 rooms of 12 objects?

What the search buys, measured

SayPlan, Tables 1 and 2

- Office graph: 6,731 tokens in full, 878 collapsed, 87% fewer before the search begins.
- Finding the right subgraph with **GPT-4**: 86.7% of simple searches and 73.3% of complex ones, in both environments.
- With **GPT-3.5**: 6.6% and 0%. It invented nodes to open, or opened the same one over and over.

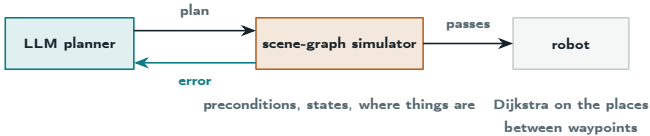
> *The hierarchy lets the model read the graph a piece at a time, and only a strong model can use it.*

2 of 3: the LLM does not plan paths

```
# an LLM asked for the whole plan writes every waypoint
goto(pose13) goto(pose14) goto(pose23) goto(pose24) goto(pose25)
goto(kitchen) access(kitchen_bench) pickup(orange1) ...
# SayPlan: the LLM names the room, Dijkstra fills in the poses
goto(kitchen) access(kitchen_bench) pickup(orange1) ...
```

- Waypoints are the places layer. Writing them out is long, and one wrong pose breaks the plan.
- So the LLM names the destination room or object, and **Dijkstra on the places graph** fills in the route.
- The LLM keeps the part a graph search cannot do: which objects, in which order, and why.

3 of 3: check the plan before the robot runs it



- `verify_plan` runs the plan forward **on the graph**: each action against the objects' states, predicates and affordances.
- A failure comes back as text, *cannot pick up banana*, if the fridge holding it is closed. The LLM rewrites the plan and the loop repeats.

One round of replanning

illustrative

```
plan 1: goto(kitchen) access(kitchen_bench) pickup(orange1)
        access(fridge) release(orange1)
check:  "cannot release orange1: fridge is closed"
plan 2: goto(kitchen) access(kitchen_bench) pickup(orange1)
        access(fridge) open(fridge) release(orange1) close(fridge)
check:  success
```

The first plan reads well and would fail on the robot. The check catches the missing `open(fridge)` on the graph, in milliseconds, before anything moves.

What each part buys

SayPlan, Table 3

	Simple, correct	Long, correct	Long, executable
LLM+P	93.3%	33.3%	0.0%
LLM as planner	93.3%	66.7%	13.3%
SayPlan	93.3%	73.3%	86.6%

- **Correctness:** the plan would do the task. **Executability:** every action can actually run.
- On short tasks all three are right as often. On long ones the others write plans that are *right in spirit* and cannot run; the path planner and the check fix that.

Where the LLM still fails

- **Negation:** *find me an office that does not have a cabinet* failed every time.
- **Counting and distance** across several nodes: the LLM miscounts, or guesses which room is nearer from coordinates.
- **A fixed map:** the graph is built beforehand and assumed static, so a moved orange breaks the plan.

> *The graph supplies the facts. Reasoning about space with them is still up to the LLM, and it is weak at it.*

Three places to decide what matters

When	System	What it costs
When the map is built	Clio	a task not on the list cannot be answered later
When a question is asked	ConceptGraphs	every object kept; long prompts
While a plan is made	SayPlan	the LLM must search well; only strong models can

A map, or a diary?

A scene graph holds the world as it is now. Some questions are about what the robot saw, and when.

- *Where did you last see my keys? How long was the loading dock blocked this morning? Take me somewhere with a nice view.*
- These need **time**, and things that are not objects: events, people, views.
- Next: **retrieval over a spatial memory**.

ReMEmbR

Anwar et al., ICRA 2025

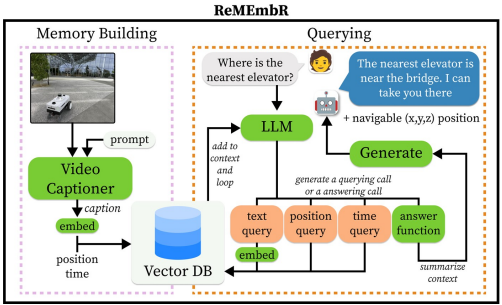


Figure: ReMEmbR (Anwar et al.), CC BY 4.0

Build: every 3 s of video becomes a caption, stored with where and when. **Query:** an LLM searches the store with tools, up to three rounds, then answers.

