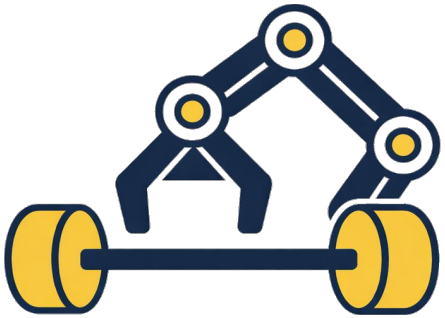


VisGym: Diverse, Customizable, Scalable Environments for Multimodal Agents

Zirui (Colin) Wang*, Junyi Zhang*, Jiaxin Ge*, Long Lian, Letian Fu, Lisa Dunlap,
Ken Goldberg, Xudong Wang, Ion Stoica, David M. Chan, Sewon Min, Joseph E. Gonzalez

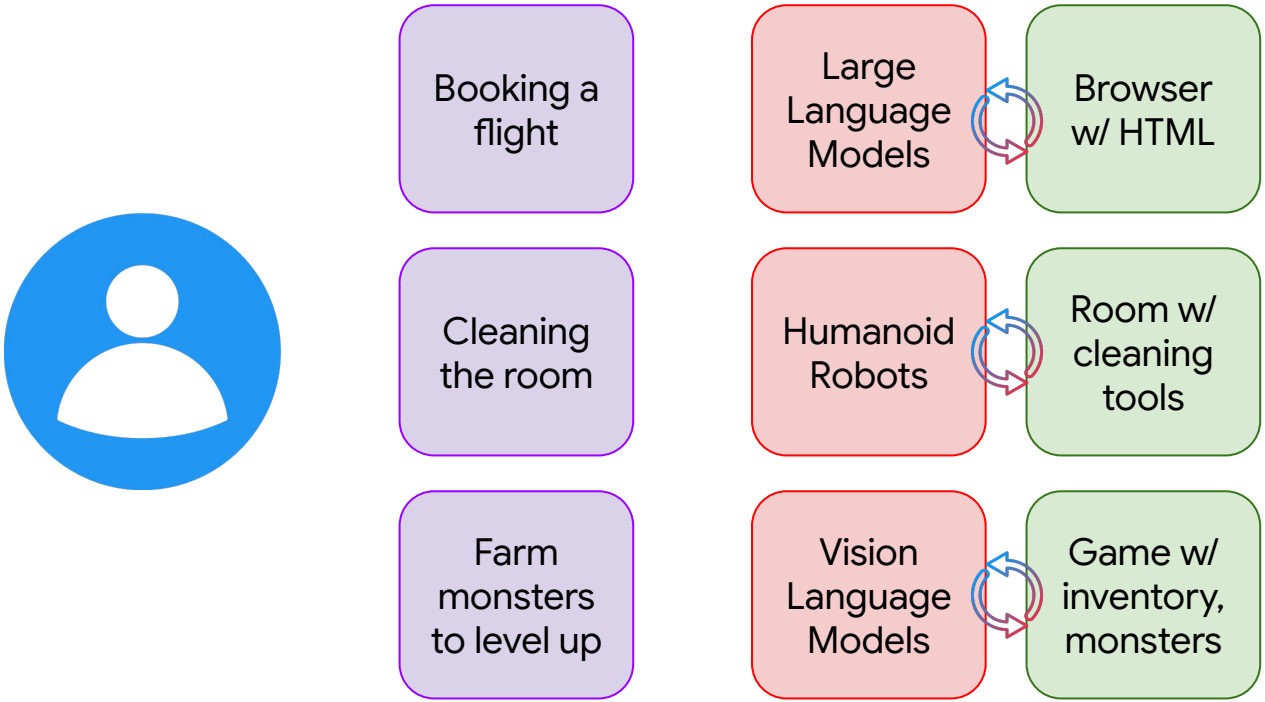
UC Berkeley

sky UC Berkeley

BAIR
BERKELEY ARTIFICIAL INTELLIGENCE RESEARCH

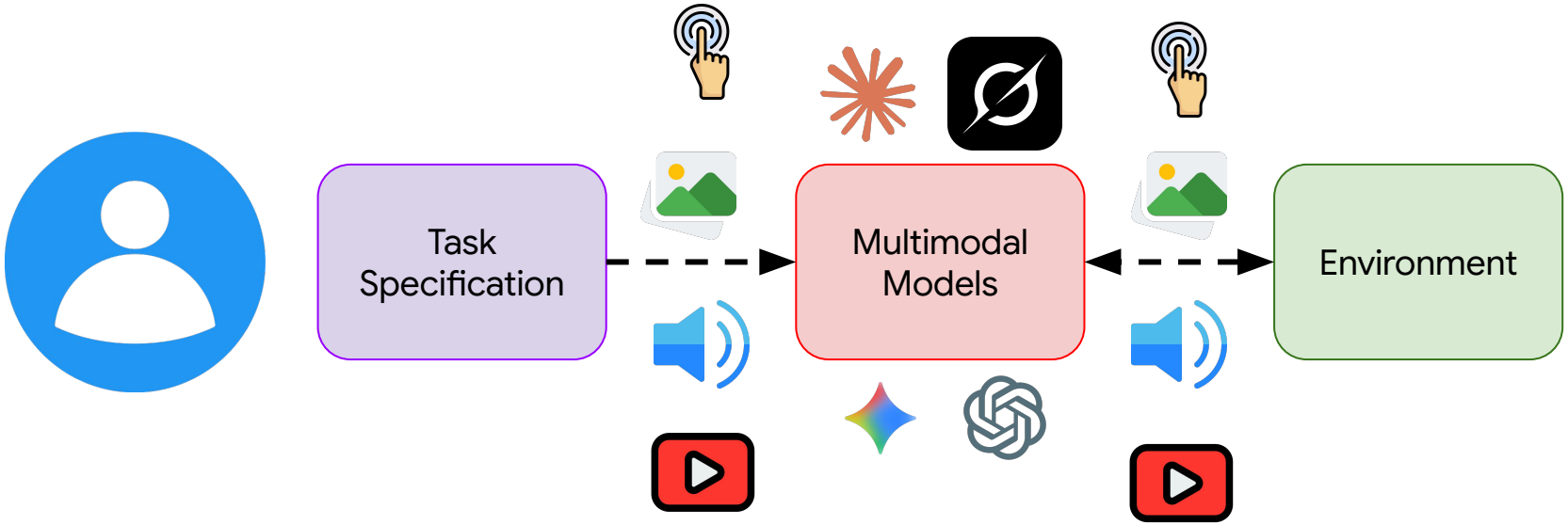
What are agents?

Agents interact with environments to solve problems/tasks for humans.



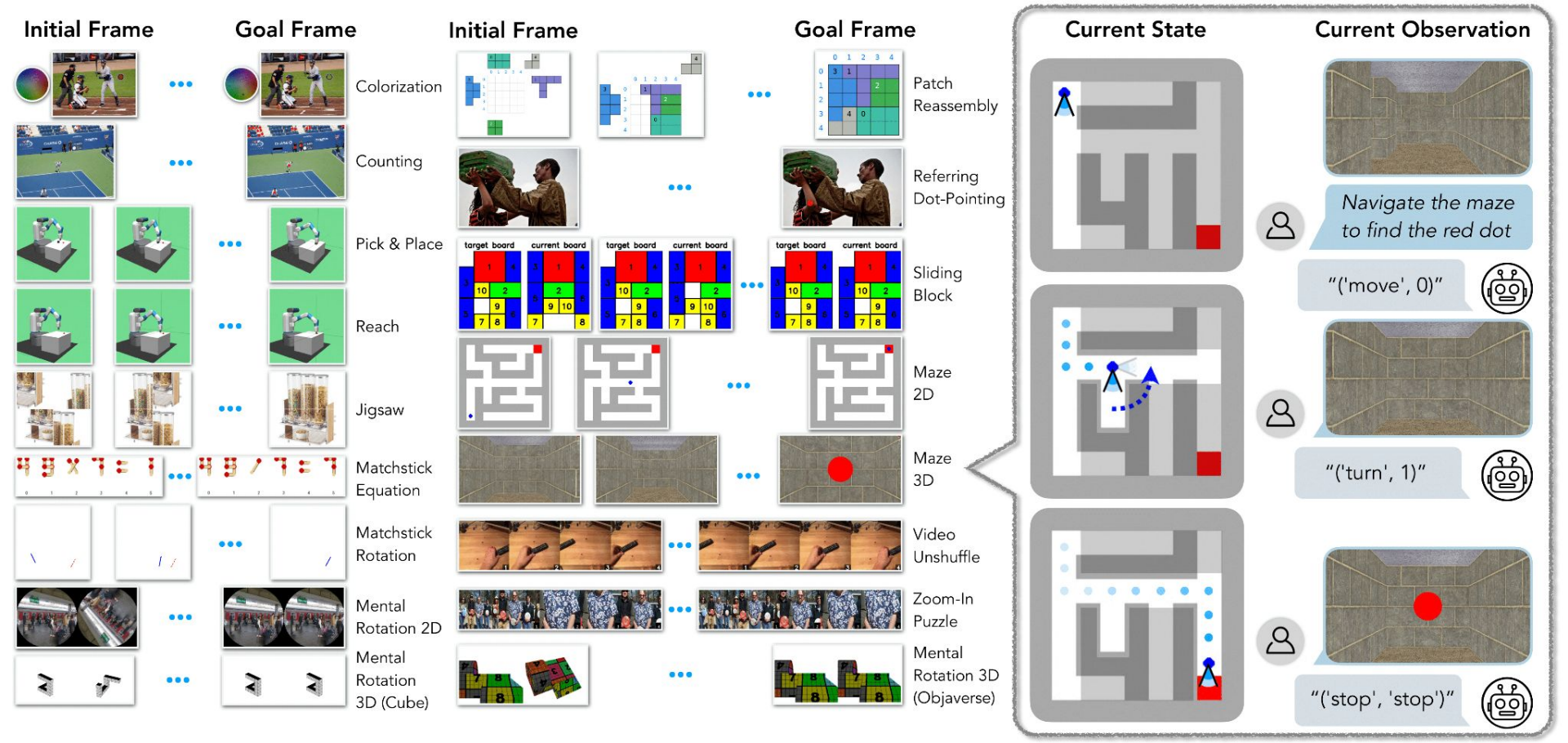
What are multimodal agents

Agents that perceive more than text from the human and/or the environment side.

