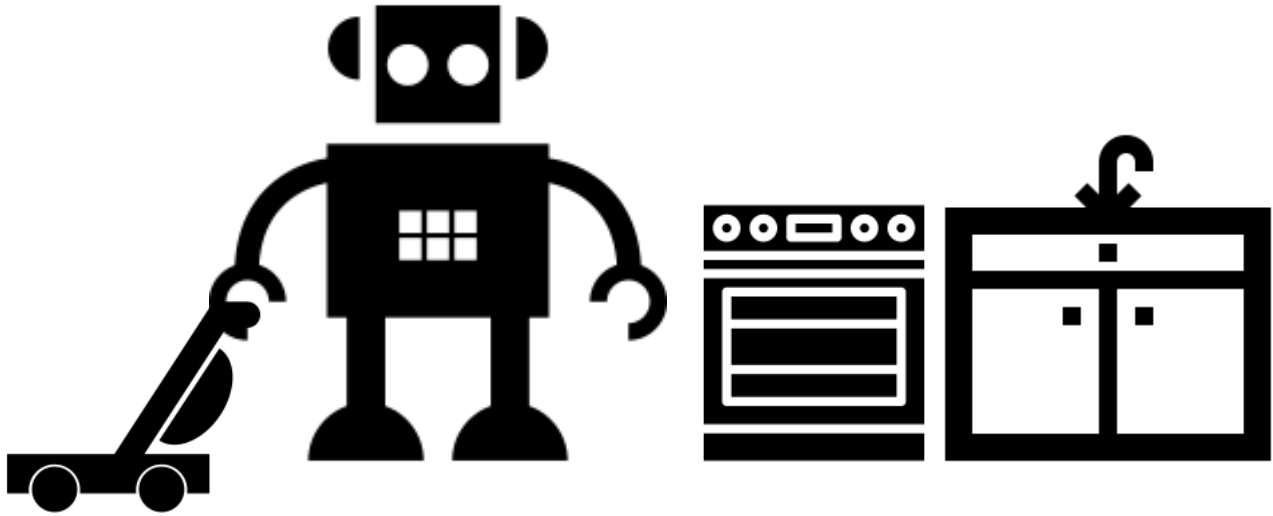
Building Robots that Understand their Worlds

Tomas Lozano-Perez, MIT
joint work with Leslie Pack Kaelbling
and LIS

Research goal:

understand the computational mechanisms
necessary to make
a general-purpose intelligent robot

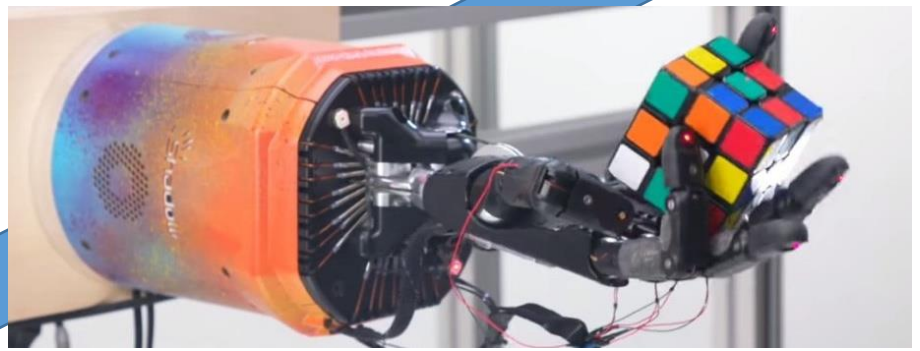


The quest for generality in robotics

(1974-2024)

an idiosyncratic view





IDEAS

History Is a Pendulum Not an Arc

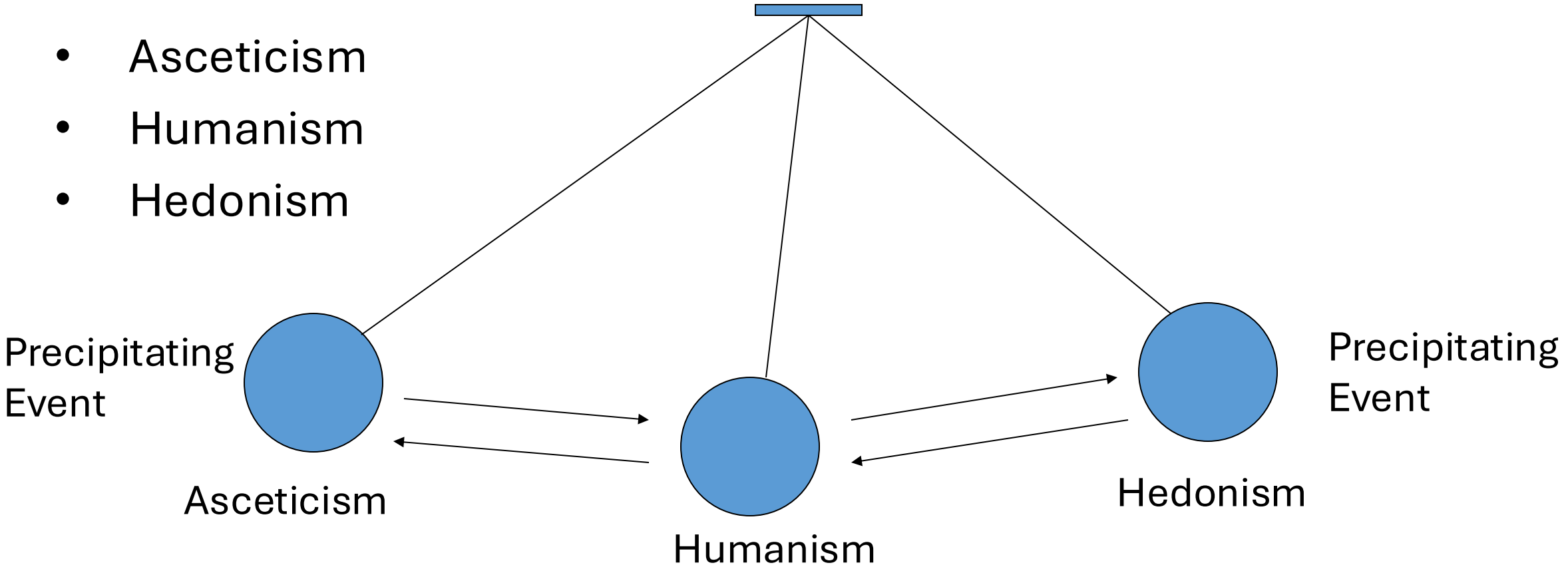
The Pendulum Theory of Art and Politics

January 20, 2017 | Artists/Creativity

Pendulum: How Past Generations Shape Our Present and Predict Our Future Hardcover – October 2, 2012

The Three Positions

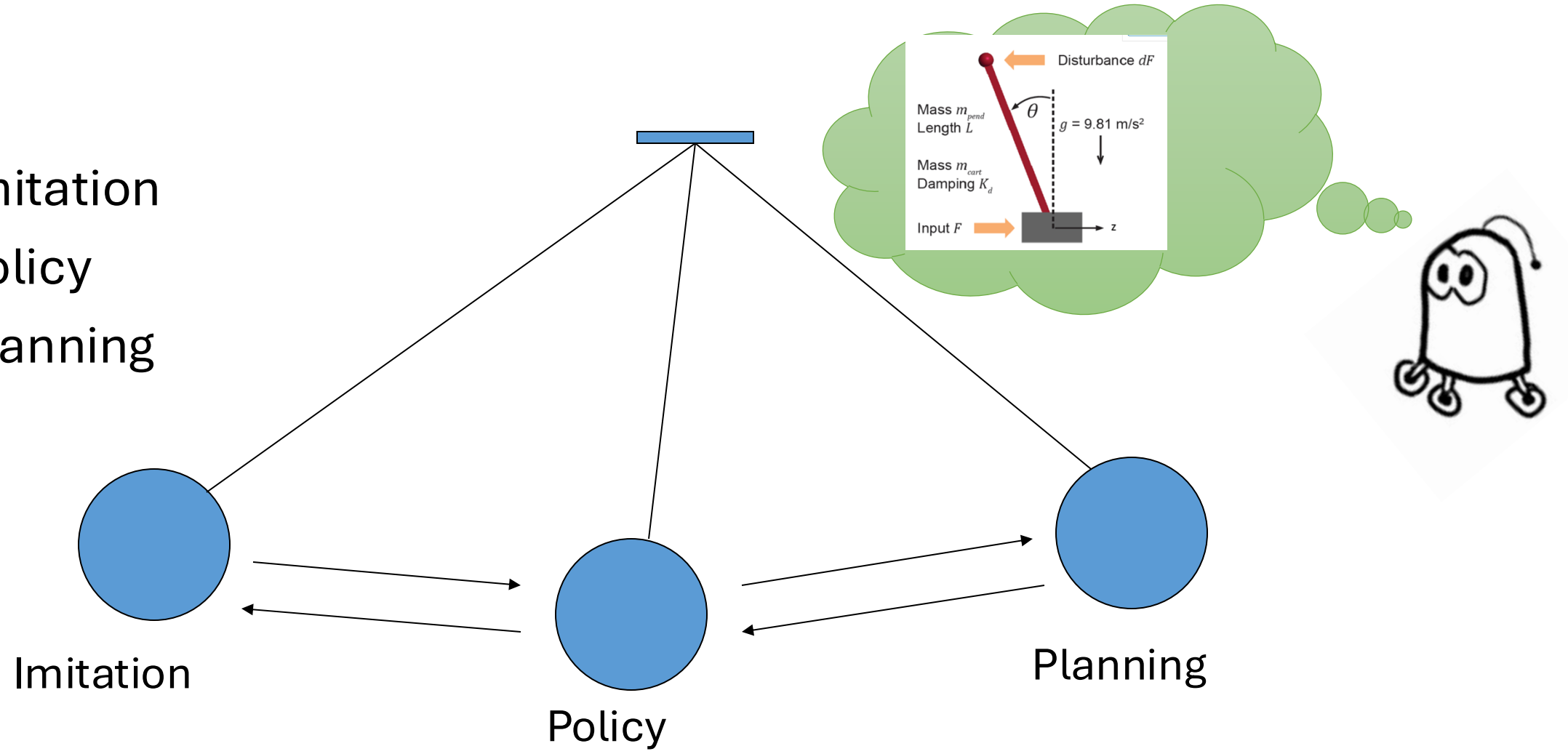
- Asceticism
- Humanism
- Hedonism



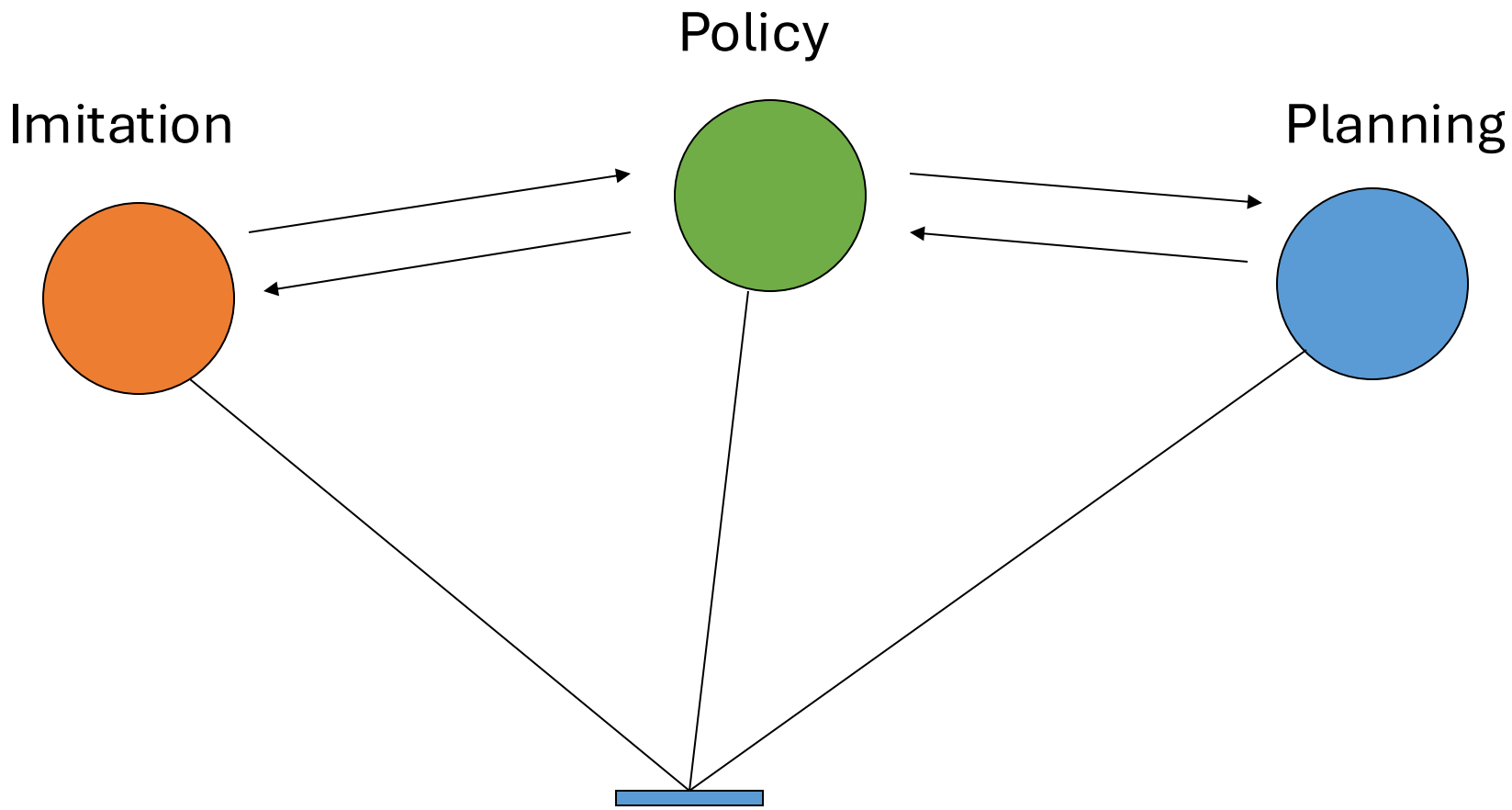
The Lakeview School District is a school district in suburban Battle Creek, Michigan, United States. I

The pendulum theory of robotics

- Imitation
- Policy
- Planning



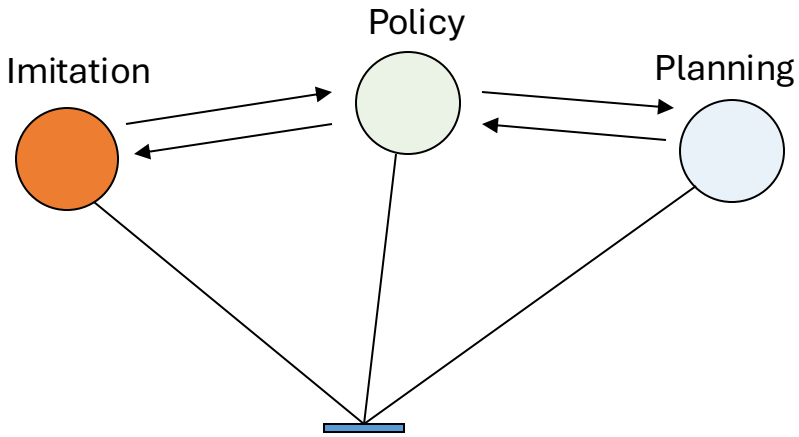
The (inverted) pendulum theory of robotics



Unimate (1961) – The first “industrial” robot

George DeVol





Basically, a tape recorder!

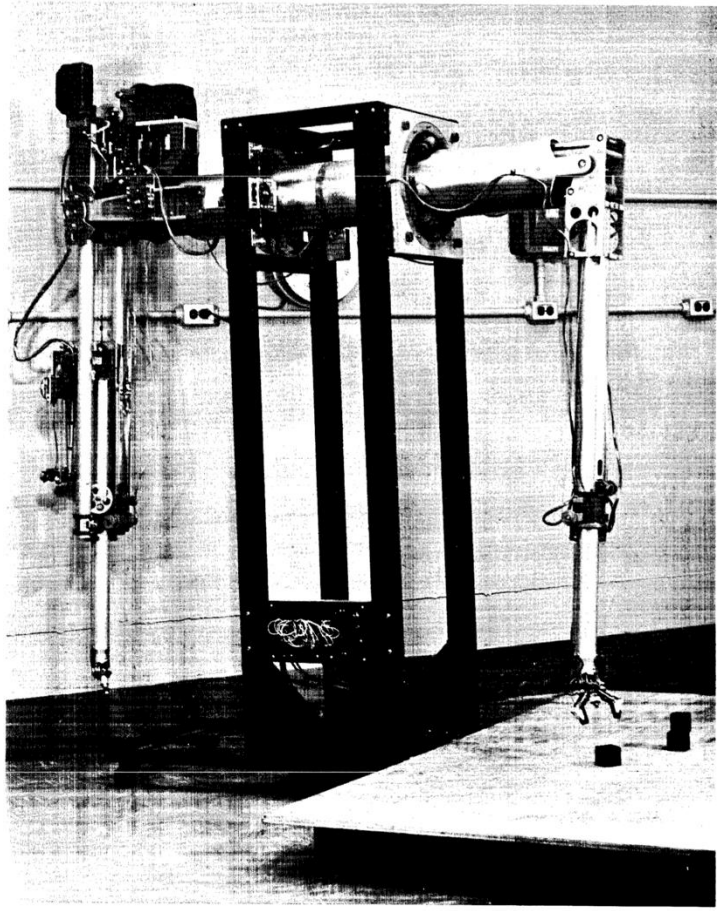


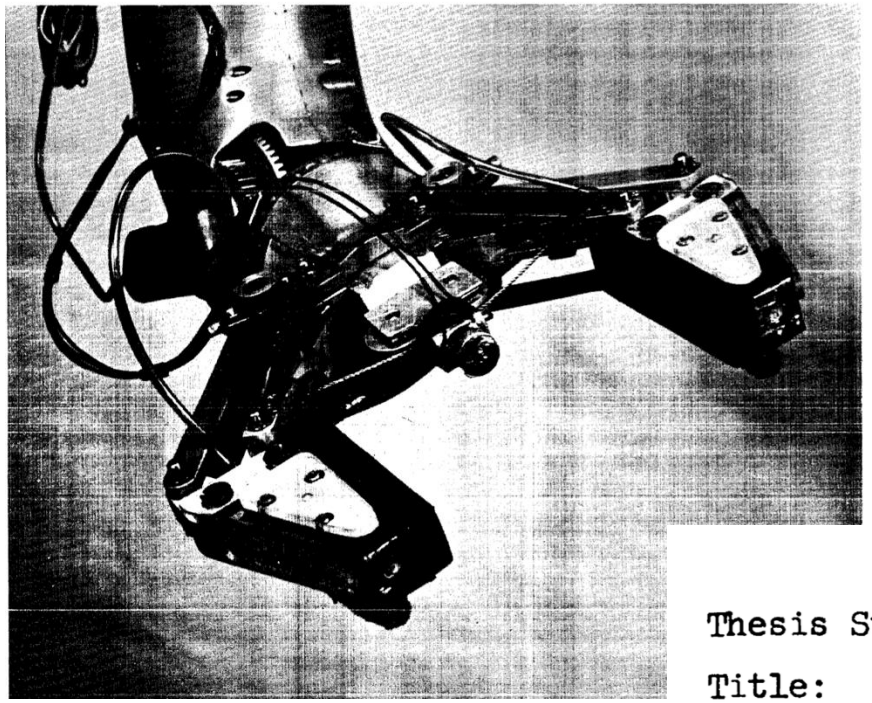


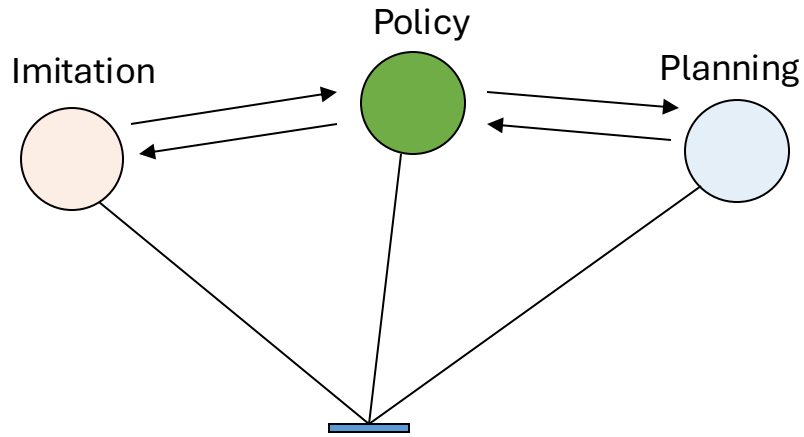
Autonomous
(not teleop)

MH-1 (1961) – The first computer-controlled manipulator

H. Ernst (MIT)







Thesis Supervisor: Claude E. Shannon

Title: Donner Professor of Sc[illegible]e

A sensory-motor policy

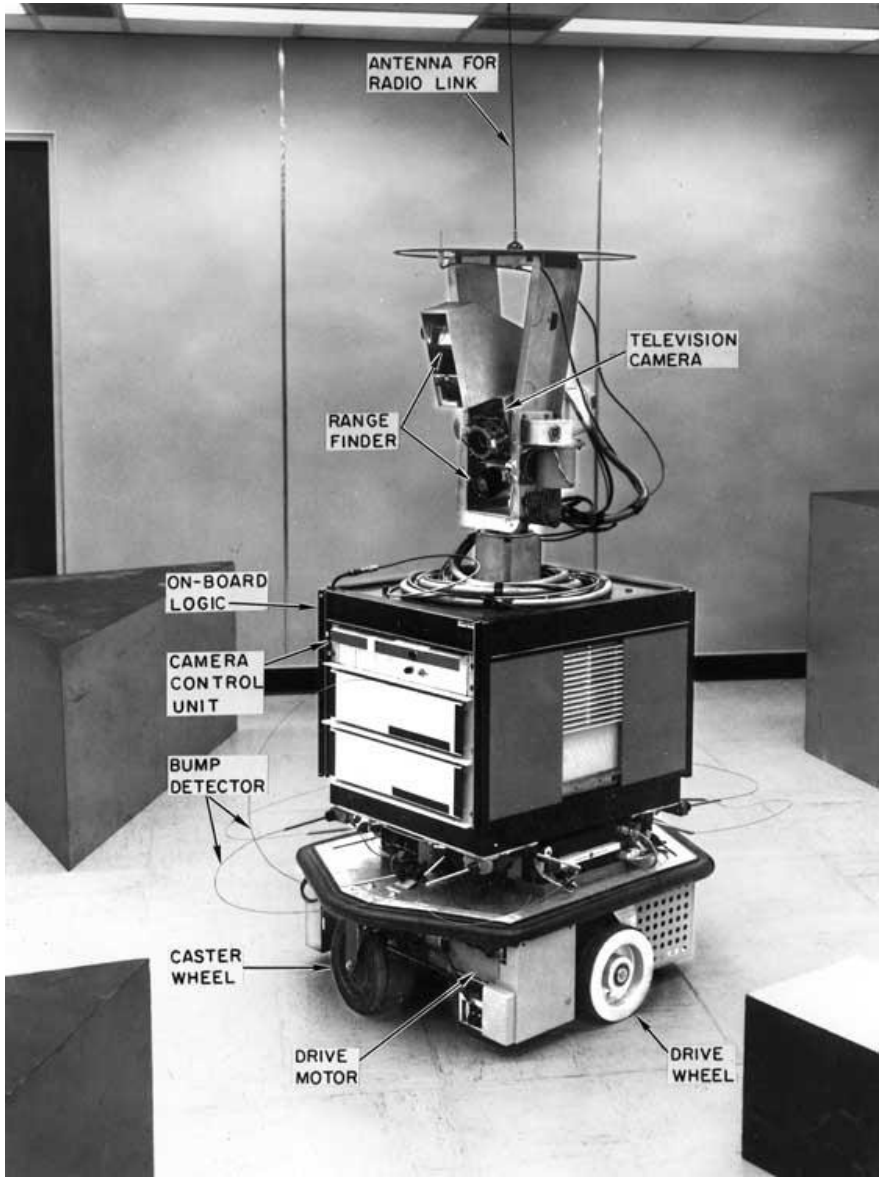


MH-1 Sensory-motor policy

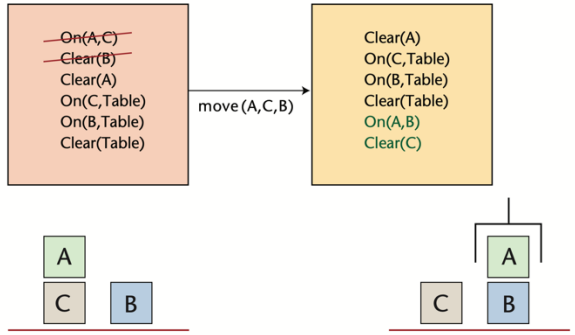


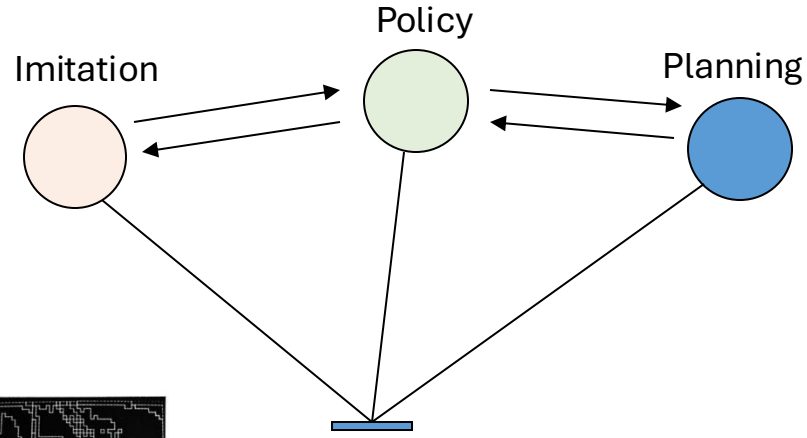
Shakey (1966 – 1972) - The first “intelligent” robot

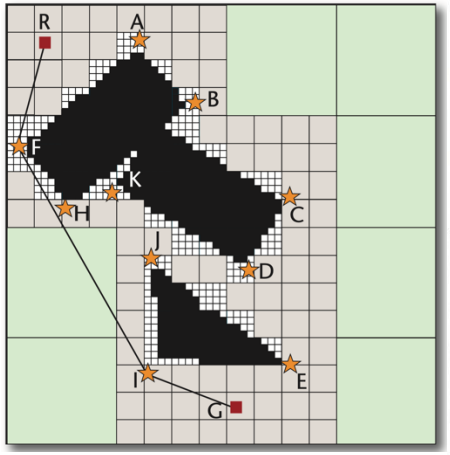
Nilsson et al. (SRI)

