

# From Words to Actions\*: How can LLMs help robotics?

14/12/2023  
Shin Watanabe

# Recap from Kai's presentation: Foundation Models for robotics

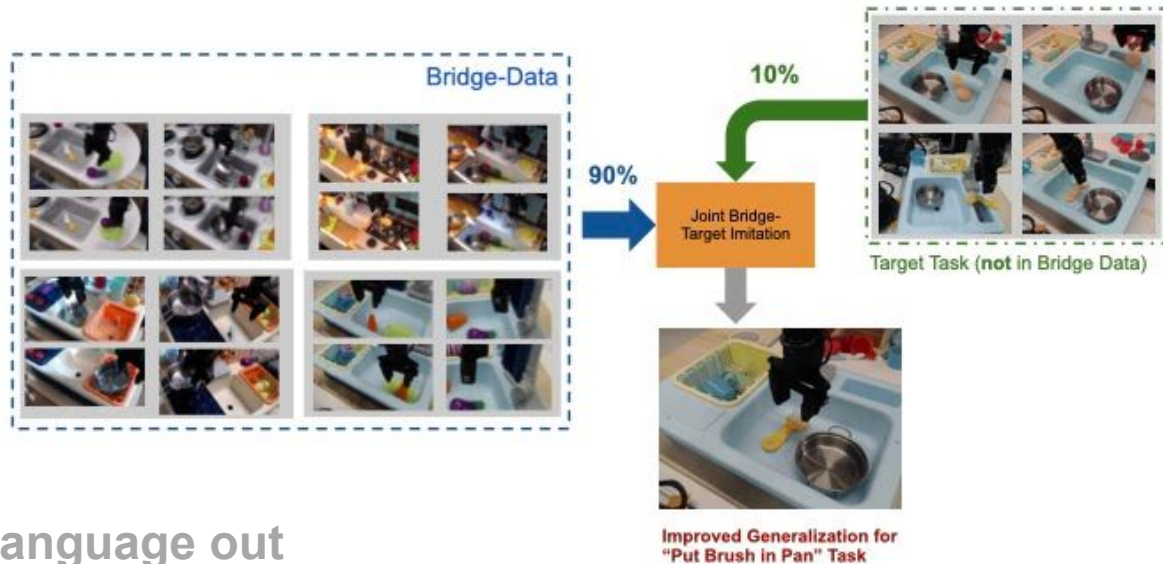
- Foundation models **pretrained** on diverse robot **demonstrations** & **fine-tuned** on different tasks show **skill generalization & transfer**

- Demo & tasks **share embodiment**

- Observations
- Actions
- Dynamics

- Output of the model is a low-level control policy

- LLMs are **language in, language out**



# The Big Question: What do LLMs “know” about robotics?

Do they know how to:

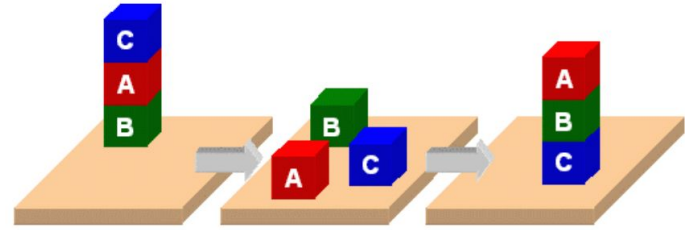
- Perform abstract reasoning?
  - c.f. classical AI planners (e.g. FastDownward)
- Perform geometric reasoning?
  - c.f. motion planners (e.g. RRT)
- Execute low-level actions in closed loop?
  - c.f. model-based control (e.g. MPC) /  
learned policies (e.g. RL, Behavioral Cloning)