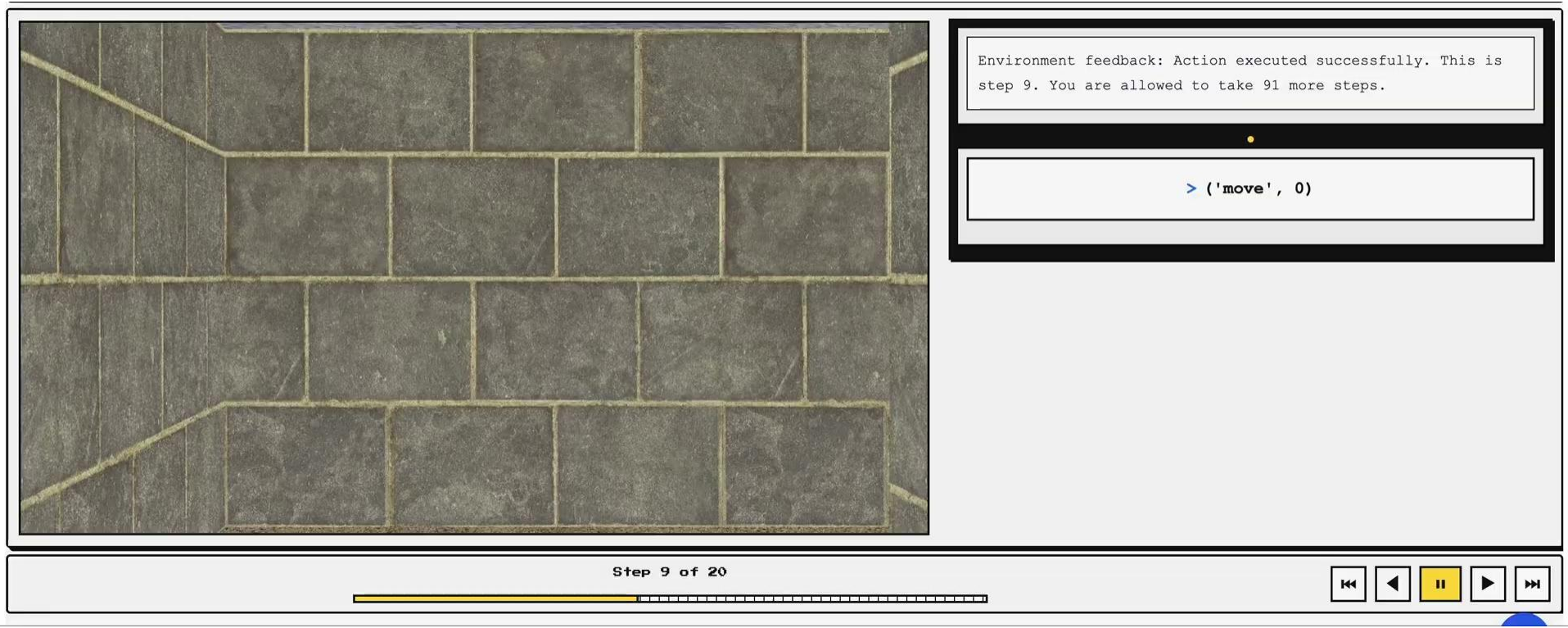


What gap are we filling in the multimodal agent space

- Many real-world problems/tasks are multimodal (in particular, vision)
  - Web agents; Robotics; GUI design, frontend
- We need **evaluations** that are indicative of their capabilities.
  - OSWorld, Libero, etc
- We need **training data** to teach the models how to become good multimodal agents.
  - MMAT, CALVIN, etc
- But more importantly, we need a **playground** to understand current limitations of models and data so we can develop better methods and data recipes.
  - **VisGym**


Initial Frame
Goal Frame
Initial Frame
Goal Frame
Colorization
Patch Reassembly
Counting
Referring Dot-Pointing
Pick & Place
Sliding Block
Reach
Maze 2D
Jigsaw
Matchstick Equation
Maze 3D
Matchstick Rotation
Video Unshuffle
Mental Rotation 2D
Zoom-In Puzzle
Mental Rotation 3D (Cube)
Mental Rotation 3D (Objaverse)
target board
current board
Current State
Current Observation
Navigate the maze to find the red dot
"('move', 0)"
"('turn', 1)"
"('stop', 'stop')"

Demo - Maze 3D

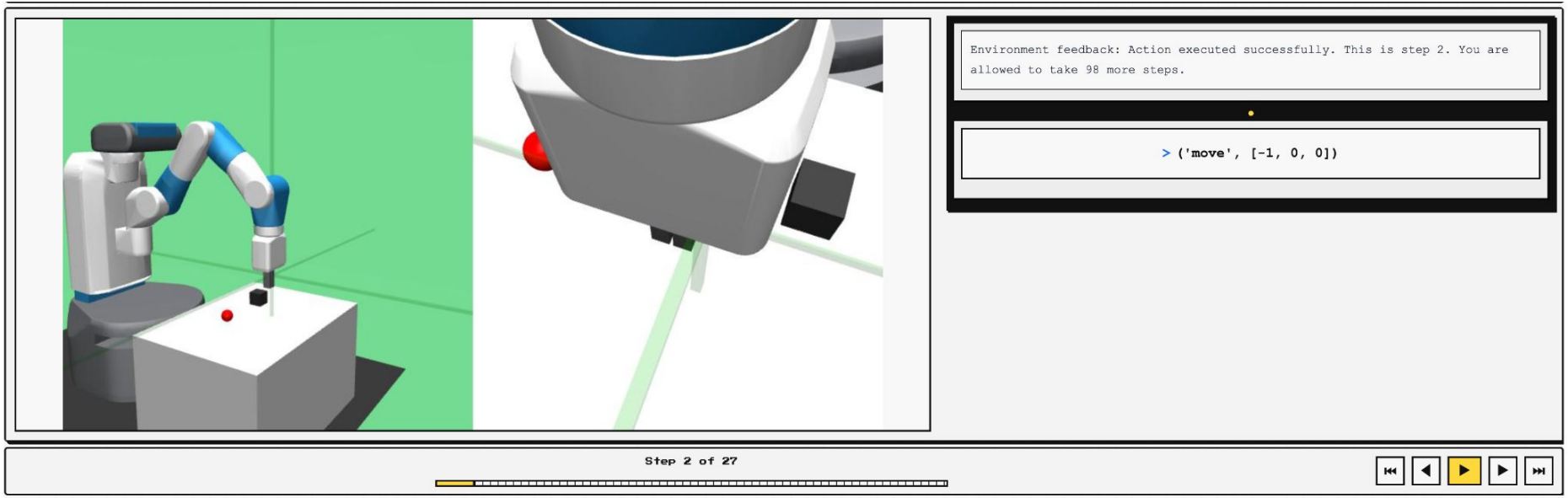

Environment feedback: Action executed successfully. This is step 9. You are allowed to take 91 more steps.
> ('move', 0)
Step 9 of 20

Colorization

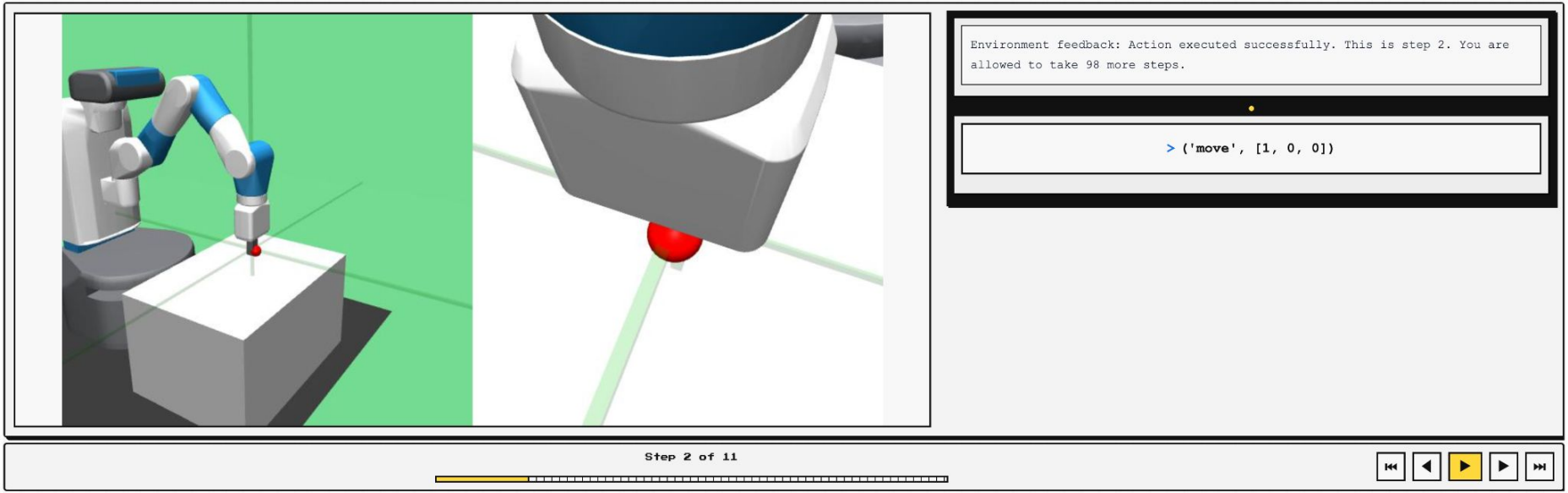
Counting


Environment feedback: Action executed successfully. This is step 3. You are allowed to take 97 more steps.
> ('mark', (0.8144, 0.6096))
Step 3 of 7