1 of 2: why not put it all in the prompt?

illustrative sizes

One caption every 3 s for an 8-hour day, 60 tokens each. How many tokens to hand an LLM the whole day? Storage, at a 1024-d float32 embedding, 300 B of text, and x, y, z, t per entry?

2 of 2: answering with three tools

```
Q:    Where can I get some chips?
LLM:  text_query("snacks")      # nearest captions by embedding
      14:02:11 (31.0, -4.2)  "shelves of chips and drinks by a counter"
      14:20:45 (2.5, 17.8)   "a vending machine in a hallway"
LLM:  text_query("cafeteria")   # a second look before answering
LLM:  answer({"position": [31.0, -4.2, 0.0],
              "text": "the cafeteria shelves, past the lobby"})
```

- Tools: **text** (nearest captions by embedding), **position** (memories near (x, y, z)), **time** (memories near a clock time). Up to three rounds.
- The answer is typed: a position to drive to, a time, a duration, yes or no, or a description.

Does retrieval beat reading everything?

ReMEmbR, NaVQA, Table I

Position error (m)	Short	Medium	Long
ReMEmbR	5.1	27.5	46.3
LLM with every caption	5.1	33.3	56.0
Video VLM, 2 frames/s	7.5	×	×

- NaVQA: 210 questions on campus drives, in clips under 2 min, 2 to 7 min, and over 7 min.
- On short clips reading everything is as good. As the history grows, retrieval pulls ahead, and the VLM cannot take the video at all (×).

ReMEmbR on a robot

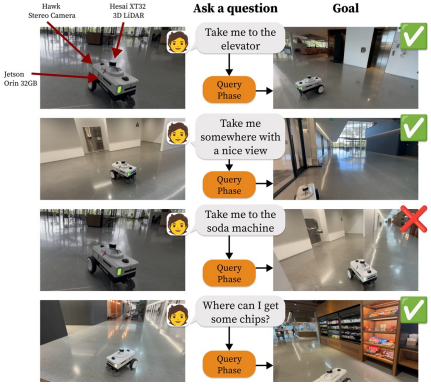


Figure: ReMEmbR (Anwar et al.), CC BY 4.0

- 25 minutes of driving in an office builds the memory, captioned on the robot's own GPU.
- *Where can I get some chips?* goes to the snack shelf. *Somewhere with a nice view* goes to a glass-walled corridor.
- *Take me to the soda machine* goes to a water fountain. The small on-board captioner had called it a **silver machine**.

Scene graph or memory log?

	Scene graph	**Caption memory**
Holds	objects, places, rooms, now	what was seen, where, when
Grows with	the size of the building	time spent driving
Good at	where is X, plans across rooms	when did, how long, what happened
Weak at	events, history	geometry, counting, anything not captioned

Embodied question answering

OpenEQA: Majumdar et al., CVPR 2024

- Ask the robot a question about a place it has been, in plain language, and grade the answer.
- **Episodic memory (EM-EQA):** answer from a recorded walk through the scene. **Active (A-EQA):** the agent is dropped in the scene and must explore before answering.
- 1,636 questions written by people, in over 180 real homes and offices from ScanNet and HM3D.

> *A benchmark for the question every map in this unit claims to answer: does the robot understand the place?*

Seven kinds of question

examples from the paper

Category	Example
Object recognition	What is left of the kitchen pass-through?
Attribute recognition	What colour is the kitchen backsplash?
Object state	Is the microwave door propped open?
Object localisation	Where is the checkers board?
Spatial understanding	Can another cookie jar fit on the cookie jar shelf?
Functional reasoning	Where can I store the house key?
World knowledge	Does this house have forced-air heating?