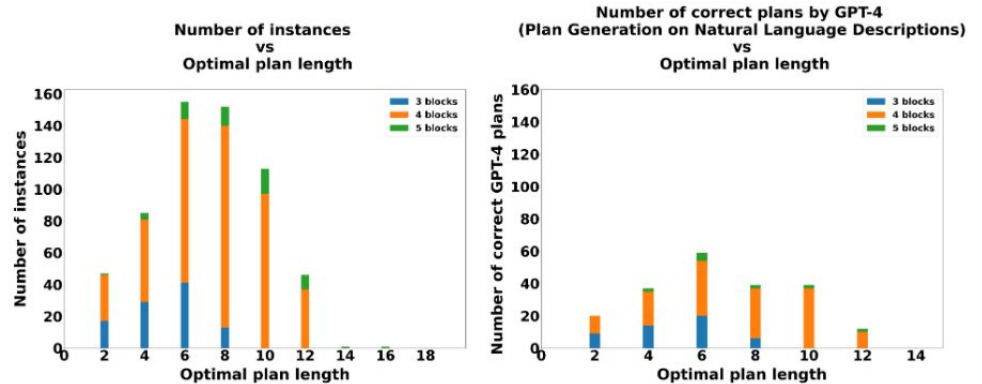


# LLMs perform poorly on planning benchmarks

- Blocksworld is a popular classical planning benchmark
- Considered an *easy* problem for standard planners
- GPT-4 succeeds only *a third* of the time
- Performance drops to **2%** (!) when the labels for actions and states are randomized (e.g. “stack” → “overcome”) → “looking up” examples instead of planning?



Blocksworld domain [UMBC]



# Do As I Can, Not As I Say: Grounding Language in Robotic Affordances

<sup>1</sup> Michael Ahn\*, Anthony Brohan\*, Noah Brown\*, Yevgen Chebotar\*, Omar Cortes\*, Byron David\*, Chelsea Finn\*, Chuyuan Fu\*, Keerthana Gopalakrishnan\*, Karol Hausman\*, Alex Herzog\*, Daniel Ho\*, Jasmine Hsu\*, Julian Ibarz\*, Brian Ichter\*, Alex Irpan\*, Eric Jang\*, Rosario Jauregui Ruano\*, Kyle Jeffrey\*, Sally Jesmonth\*, Nikhil J. Joshi\*, Ryan Julian\*, Dmitry Kalashnikov\*, Yuheng Kuang\*, Kuang-Huei Lee\*, Sergey Levine\*, Yao Lu\*, Linda Luu\*, Carolina Parada\*, Peter Pastor\*, Jornell Quiambao\*, Kanishka Rao\*, Jarek Rettinghouse\*, Diego Reyes\*, Pierre Sermanet\*, Nicolas Sievers\*, Clayton Tan\*, Alexander Toshev\*, Vincent Vanhoucke\*, Fei Xia\*, Ted Xiao\*, Peng Xu\*, Sichun Xu\*, Mengyuan Yan\*, Andy Zeng\*

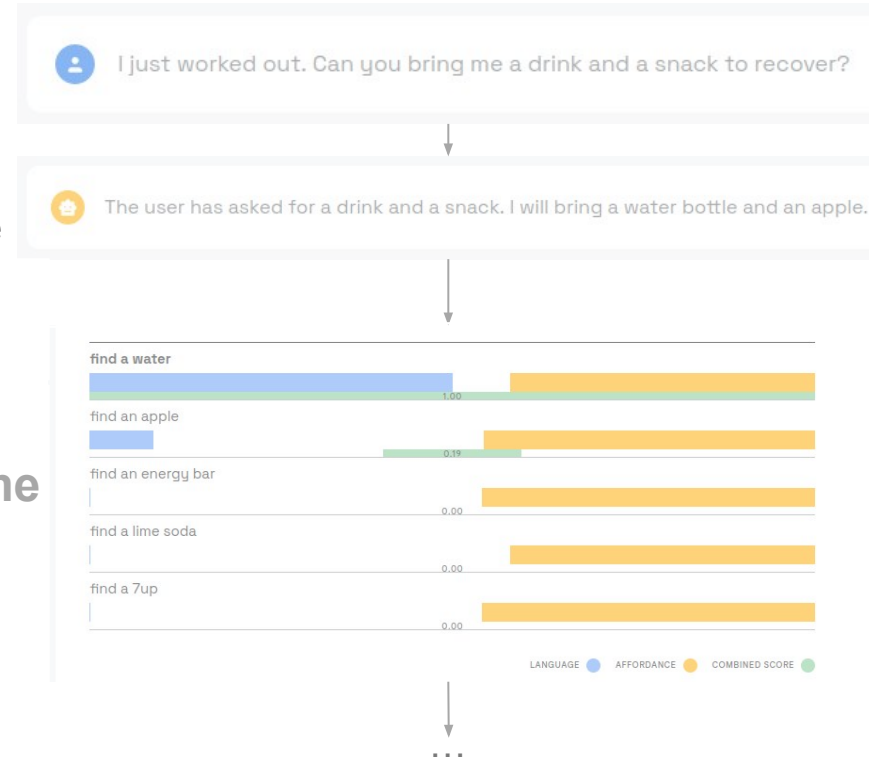
## Abstract Reasoning (Planning)

