Learning Hierarchical Goal-Oriented Tasks from Situated Interactive Instruction

Shiwali Mohan

Computer Science and Engineering
University of Michigan, Ann Arbor

June 11th, 2014

Artificial Autonomous Collaborators

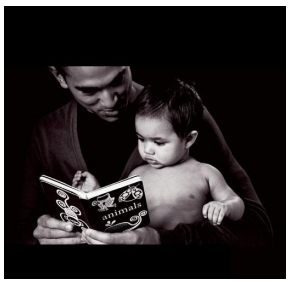
intelligent co-operative behavior in novel environments

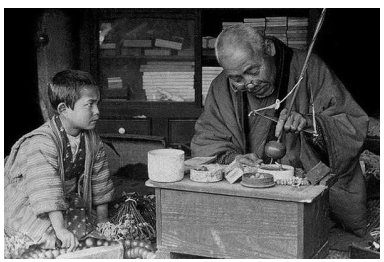
- Diversity in tasks, environments, user preferences
- All usage cannot be predicted at design time
- Designers cannot pre-program all knowledge, skills
- Agents must learn online, continuously, robustly
- How to design adaptive, *taskable* agents?

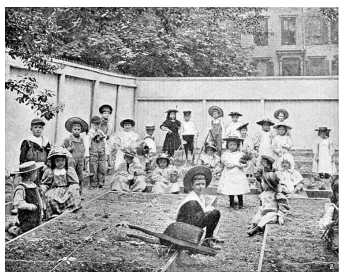
Human Learning

occurs in a variety of social constructs

direct indirect



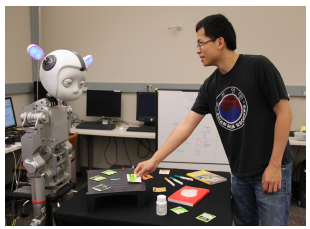



social constructs: reduce perceptual, semantic, learning complexity; encourage discovery; validate learning; facilitate scaffolding; help in knowledge assimilation

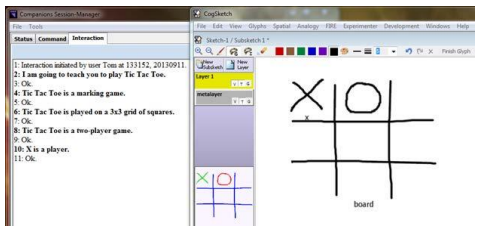
Learning from Human-Agent Interaction

Concept Learning

Grounded Words
Matuszek et al. (2012)



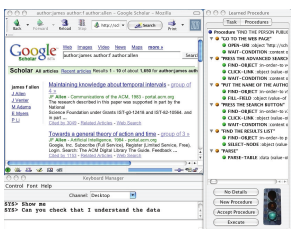
Goals
Chao et al. (2012)



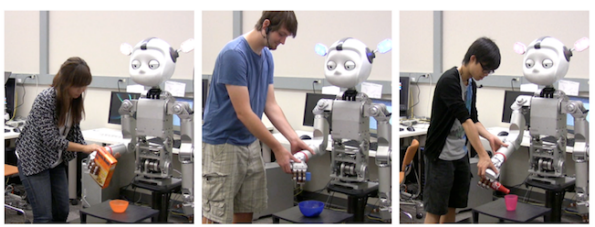
Game Rules
Hinrichs and Forbus (2013)

Categorization
Krause et al. (2014)

Action Learning

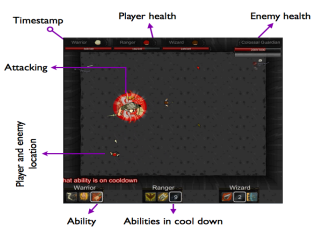


Programs
Allen et al. (2007)



Skills
Cakmak et al. (2011)

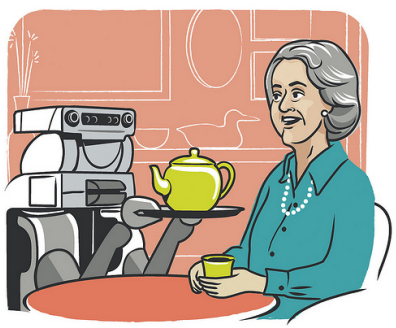
Control
Cantrell et al. (2011)



Game Play
Silva et al. (2013)