Grading an open answer

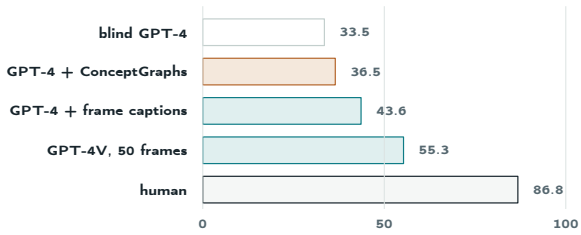
- Answers are free text, so exact match fails: *entryway table* and *the table by the front door* are both right.
- **LLM-Match:** GPT-4 compares each answer with the human one and scores it $\sigma \in \{1, \dots, 5\}$.

$$C = \frac{1}{N} \sum_{i=1}^{N} \frac{\sigma_i - 1}{4} \times 100\%$$

Three answers scored $5, 3, 1$ give $C = 50\%$. Against four human graders on 300 questions, Spearman $\rho = 0.909$, as high as the graders agree with each other.

How well do they do?

OpenEQA, EM-EQA



- **Blind:** GPT-4 given only the question, no memory at all.
- **+ ConceptGraphs:** L13's map, serialised as in L13's *What the language model actually sees*.
- The best system is about 32 points short of people.

A blind model scores a third

- Homes are regular. Asked the colour of a staircase railing with no memory at all, GPT-4 says *brown*, and is often right.
- So 33.5 is the floor, and ConceptGraphs adds only 3.0 points to it.
- On **spatial understanding** the scene graph scores 29.1, below the blind model's 31.4.

> *A map has to beat the guess a language model makes without one.*

Why the coordinates did not help

- The authors deleted every box and size from the serialised graph. The scores **barely moved**.
- So GPT-4 was answering from the captions, and not using the 3D numbers next to them.
- Same finding as SayPlan's failures on distance and counting: geometry written out as text is mostly ignored.

Last part of today

What a language model can and cannot infer about space, and what to compute outside it: distances, containment and free space done by geometry, handed to the model as answers rather than coordinates.

What the graph cannot do

- **Collision checking:** a box per object is not the free space. Keep the mesh or ESDF underneath.
- **Uncertainty:** most systems store one feature and one caption per node, with no confidence.
- **Relations** named by an LLM from captions and boxes are plausible, not measured.
- **Evaluation** is hard: there is rarely a ground-truth graph, so papers report object recall and query success instead.

What a language model brings

	From today's three systems
Good at	where things usually are (the blind model's 33.5), what an object is for, which room to search, the order of steps
Weak at	distances, counting, negation, left and right, anything that needs the numbers in the graph

Is that the input's fault or the model's? Test the model with the input taken away.

Space described in words

Yamada et al., TMLR 2024

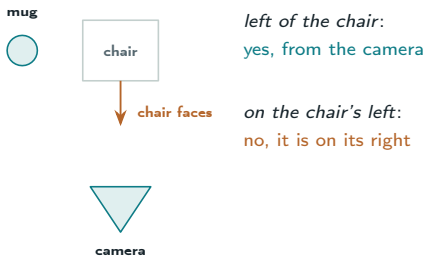
GPT-4 is told a map one step at a time (*you move up and see a cat*), then asked where it is after 8 moves. Chance is 1/8.

Map	GPT-4	People
3 by 3 square grid	0.69	0.90
ring of 12	0.22	0.78
triangular grid	0.20	0.58
hexagonal grid	0.05	0.41
all structures	0.29	0.67

The square grid, the layout of most text on the web, is easiest. On the hexagonal grid GPT-4 does worse than guessing.

Left of what?

What'sUp: Kamath et al., EMNLP 2023



- Photos of one object *on*, *under*, *left of* and *right of* another. Pick the right caption of 4: chance is 25%.
- CLIP: 31.0%. The best of 18 models: 47.8%. People: 100%.
- Only 0.2% of LAION-2B captions use a spatial preposition, and *left* is ambiguous when they do.

VSI-Bench: space from video

Yang et al., Thinking in Space, CVPR 2025



Figure: Yang et al., Thinking in Space (VSI-Bench), CC BY 4.0

Over 5{,}000 questions on 288 walk-through videos of real rooms (ScanNet, ScanNet++, ARKitScenes): counting, sizes, distances, directions, routes, order.