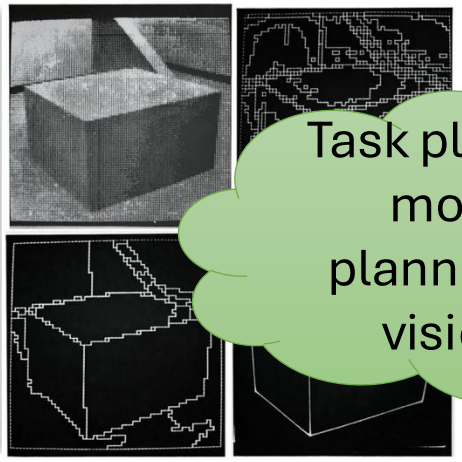


Preconditions: On(A,C), Clear(A), Clear(B)
Delete List: (On(A,C), Clear(B)
Add List: On(A,B), Clear (C)

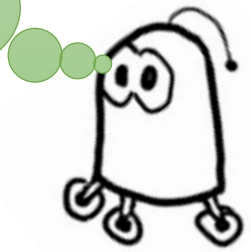




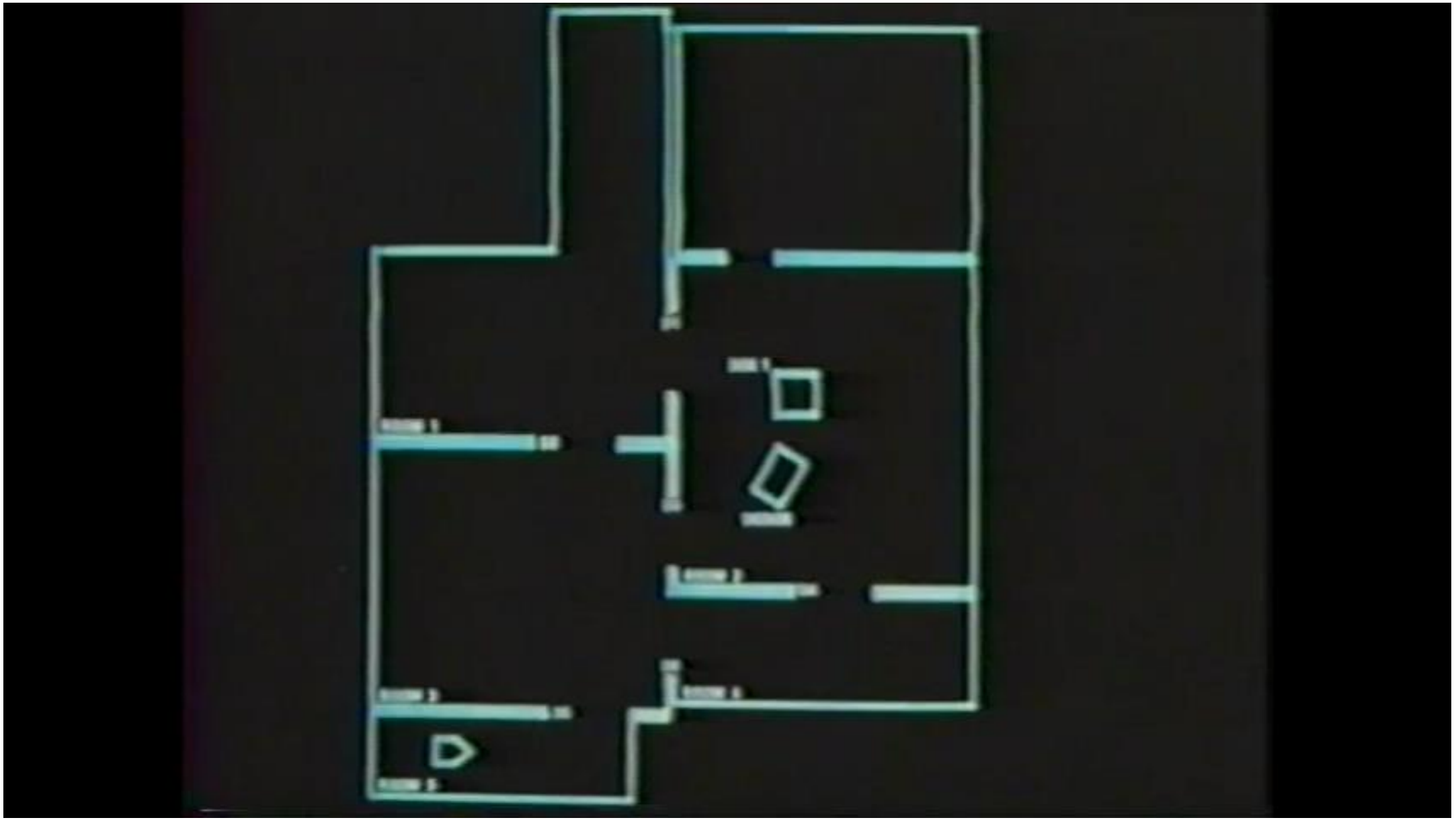




Task planning, motion planning, A*, vision…



Shakey - Plan, Perceive, Execute, Perceive, Re-Plan



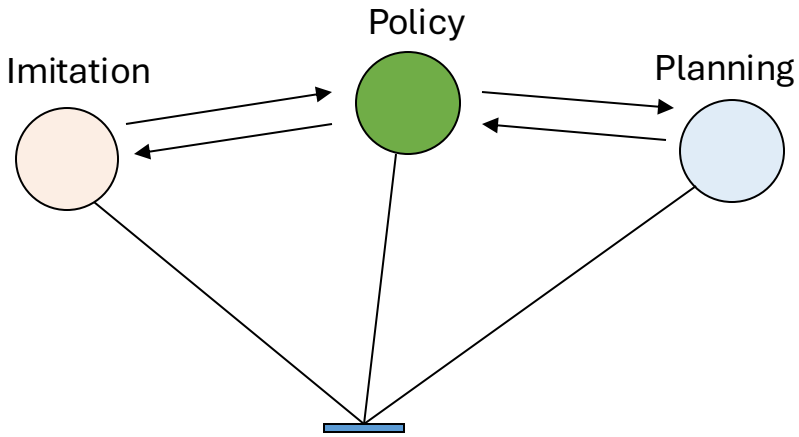
Robot programming languages (1970s, 1980s)

```
PICKUP: SUBR (PART_DATA, TRIES);
  MOVE(GRIPPER, DIAMETER(PART_DATA)+0.2);
  MOVE(<1, 2, 3>, XYZ_POSITION(PART_DATA)+<0, 0, 1>);
  TRY_PICKUP(PART_DATA, TRIES);
  END;

TRY_PICKUP: SUBR(PART_DATA, TRIES);
  IF TRIES LT 1 THEN RETURN('NO PART');
  DMOVE(3, -1.0);
  IF GRASP(DIAMETER(PART_DATA)) = 'NO PART'
    THEN TRY_PICKUP(PART_DATA, TRIES - 1);
  END;

GRASP: SUBR(DIAMETER, F);
  FMONS: NEW APPLY($MONITOR, PINCH_FORCE(F));
  MOVE(GRIPPER, 0, FMONS);
  RETURN ( IF QPOSITION(GRIPPER) LE DIAMETER/2
             THEN 'NO PART'
             ELSE 'PART' );
  END;

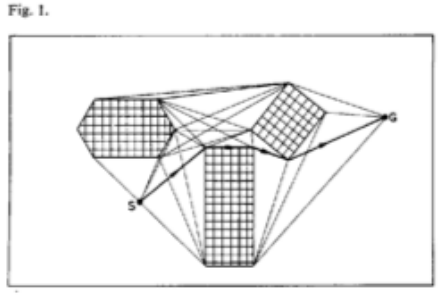
INSERT: SUBR(PART_DATA, HOLE);
  FMONS: NEW APPLY($MONITOR,
                   TIP_FORCE(LANDING-FORCE));
```

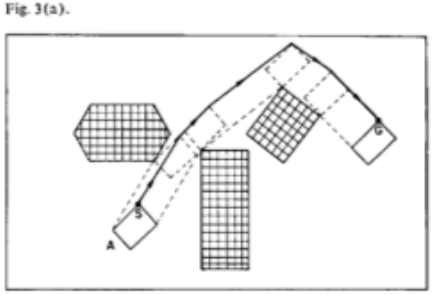


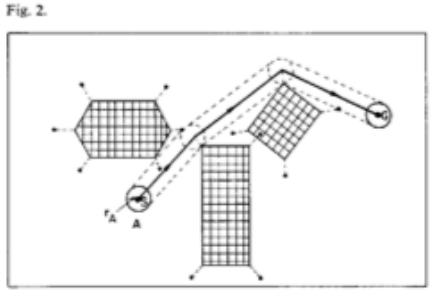
TLP “Robot Programming”, 1983

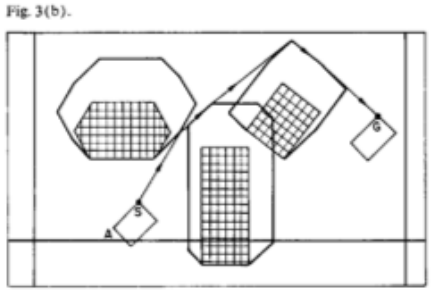
Configuration Space Motion Planning (1980s)

Lozano-Perez et al (MIT)

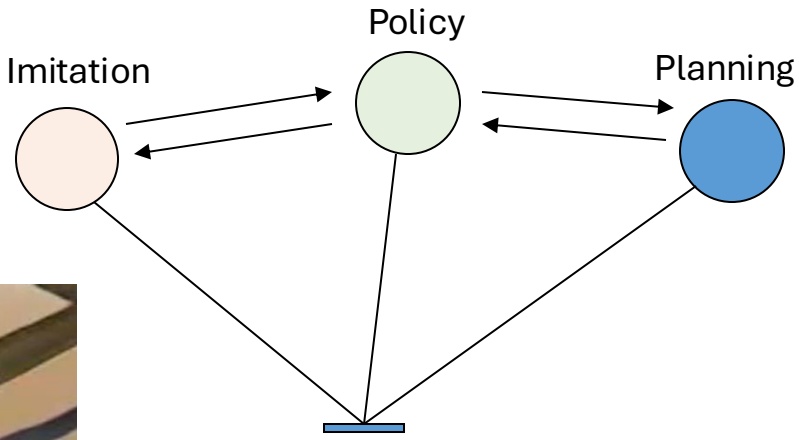






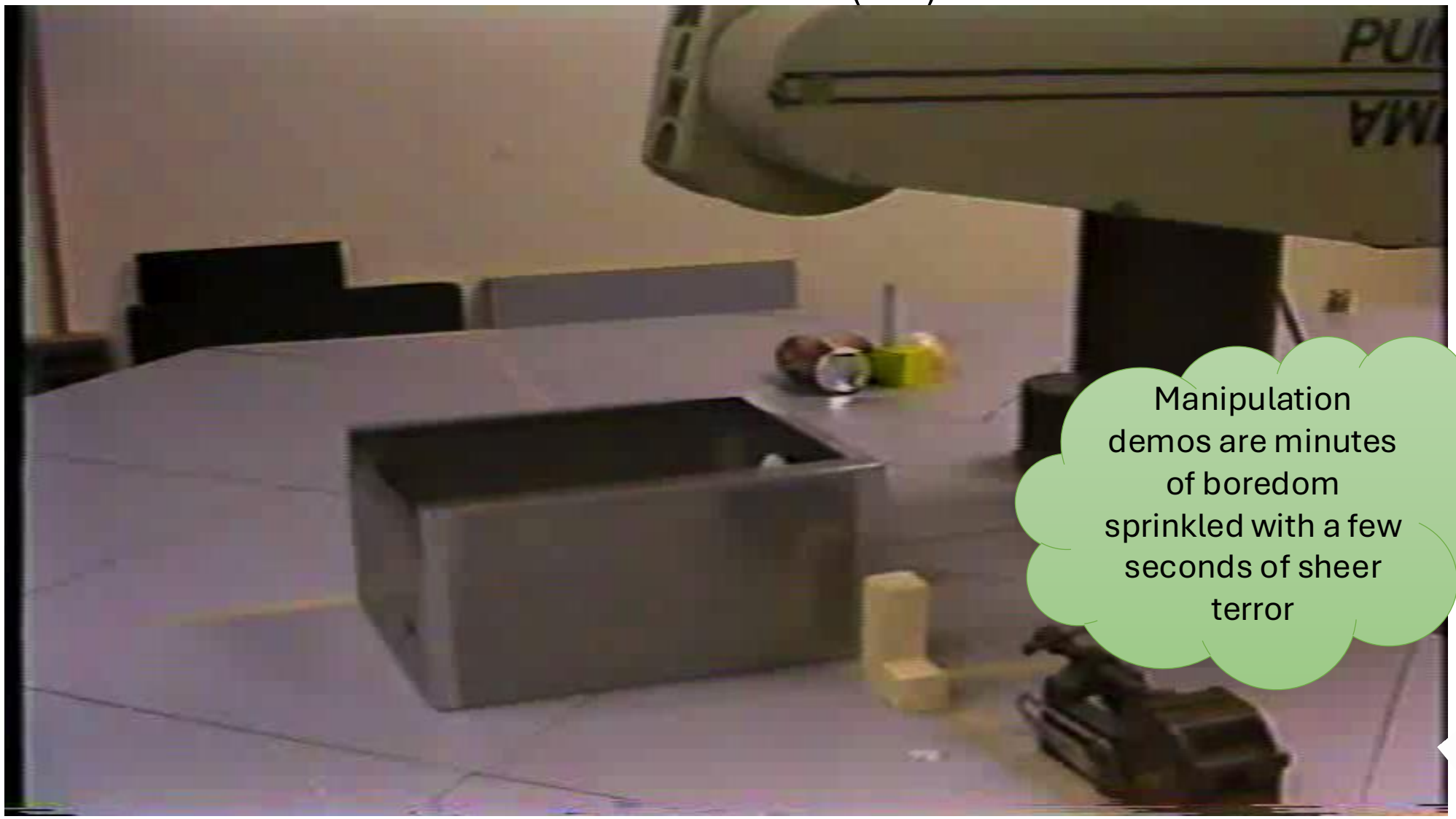






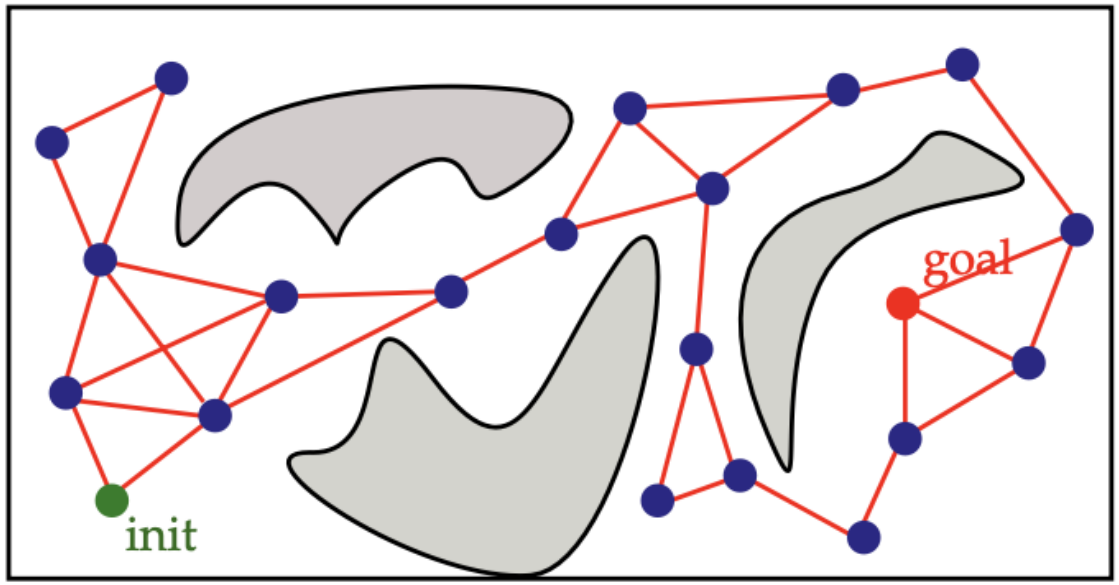
C-Space Motion Planning (1980s)

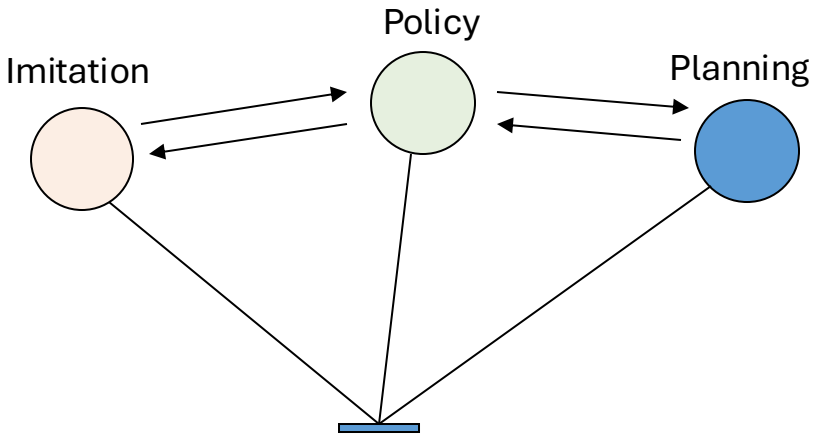
Lozano-Perez et al. (MIT)

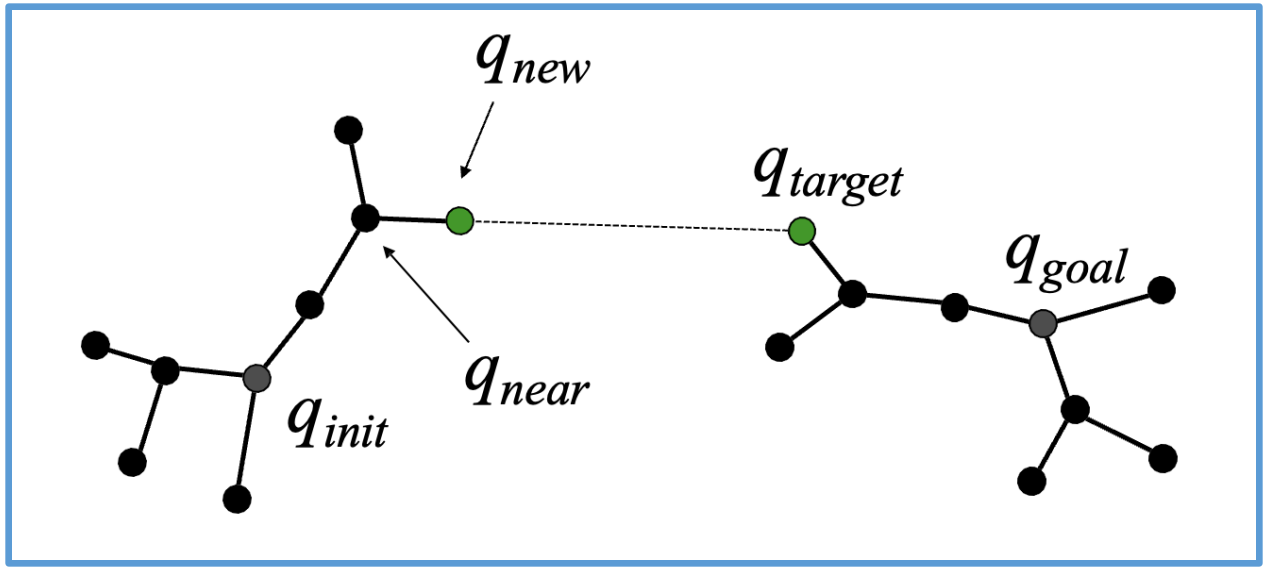


Sampling-Based Motion Planning: PRM, RRT (1996 -)

Kavraki, Latombe, Lavalle, Kuffner et al

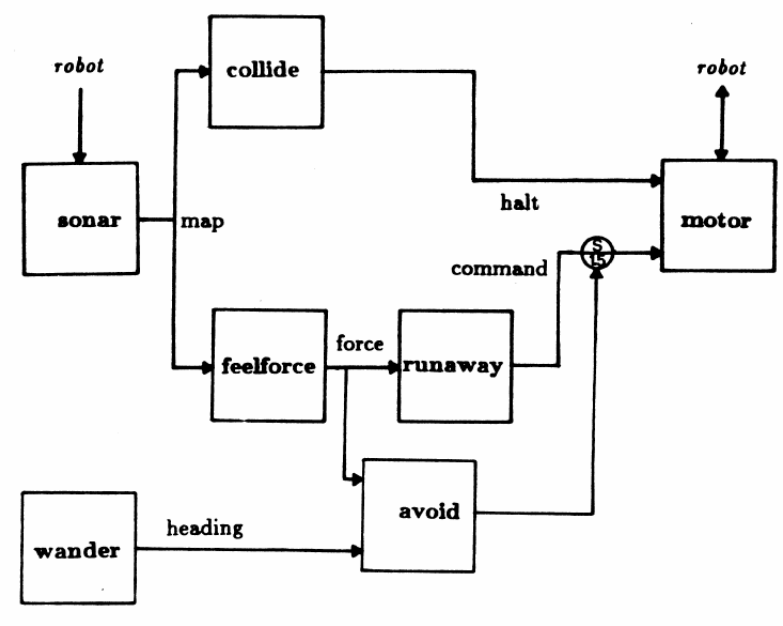






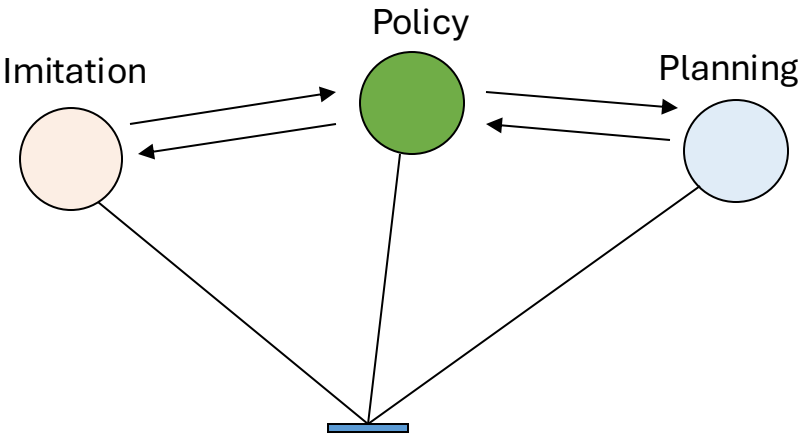
Behavior-based robotics (1990s)

Brooks (MIT, IRobot)









Probabilistic planning: POMDP (2000s)

Sondik, Kaelbling, Littman, Cassandra, et al.

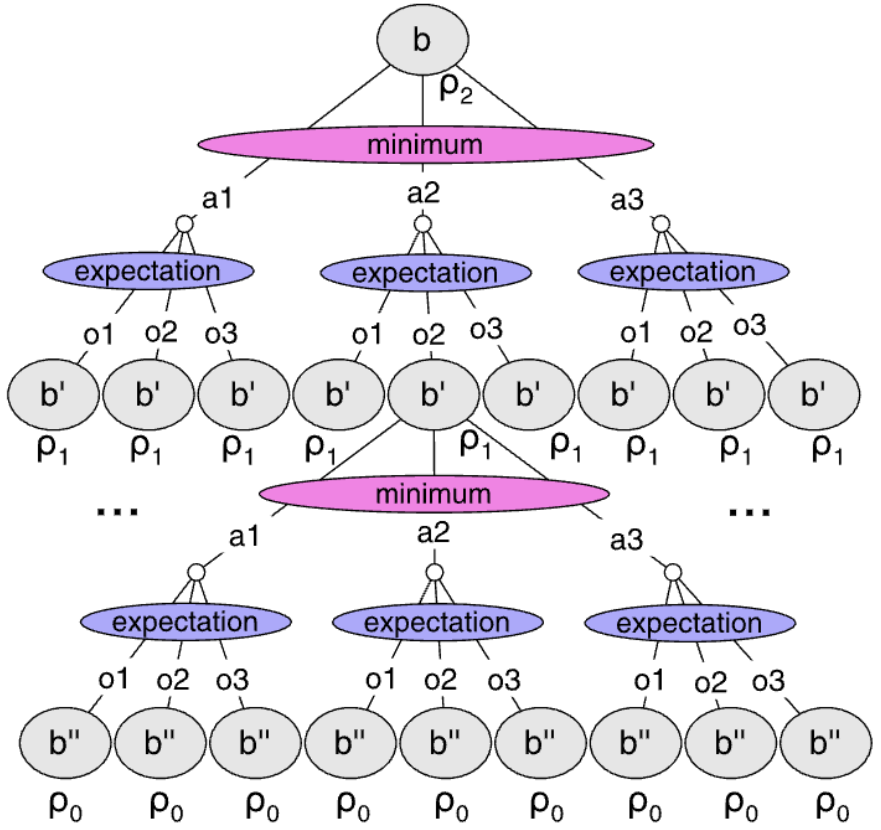
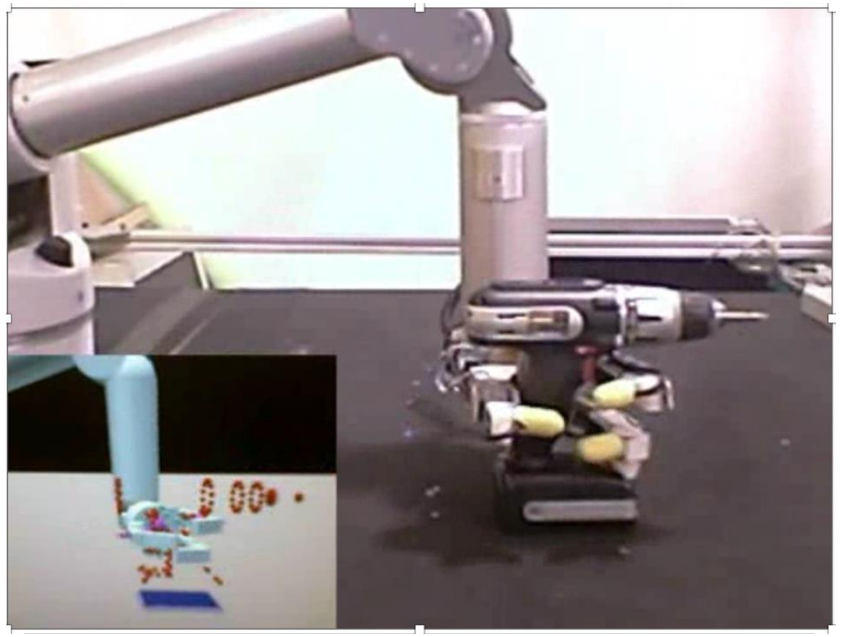
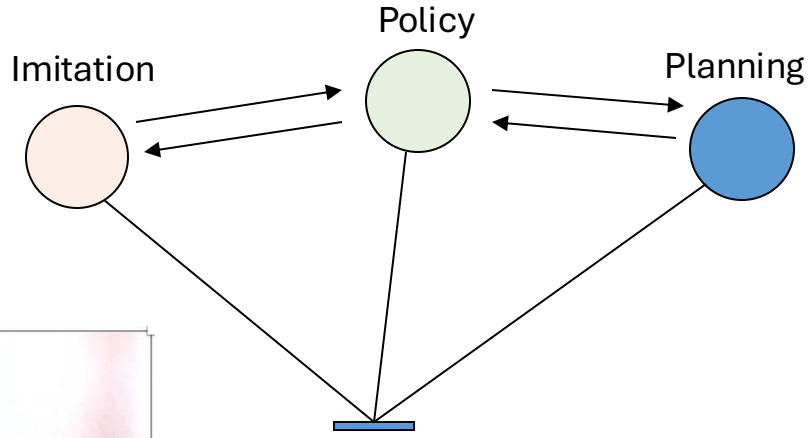