Task Learning

- Tasks in domestic environments*
  - *set the table*, *serve dinner* etc.
  - goal achievement
  - hierachical control structure
- Task knowledge
  - what is the structure of the task?
  - how is it performed?
  - when is it appropriate to perform the task?
- Intelligent interactive learning
  - natural interactions with naive users
  - mixed-initiative learning
  - multi-task acquisition
  - fast, informed generalization

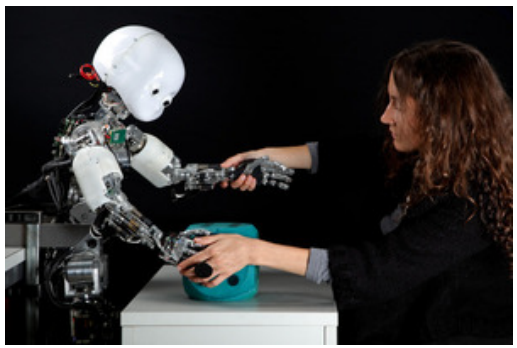


* Cakmak, M. and Takayama, L. Towards a Comprehensive Chore List for Domestic Robots. *Proceedings of the Eighth International Conference on Human-Robot Interaction.* 2013.

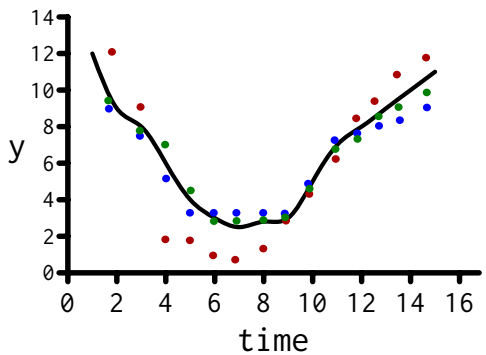
* Mohan, S. *Towards a Comprehensive Chore List for Kitchen Tasks.* 2013. URL: `http://www.shiwali.me/kitchen-data.html`.

Previous Approaches

Learning from Demonstration

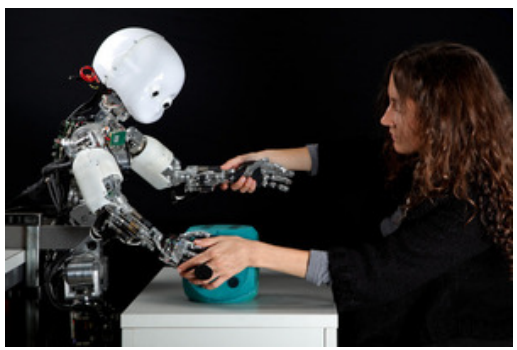


Argall et al. (2009)

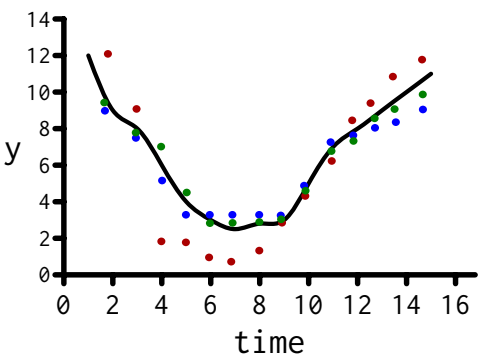


Previous Approaches

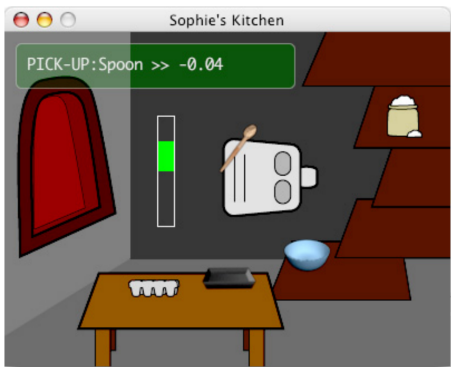
Learning from Demonstration



Argall et al. (2009)