Scoring a number

mean relative accuracy, VSI-Bench Eq. 1

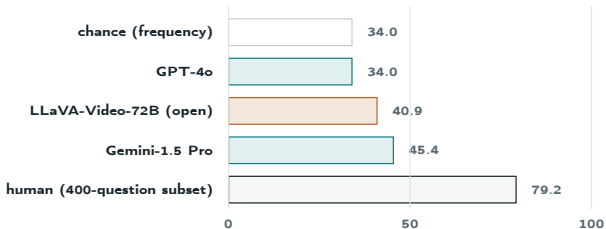
$$\mathrm{MRA} = \frac{1}{10} \sum_{\theta \in \{0.50,\, 0.55,\, \ldots,\, 0.95\}} \mathbb{1}\left(\frac{|\hat{y} - y|}{y} < 1 - \theta \right)$$

The bed is $y = 3.2$ m from the sofa. Score the answers $\hat{y} = 4.0$ m and $\hat{y} = 3.5$ m.

θ	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
$1-\theta$	0.50	0.45	0.40	0.35	0.30	0.25	0.20	0.15	0.10	0.05

How well do video models do?

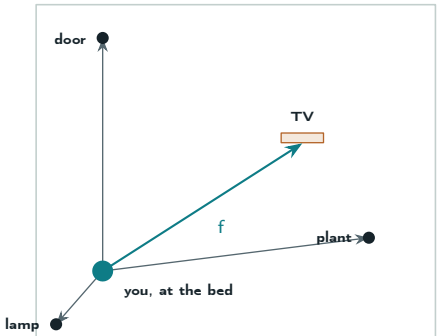
VSI-Bench, Fig. 6, average over the eight tasks



- GPT-4o scores the same as always giving the most common answer.
- The best model is 34 points short of people.
- 7 of the 12 open models score below chance.

Left, right or behind, by hand

an illustrative room



6 m by 5 m, seen from above

You stand at the bed $(1, 1)$ facing the TV $(4, 3)$. Is the door $(1, 4.5)$, the plant $(5, 1.5)$, the lamp $(0.3, 0.2)$ to your left, right, or behind (a turn of 135° or more)?

Thinking harder makes it worse

VSI-Bench, Secs. 5.1 and 5.2

- Chain-of-thought and tree-of-thought prompting **lower** the average by about 4%; self-consistency ends 1.1% below.
- On ordinary video questions the same prompting helps.
- Of the best model's wrong answers, about 71% are spatial reasoning: relating objects to each other, and switching between *my view* and *the room's layout*.

> *More words do not help. The failure is in the map the model holds.*

Draw the map first

VSI-Bench, Sec. 6, Table 2

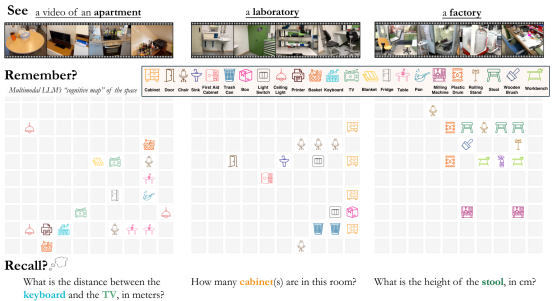
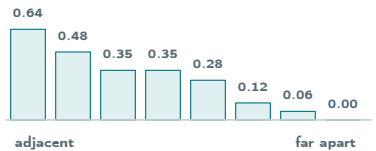


Figure: Yang et al., Thinking in Space (VSI-Bench), CC BY 4.0

Ask the model to place each object in a 10×10 grid, then answer. Relative distance: 46 with no map, 56 with its own map, 66 with the true map.

Local maps, not a global one

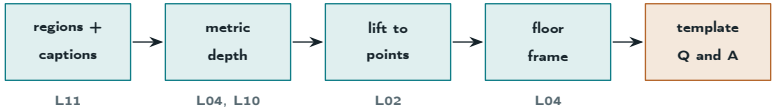
VSI-Bench, Fig. 11



- Score each pair of objects in the model's map: right if their distance is within one cell of the truth.
- Neighbours: 0.64. Objects far apart: near 0.
- The model keeps small local maps, and does not join them into one.

Train on geometry: SpatialVLM

Chen et al., CVPR 2024



- Run 10 million internet images through the pipeline above, and write 2 billion questions with metric answers: *how far is the chair from the table?*
- Every box except the last is from this course. The labels are only as good as monocular metric depth.

Does it help?