Fig. 2 Parts of a depth-2 search tree and associated value computation



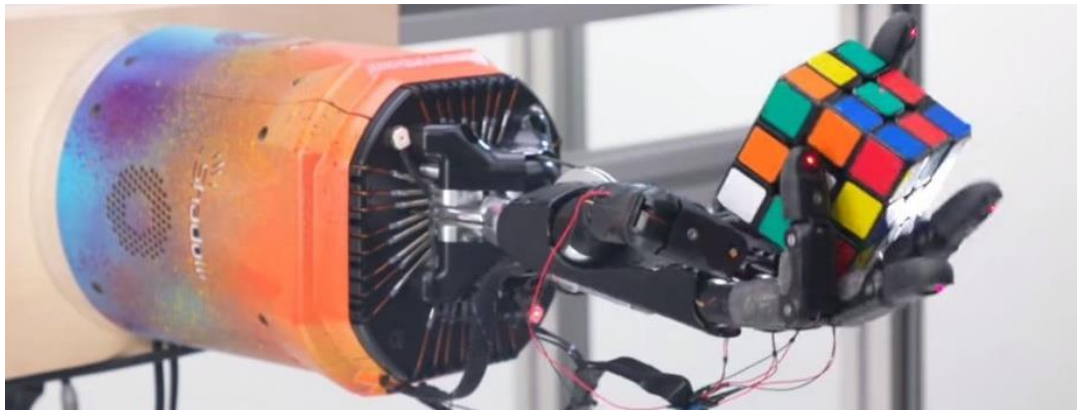


Grasping under uncertainty

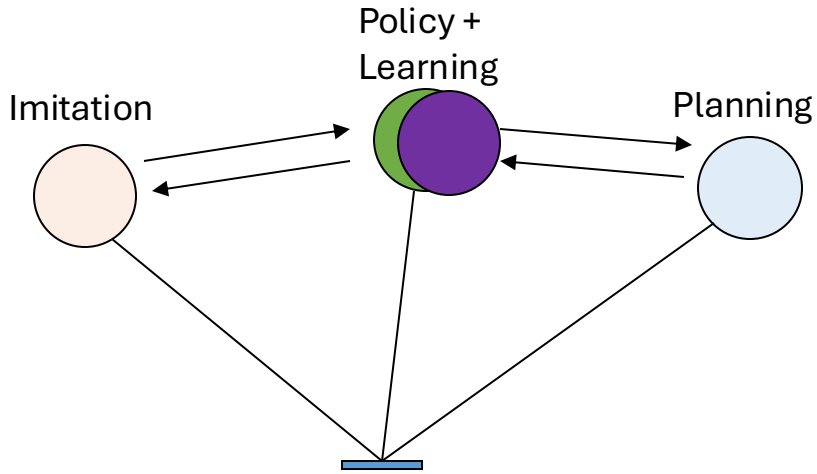
Hsiao, Lozano-Perez and Kaelbling 2009



Learn a policy via trial and error: Deep RL (2010s)

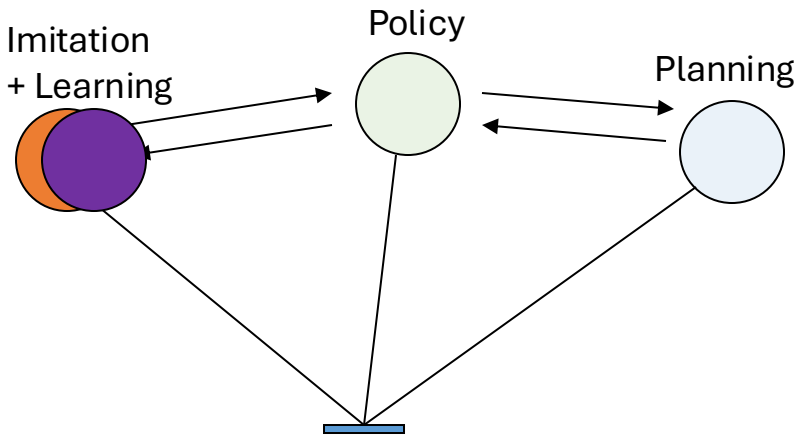






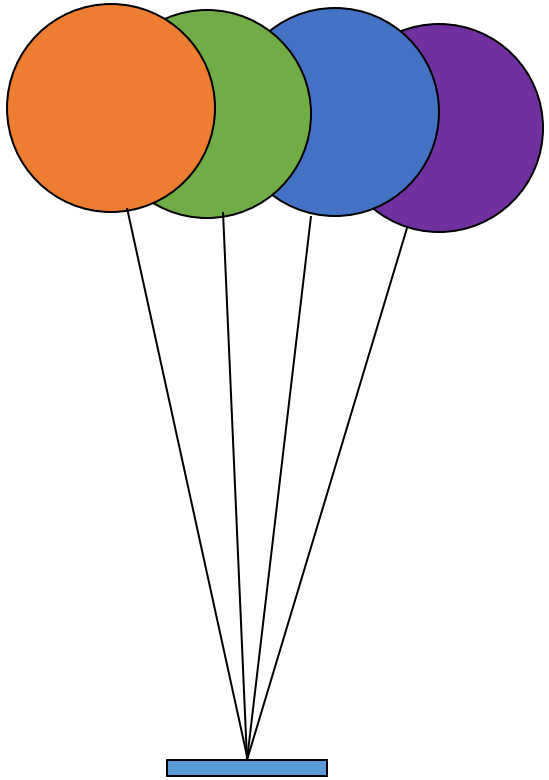
Learn a policy via imitation: BC (2020s)



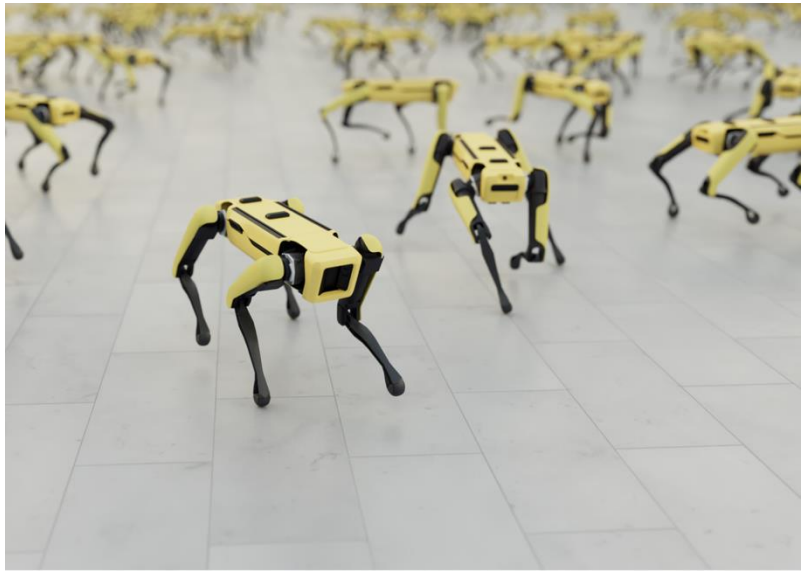


Imitation + Policy + Planning + Learning

is all you need



Motion



Task



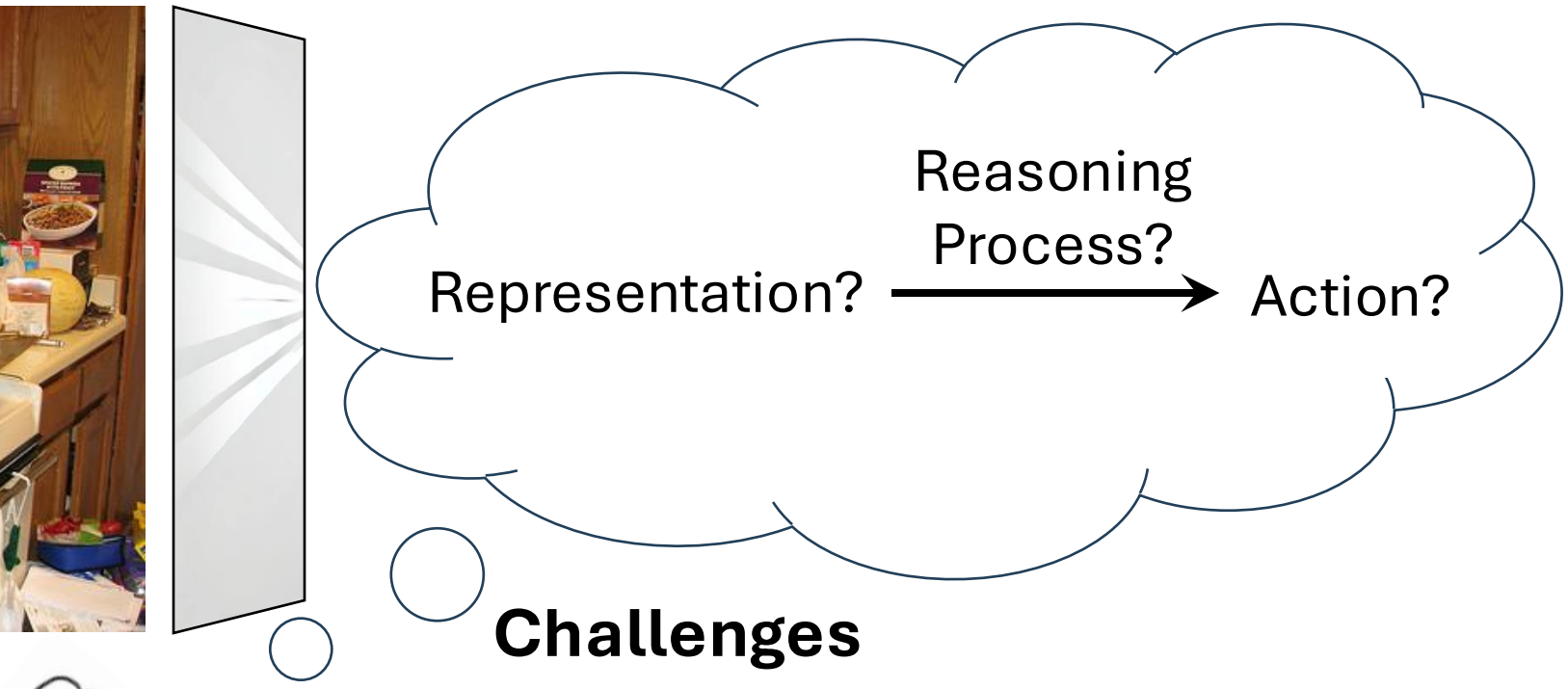
Task and Motion



Physical Decision Making = Task and motion



Task: Clean this messy kitchen for me.

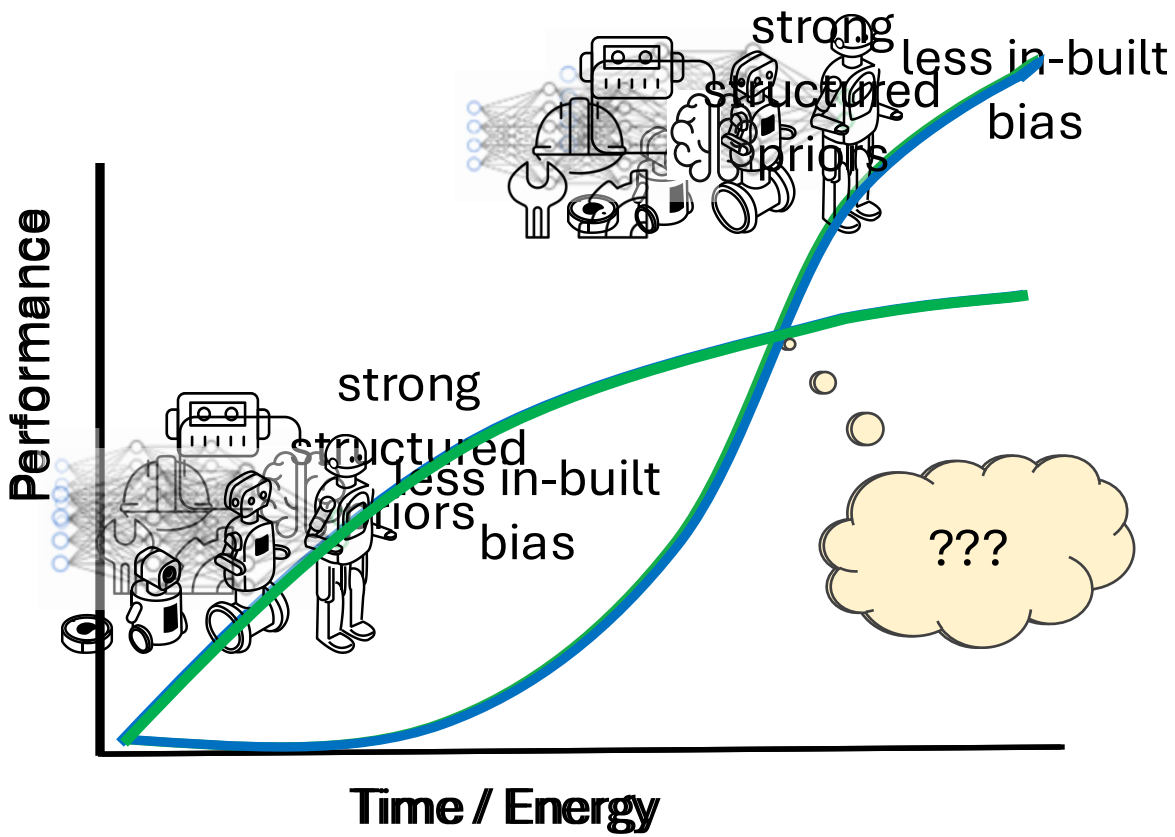


Challenges

- Sense geometric and physical properties
- Control a high-dof robot
- Make long-horizon plans
- Do all these in highly-varying environments

Two paths diverged…

Learn a single, massive, goal conditioned policy from huge amounts of data.

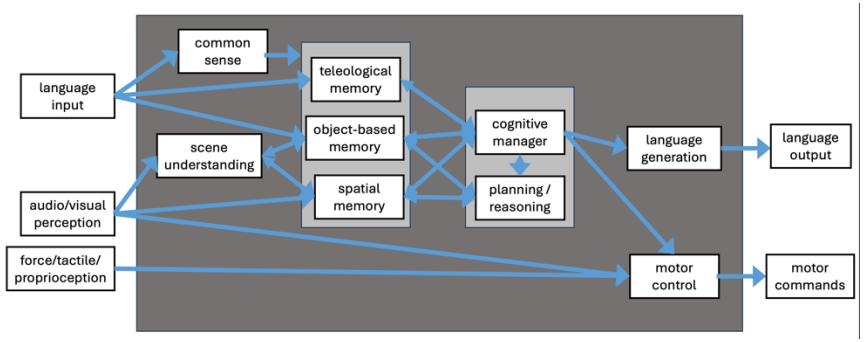


Take all possible advantage of:

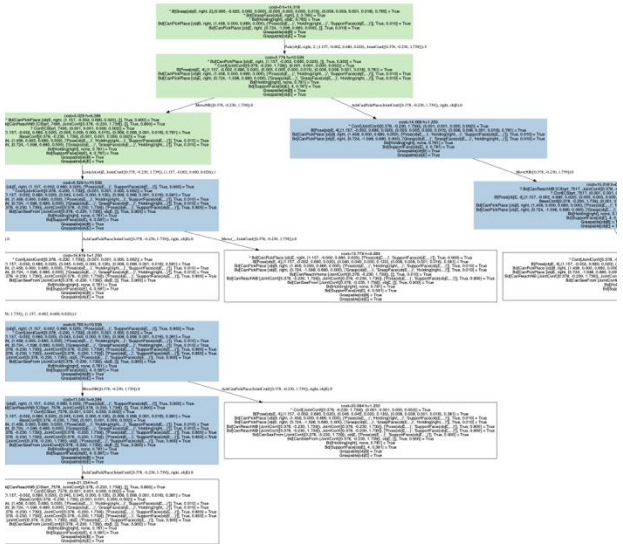
- the structure of the physical (and social) world
- algorithmic insights from computer science and cognitive science
  - abstraction, modularity, composition, inference
- our accumulated cultural knowledge (in LLMs, etc.)

Three Key Ingredients for Generality

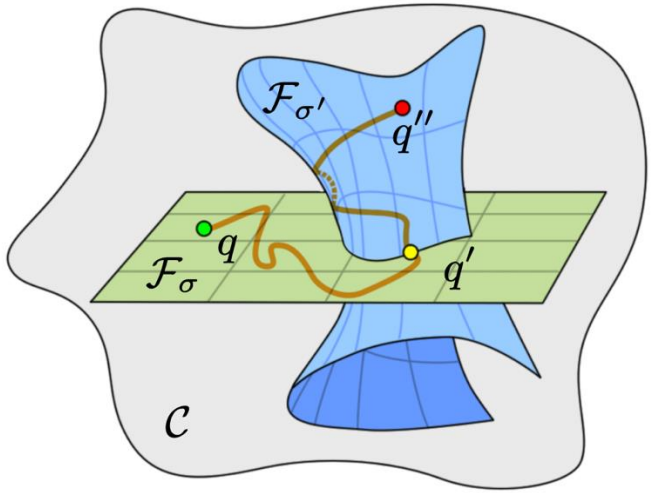
Modularity



Planning



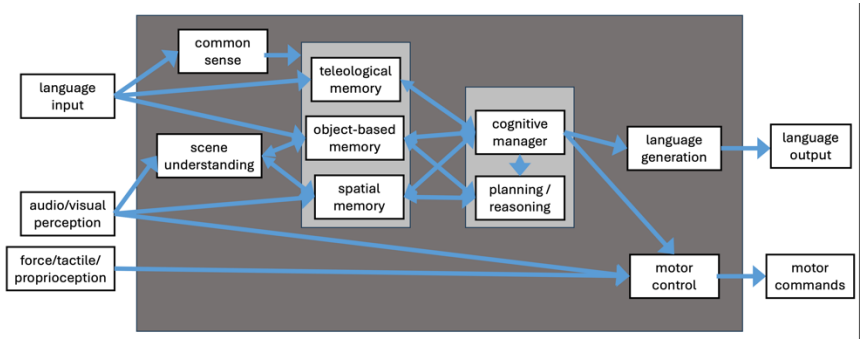
Compositional Abstraction



Best when taken together!

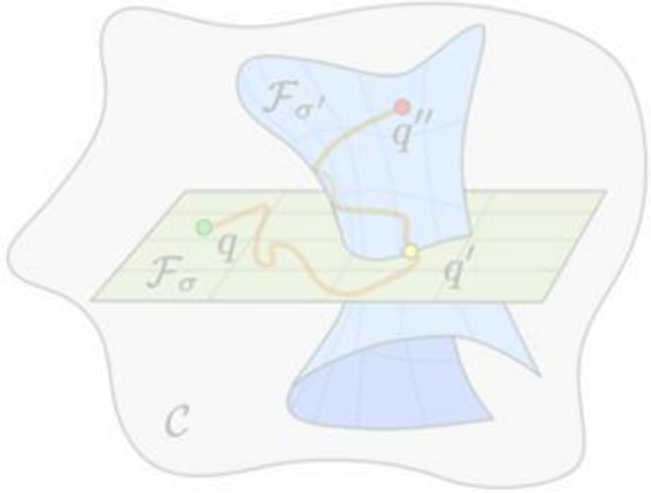
Three Key Ingredients for Generality

Modularity



Planning

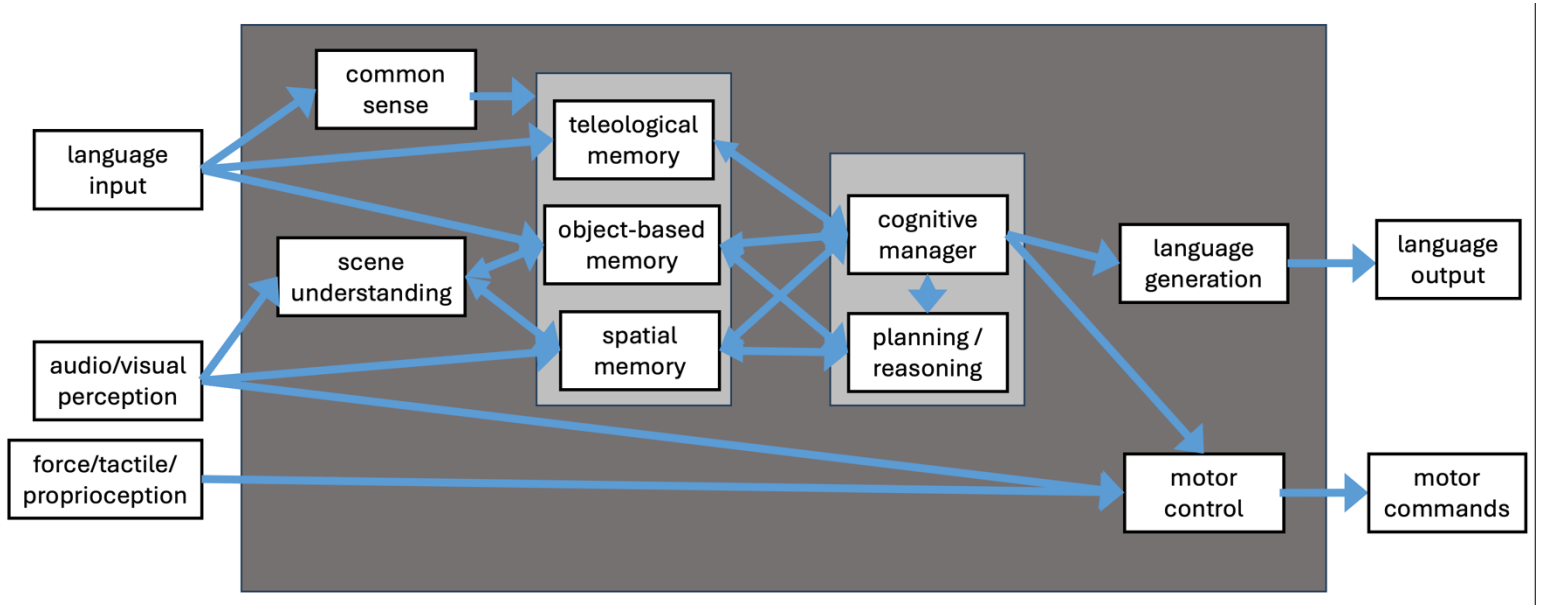
Compositional Abstraction



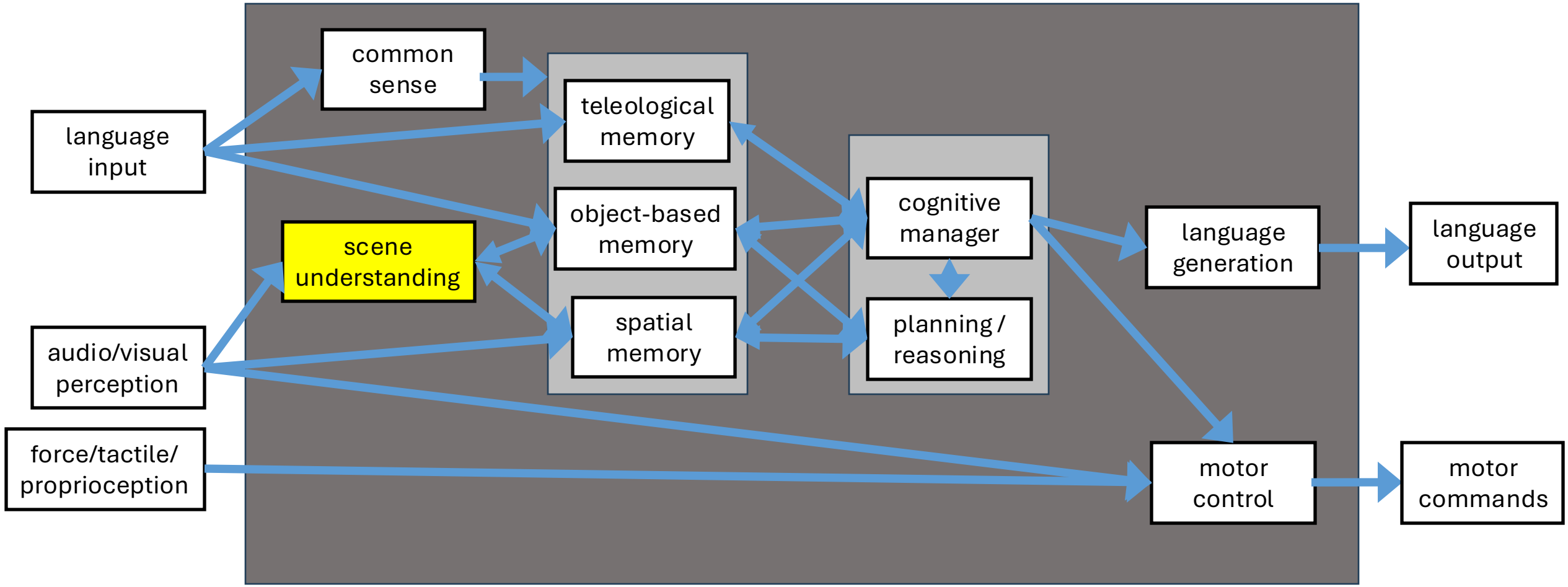
Best when taken together!