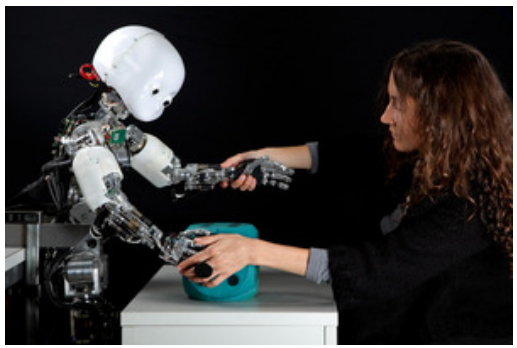
Interactive Reinforcement Learning



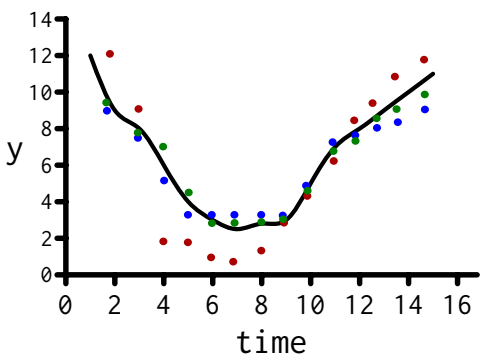
Thomaz et al. (2005)

Previous Approaches

Learning from Demonstration



Argall et al. (2009)



Interactive Reinforcement Learning

Thomaz et al. (2005)

Learning from Dialog

Mericli et al. (2014)

```
User : While landmark 1 is visible
Robot: What should I do in this loop?
User : Turn until landmark 1 is ahead
Robot: I will turn until I am facing
Landmark 1.
Robot: What should I do next?
User : Forward until 0.5 meters from
Landmark 1 max 0.2 meters
...
```

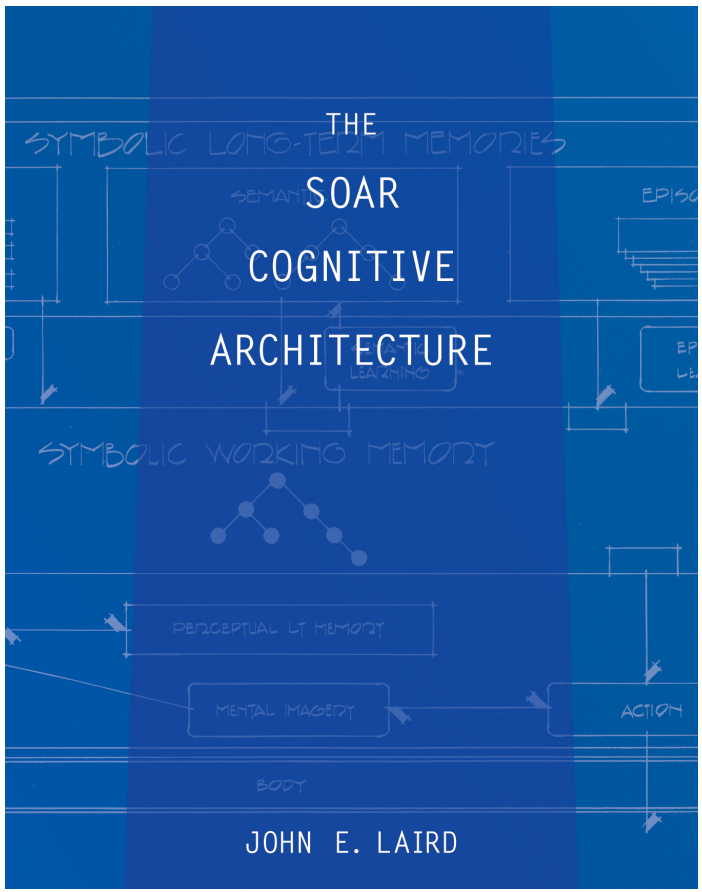
Situated Task-oriented Dialog

- Participants achieve a joint understanding
  - of the task and its components
  - of information and skills
  - Bobrow et al. (1977), Grosz and Sidner (1986), Rich and Sidner (1998), Scheutz et al. (2011)
- Sub-dialogs are aligned with
  - information gathering actions
  - subtasks and subgoals
  - procedures
- Contains information useful for learning

```
E: First you have to remove the
   flywheel.
A: How do I remove the flywheel?
E: First, loosen the two allen setscrews
   holding it to the shaft, then
   pull it off.
A: OK. I can only find one screw.
   Where is the other one?
E: on the hub of the flywheel.
A: Thats the one I found.
   Where is the other one?
...
```

Rosie: A Situated Interactive Instruction Framework

based on Soar Cognitive Architecture



Rosie: Mohan, S., Mininger, A., Kirk, J., and Laird, J. Acquiring Grounded Representations of Words with Situated Interactive Instruction. *Advances in Cognitive Systems* 2, 2012.