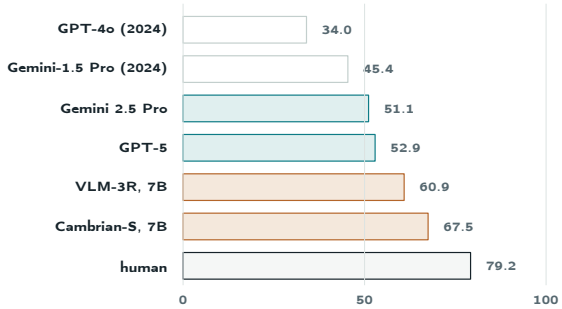
SpatialVLM, Tables 1 and 2

	Yes/no relations	**Gives a number**	**Within 2×**
GPT-4V	68.0	1.0	0.0
LLaVA-1.5	71.3	20.9	13.0
PaLM 2-E		88.8	33.9
SpatialVLM	75.2	99.0	37.2

- *Within 2×*: the answer is between half and double the true distance.
- Better than every baseline, and still off by more than a factor of 2 on most questions.

A year later on VSI-Bench

average score; sources in the notes



- Frontier models gained about 8 points in a year.
- Small open models **trained for space** gained more: a 7B model passes GPT-5 by 15.
- People are still ahead of everything.

Recipe 1: give it geometry tokens

VLM-3R (2025); Spatial-MLLM (2025)

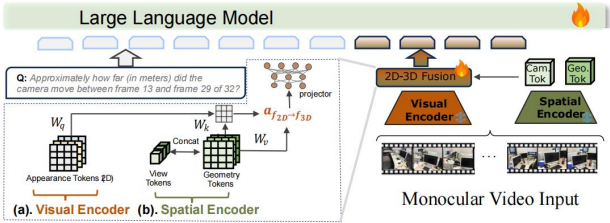


Figure: VLM-3R (Fan et al.), Apache-2.0

- A feed-forward 3D model from L10 (CUT3R in VLM-3R, VGGT in Spatial-MLLM) reads the same frames and hands the LLM **camera and geometry tokens** next to the image tokens.
- VLM-3R: 60.9. The same base model fine-tuned on the same data **without** the geometry tokens: 57.7.

Recipe 2: train on spatial questions

Cambrian-S: Yang et al., Nov 2025

- **VSI-590K:** 590,667 spatial question-answer pairs over 5,963 videos and 44,858 images from 10 sources.
- Cambrian-S-7B scores 67.5, over 16 points above Gemini 2.5 Pro. Its 0.5B version scores 50.6.
- **VSI-Super:** recall where things were, and keep a count, over videos of an hour and more. Gemini 2.5 Flash counts at 10.9% on hour-long video and runs out of context at two hours.
- Their proposal: keep a model that **predicts the next frame**, and store only what surprised it.

How much of that is reading the question?

Brown et al., Nov 2025

- Train a **blind** language model on part of VSI-Bench's own questions, test on the rest: it gains 17.9 points. Over half the counting answers are *2*.
- Fine-tune a video model on spatial QA: its blind score rises 18.8, its score with video 20.4. Most of the gain needed no video.
- **VSI-Bench-Debiased** drops the 54% of questions a blind model can game. Cambrian-S: 67.5 $\rightarrow$ 59.9.

> *OpenEQA's blind baseline, again. Report it for any spatial benchmark you use.*

Recipe 3: train for the robot's questions

Gemini Robotics-ER, Mar 2025

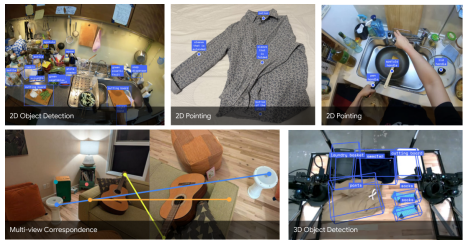


Figure: Gemini Robotics Team (arXiv 2503.20020), CC BY 4.0

- Point at the handle, box the socks in 3D, match points across views: outputs a controller can use, in pixels and metres.
- 3D boxes on SUN-RGBD: 48.3 AP, against 30.7 for the base Gemini and 43.7 for ImVoxelNet, a trained 3D detector.