Modularity

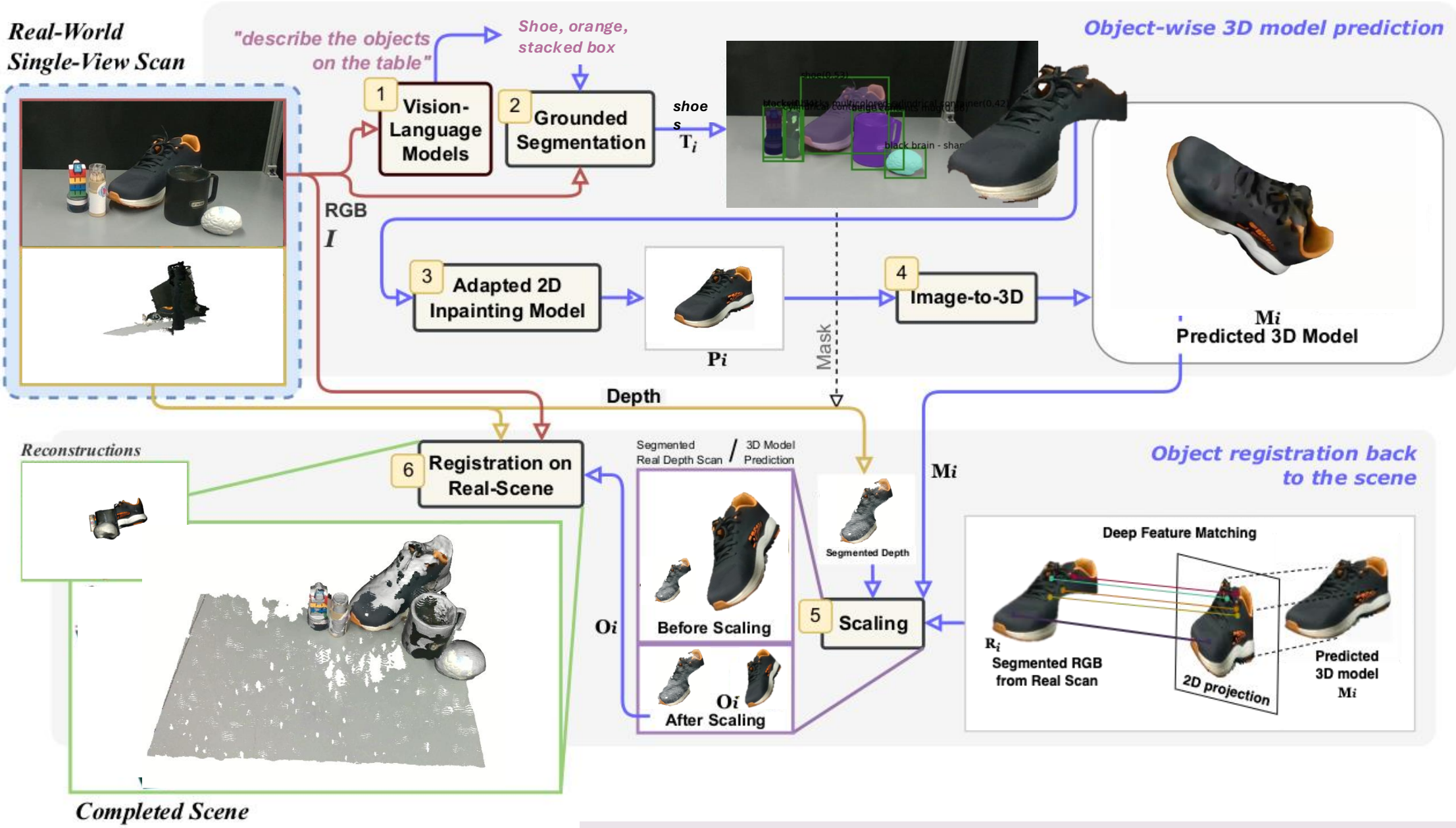
- Independence structure in computational problems often enables
  - compositional generalization
  - sample-efficient learning
  - non-destructive incremental learning
- Representations are key enablers
  - Text
  - Images
  - 3D



One possible policy representation



Object-based scene understanding: no prior object models







Aditya Agarwal, ..., Bipasha Sen ..., SceneComplete, submitted

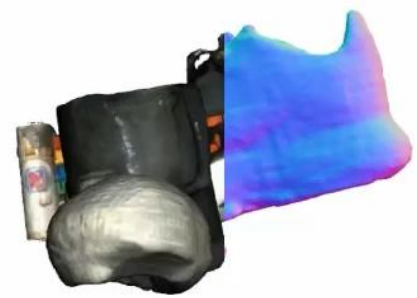
SceneComplete: Open-World 3D Scene Completion in Complex Real-World Environments for Robot Manipulation

Aditya Agarwal[1], Gaurav Singh[2], Bipasha Sen[1],
Tomás Lozano-Pérez[1], Leslie Pack Kaelbling[1]

[1]MIT CSAIL, [2]IIIT-Hyderabad

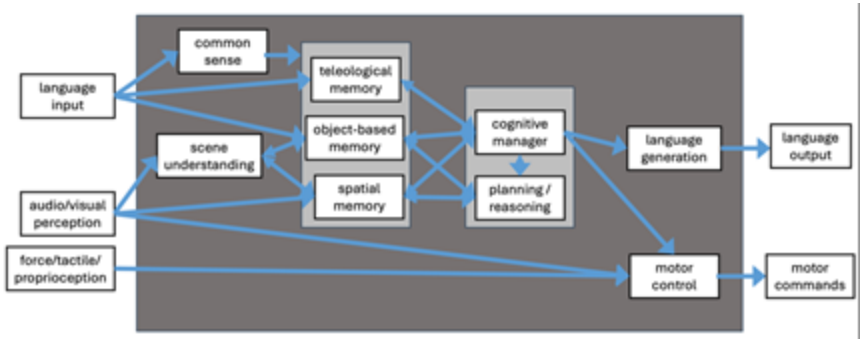




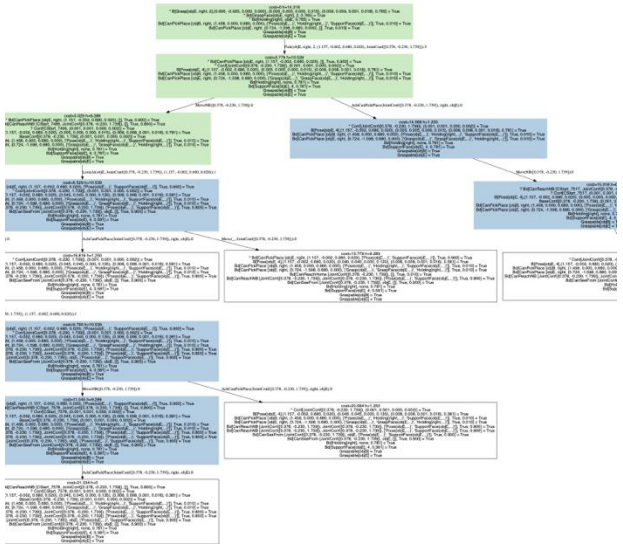


Three Key Ingredients for Generality

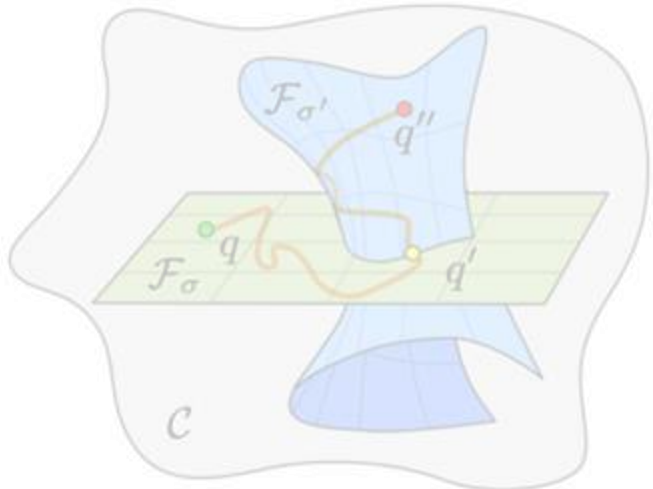
Modularity



Planning



Compositional Abstraction

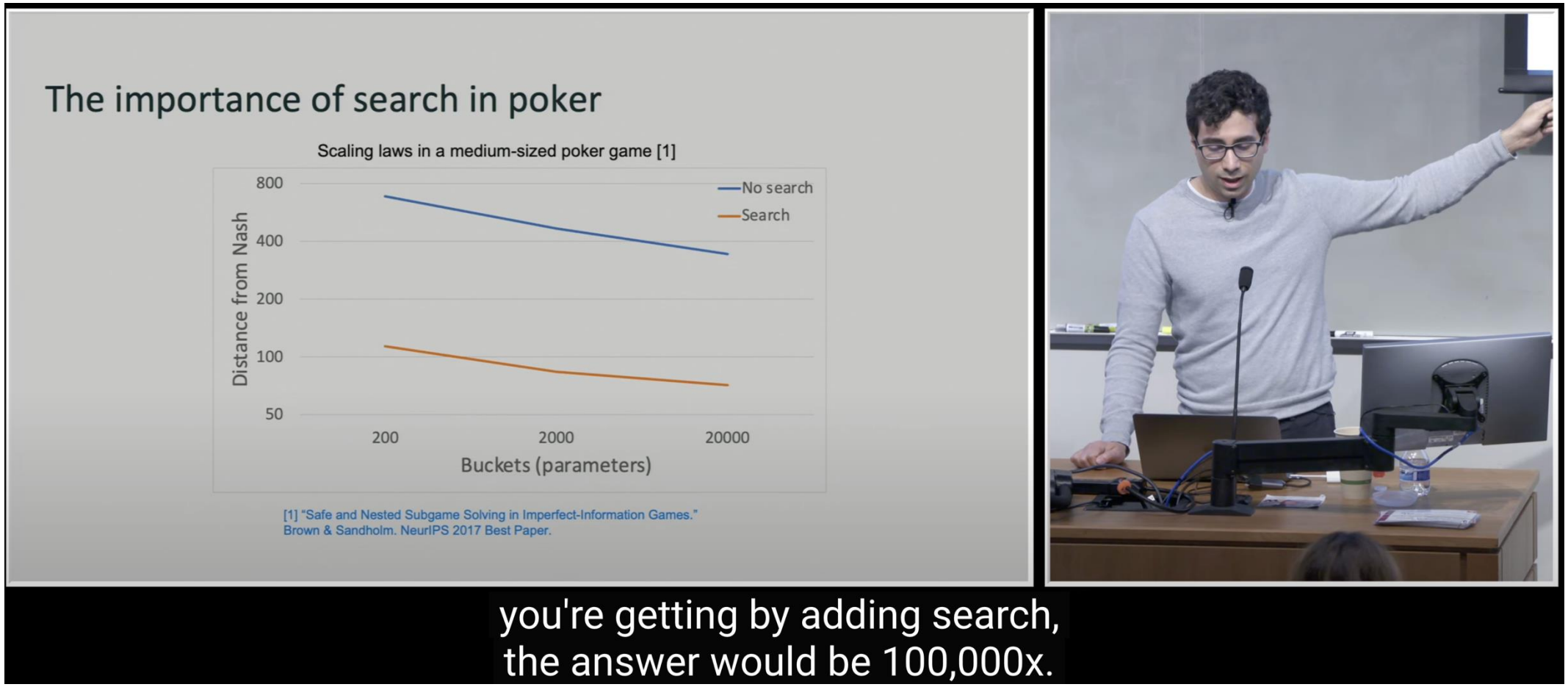


Best when taken together!

The power of search

Noam Brown, Open AI

“It turned out that having a bot think for just 20 seconds in a hand of poker got the same boosting performance as scaling up the model by 100,000x and training it for 100,000 times longer”



Solve the problem that you have!



The set of situations any individual is likely to encounter in its lifetime is orders of magnitude smaller than set of actually possible situations

Policies contain answers to all the problems: exponentially large circuit complexity for some domains

Planners find the answer to the problem at hand at some computational cost

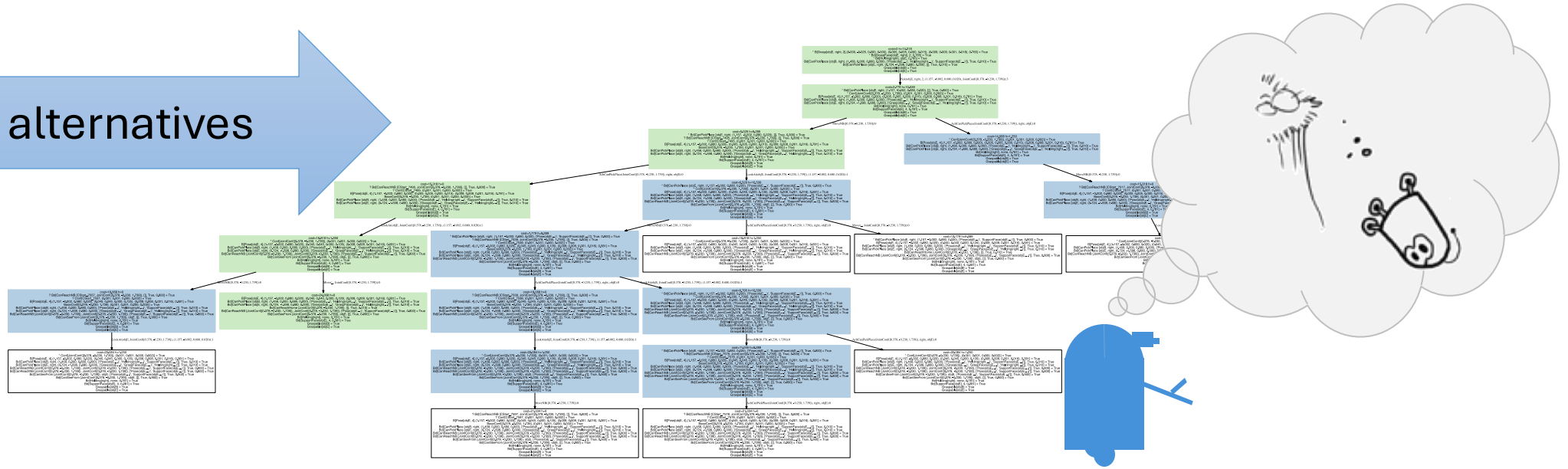
Jiayuan Mao, NeurIPS 2023

Planning: Letting our theories die in our stead
(Karl Popper)

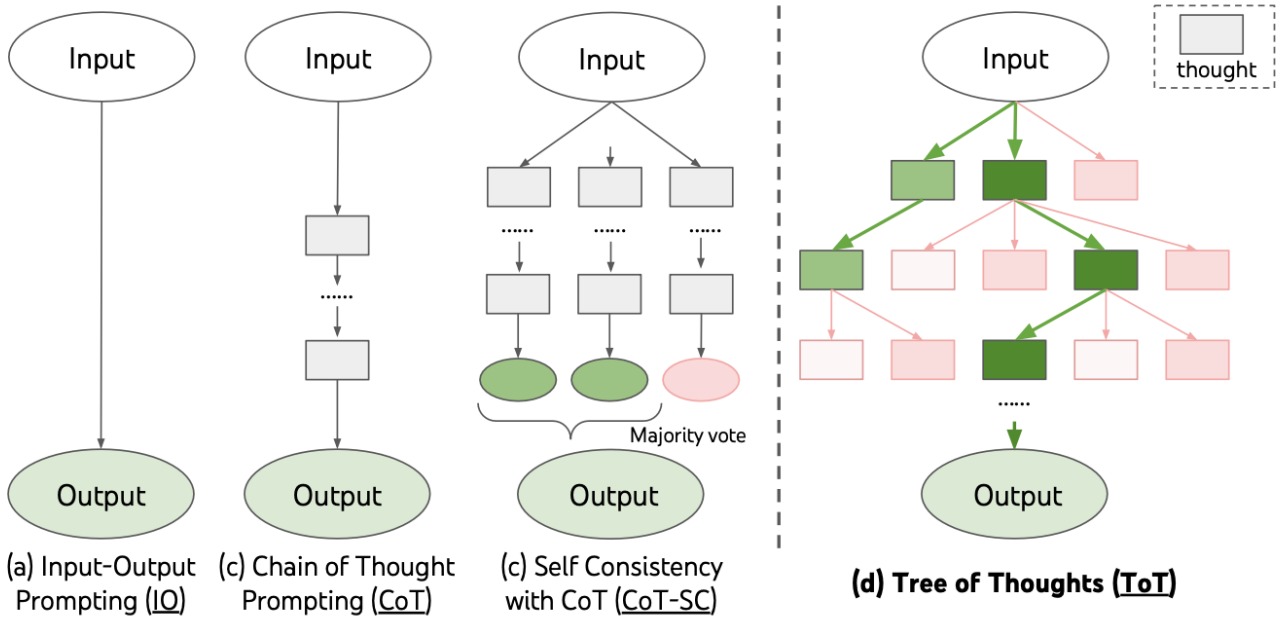
Execute a learned policy



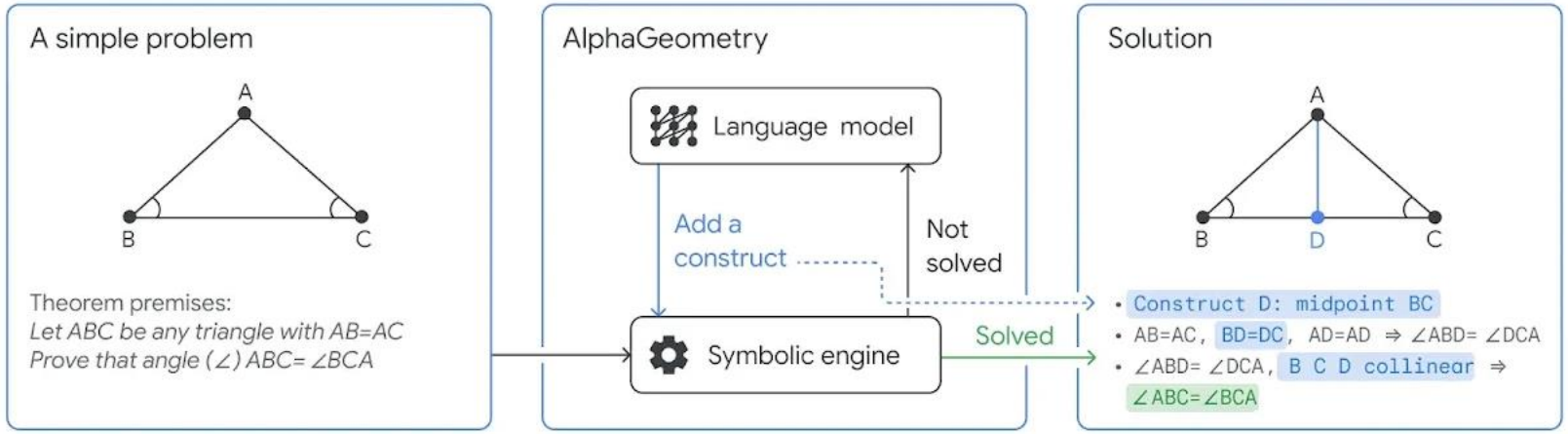
Search over alternatives



Run-time inference – some examples

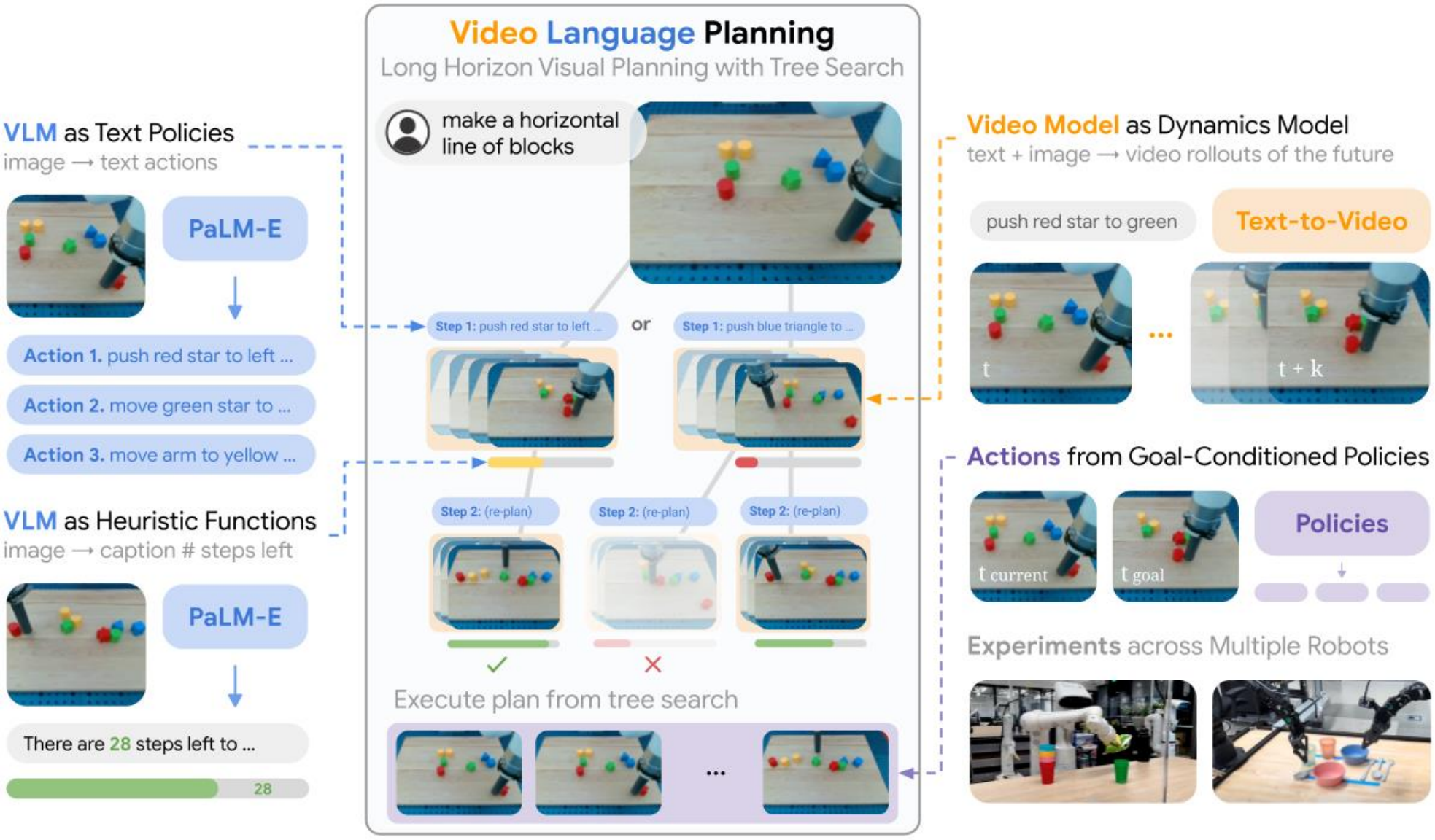


(a) Input-Output Prompting (IO) (c) Chain of Thought Prompting (CoT) (c) Self Consistency with CoT (CoT-SC) **(d) Tree of Thoughts (ToT)**



Video Language Planning





Yilun Du, <many Google collaborators>,

Increasing horizon with good old fashioned search!

Combine

- **Abstract policy**: P(text action | image, goal) --- PalmE, finetuned
- **Abstract heuristic**: P(# abstract steps to go | image, goal) --- PalmE, finetuned
- **Abstract transition model and subgoal policy**:
  P(image sequence | image, text action)
- **Control policy**: P(motor action | image, next image)

Abstract high-level search:

- query abstract policy for K possible actions
- compute next-state image using transition model
- do beam search to get abstract action sequence

Yilun Du, <many Google collaborators>,

Increasing horizon with good old fashioned search!

Video Language Planning

Video Generation

Real Execution

Put the fruits into the top drawer

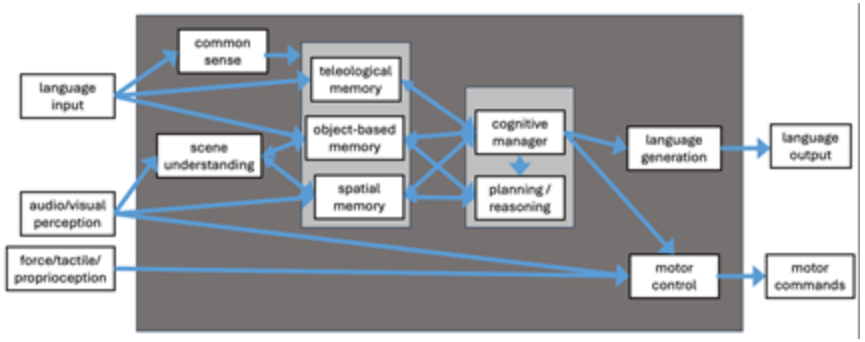


Execution

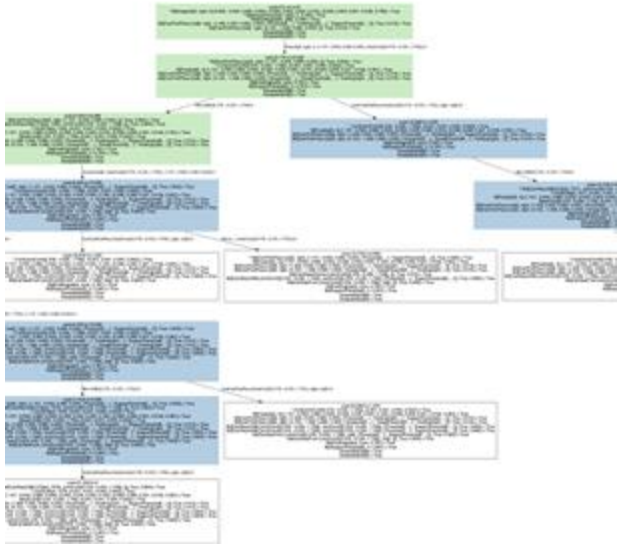


Three Key Ingredients for Generality

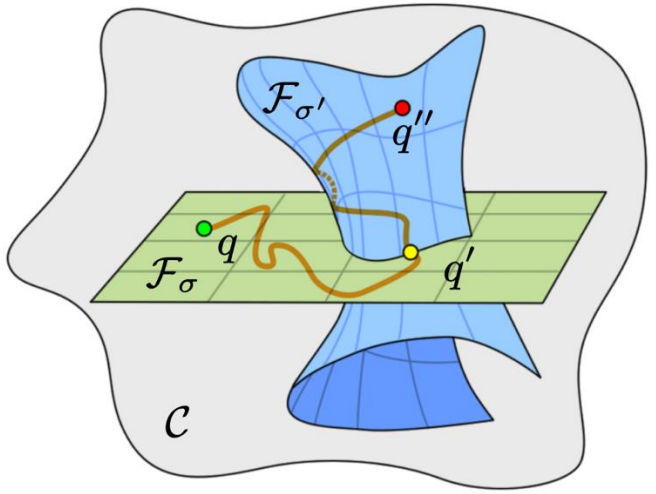
Modularity



Planning



Compositional Abstraction



Best when taken together!