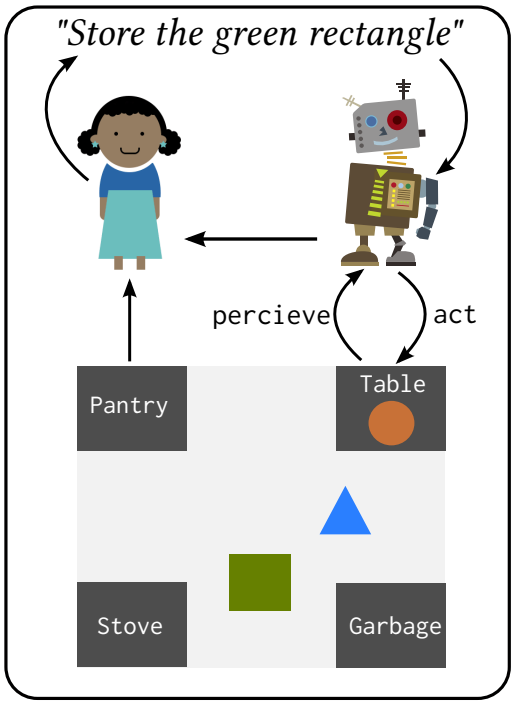
Soar: Laird, J. *The Soar Cognitive Architecture*. MIT Press, 2012.

Computational Problems of SII

Mohan, S., Kirk, J., and Laird, J. A Computational Model for Situated Task Learning with Interactive Instruction. *International Conference on Cognitive Modeling*. 2013.

Computational Problems of SII

Integrative Interaction

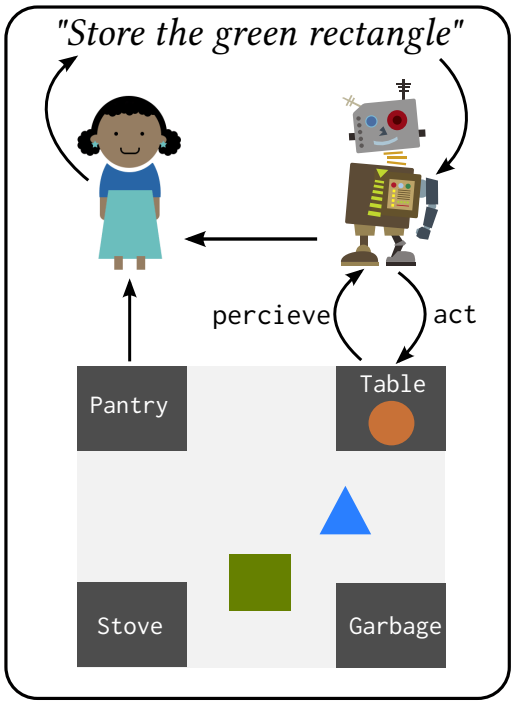


Mohan et al. (2012): interaction model for learning extends Rich and Sidner (1998)

Mohan, S., Kirk, J., and Laird, J. A Computational Model for Situated Task Learning with Interactive Instruction. *International Conference on Cognitive Modeling*. 2013.

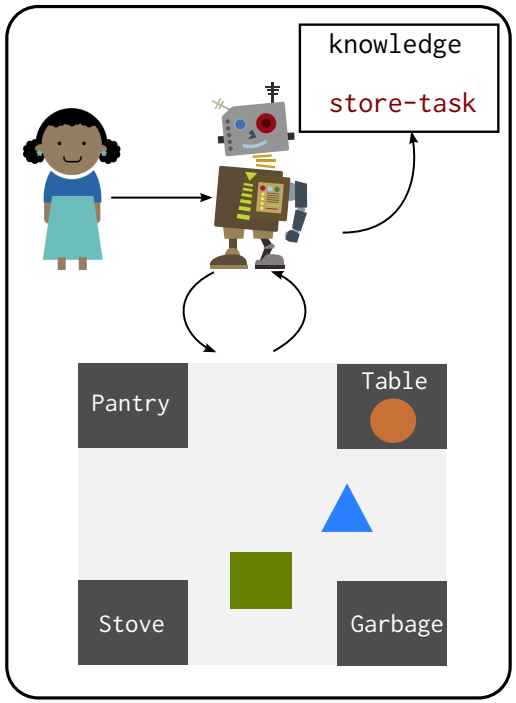
Computational Problems of SII

Integrative Interaction



Mohan et al. (2012): interaction model for learning extends Rich and Sidner (1998)