Pick and Place



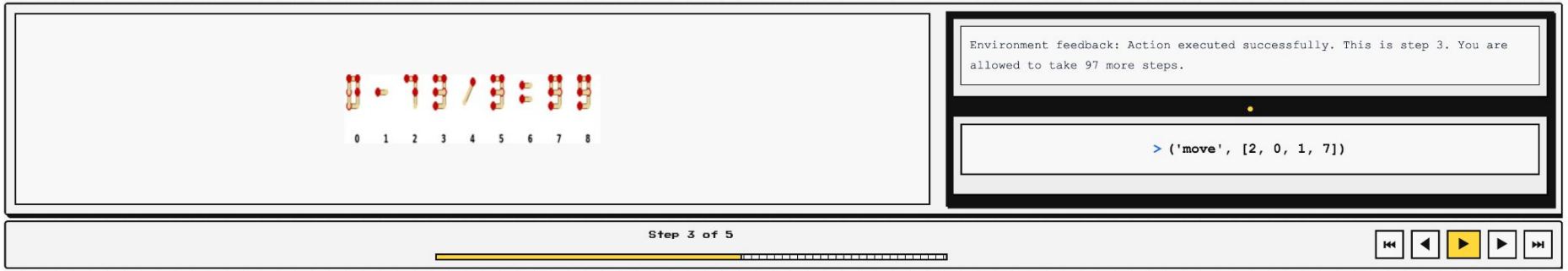
Reach


Environment feedback: Action executed successfully. This is step 2. You are allowed to take 98 more steps.
> ('move', [1, 0, 0])
Step 2 of 11

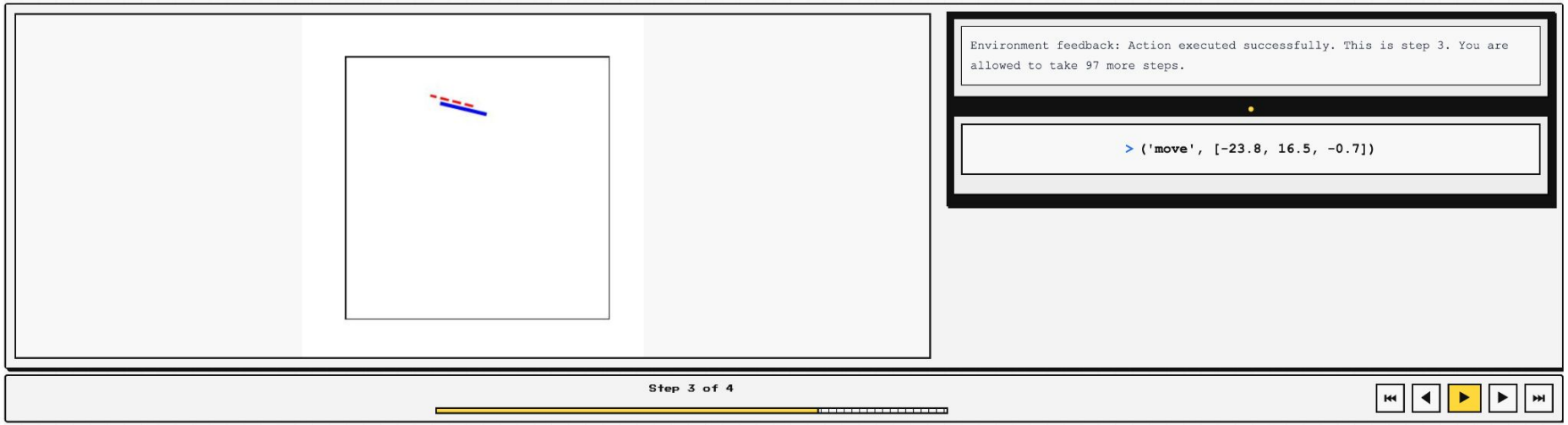
Jigsaw


Environment feedback: Action executed successfully. This is step 5. You are allowed to take 95 more steps.
> ('swap', ((2, 2), (0, 1)))
www.bigstock.com · 383370383
Step 5 of 8