Compositional Abstractions

States are described using state abstractions (classifiers on state):

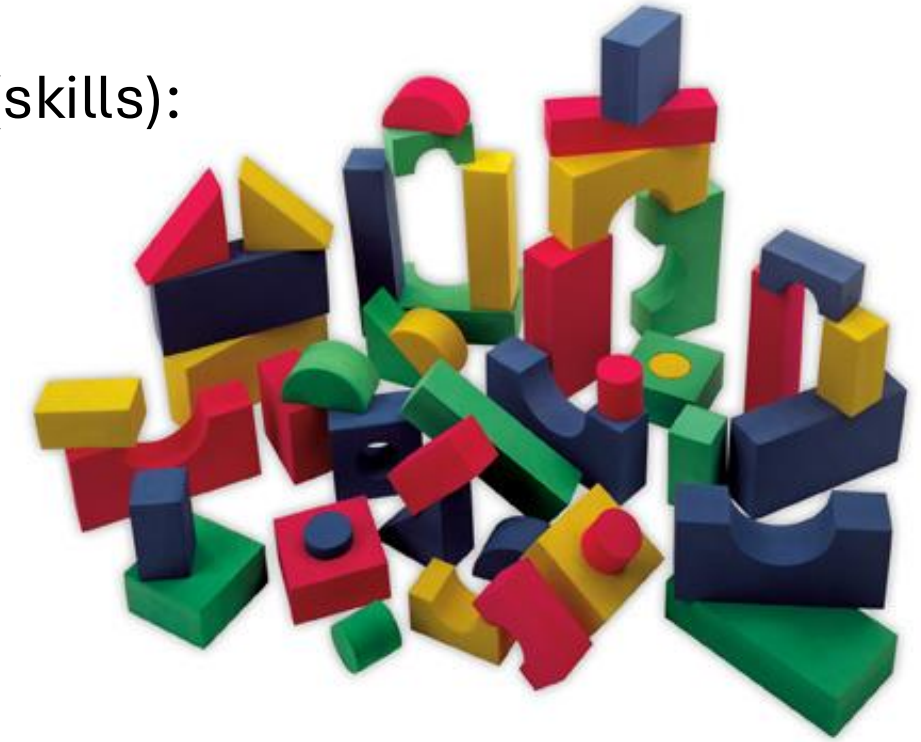
holding(object)
in(object, cup)

They can be composed to describe preconditions, effects and goals

Actions are described using temporal abstractions (skills):

grasp(object)
place-in(object, container)

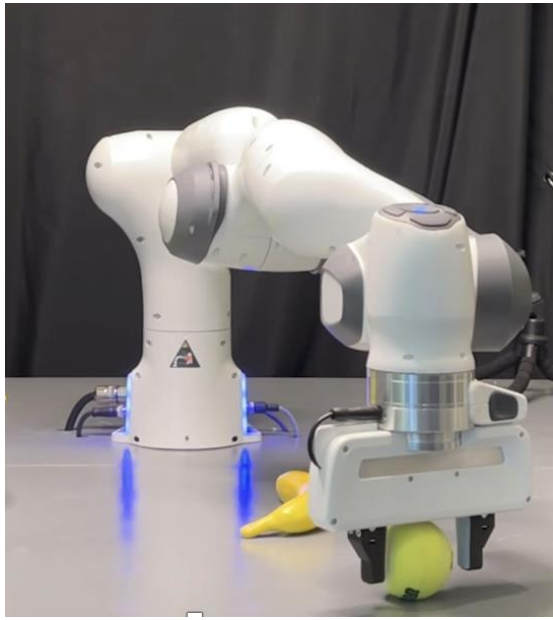
They can be composed to achieve goals efficiently

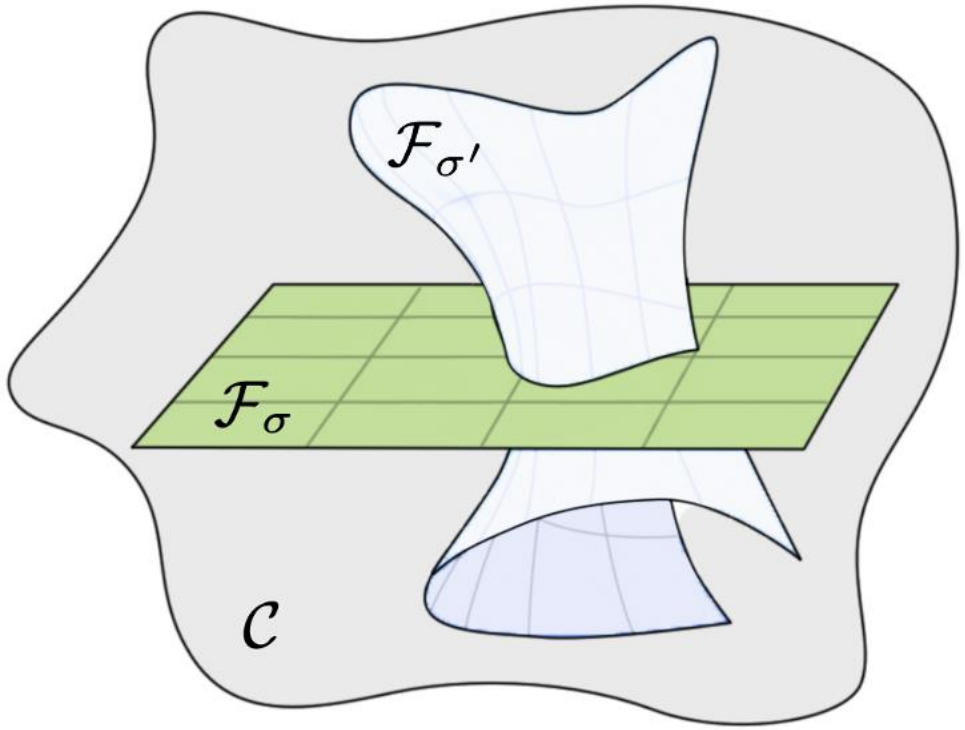


Physical structure in task and motion planning

Compositional abstraction exposes **sparsity** and **temporal decomposition**

Search space is set of **low-dimensional manifolds** in the **configuration space**





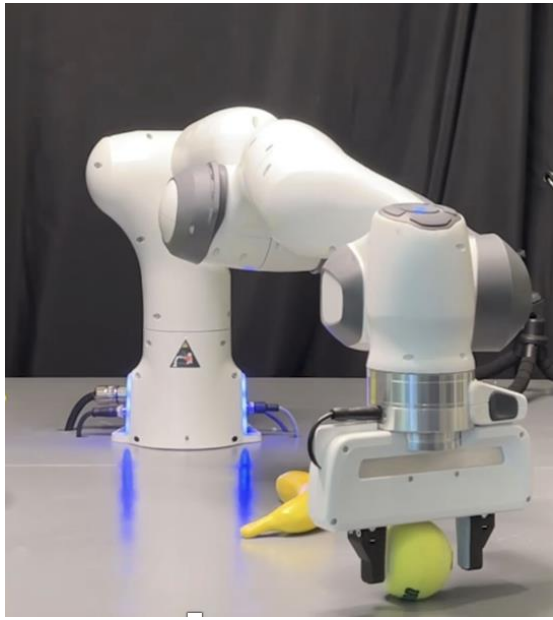

```
action move-to-grasp(o: obj, g: grasp, t: traj)
  pre: robot-at(t[0]), valid-g(t[-1], pose(o), g)
  eff: robot-at(t[-1]), holding(o, g)
  controller: ...
```

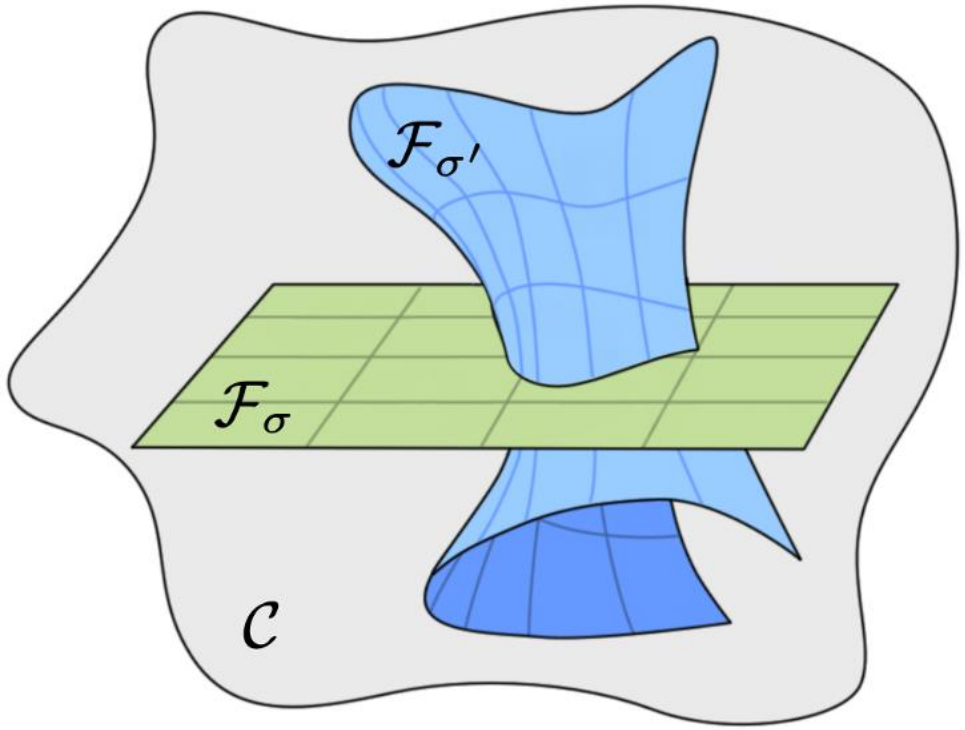
Figure: Hauser and Latombe. Multi-Modal Motion Planning in Non-Expansive Spaces.

Physical structure in task and motion planning

Compositional abstraction exposes **sparsity** and **temporal decomposition**

Search space is set of **low-dimensional manifolds** in the **configuration space**






```
action move-to-grasp(o: obj, g: grasp, t: traj)
  pre: robot-at(t[0]), valid-g(t[-1], pose(o), g)
  eff: robot-at(t[-1]), holding(o, g)
  controller: ...
action move-while-holding(o: obj, g: grasp, t: traj)
  pre: robot-at(t[0]), holding(o, g), valid-obj-t(o, t)
  eff: robot-at(t[-1]), obj-at(...)
  controller: ...
```

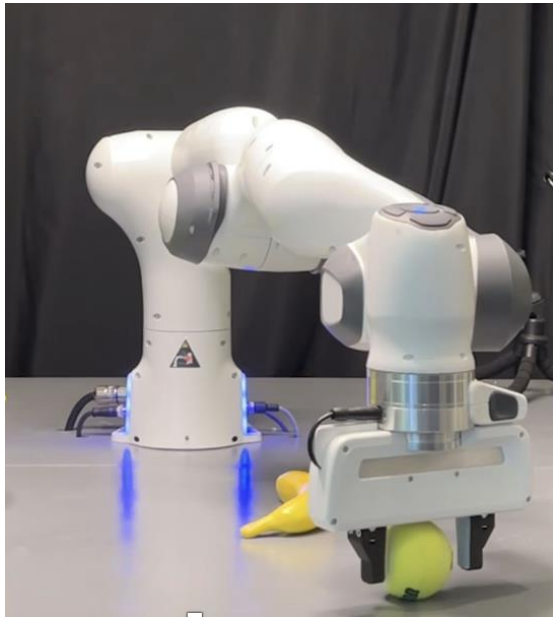
Figure: Hauser and Latombe. Multi-Modal Motion Planning in Non-Expansive Spaces.

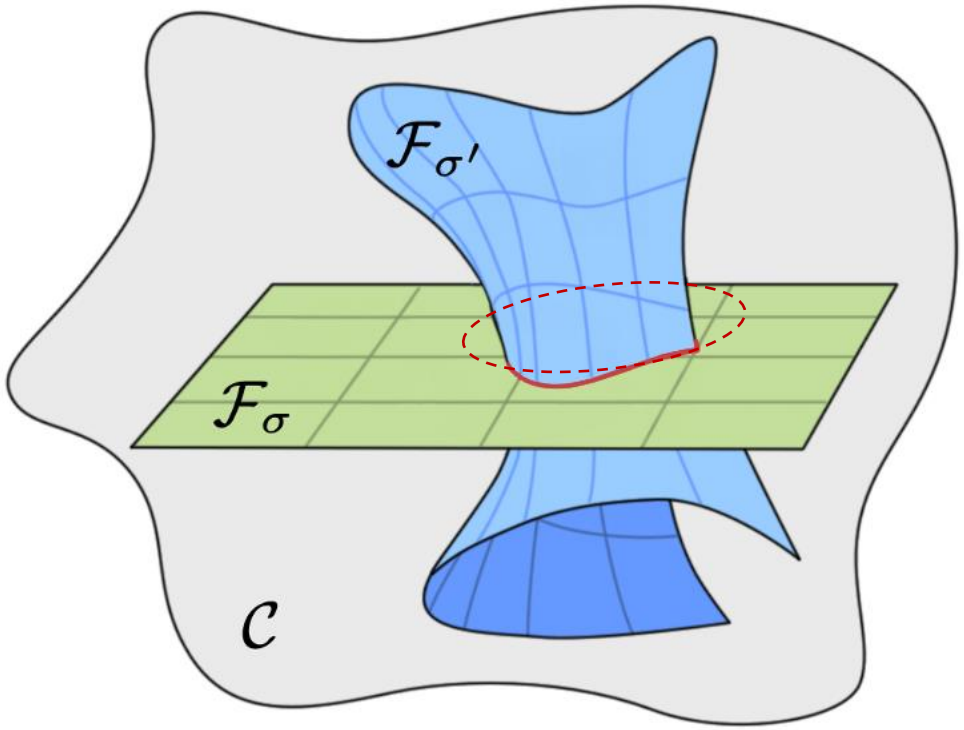
Physical structure in task and motion planning

Compositional abstraction exposes **sparsity** and **temporal decomposition**

Search space is set of **low-dimensional manifolds** in the **configuration space**

Connected at regions modeled by preconditions and effects

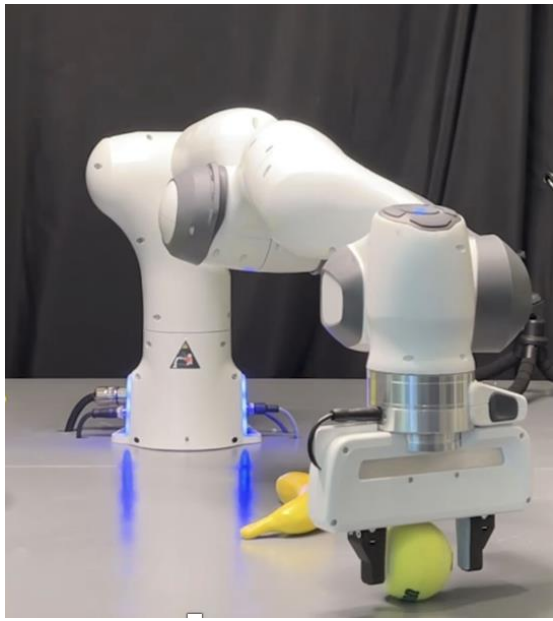





```
action move-to-grasp(o: obj, g: grasp, t: traj)
  pre: robot-at(t[0]), valid-g(t[-1], pose(o), g)
  eff: robot-at(t[-1]), holding(o, g)
  controller: ...
action move-while-holding(o: obj, g: grasp, t: traj)
  pre: robot-at(t[0]), holding(o, g), valid-obj-t(o, t)
  eff: robot-at(t[-1]), obj-at(...)
  controller: ...
```

Figure: Hauser and Latombe. Multi-Modal Motion Planning in Non-Expansive Spaces.

Physical structure in task and motion planning

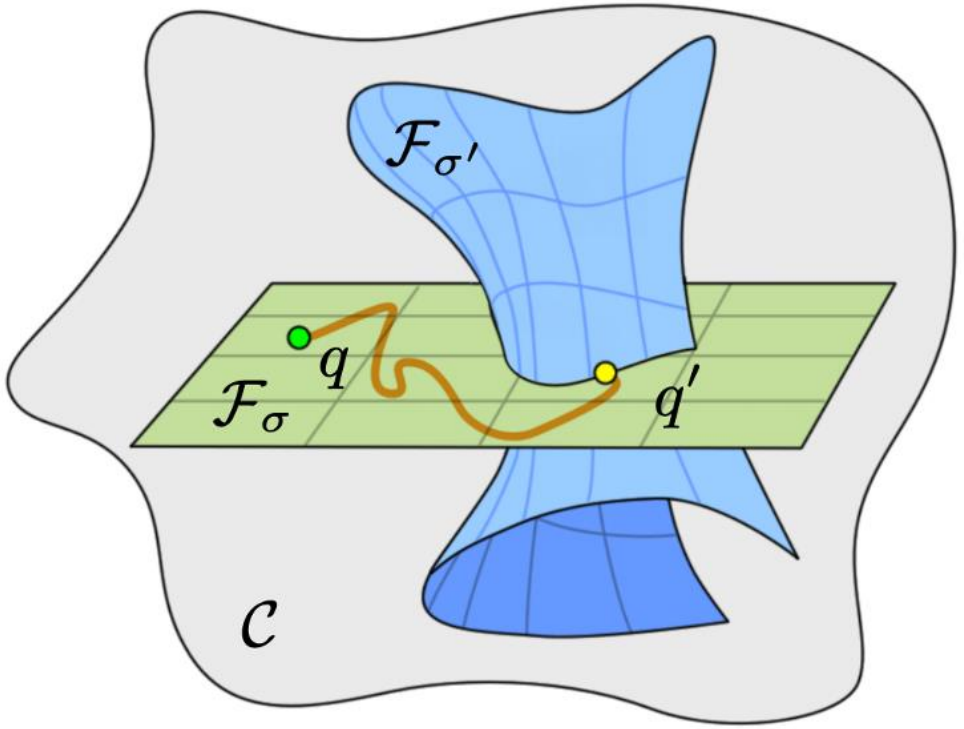


Compositional abstraction exposes **sparsity** and **temporal decomposition**

Search space is set of **low-dimensional manifolds** in the **configuration space**

Connected at regions modeled by preconditions and effects

They make planning easier by suggesting subgoals for planning

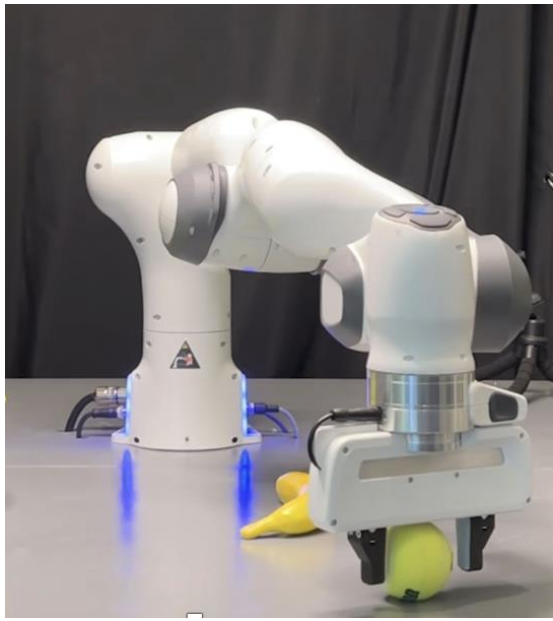



```
action move-to-grasp(o: obj, g: grasp, t: traj)
  pre: robot-at(t[0]), valid-g(t[-1], pose(o), g)
  eff: robot-at(t[-1]), holding(o, g)
  controller: ...

action move-while-holding(o: obj, g: grasp, t: traj)
  pre: robot-at(t[0]), holding(o, g), valid-obj-t(o, t)
  eff: robot-at(t[-1]), obj-at(...)
  controller: ...
```

Figure: Hauser and Latombe. Multi-Modal Motion Planning in Non-Expansive Spaces.

Physical structure in task and motion planning

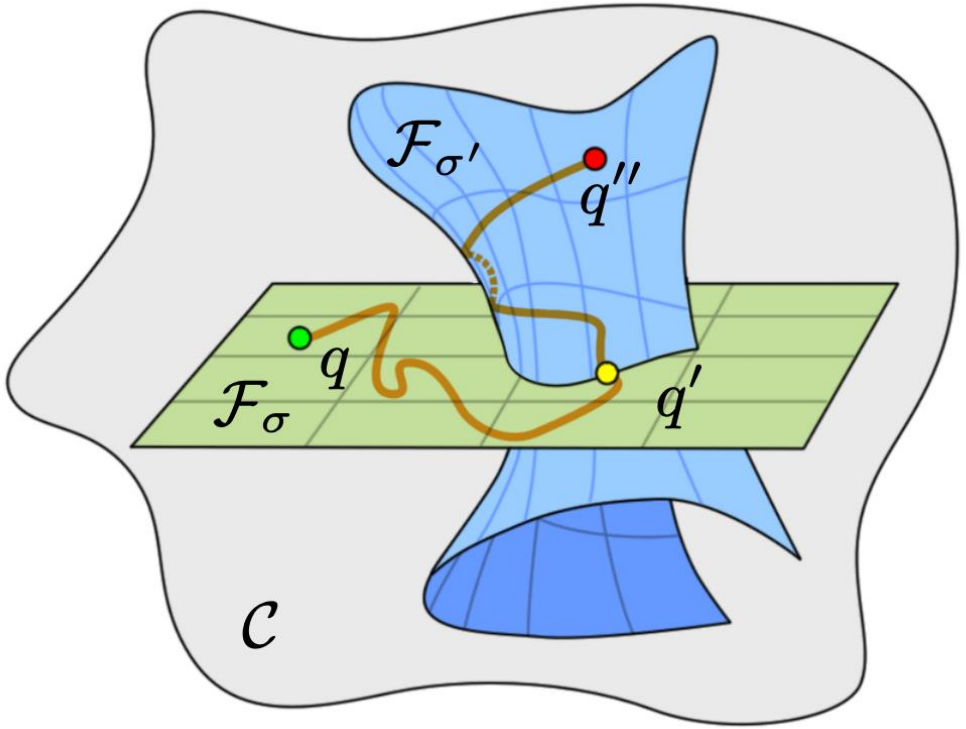


Compositional abstraction exposes **sparsity** and **temporal decomposition**

Search space is set of **low-dimensional manifolds** in the **configuration space**

Connected at regions modeled by preconditions and effects

They make planning easier by suggesting subgoals for planning




```
action move-to-grasp(o: obj, g: grasp, t: traj)
  pre: robot-at(t[0]), valid-g(t[-1], pose(o), g)
  eff: robot-at(t[-1]), holding(o, g)
  controller: ...

action move-while-holding(o: obj, g: grasp, t: traj)
  pre: robot-at(t[0]), holding(o, g), valid-obj-t(o, t)
  eff: robot-at(t[-1]), obj-at(...)
  controller: ...
```

Figure: Hauser and Latombe. Multi-Modal Motion Planning in Non-Expansive Spaces.

Task and Motion Planning for Robotics



Instruction: Put all food items in the fridge.
Initial State: in(Cabbage, Pot),
on(Potato, Table), …

Task Plan:

① *Open the left fridge door* ② *Remove the pot lid* ③ *Move the cabbage from pot to fridge* ④ *Move potato to fridge*

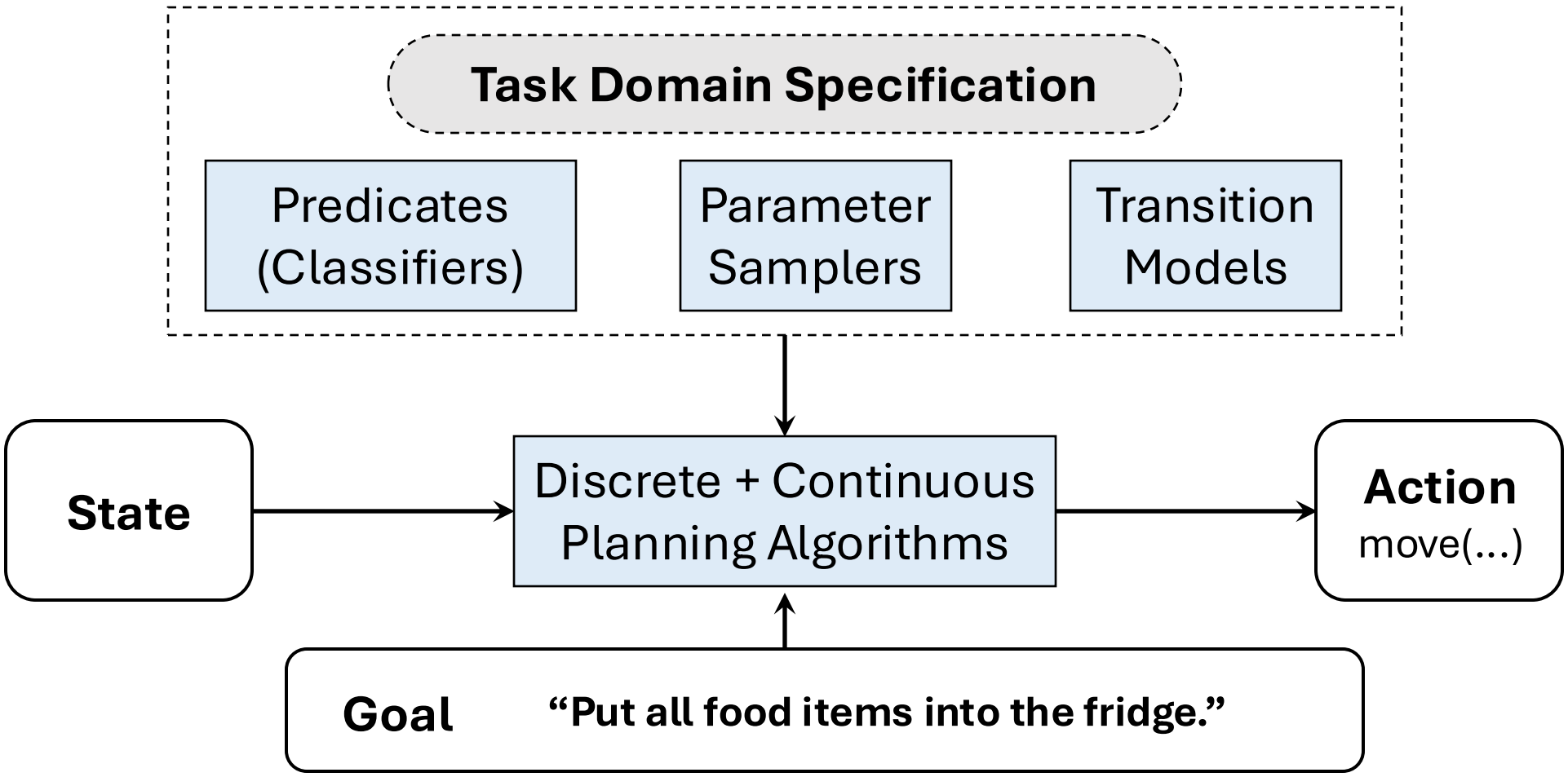
Refine + Feedback

Motion Plan:



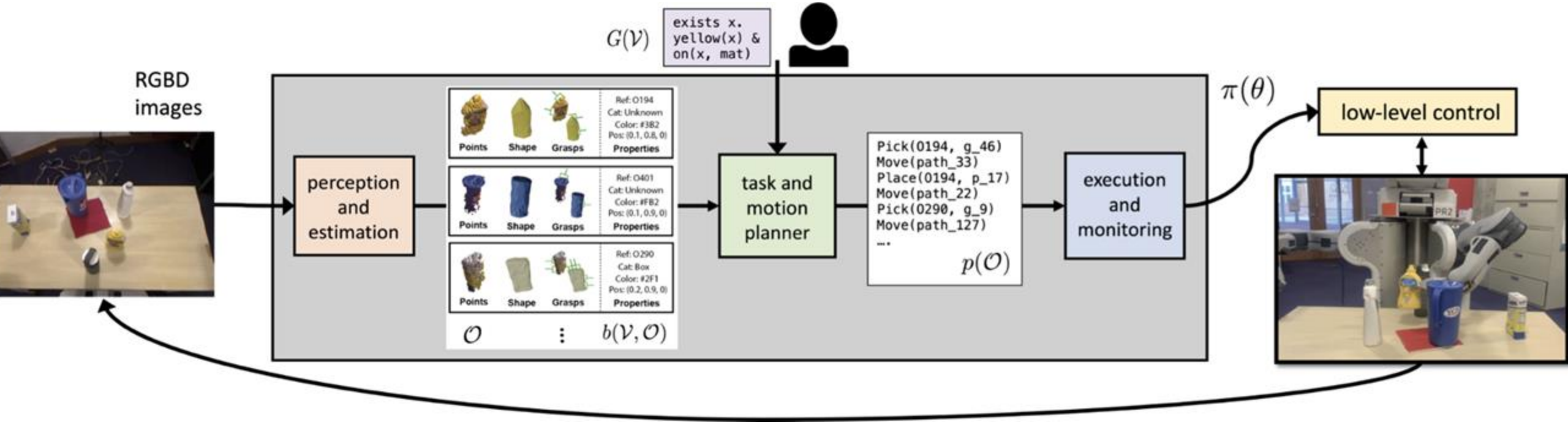
Photo Credit: Yang et al, RSS 2023

Basic Elements in Task and Motion Planning



Closed-loop planning-driven end-to-end policy using these abstractions





It works with completely novel objects!