### SayCan (Arxiv 2022)

1. Assume a set of skills, equipped with:
  - a. A language label
  - b. Skill completion probability\* from current state
2. LLM generates a list of **tasks** from user instructions
3. Skills to be executed =  
Similarity between **skill label** & **task name**

\*

Skill completion probability



...

\*value function of the learned policy



# Inner Monologue: Embodied Reasoning through Planning with Language Models

Wenlong Huang<sup>†</sup>, Fei Xia<sup>†</sup>, Ted Xiao<sup>†</sup>, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Noah Brown, Tomas Jackson, Linda Luu, Sergey Levine, Karol Hausman, Brian Ichter

## Geometric Reasoning

### Inner Monologue (CoRL 2023)

- SayCan + Task success feedback from scene descriptor\* = replanning!

### Socratic Models (Arxiv 2022)

- Combine LLM with:
  1. **vision-language model\*\***
  2. **language-conditioned skills\*\*\***
- All modules communicate via language

\* MDETR (ICCV 2021)

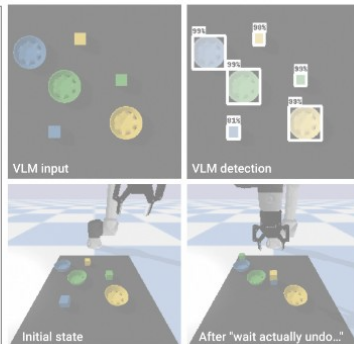
\*\* ViLD (ICLR 2021)

\*\*\* CLIPort (CoRL 2022)

### Socratic Models: Composing Zero-Shot Multimodal Reasoning with Language

Andy Zeng, Maria Attarian, Brian Ichter, Krzysztof Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aavek Purohit, Michael S. Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, Pete Florence

```
objects = ["green block", "blue block", "yellow block", "green bowl", "blue bowl", "yellow bowl"]
# move all the blocks to different corners.
Step 1. robot.pick_and_place("green block", "top left corner")
Step 2. robot.pick_and_place("blue block", "top right corner")
Step 3. robot.pick_and_place("yellow block", "bottom left corner")
# now move the blue block to the middle.
Step 1. robot.pick_and_place("blue block", "middle")
# stack the blocks on top of each other.
Step 1. robot.pick_and_place("yellow block", "blue block")
Step 2. robot.pick_and_place("green block", "yellow block")
# wait actually undo that last step.
Step 1. robot.pick_and_place("green block", "top left corner")
# put the yellow block in the bowl you think it best fits.
Step 1. robot.pick_and_place("yellow block", "yellow bowl")
# ok now sort the remaining blocks in the same way.
Step 1. robot.pick_and_place("blue block", "blue bowl")
Step 2. robot.pick_and_place("green block", "green bowl")
```



Video of results: <https://innermonologue.github.io/>  
<https://socraticmodels.github.io/>