Matchstick Equation

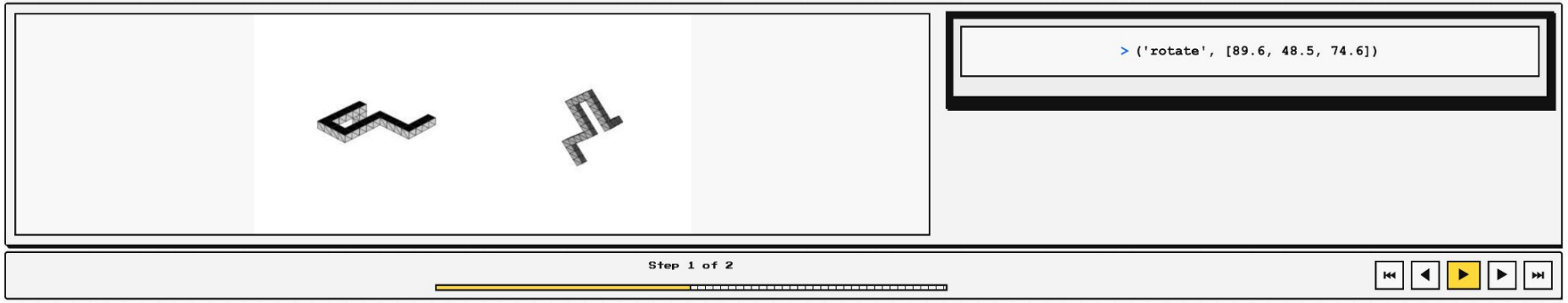


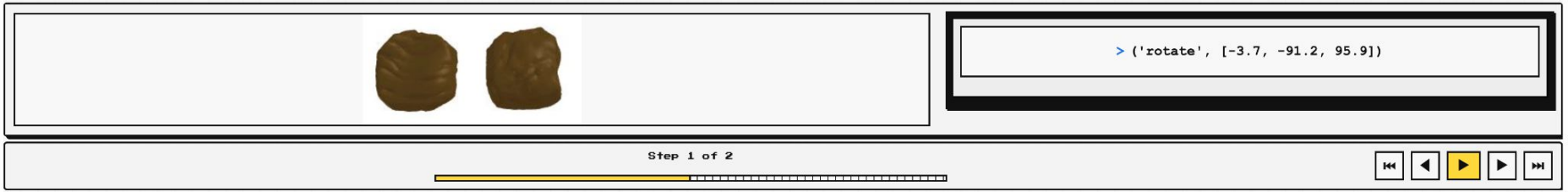
Matchstick Rotation



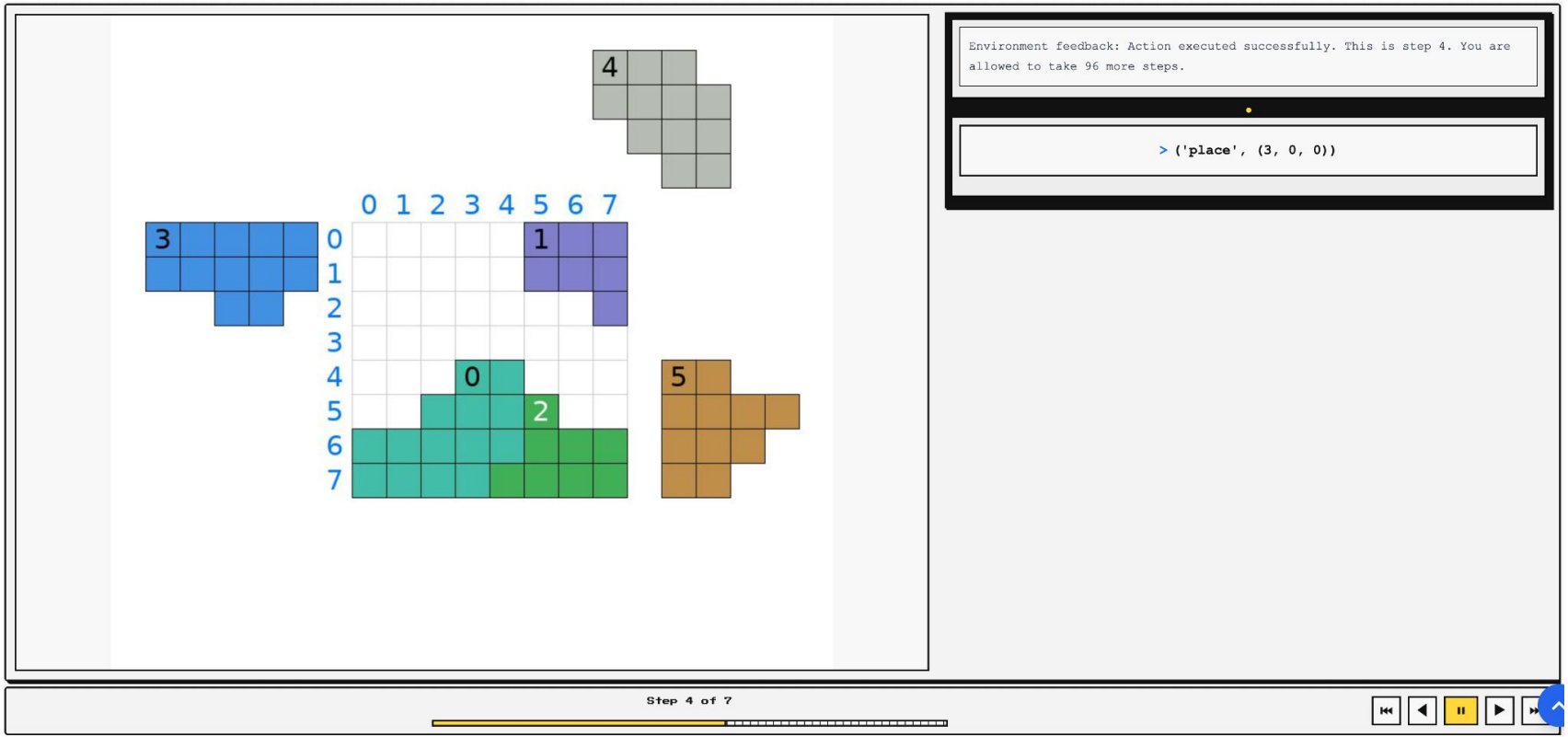
Mental Rotation 2D

Mental Rotation 3D Cube & Objaverse





Patch Reassembly



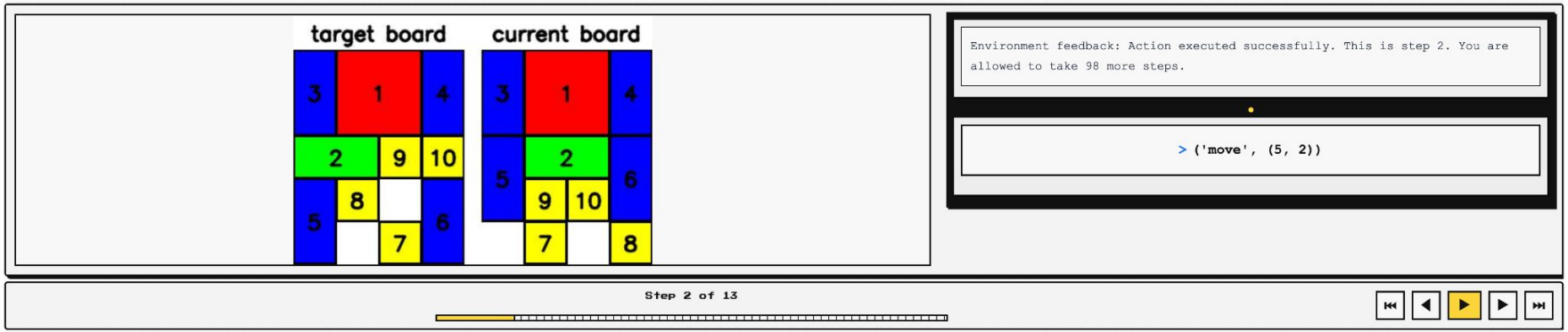
Referring Dot-Pointing

Environment feedback: Action executed successfully. This is step 2. You are allowed to take 98 more steps.

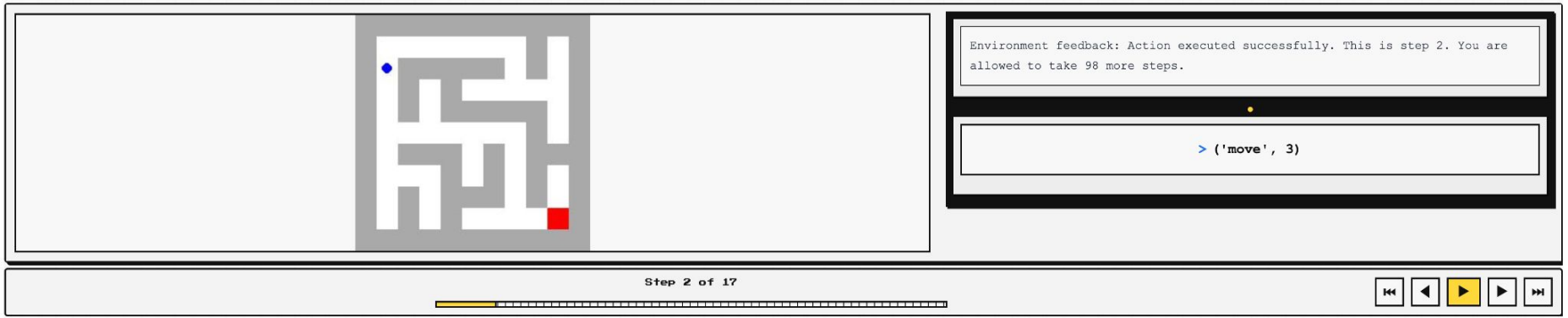
> ('mark', (0.3530, 0.4949))

Step 2 of 4

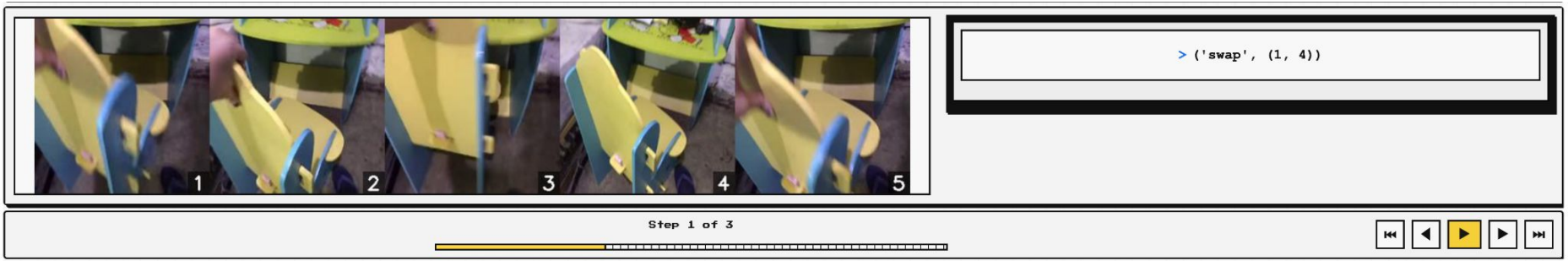
Sliding Blocks



Maze 2D

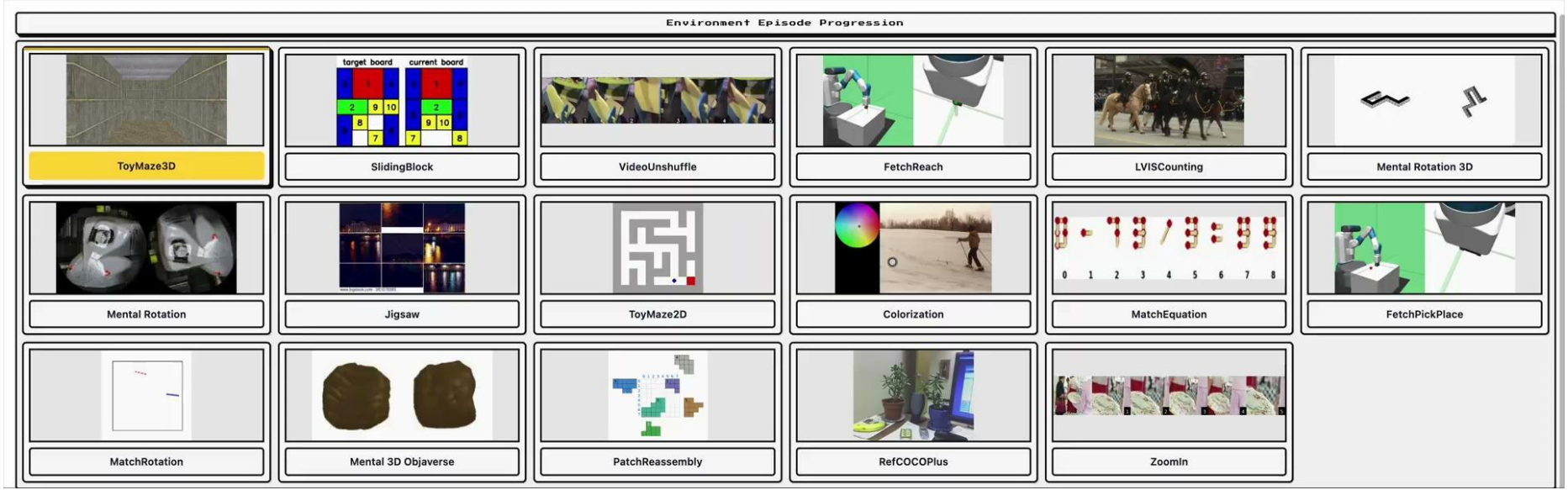

Environment feedback: Action executed successfully. This is step 2. You are allowed to take 98 more steps.
> ('move', 3)
Step 2 of 17