Aidan Curtis, Xiaolin Fang, …, Caelan Garrett, ICRA 2022

Immediate generalization to novel objects, goals, arrangements



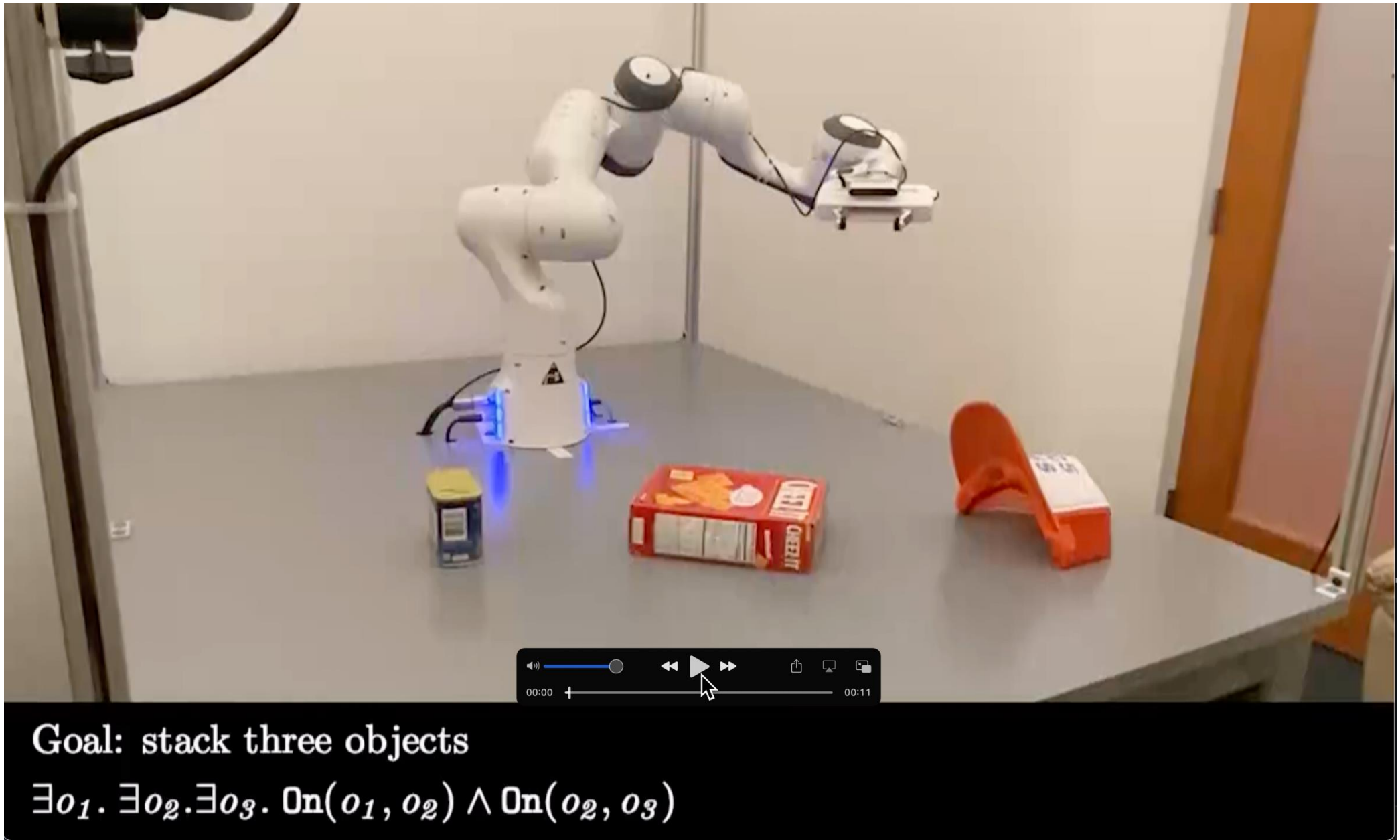
Put each object in the bowl of the closest color



Put all the objects on the blue mat



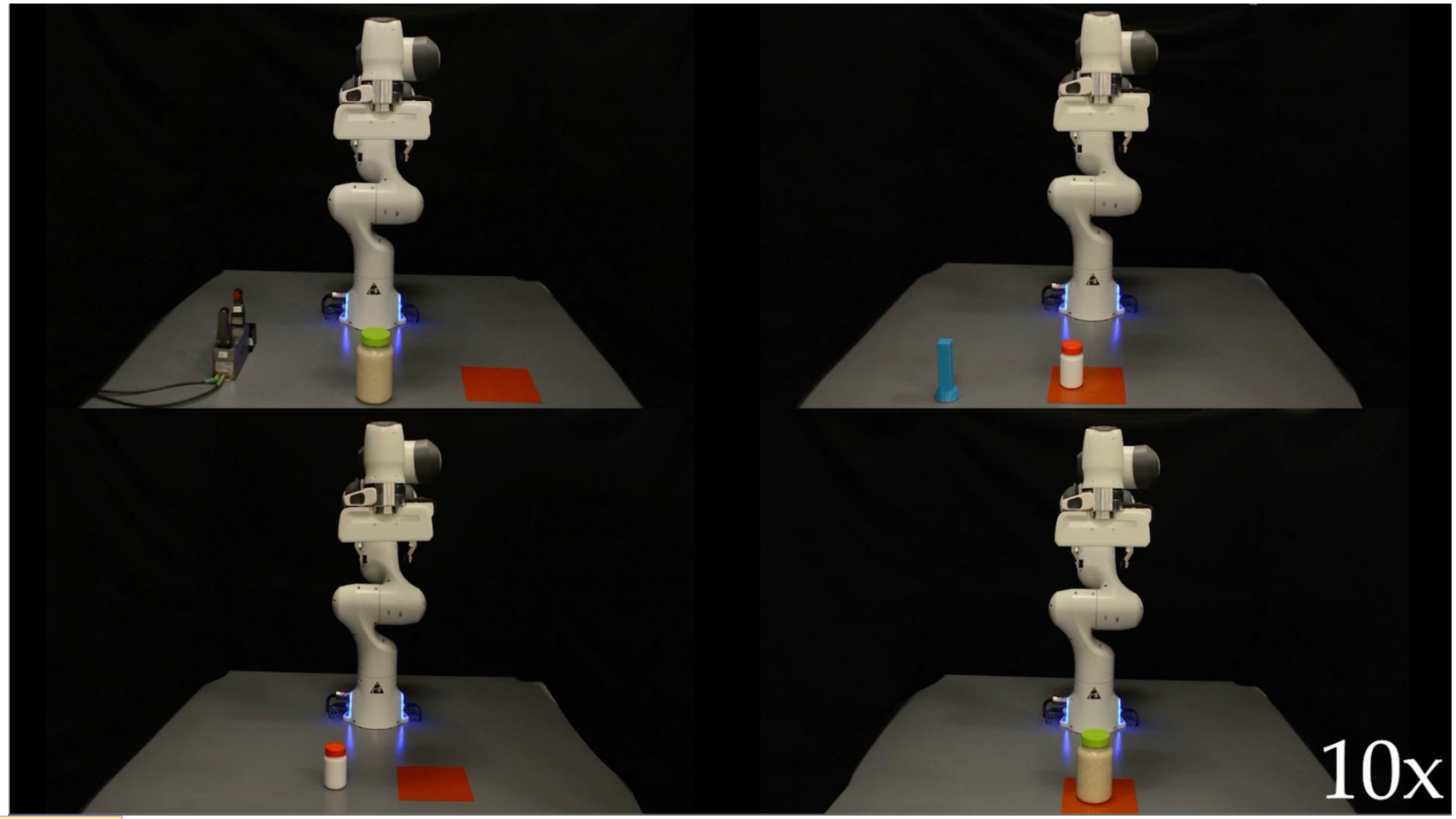
Cross-embodiment generalization



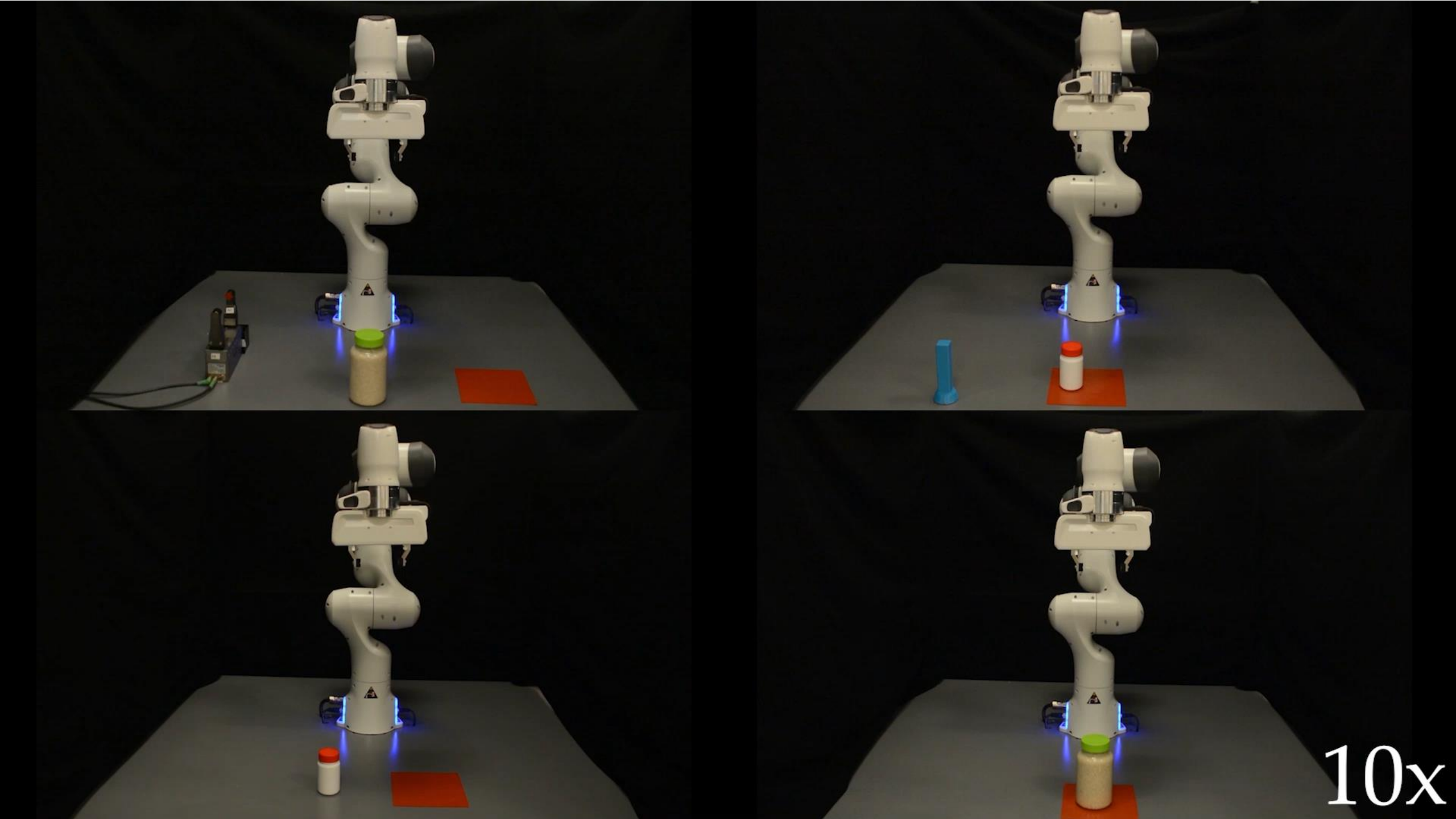


Forceful Manipulation

- Given object models…
- Four ways of opening a child-proof bottle, depending on model parameters



Holladay, … IJRR 2023


10x

Task and motion planning in belief space: information gathering integrated with geometric reasoning





Caelan Garrett, <Nvidia collaborators>, ..., ICRA2020

VLM and TAMP >> VLM or TAMP

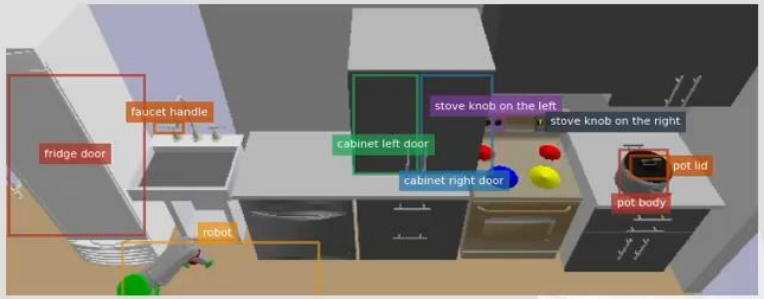




Goal given to VLM:

make a chicken soup

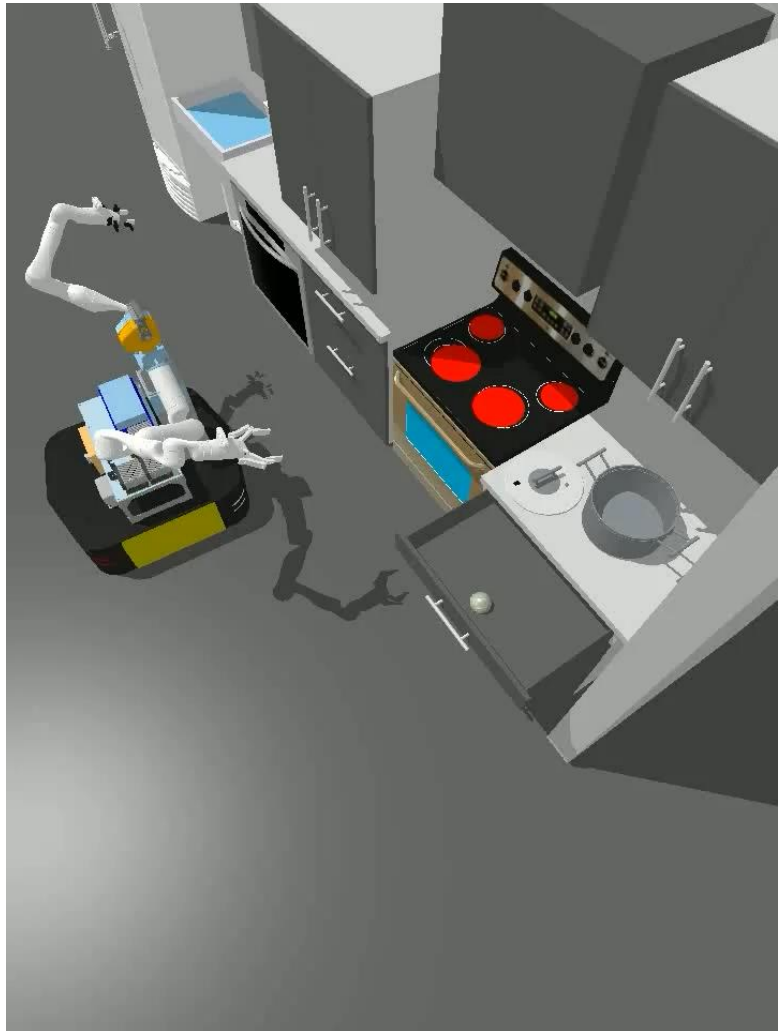
Image given to VLM:

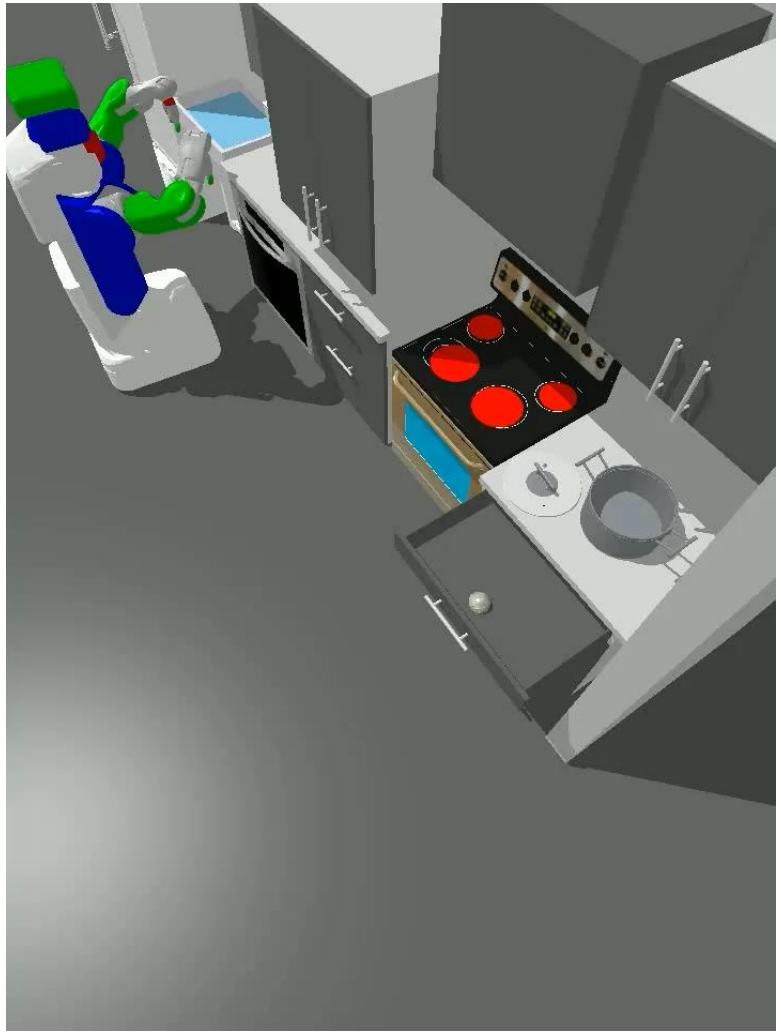


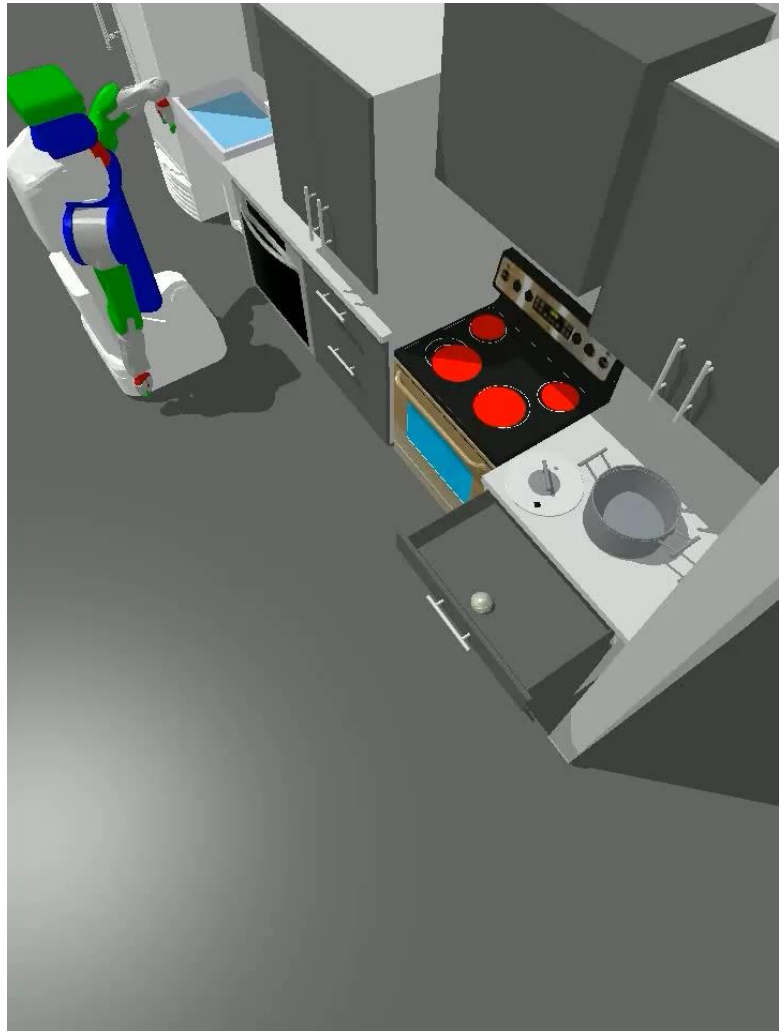


Zhutian Yang, Caelan Garrett, ..., Submitted 2024

Different embodiment -> Different task plans



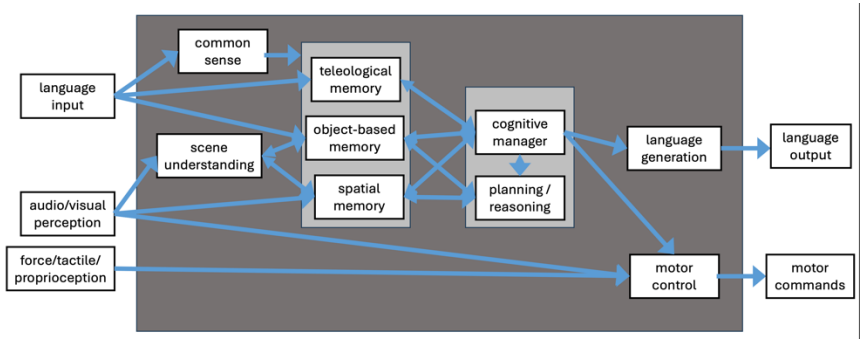




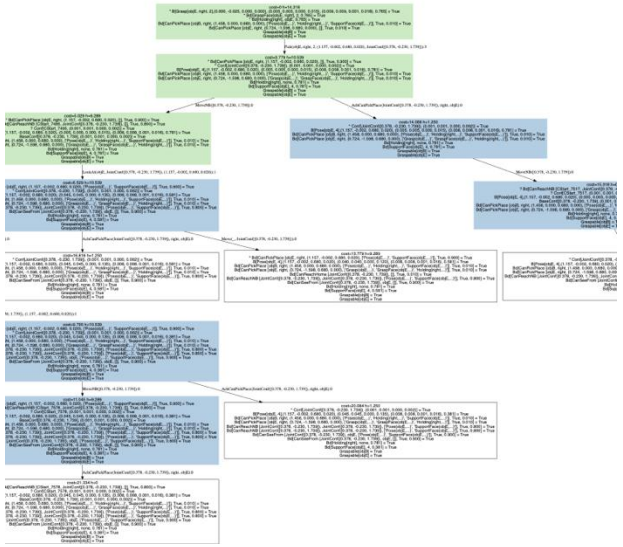
Zhutian Yang, Caelan Garrett, ..., Submitted 2024

Three Key Ingredients for Generality

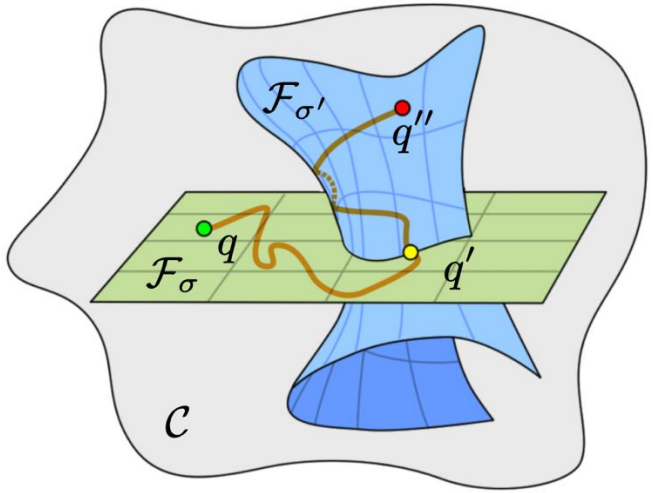
Modularity



Planning



Compositional Abstraction

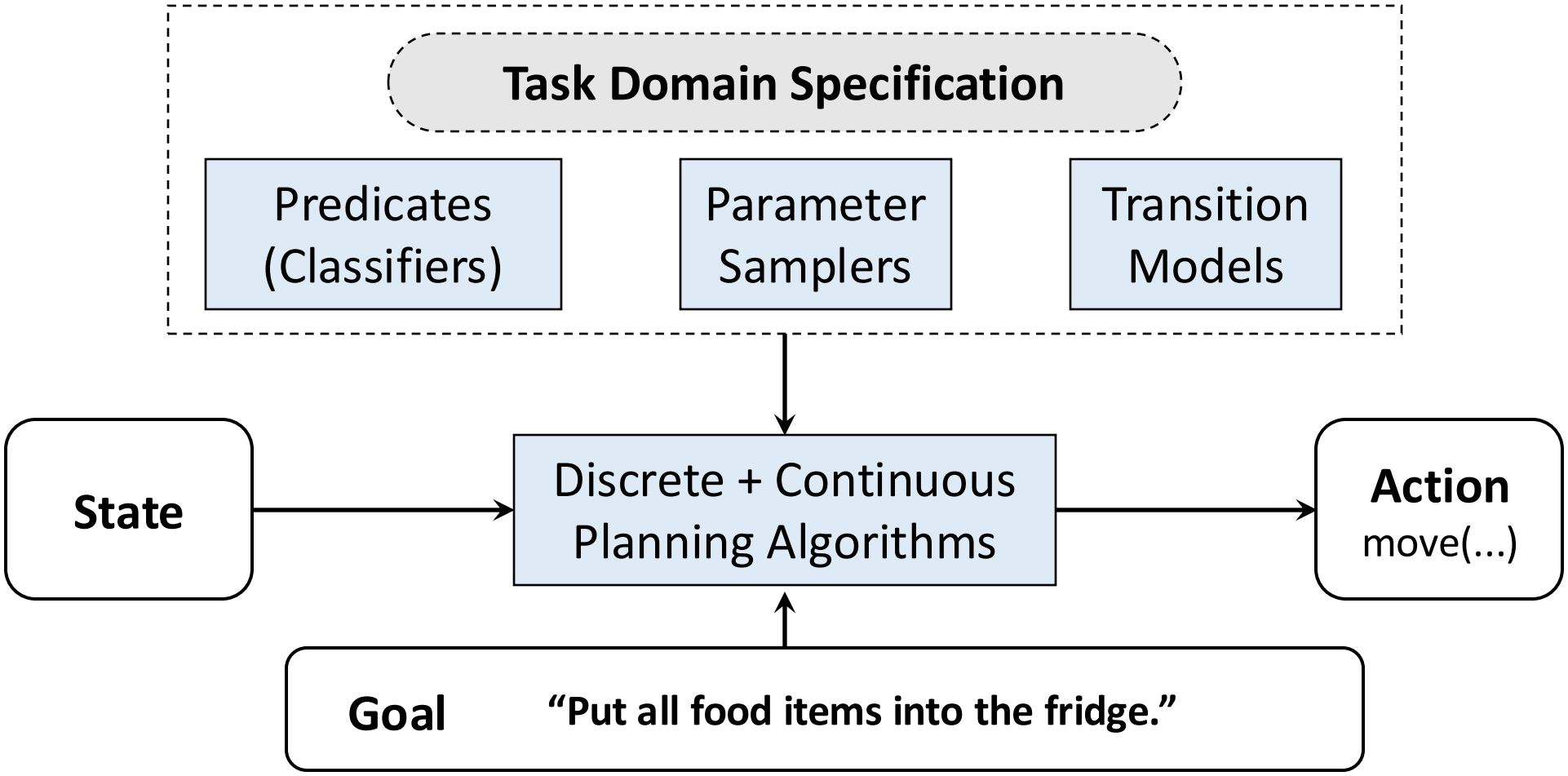


Best when taken together!