Where the gap is still wide

MindCube: Yin et al., v2 Mar 2026

- A few photos of one scene from different sides, then questions that need a viewpoint none of the photos has: *what would be on your left if you stood over there?*
- 21,154 questions. Chance: 32.4. GPT-5: 47.6. People: 94.5.
- Train a 3B model to **write a map first, then reason**: 37.8 $\rightarrow$ 57.8; with RL on top, 61.3.

> *Writing the map down helps, as in VSI-Bench. Taking a viewpoint the camera never had is still hard.*

GPT-6 Astra

OpenAI, released Sep 3, 2026

- A general model, launched on computer use and 3D design. Robot demos followed within weeks, mostly from outside OpenAI.
- It writes the gripper's target positions itself. A planner over a separate VLA, as in SayPlan, is no longer the only option.
- Demos: painting the Golden Gate Bridge with a cheap arm, block pick-and-place, slicing a cucumber, a Rubik's cube in simulation.

> *Under eight weeks of evidence. Read the numbers that follow as early reports.*

Astra on a robot arm

StationeryBench (Robocurve); RoboDojo

StationeryBench, 100 tasks	GPT-6 Astra	MolmoAct2
fully completed	7	0
median progress, of 100	46	12

- Two YAM arms, five desk tasks: uncap a marker, pour out paper clips, hand a ruler from one arm to the other.
- RoboDojo: briefly first among robot-control models, ahead of $\pi_{0.5}$ and MolmoAct2, in simulation. The real-robot runs stopped after it damaged hardware. It has since fallen to seventh.
- Yoav Artzi called it a *step change in spatial reasoning*. Note that it still completed only 7% of the tasks.

Astra navigating from one camera

Dai et al., arXiv 2609.29861, Sep 2026

- Vision-and-language navigation in continuous environments (R2R-CE): *walk past the sofa, turn left at the kitchen, stop by the fridge*.
- Monocular RGB only. No navigation training, no waypoint predictor, no map built beforehand.
- Success 81.3% on 100 episodes: 15.3 points above the best zero-shot method and 9.2 above the best **trained** one.
- Failures left: executing the move it chose, and deciding it has arrived.

Astra with a spatial memory: HomeBody

Huh, Gu, Truong, Liu, Tevet (Stanford, Caltech), Sep 2026

1. **Map first:** the robot explores; SLAM on lidar fixes the room's size, and Astra builds an Isaac Sim twin from the video. ICP aligns the two (L07).
2. **Memory:** camera frames stored in one shared frame with descriptions, recalled when an object is out of view (ReMEmbR's idea).
3. **Skills as tools:** pick, place, open drawer, navigate. Astra picks a skill and a target by a tool call; a failure comes back as text (SayPlan's loop).
4. **Perception underneath:** stereo depth, SAM 2.1 tracking, 250 Hz arm control on a laptop GPU.

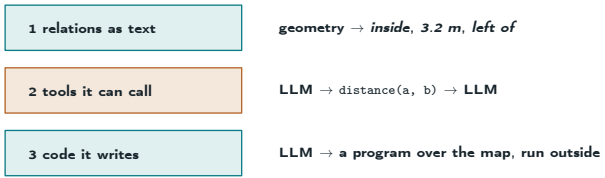
> *The strongest model of 2026 still gets its metric geometry from SLAM, through a tool call.*

What Astra still cannot do

- **Speed:** it pauses for seconds to think before each move. π_0 produces a second of actions in 73 ms.
- **Reaction:** slow to respond when the scene changes mid-motion, and too large to run on the robot.
- **Contact:** weak at precise two-handed coordination and at dexterous hands. An overnight shirt fold ended in a ball.
- **Cost:** HomeBody lists API cost, reasoning pauses and overheating finger servos as its limits.

> *Better spatial reasoning moves rows in the Who answers table. It does not remove the geometry column.*

Three ways to give it the geometry



- Each moves the spatial step out of the model, into code that does it exactly.
- You have already seen route 2: SayPlan's Dijkstra, ReMEmbR's position query.

Route 1: relations as text

an illustrative kitchen wall



- Each node keeps an axis-aligned box: an interval in x, in y and in z.
- **Inside:** each of the object's intervals lies within the container's.
- **On:** the object's bottom is within 2 cm of the support's top, and its x, y footprint lies within the support's.
- The map builder runs these on nearby pairs and writes the true ones as words.