Video Unshuffle



Zoom-In Puzzle

VisGym

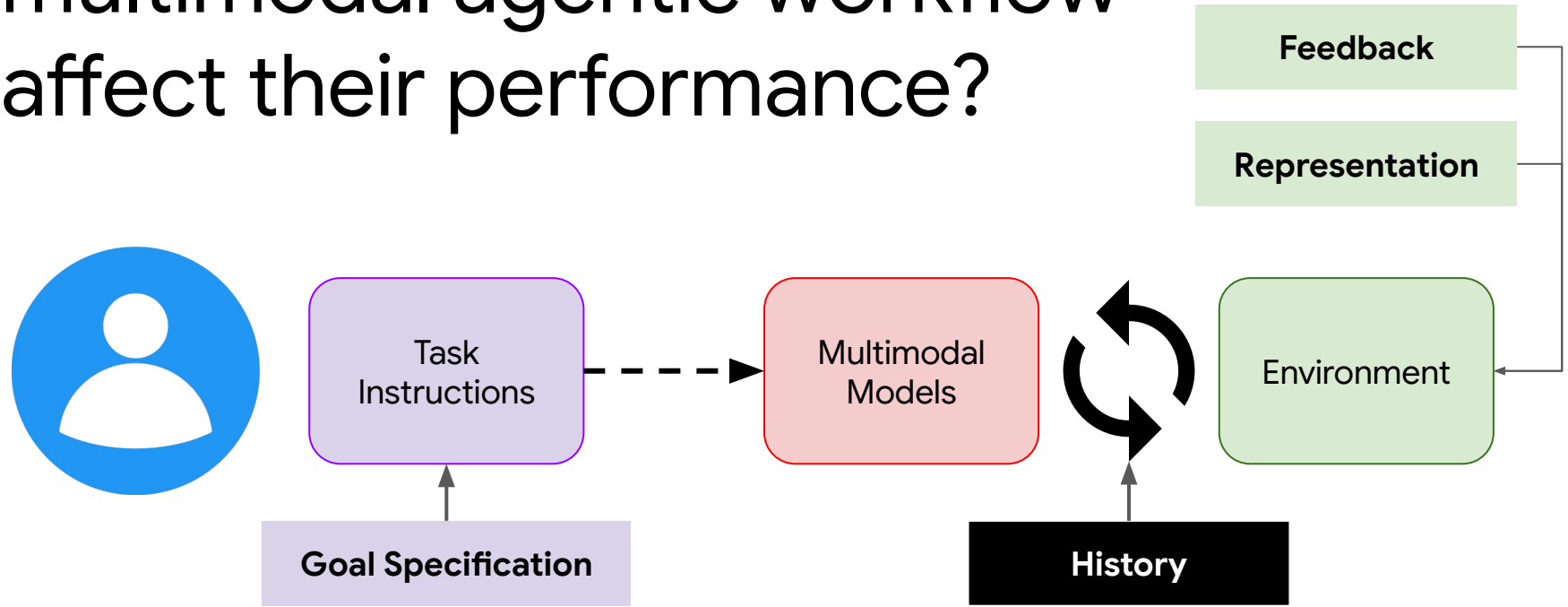

Environment Episode Progression
ToyMaze3D
target board
current board
SlidingBlock
VideoUnshuffle
FetchReach
LVISCounting
Mental Rotation 3D
Mental Rotation
Jigsaw
ToyMaze2D
Colorization
0 1 2 3 4 5 6 7 8
MatchEquation
FetchPickPlace
MatchRotation
Mental 3D Objaverse
PatchReassembly
RefCOCOPlus
ZoomIn

Task Categorization

Table 2. **VisGym environments.** For each environment, we specify (1) **Domain**: whether observations come from **Real** or **Synthetic** images, (2) **Observability (Obs.)**: **Full** or potentially **Partial**, (3) **Dynamics (Dyn.)**: **Known** vs. **Unknown** dynamics, (4) **Parameters (P.)**: number of difficulty parameters, and (5) **Available Actions**.

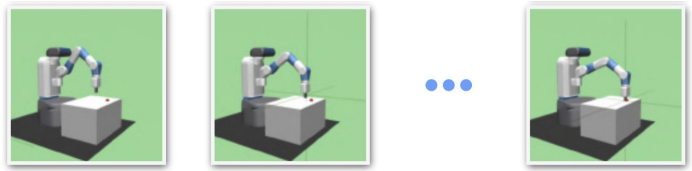
Environment	Domain	Obs.	Dyn.	P.	Available Actions
Colorization (102)	Real	Full	Known	1	`rotate`(θ); `saturate`(δ); `stop`()
Counting (30)	Real	Full	Known	2	`mark`(x, y); `undo`(); `guess`(N); `stop`()
Jigsaw (27)	Real	Full	Known	2	`swap`($(r_1, c_1), (r_2, c_2)$); `reorder`([...]); `stop`()
Matchstick Equation (42)	Synthetic	Full	Known	1	`move`($[i, s, j, t]$); `undo`(); `stop`()
Matchstick Rotation (44)	Synthetic	Full	Unknown	3	`move`($[dx, dy, d\theta]$); `stop`()
Maze 2D (43)	Synthetic	Full	Known	2	`move`(d); `stop`()
Maze 3D (43)	Synthetic	Partial	Known	2	`move`(0); `turn`(d); `stop`()
Mental Rotation 2D (18)	Real	Full	Known	1	`rotate`(θ); `stop`()
Mental Rotation 3D (Cube) (66; 70)	Synthetic	Partial	Known	3	`rotate`($[dy, dp, dr]$); `stop`()
Mental Rotation 3D (Objaverse) (70; 20)	Synthetic	Partial	Known	1	`rotate`($[dr, dp, dy]$); `stop`()
MuJoCo Fetch (Pick-and-Place) (85)	Synthetic	Partial	Unknown	0	`move`($[x, y, z]$); `gripper`(g); `stop`()
MuJoCo Fetch (Reach) (85)	Synthetic	Partial	Unknown	0	`move`($[x, y, z]$); `stop`()
Patch Reassembly (28)	Synthetic	Full	Known	2	`place`(p, r, c); `remove`(p); `stop`()
Referring Dot-Pointing (39)	Real	Full	Known	0	`mark`(x, y); `stop`()
Sliding Block (75)	Synthetic	Full	Known	1	`move`(b, d); `stop`()
Video Unshuffle (29; 60)	Real	Full	Known	3	`swap`(i, j); `reorder`([...]); `stop`()
Zoom-In Puzzle (6)	Real	Full	Known	5	`swap`(i, j); `reorder`([...]); `stop`()

How design factors of multimodal agentic workflow affect their performance?



Do we keep the full context history for agents?

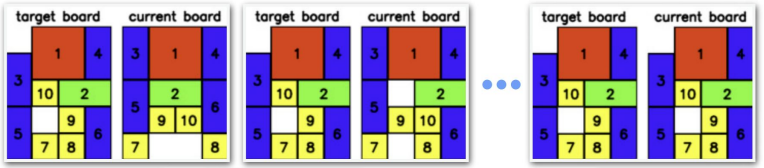
Learned Dynamics



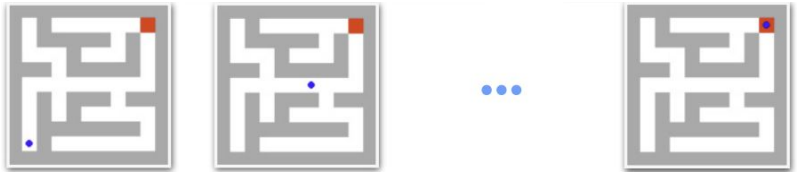
Reach

Matchstick Rotation

Learned Feedback

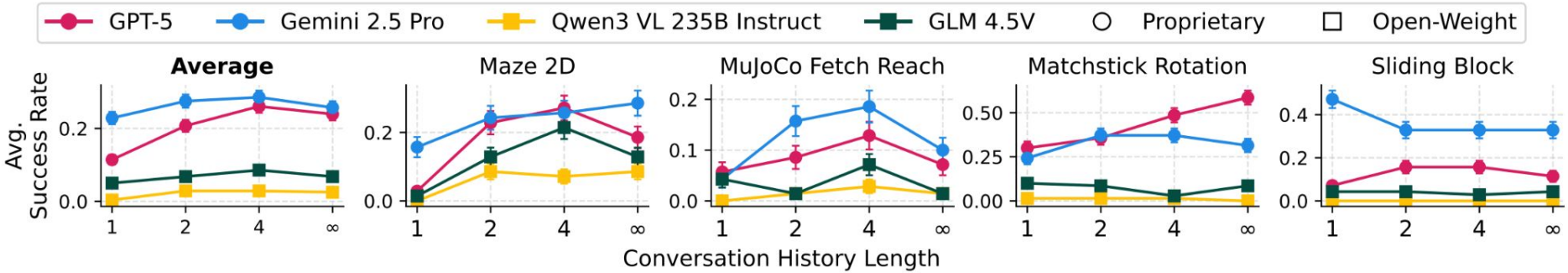


Sliding Block



Maze 2D

Do we keep the full context history for agents?