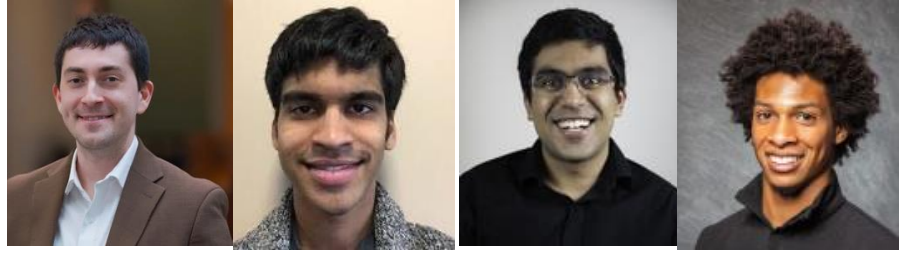
So, let's…

- Start with built-in fundamental learning and inference algorithms
- Build and pre-train general, compositional world models at design time
- Perform runtime inference (planning) to select actions
- Continually and compositionally improve and augment these models at runtime

Basic Elements in Task and Motion Planning



Neuro-symbolic skill learning



Symbolic Operators

```
Operator-PickTool:

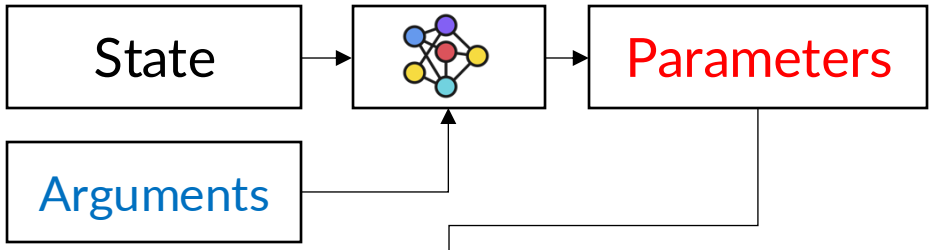
Arguments: [ ⊓, 🤖 ]

Preconditions: {GripperOpen(🤖)}

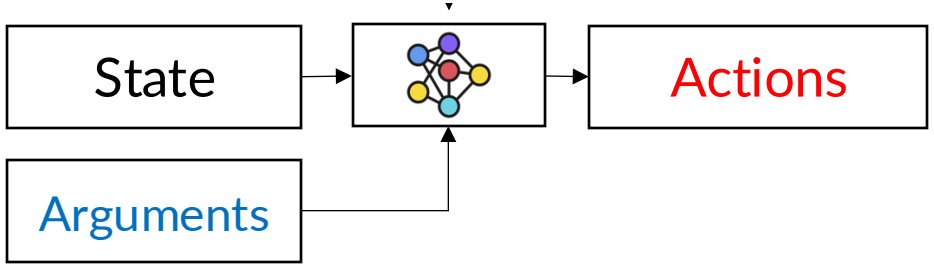
Add effects: {Holding( ⊓, 🤖)
Delete effects: {GripperOpen(🤖)}
```

*Learned with **inductive logic programming** and related techniques*

Neural Parameter Samplers



Neural Parameterized Policies



*Learned with **relational deep learning**, other statistical learning techniques*

Tom Silver, Rohan Chitnis, Nishanth Kumar, Willie McClinton,..., AAAI 2023

Leverage LLMs to get operator sketches

Give a natural-language instruction and set of available objects to the LLM

Ask it to provide a symbolic plan

Bring a hot egg to the table.

Objects: egg, stove, ...
State: on(egg, table), cold(egg), open(microwave), ...

Observe that we don't have operators for some steps in the plan

Ask LLM to suggest an operator sketch

heat-object ? ? ?

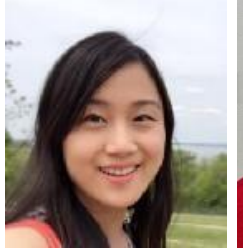






Lio Wong, Jiayuan Mao, Pratyusha Sharma, ..., Jacob Andreas, ICLR 2024

Add a newly learned motor skill to our repertoire by grounding operator sketch

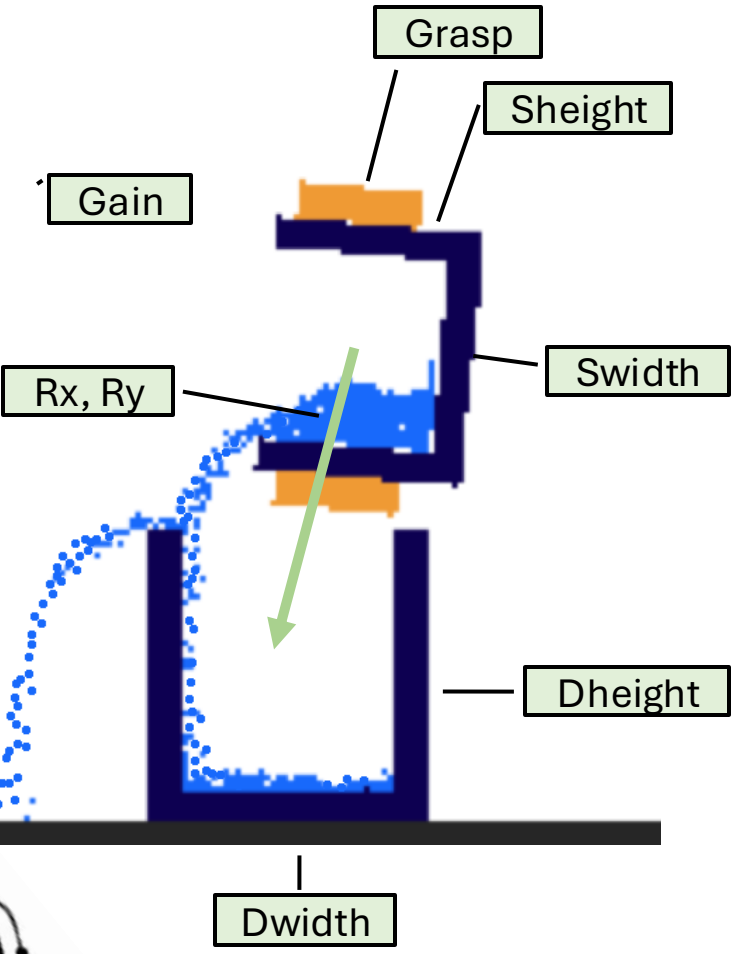



Result: Contains(Dest, Liquid)

Skill: Pour(**Gain**)

Preconditions:

- Contains(Source, Liquid)
- Holding(Source, **Grasp**)
- Shape(Source) = (**Swidth, Sheight**)
- Shape(Dest) = (**Dwidth, Dheight**)
- RelPose(Source, Dest) = (**Rx, Ry**)
- **Constraint(Sw, Sh, Dw, Dh, Rx, Ry, Grasp, Gain)**



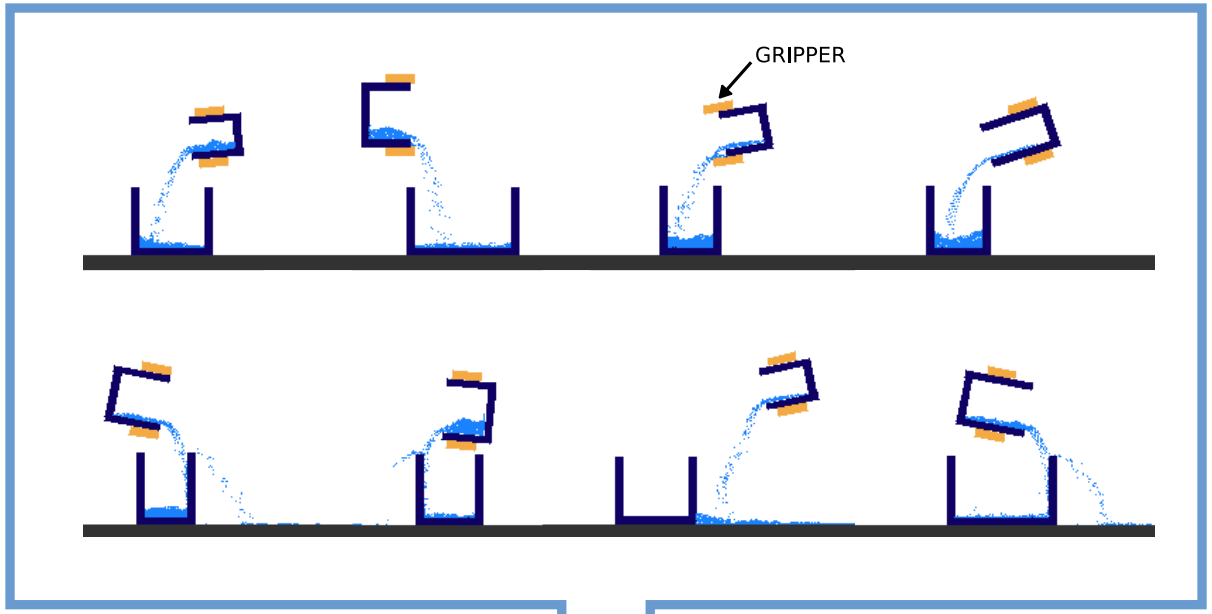
Incrementally add new skills to model



Zi Wang, Caelan Garrett, …, IROS 2018

Learning the operator constraint: supervised training

labeled training data



$$\text{constraint}(\theta) \equiv g(\theta) > 0$$

Sw, Sh, Dw, Dh, Rx, Ry, Grasp, Gain

θ

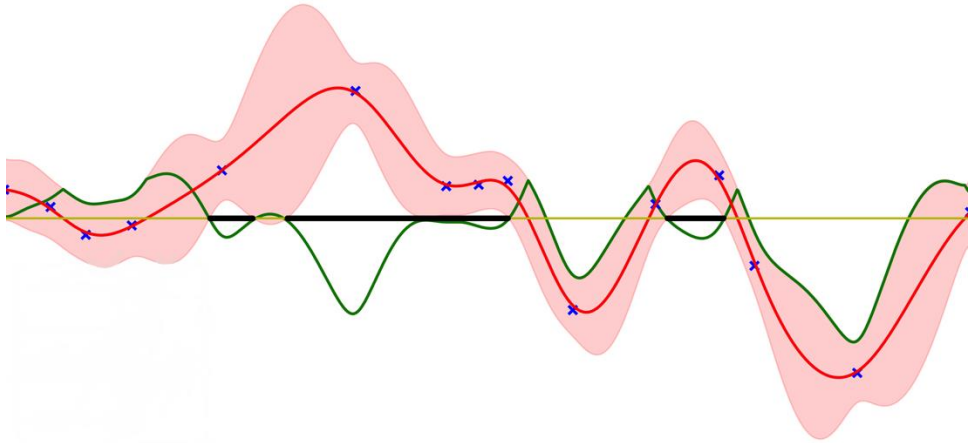
Gaussian Process Regression

score

$g(\theta)$

Active learning: training on real robot is expensive!





Zi Wang, Caelan Garrett, …, IROS 2018

Note that we do grasp and motion planning during data acquisition!

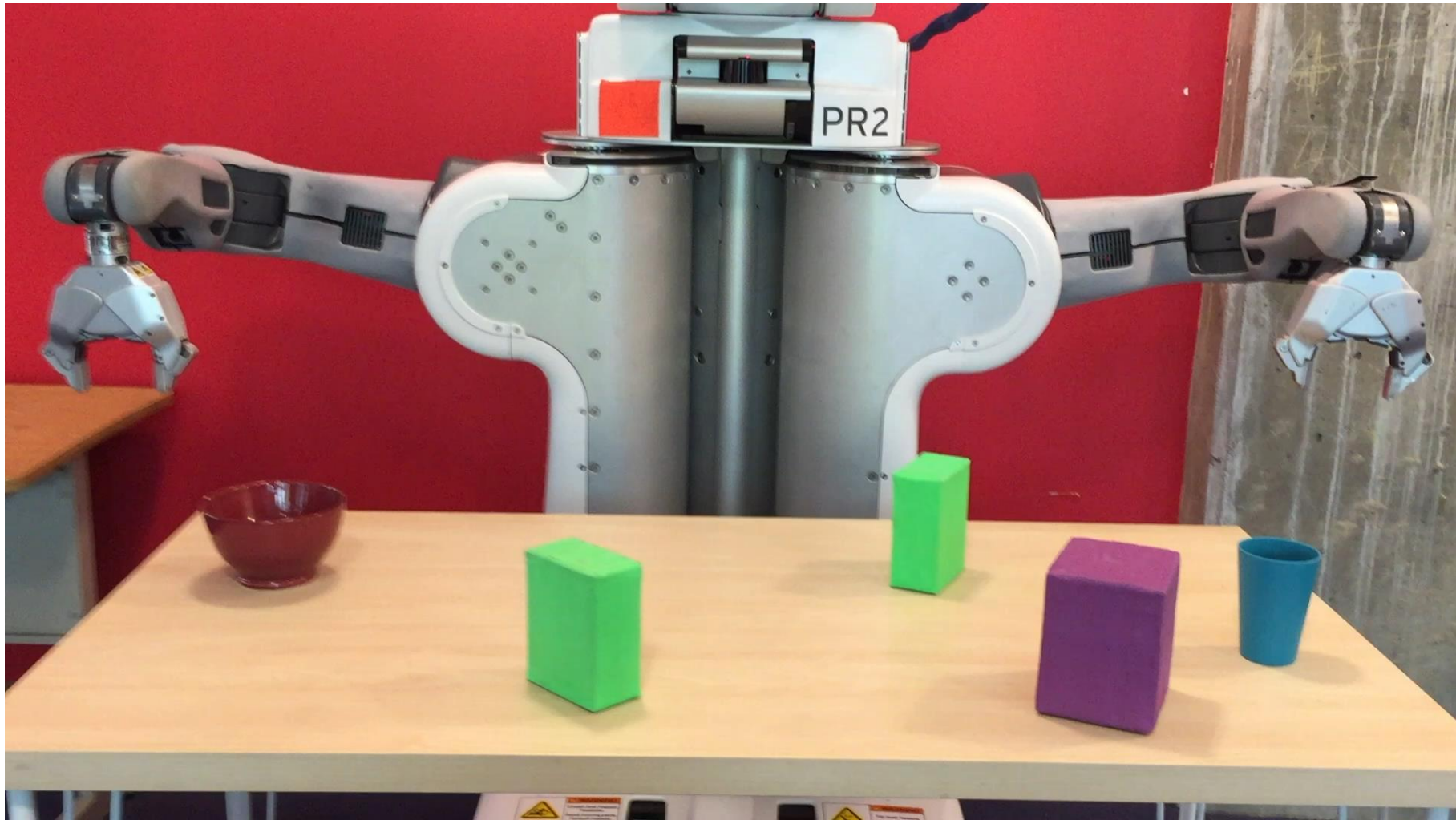


New operation combines smoothly with existing ones

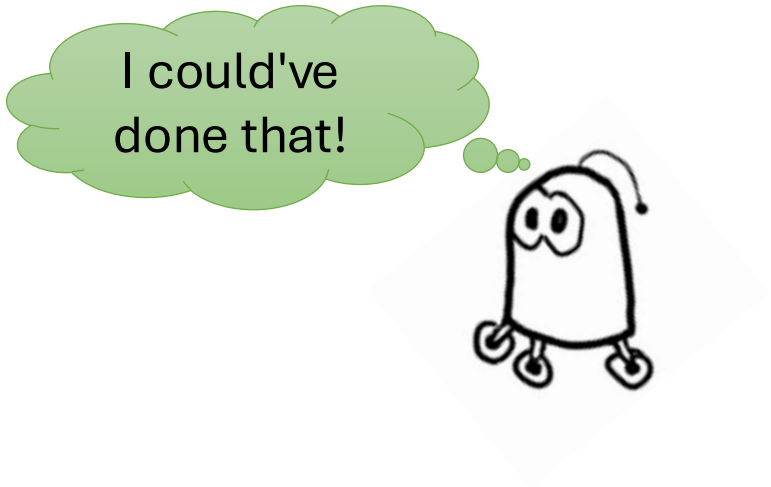
Novel starting arrangement, goal

Given pick/place operators

Learned pour and push



What Can We Learn from One Demonstration?





What Can We Learn from One Demonstration?

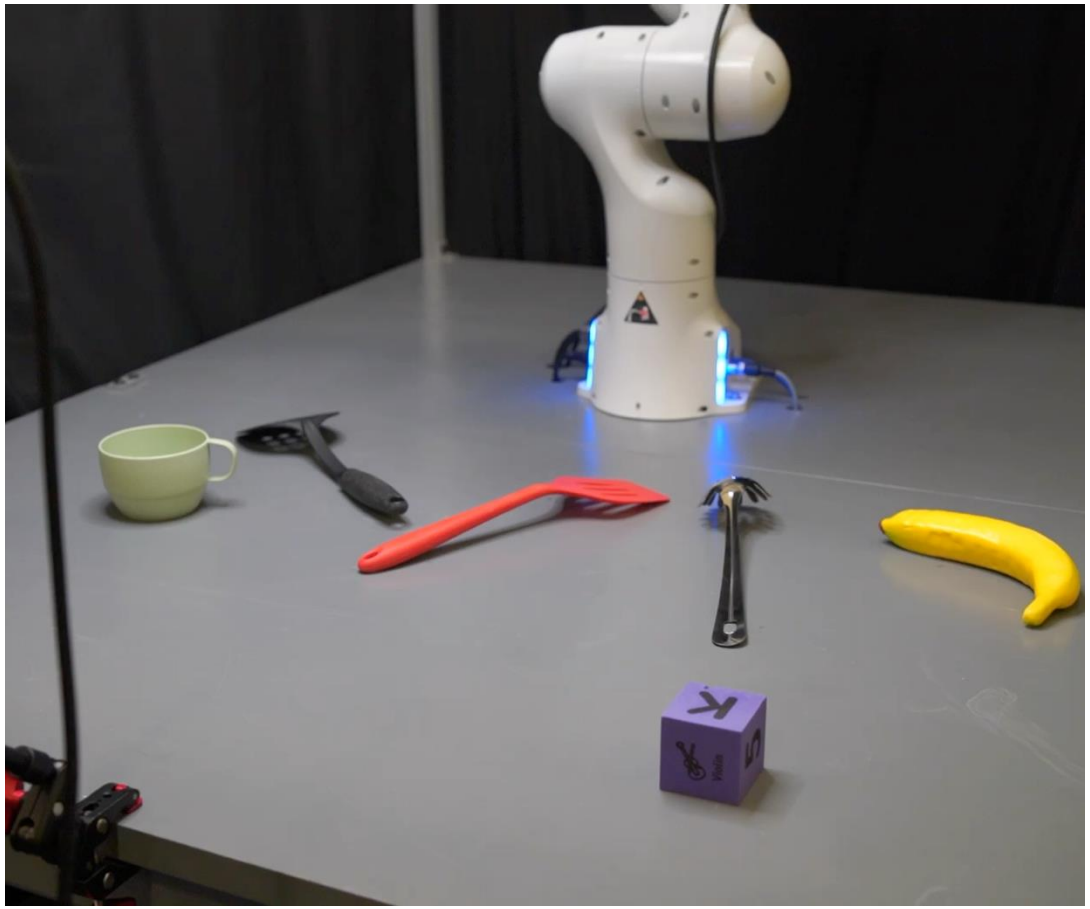
A “**strategy**” for picking up the cylinder:

- Push to rotate
- Exert force on one end so that it tilts
- Move the bucket



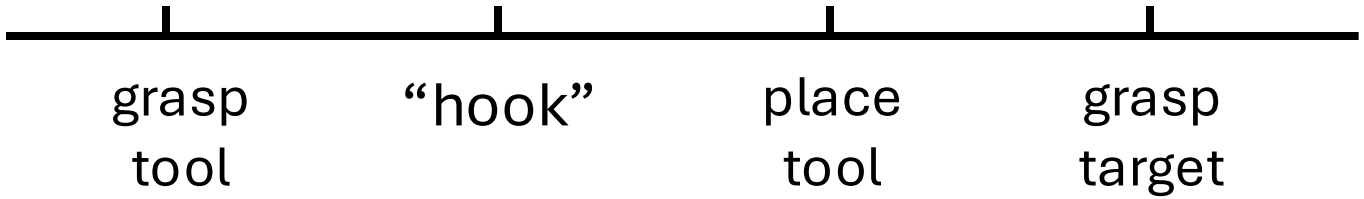
What Can We Learn from One Demonstration?





hook (r: robot, tool: object, target: object)
→ action_sequence

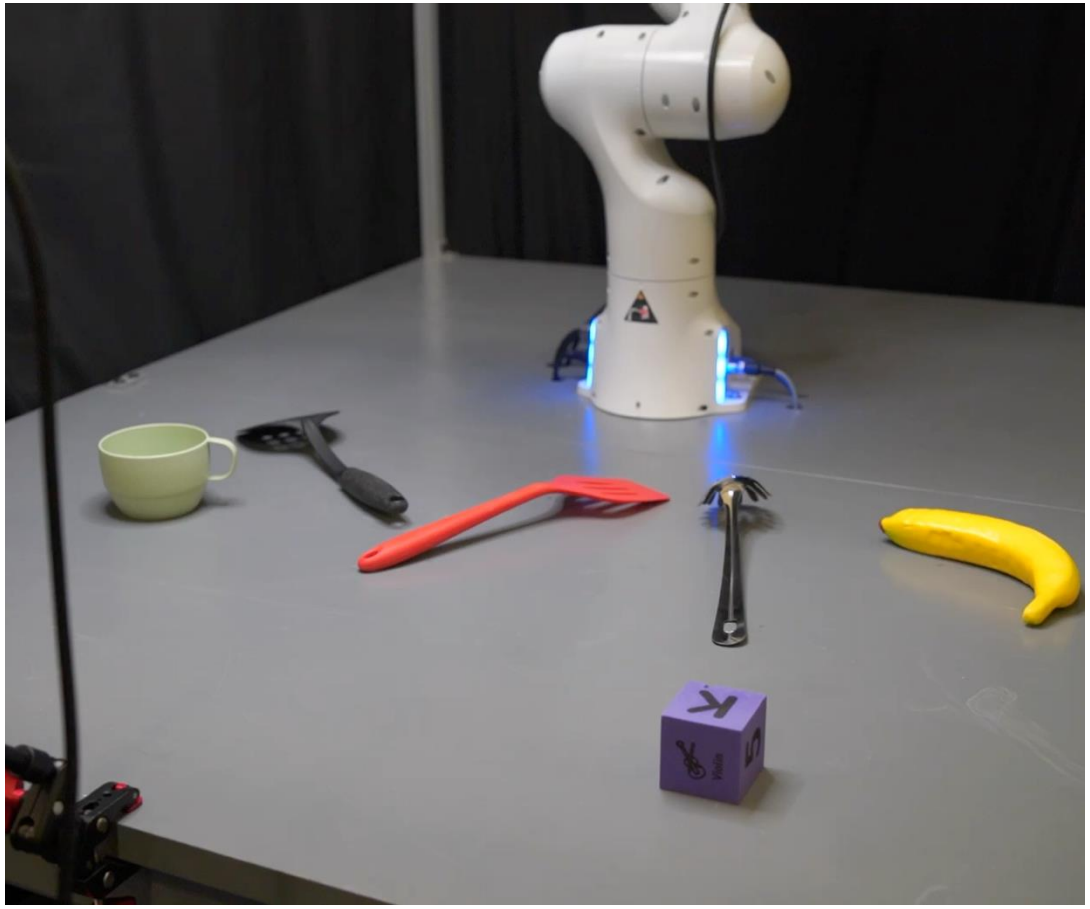
Contact-mode sequence is a type of operator sketch!