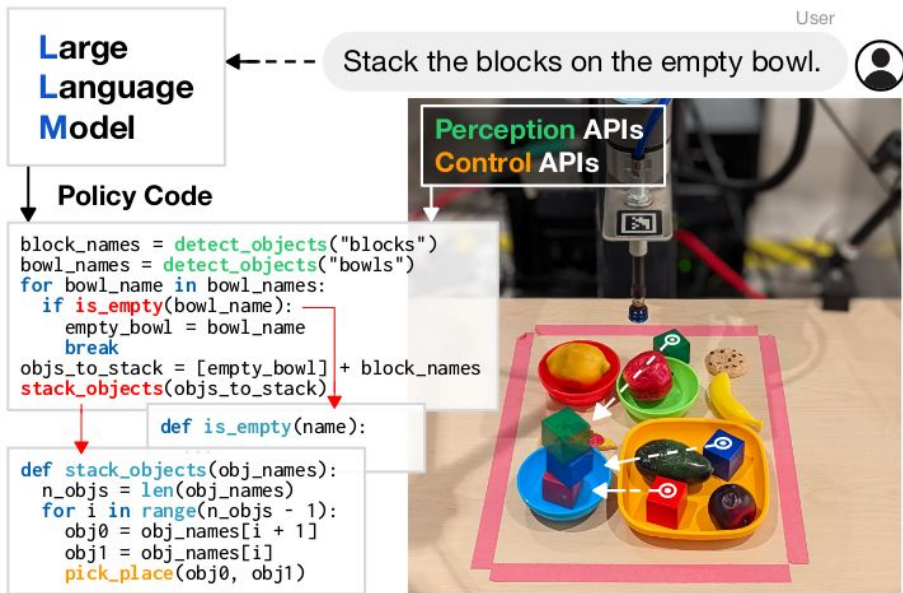
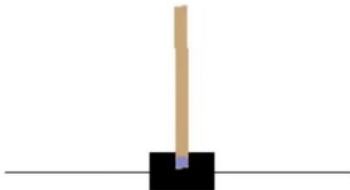




## Skill Execution

### Code as Policies (ICRA 2023)

- Have the LLM *program* a controller
- Can code up low-level controllers from scratch (e.g. cartpole)
- Still relies on perception & control modules






Video of results: <https://code-as-policies.github.io>

# The Big Question: What do LLMs “know” about robotics?

Do they know how to:

Kind of, **but**:

- Perform abstract reasoning?  With pretrained skills
  - e.g. classical AI planners (e.g. FastDownward)
- Perform geometric reasoning?  With perception modules
  - e.g. motion planners (e.g. RRT)
- Execute low-level actions in closed loop?  With examples (instructions/code)
  - e.g. model-based control (e.g. MPC)

As long as [Planning, Perception, Control] can be expressed in language (or code),  
then LLMs can help out!

## Limitations

or

## Advantages?



- LLM-based plans have no formal guarantees of task success
- LLMs don't seem to do compositional reasoning (Dziri et al. 2023 [10])



- Grounding needs external modules to be conditioned on language



- Do we want unverified code running on robots?

- User-friendly (dealing with ambiguities inherent in language)

- Language as a common interface (e.g. socratic models)

- Leveraging large datasets / pretrained models  
→ accelerate research?