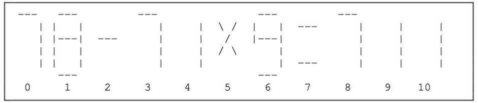


Models benefit from having some context in its history to understand the consequences of their actions (e.g., from feedback, visual differences, etc). However, performance saturates and even **degrades** as we keep all their decision makings!

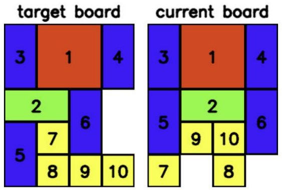
Do we represent visuals in text?

Image-based?

Visual Rendering (default)



Text Representation (variant)
Matchstick Equation



Visual Rendering (default)

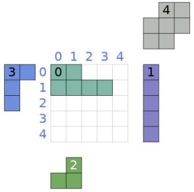
```
Target Current
-----------
3114 | 3114
3114 | 3114
226. | 5226
576. | 5906
5890 | 7.8.
```

Text Representation (variant)
Sliding Block

Visual Rendering (default)

```
#############
#############
## #      A##
## ####### ##
## #     # ##
## ### ### ##
##         ##
## ####### ##
##   #     ##
## # ########
##T#       ##
#############
#############
```

Text Representation (variant)
Maze 2D



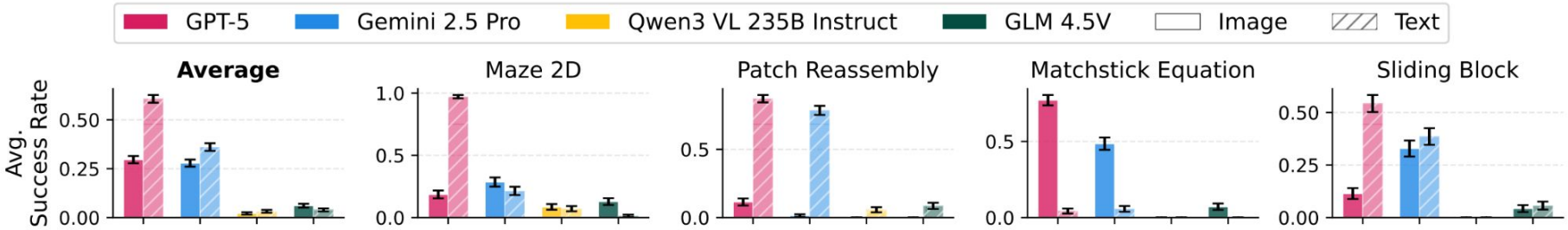
Visual Rendering (default)

```
   0  1  2  3  4
  ---------------
0 | 0  0  .  .  .
1 | 0  0  0  0  .
2 | .  .  .  .  .
3 | .  .  .  .  .
4 | .  .  .  .  .

--- Parked Patches ---

Patch 1:
  *
  1
  1
  1
  1

Patch 2:
   *
  22

Patch 3:
  *3
  3
  3

Patch 4:
   *4
  444
  44
```

Text Representation (variant)
Patch Reassembly

Text-based?

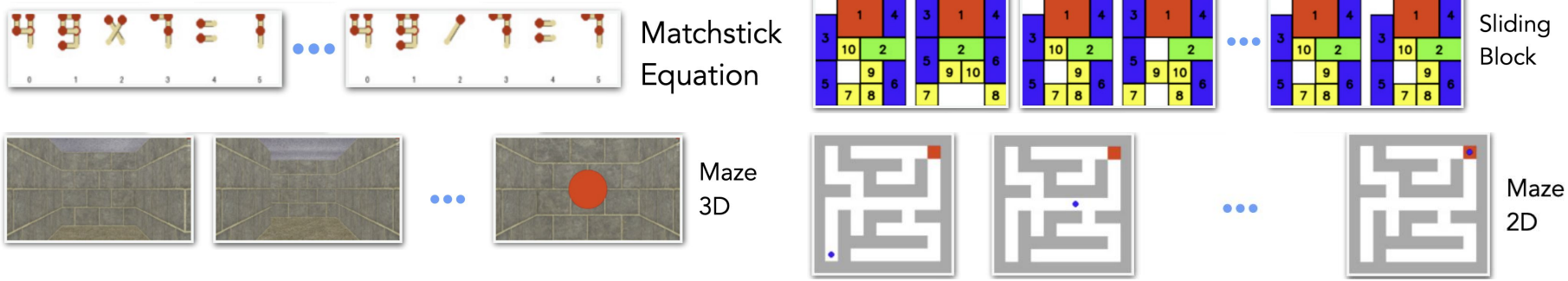
Do we represent visuals in text?



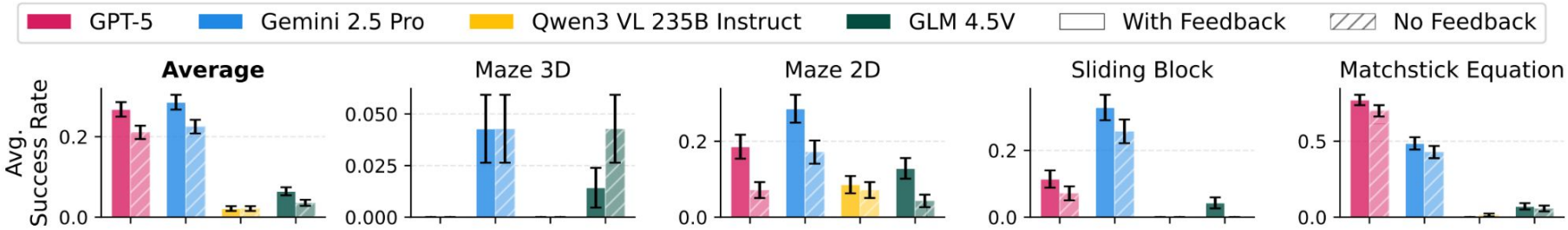
Models perform well when visually structured inputs such as grids can be faithfully represented in text, revealing a gap between perception and problem solving. However, when representations are unstructured, such as figlet-style matchstick equations, visual priors still provide a clear benefit.

Do we need text feedback from environment?

- Fine-grained vision
- Common for models to make illegal moves

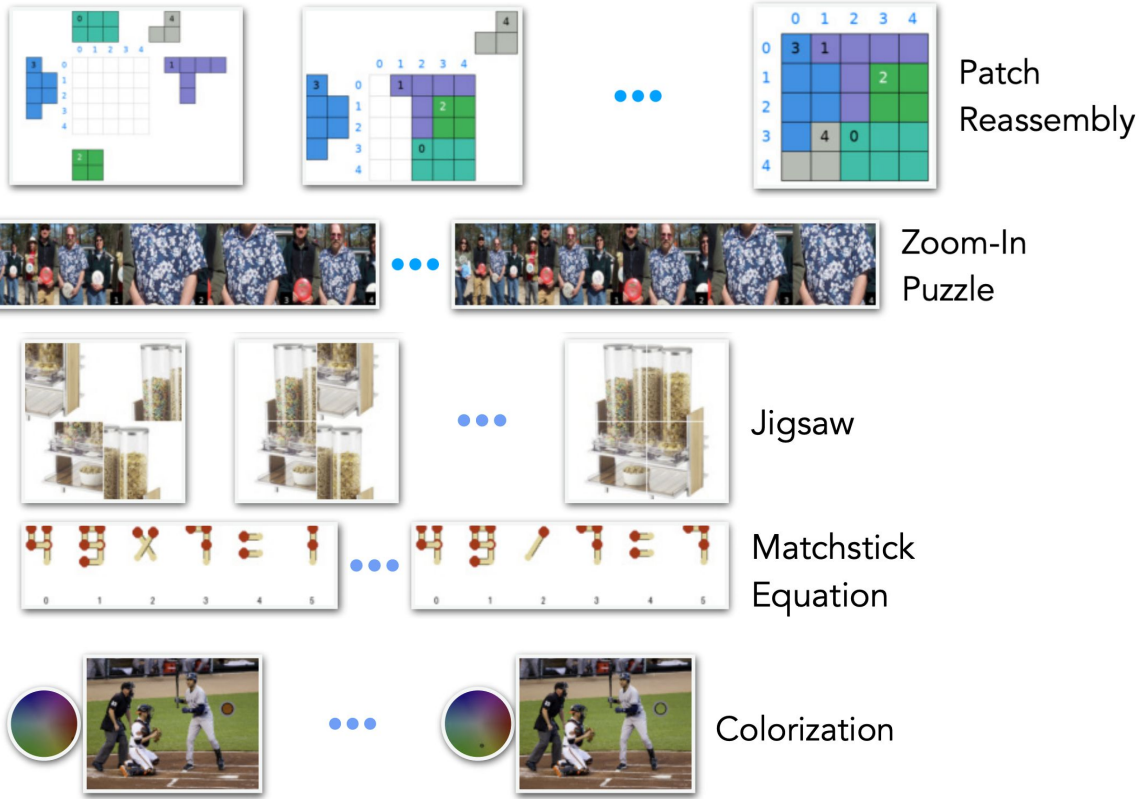


Do we need text feedback from environment?