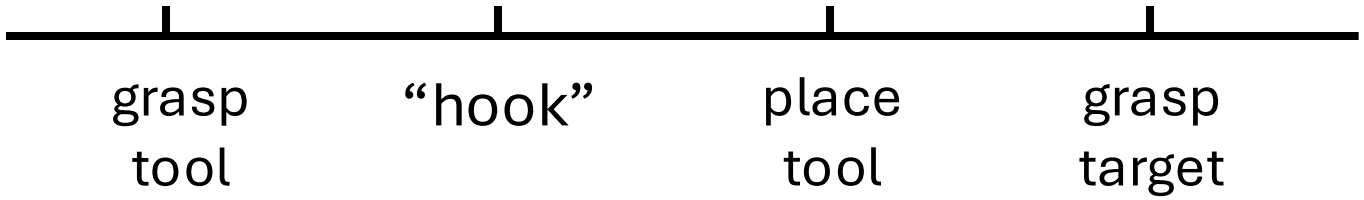
Jiayuan Mao, ..., CORL 2023

The Contact Mode Subgoals in Hook-Using

Key idea: model manipulation "strategies" as
a sequence of subgoals about contacts among objects

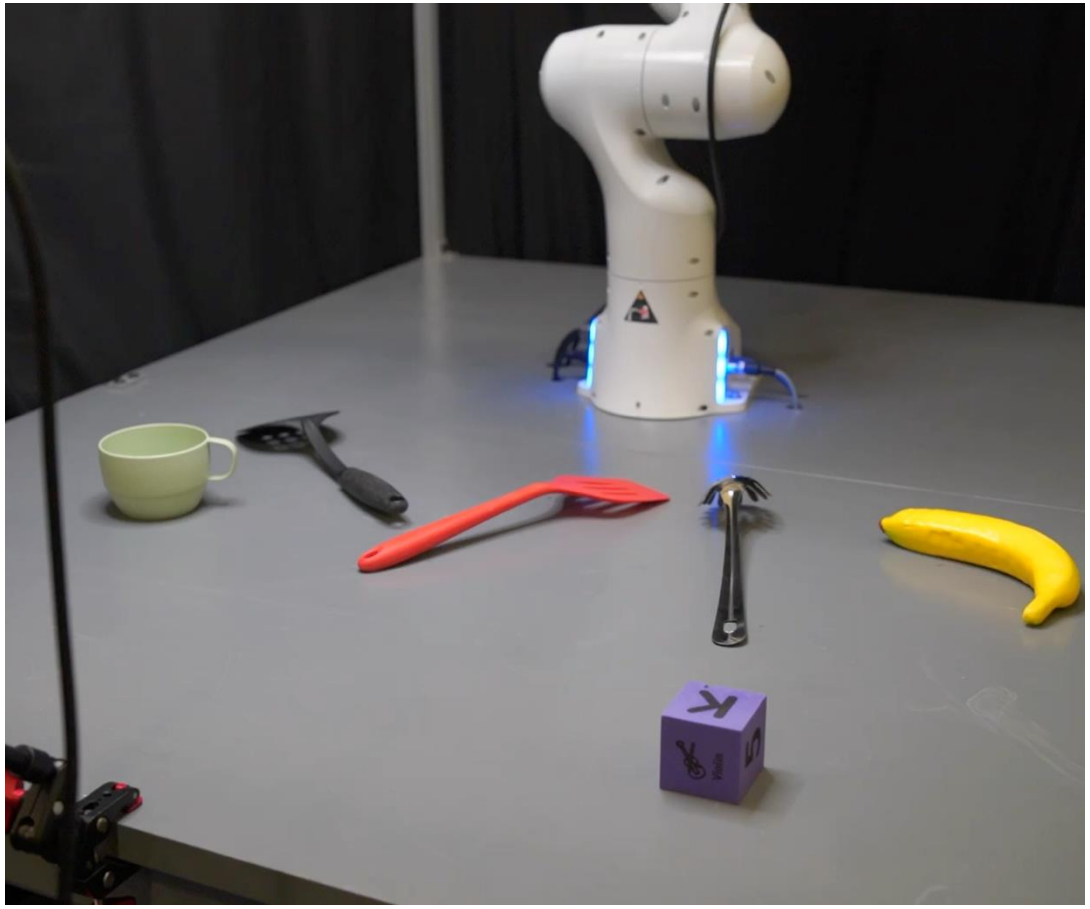


```
behavior hook(target, tool):
  body:
    achieve holding(tool, ?grasp1)
    move-with-contact(tool, target, ?traj)
    place_down(tool)
    grasp(target, ?grasp2)
```

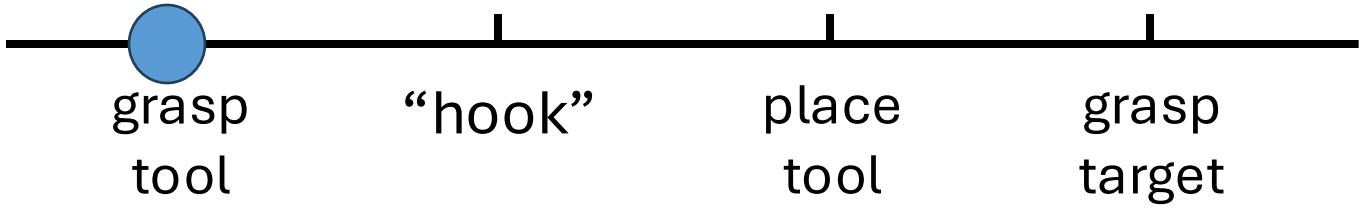


The Contact Mode Subgoals in Hook-Using

Key idea: model manipulation "strategies" as a program of a sequence of subgoals about contacts among objects

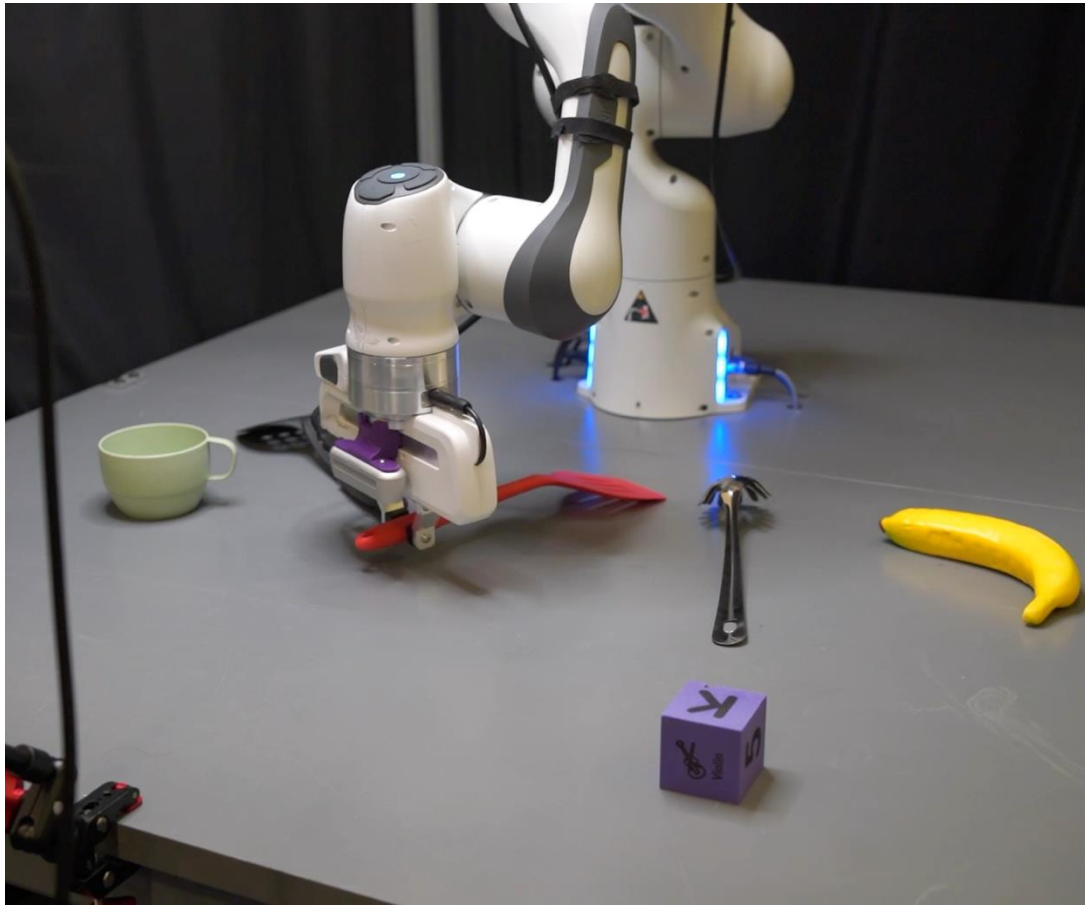


```
behavior hook(target, tool):
  body:
    achieve holding(tool, ?grasp1)
    move-with-contact(tool, target, ?traj)
    place_down(tool)
    grasp(target, ?grasp2)
```

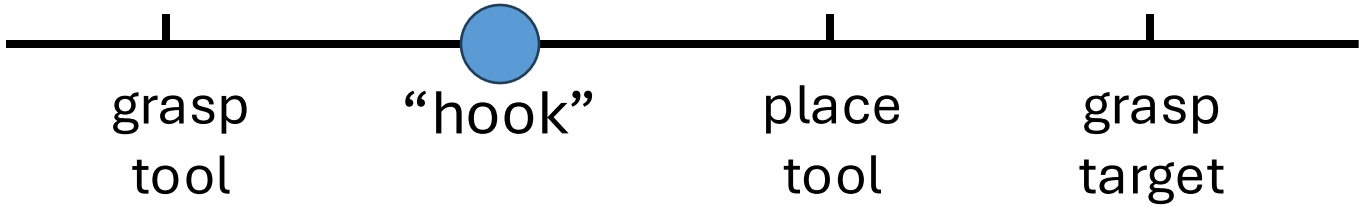


The Contact Mode Subgoals in Hook-Using

Key idea: model manipulation "strategies" as a program of a sequence of subgoals about contacts among objects

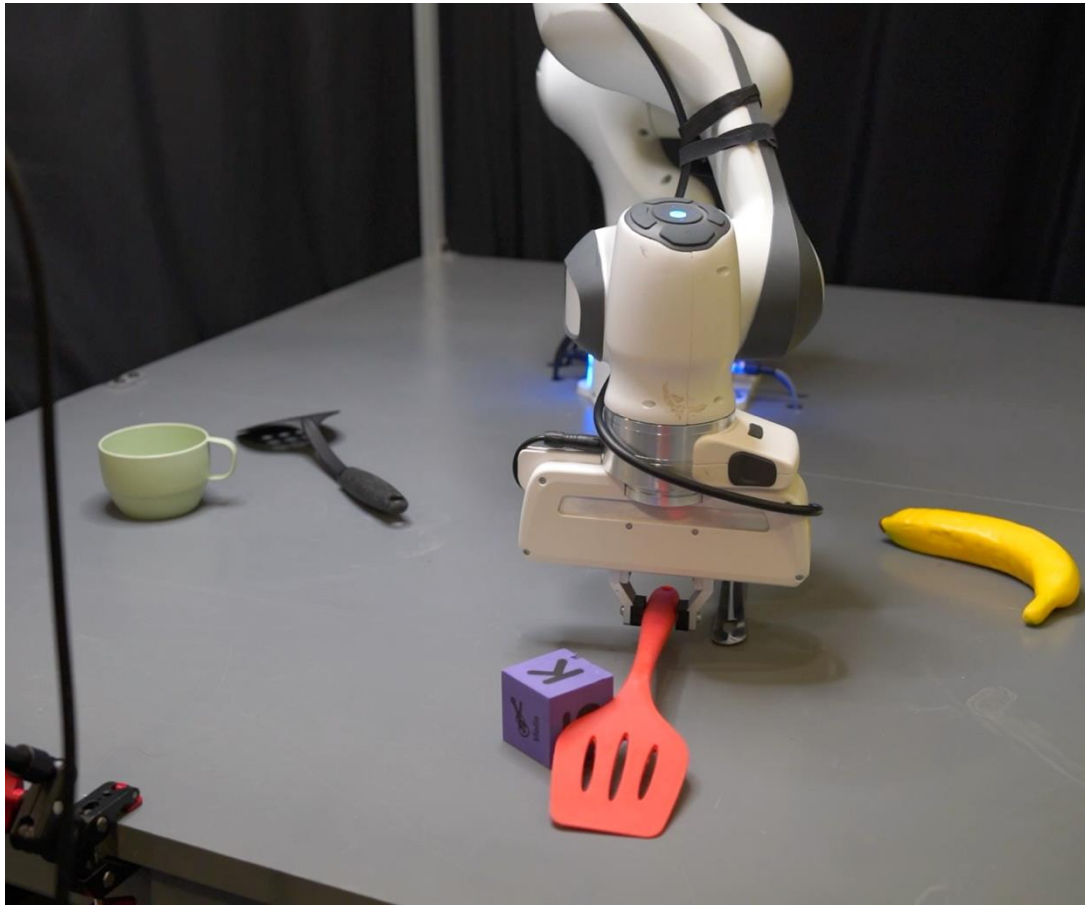


```
behavior hook(target, tool):
  body:
    achieve holding(tool, ?grasp1)
    move-with-contact(tool, target, ?traj)
    place_down(tool)
    grasp(target, ?grasp2)
```

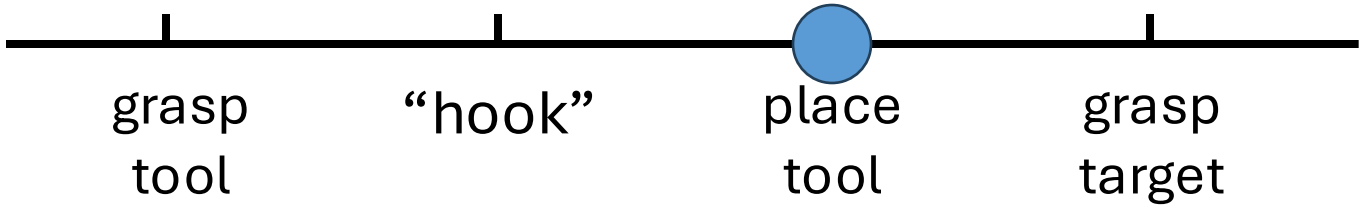


The Contact Mode Subgoals in Hook-Using

Key idea: model manipulation "strategies" as a program of a sequence of subgoals about contacts among objects

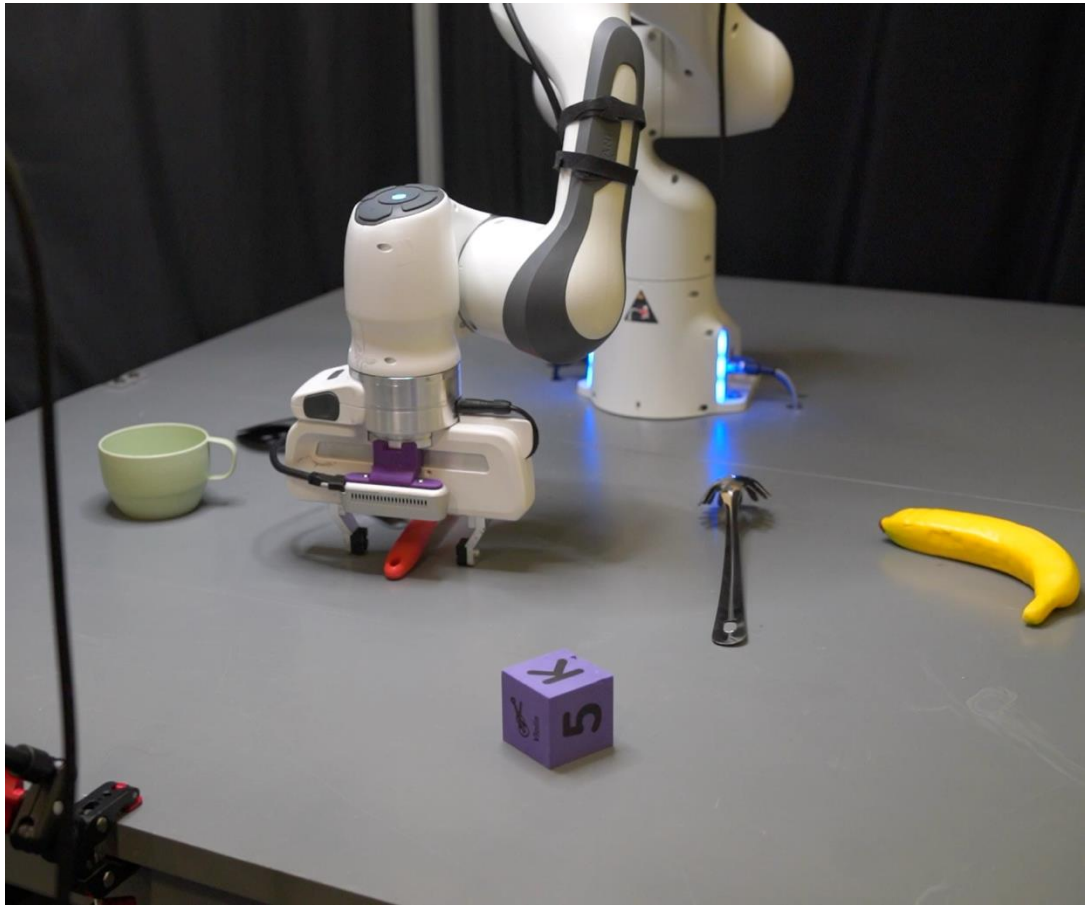


```
behavior hook(target, tool):
  body:
    achieve holding(tool, ?grasp1)
    move-with-contact(tool, target, ?traj)
    place_down(tool)
    grasp(target, ?grasp2)
```

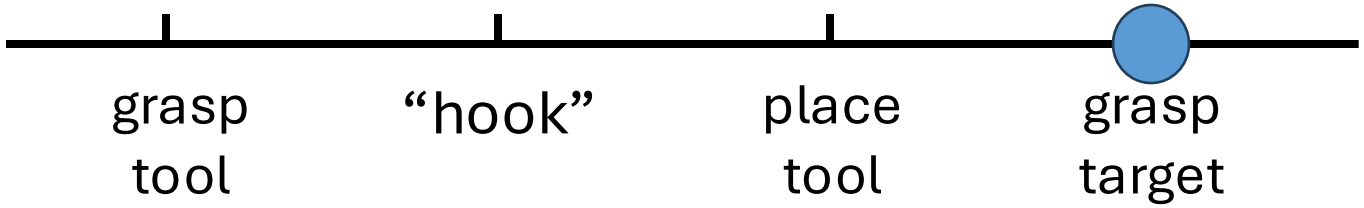


The Contact Mode Subgoals in Hook-Using

Key idea: model manipulation "strategies" as a program of a sequence of subgoals about contacts among objects

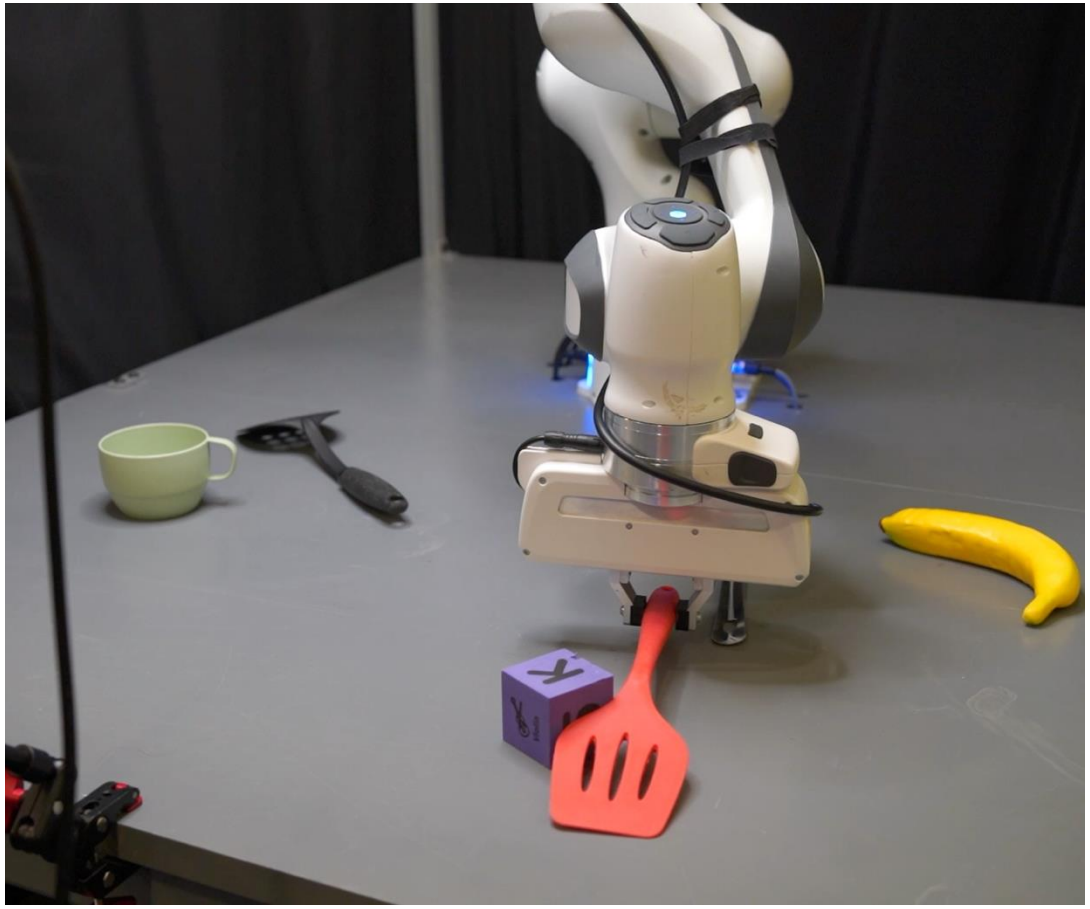


```
behavior hook(target, tool):
  body:
    achieve holding(tool, ?grasp1)
    move-with-contact(tool, target, ?traj)
    place_down(tool)
    grasp(target, ?grasp2)
```



The Contact Mode Subgoals in Hook-Using

Key idea: model manipulation "strategies" as a program of a sequence of subgoals about contacts among objects

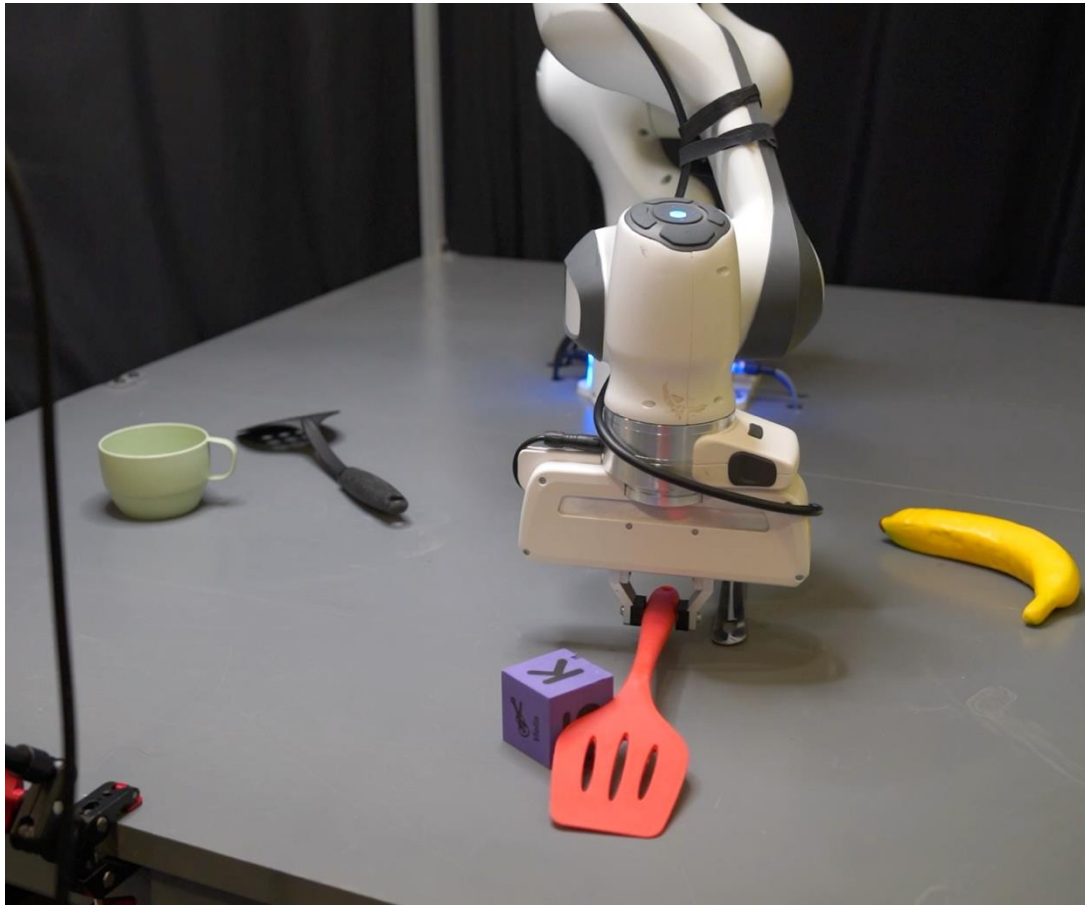


```
behavior hook(target, tool):
  body:
    achieve holding(tool, ?grasp1)
    move-with-contact(tool, target, ?traj)
    place_down(tool)
    grasp(target, ?grasp2)
```

The sequence of subgoals can be learned from a single demonstration!

The Contact Mode Subgoals in Hook-Using

Key idea: model manipulation "strategies" as a program of a sequence of subgoals about contacts among objects



```
behavior hook(target, tool):
  body:
    achieve holding(tool, ?grasp1)
    move-with-contact(tool, target, ?traj)
    place_down(tool)
    grasp(target, ?grasp2)
```

The sequence of subgoals can be learned from a single demonstration!

Concrete values for grasping and trajectories vary across instances

Real Robot Execution of the Learned Strategies

Goal: in(cube, cup)



Generalization to new tools, with no 3D model required.
We apply our structured model and planner based on point cloud inputs.

Planning to practice:

Self-motivated, fully independent, task driven learning

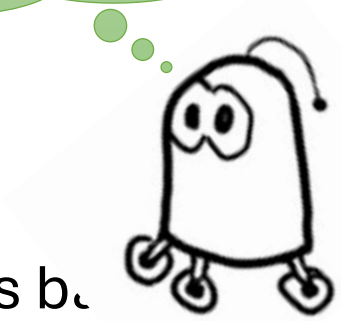





Robot

- Has a set of parameterized skills
- Knows a task distribution
- Is not very good at deploying those skills
- Has some free time on its hands
- Decides to practice:
  - Choose to practice skills that are important for its tasks and that it's b
  - Plans, using its current skills:
    - to decide which skills are important for its tasks
    - to set up the environment so that it can practice the skill
  - Plan to **practice important** skills you think you **can improve at** at!
  - Plan to **use** skills you think you are **good** at!





Boston Dynamics AI INSTITUTE

Nishanth Kumar, Tom Silver, Willie McClinton, Linfeng Zhao, BDAII colleagues, RSS 2024


Cleanup Playroom Environment

So, let's…

- Start with built-in fundamental learning and inference algorithms
- Build and pre-train general, compositional world models at design time
- Perform runtime inference (planning) to select actions
- Continually and compositionally improve and augment our models at runtime

Enormous questions remain!

- Deciding on interfaces:
  - Language and images are good (and have agreed-upon semantics)
  - Will they suffice? What about 3D? Or configuration space?

- How can we get good uncertainty quantification in all models so we can combine them dynamically and effectively online?

- How can we design flexible memory systems for space and objects that play well with other modules?

- How can we address all of these problems when there is substantial, long-term partial observability?

- Etc.

Huge thanks to LIS students and to colleagues!

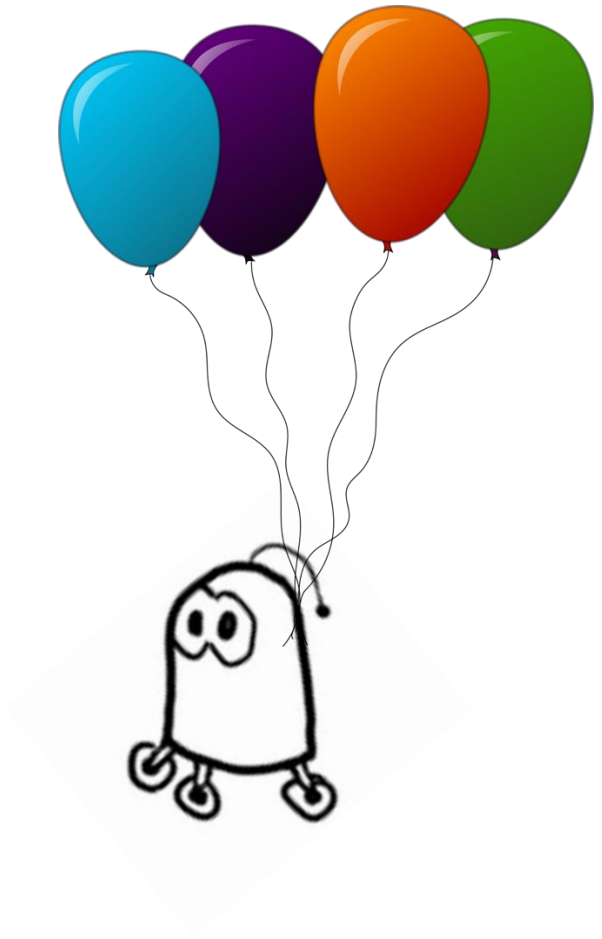
- Aditya Agarwal
- Rohan Chitnis
- Aidan Curtis
- Yilun Du
- Xiaolin Fang
- Caelan Garrett
- Nishanth Kumar
- Jiayuan Mao
- Willie McClinton
- Tom Silver
- Xi Wang
- Zhutian Yang
- Leslie Kaelbling
- Josh Tenenbaum

And collaborators!

We gratefully acknowledge support from
- NSF grant 2214177
- AFOSR grant FA9550-22-1-0249
- ONR MURI grant N00014-22-1-2740
- ARO grant W911NF-23-1-0034
- the MIT-IBM Watson Lab
- the Boston Dynamics Artificial Intelligence Institute
- the MIT Quest for Intelligence

Imitation + Policy + Planning + Learning

is all you need



The End