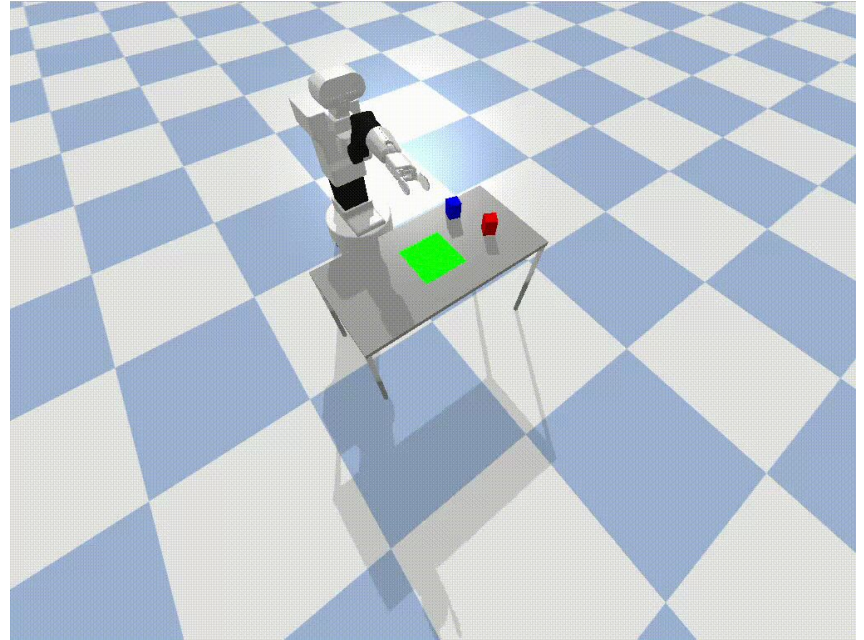
# How does this affect my research?

I use a combination of:

- Classical planning
- Motion planning
- Low-level controllers

Each of which can be replaced / augmented by language-based models

## Should I?



TIAGo using task planning, motion planning and a learned push skill [11]

# References

- [1] M. Ahn *et al.*, “Do As I Can, Not As I Say: Grounding Language in Robotic Affordances,” p. 34, Aug. 2022. doi: [10.48550/arXiv.2204.01691](https://doi.org/10.48550/arXiv.2204.01691).
- [2] A. Zeng *et al.*, “Socratic Models: Composing Zero-Shot Multimodal Reasoning with Language.” arXiv, May 27, 2022. doi: [10.48550/arXiv.2204.00598](https://doi.org/10.48550/arXiv.2204.00598).
- [3] W. Huang *et al.*, “Inner Monologue: Embodied Reasoning through Planning with Language Models.” arXiv, Jul. 12, 2022. doi: [10.48550/arXiv.2207.05608](https://doi.org/10.48550/arXiv.2207.05608).
- [4] J. Liang *et al.*, “Code as Policies: Language Model Programs for Embodied Control.” arXiv, Sep. 19, 2022. doi: [10.48550/arXiv.2209.07753](https://doi.org/10.48550/arXiv.2209.07753).
- [5] N. Dziri *et al.*, “Faith and Fate: Limits of Transformers on Compositionality.” arXiv, Oct. 31, 2023. Accessed: Nov. 12, 2023. [Online]. Available: <http://arxiv.org/abs/2305.18654>
- [6] M. Shridhar *et al.*, “CLIPort: What and Where Pathways for Robotic Manipulation,” in *Proceedings of the 5th Conference on Robot Learning*, PMLR, Jan. 2022, pp. 894–906. Accessed: Feb. 27, 2023. [Online]. Available: <https://proceedings.mlr.press/v164/shridhar22a.html>
- [7] Gu, Xiuye, et al. "Open-vocabulary object detection via vision and language knowledge distillation." *arXiv preprint arXiv:2104.13921* (2021).
- [8] Kamath, Aishwarya, et al. "Mdetr-modulated detection for end-to-end multi-modal understanding." *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021.
- [9] K. Valmeekam, M. Marquez, A. Olmo, S. Sreedharan, and S. Kambhampati, “PlanBench: An Extensible Benchmark for Evaluating Large Language Models on Planning and Reasoning about Change.” arXiv, Nov. 25, 2023. doi: 10.48550/arXiv.2206.10498.
- [10] N. Dziri et al., “Faith and Fate: Limits of Transformers on Compositionality.” arXiv, Oct. 31, 2023. Accessed: Nov. 12, 2023. [Online]. Available: <http://arxiv.org/abs/2305.18654>
- [11] S. Watanabe, G. Horn, J. Tørresen, and K. O. Ellefsen, “Offline Skill Generalization via Task and Motion Planning.” arXiv, Nov. 24, 2023. doi: 10.48550/arXiv.2311.14328.