Inside or on, by hand

the same kitchen wall

Boxes $[x], [y], [z]$ **in metres. Cabinet** $[0, 0.8], [0, 0.35], [1.4, 2.1]$**; counter** $[0, 2], [0, 0.6], [0, 0.9]$**; mug A** $[0.26, 0.34], [0.12, 0.2], [1.42, 1.52]$**; mug B** $[1.2, 1.28], [0.2, 0.28], [0.9, 1.0]$**. Which mug is *inside* the cabinet, and which is *on* the counter?**

Route 2: tools it can call

LLM-Grounder: Yang et al., ICRA 2024

- Task: find the object a sentence describes in a 3D scan (ScanRefer). *The chair between the window and the table.*
- GPT-4 splits it into a target and landmarks, and calls two tools on an OpenScene map. They return boxes, **volumes**, and **distances** between centroids. GPT-4 picks.

ScanRefer, zero-shot	**Acc@0.25 (\{}%)**
OpenScene	13.0
OpenScene + GPT-3.5	14.3
OpenScene + GPT-4	17.1
ScanRefer, trained on it	34.4

Route 3: code it writes

Code as Policies: Liang et al., ICRA 2023

```
# while the red block is to the left of the blue bowl,
# move it to the right 5cm at a time.
while get_pos('red block')[0] < get_pos('blue bowl')[0]:
    target_pos = get_pos('red block') + [0.05, 0]
    put_first_on_second('red block', target_pos)
```

- Generated by the model, Sec. III-C. No example like it was in the prompt.
- `get_pos` is a perception call (an open-vocabulary detector); the comparison and the 5 cm step are plain arithmetic, run outside the model.

Who answers which question?

Question	Who	How
Where would I find a can opener?	the model	priors; the blind baseline
Is the kettle to my left?	geometry	cross product from a pose
How far is the bed from the sofa?	geometry	boxes or the mesh
Will another jar fit on the shelf?	both	model parses, free space decides
How many chairs are there?	the map	count nodes, after association
Somewhere quiet to take a call?	the model	over captions, ReMEmbR style

Unit 2 in one slide

Lecture	The map gains	Open problem
L11 lifting 2D models	a meaning at every point	features disagree across views
L12 objects	pose and shape of one thing	unseen, articulated, deformable
L13, L14 scene graphs	objects, places, rooms, relations	what counts as an object
L15 memory, language	history, questions, plans	the model cannot use the geometry

Every map so far describes the world as it is. Unit 3 predicts what happens next.

Learn it again, differently

the explanations worth your time

- The SayPlan project page. The robot carrying out long plans across an office floor.
- The ReMEmbR project page and the OpenEQA project page, with videos and the benchmark data.
- Thinking in Space: VSI-Bench's code and data. Try a few questions yourself before reading the scores.
- The What'sUp repository. The left, right, on and under photo sets, small enough to look through in ten minutes.
- The Code as Policies project page. Videos of generated code driving a robot, with the prompts.
- HomeBody and the community list Awesome Astra Embodied AI: what people have built on GPT-6 Astra in its first weeks, with links to each benchmark.

What you should be able to do

- Count the tokens a collapsed graph saves, and say what SayPlan hands to the LLM, to Dijkstra and to the simulator.
- Size a day of caption memory, explain why ReMEmbR retrieves instead of reading it all, and read an OpenEQA score against the blind baseline.
- Score a numerical answer with mean relative accuracy.
- Decide left, right or behind from a position and a facing direction, and *inside* or *on* from two boxes.
- Say what text-only, image and video models get wrong about space, with one number for each, and name the three recipes that have since improved them.
- For a spatial question, say whether the model, the map or the geometry should answer it, and how the answer reaches the model.

Next

L16 Mon Nov 2

- **L16, Mon Nov 2: latent dynamics.** Unit 3 begins: what a world model is, PlaNet and Dreamer, JEPA. Response paper: DreamerV3.
- **L17, Wed Nov 4: 4D and occupancy forecasting.** Response paper: Occupancy Networks.
- **Both responses due Sun Nov 8.**
- **Seminar II, Mon Nov 9:** Unit 2 papers, student-led. Sign up on Piazza.