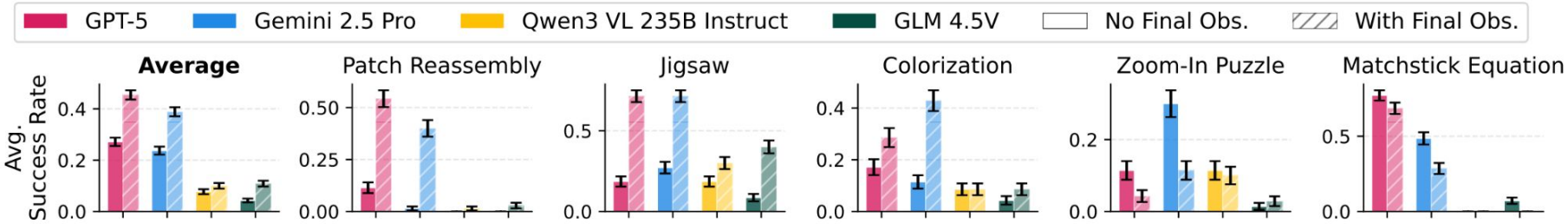
Models consistently improve if text-based feedback is provided. This indicates either a limited capability to perceive visual transitions or to infer what happened based on transitions, or both.

Do we tell what the final goal should look like?



Tasks that are easy to operate but hard to imagine the goal

Do we tell what the final goal should look like?



Models benefit from having the final goal observation, indicating limitations in visual reasoning.
However, perception issues can cause backfire, where performance degrades with model thinking that the initial observation looks the same as goal observation.

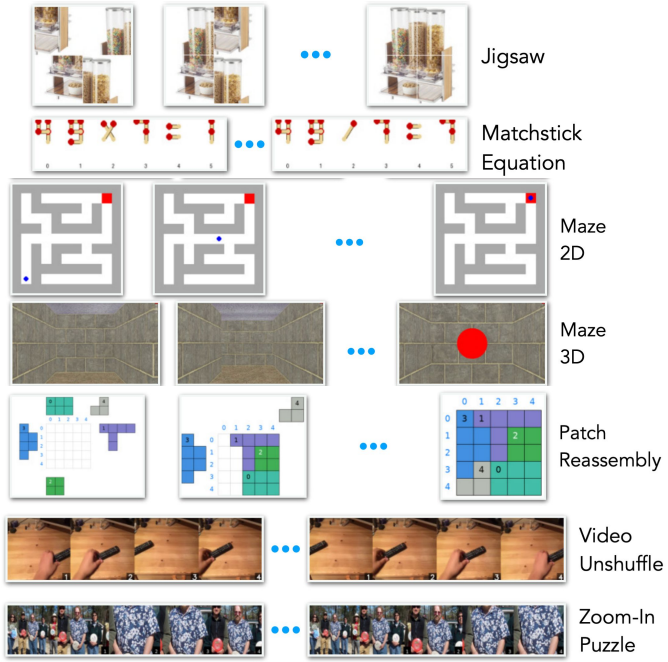
What does supervised finetuning tell us about

- generalization,
- different modules' contribution
- data curation strategies?

Supervised finetuning setup

- We use our **heuristic-based multi-step solver** to generate demonstrations for multi-step supervised finetuning.
- Two settings: **single-task** finetuning and **mixed-task** finetuning
- Qwen 2.5 VL Instruct 7b, full-parameter finetuning; bsz of 64 w/ 1500 and 5000 for single-task and mixed task finetuning respectively (~100K, ~350K demontstrations)

Is SFT alone sufficient for difficulty generalization?



Number of Pieces

Number of Moves

Maze Size

Maze Size

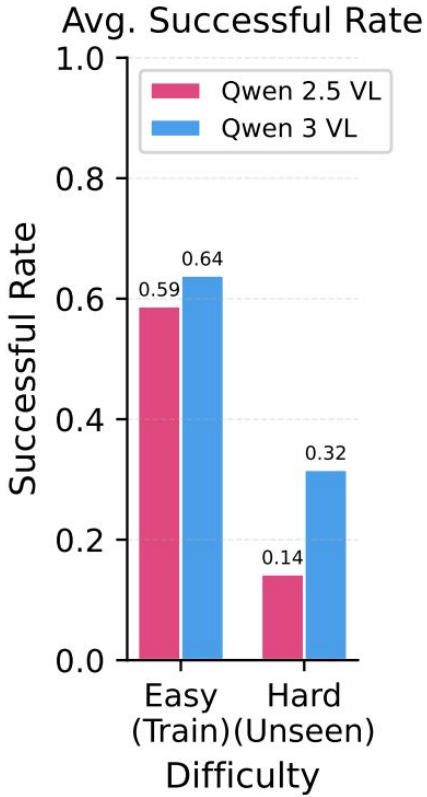
Patch & Board Size

Number of Pieces

Number of Pieces

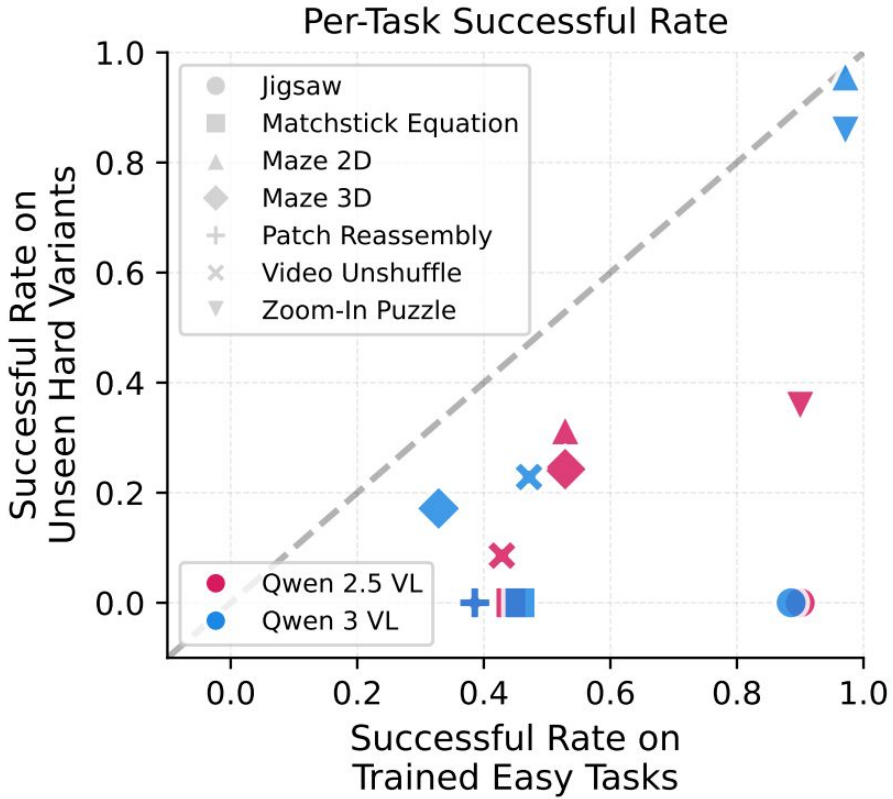
Train on easy; test on easy and hard

Is SFT alone sufficient for difficulty generalization?



Generalization emerges from SFT, and **stronger base model benefits more** from the same supervision.

Is SFT alone sufficient for difficulty generalization?



Generalization is **strongly data-dependent** more than model-dependent

What matters for multimodal multiturn capabilities?

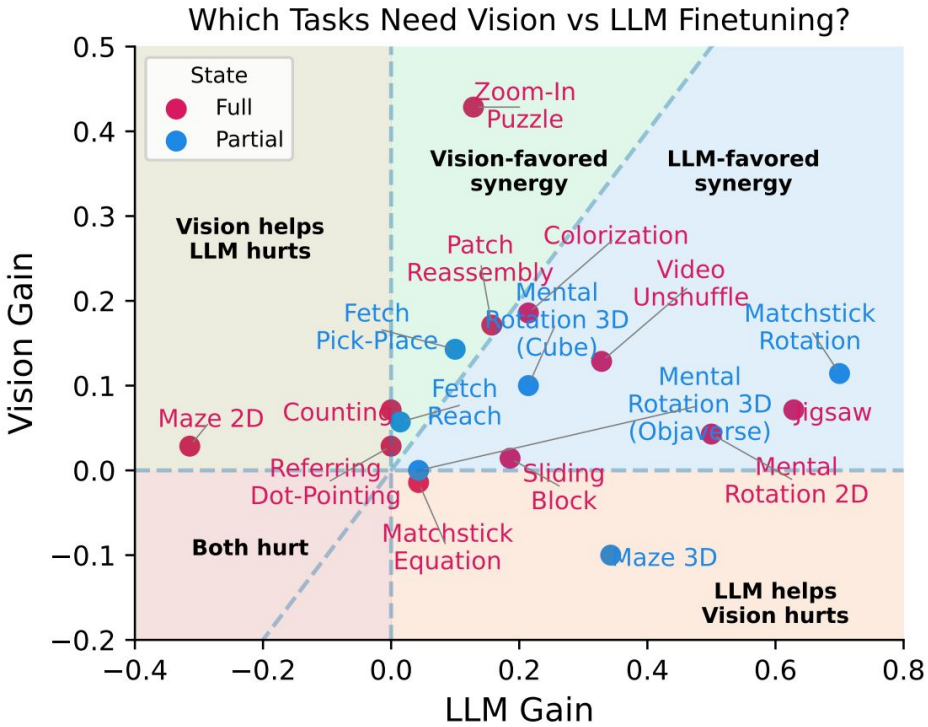
We hypothesize two key factors:

- **Visual encoder fidelity**: ability to capture fine-grained visual changes across turns (state transitions).
- **LLM temporal grounding**: ability to integrate and reason over past observations and actions.

What we did:

- Trained on mixed-task setting.
- Ablated learning by freezing the visual encoder (LLM-only gain) and freezing the LLM (vision-only gain).

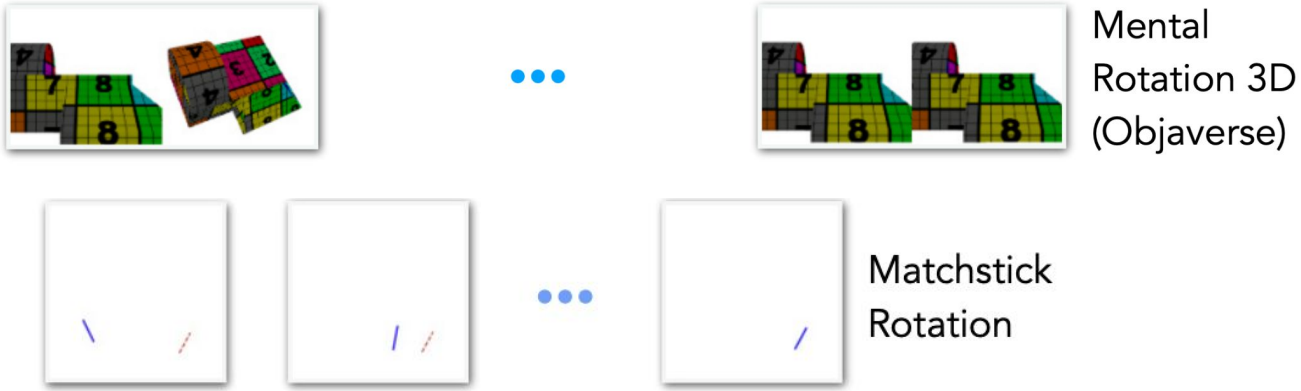
What matters for multimodal multiturn capabilities?



The vast majority of tasks benefit from finetuning both the visual encoder and the LLM.
Tasks where partial **observability and unknown dynamics** benefit more from finetuning **LLM**, and tasks where **fine-grained vision** is important benefits more from finetuning the **visual encoder**

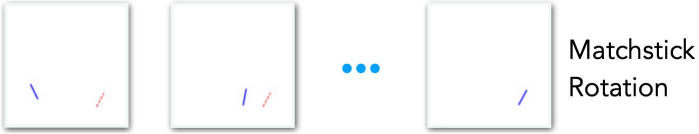
What do environments tell us about data curation?

Some environments have partial observability and unknown dynamics.

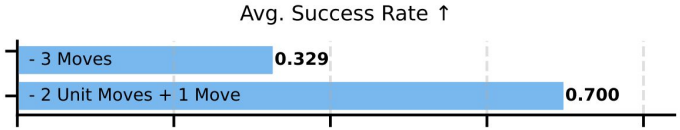


We can teach model to directly produce the steps to reach the goal, but is that ideal?

What do environments tell us about data curation?



Matchstick Rotation

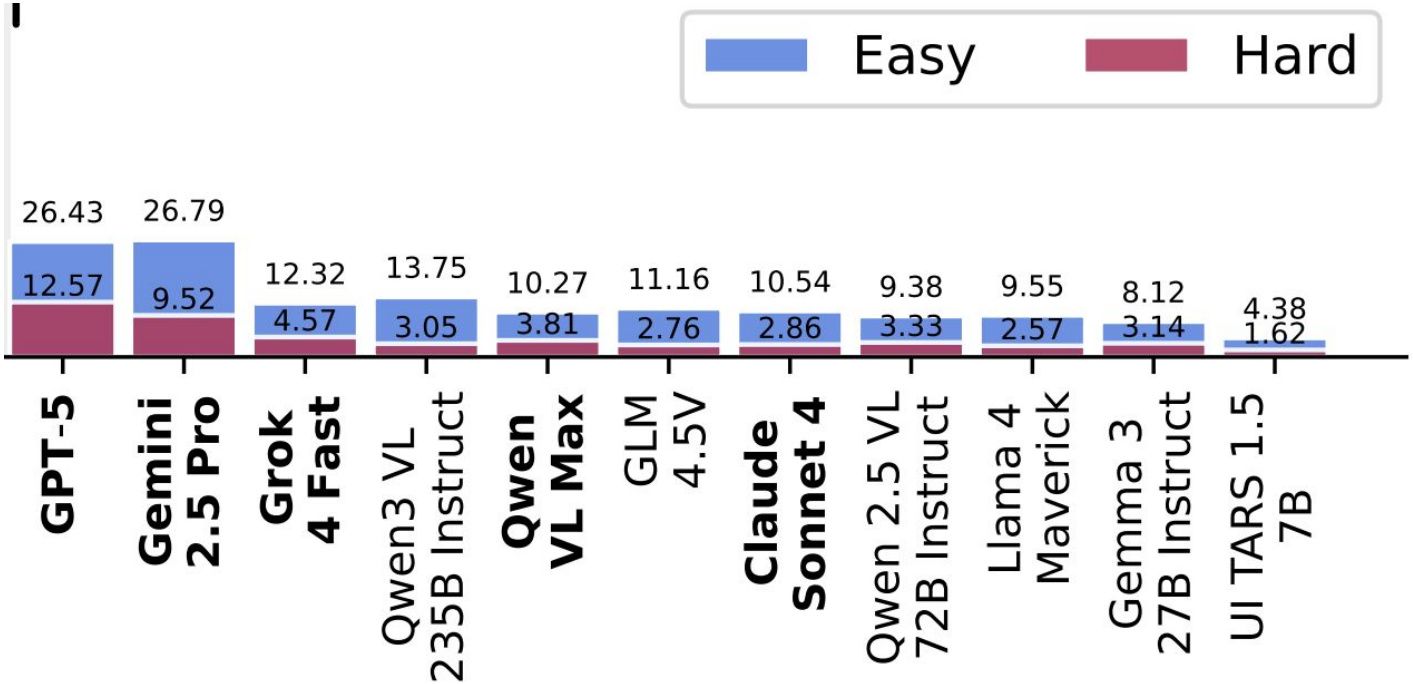


Mental Rotation 3D (Objaverse)

Directly teaching the model to produce the solution in POMDP is far from ideal, even with more data.

It is more important to teach the model to explore before exploiting, both for training learnability and inference competency.

How do frontier models perform on VisGym?



Thanks!
Questions?

Website: visgym.github.io

Email: zwcolin@eecs.berkeley.edu

VisGym: Diverse, Customizable, Scalable Environments for Multimodal Agents

Zirui Wang[†], **Junyi Zhang**[†], **Jiaxin Ge**[†], **Long Lian, Letian Fu, Lisa Dunlap,**
Ken Goldberg, Xudong Wang, Ion Stoica, David M. Chan, Sewon Min, Joseph E. Gonzalez

UC Berkeley [†]Equal contribution.

Modern Vision–Language Models (VLMs) remain poorly characterized in multi-step visual interactions, particularly in how they integrate perception, memory, and action over long horizons. We introduce VisGym, a gymnasium of 17 environments for evaluating and training VLMs. The suite spans symbolic puzzles, real-image understanding, navigation, and manipulation, and provides flexible controls over difficulty, input representation, planning horizon, and feedback. We also provide multi-step solvers that generate structured demonstrations, enabling supervised finetuning. Our evaluations show that all frontier models struggle in interactive settings, achieving low success rates in both the easy (26.8%) and hard (12.6%) configurations. Our experiments reveal notable limitations: models struggle to effectively leverage long context, performing worse with an unbounded history than with truncated windows. Furthermore, we find that several text-based symbolic tasks become substantially harder once rendered visually. However, explicit goal observations, textual feedback, and exploratory demonstrations in partially observable or unknown-dynamics settings for supervised finetuning yield consistent gains, highlighting concrete failure modes and pathways for improving multi-step visual decision-making. All the code, data, and models will be released.

Correspondence: {zwcolin,junyizhang,gejiaxin}@eecs.berkeley.edu

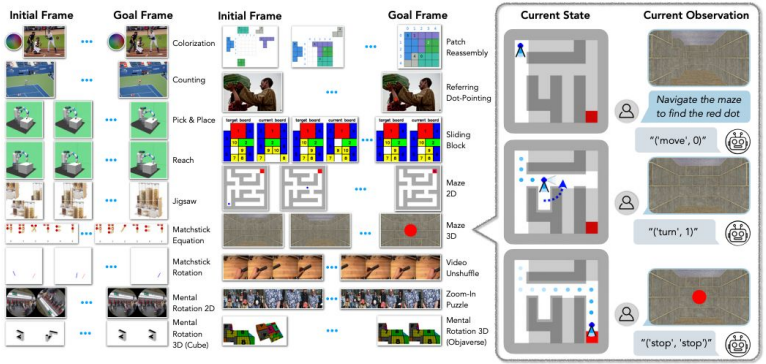


Figure 1. **An overview of VisGym.** (*Left*) VisGym consists of 17 diverse, long-horizon environments designed to systematically evaluate, diagnose, and train VLMs on visually interactive tasks with different domains, levels of state observability, and types of observations. (*Right*) An example trajectory for the Maze 3D navigation task illustrates a partially observable environment consisting of non-structured synthetic renderings. Here, a VLM is prompted with (1) the task description (*simplified in the figure*) and (2) a set of available actions to use (*not shown in the figure for simplicity*). The agent must select each action conditioned on both its past actions and observation history for its decision-making.