Task Learning



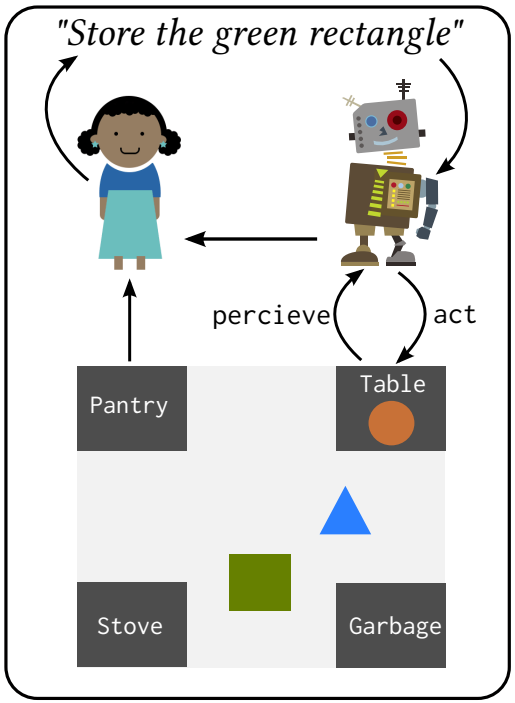
Mohan and Laird (2014): interactive Explanation-based Generalization

---

Mohan, S., Kirk, J., and Laird, J. A Computational Model for Situated Task Learning with Interactive Instruction. *International Conference on Cognitive Modeling*. 2013.

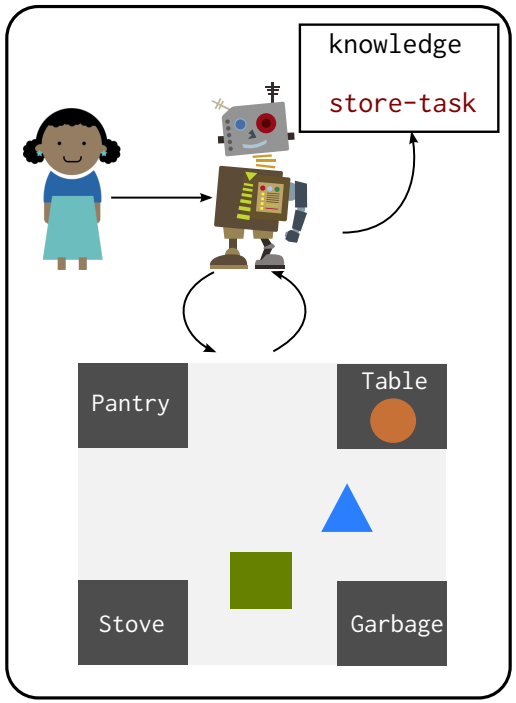
Computational Problems of SII

Integrative Interaction



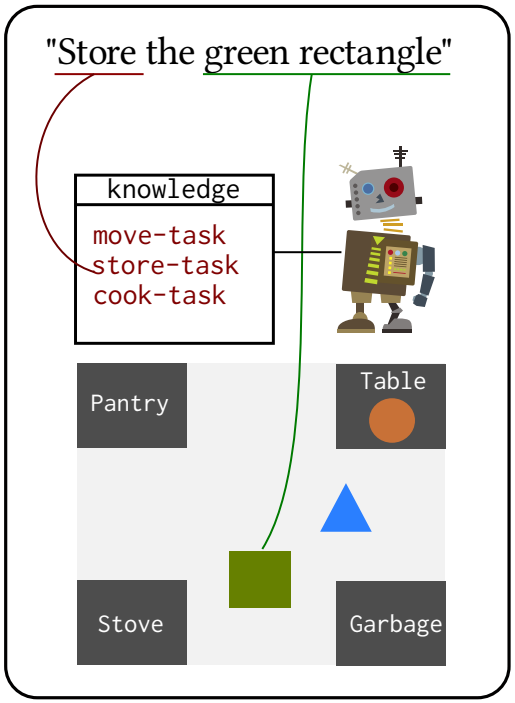
Mohan et al. (2012): interaction model for learning extends Rich and Sidner (1998)

Task Learning



Mohan and Laird (2014): interactive Explanation-based Generalization

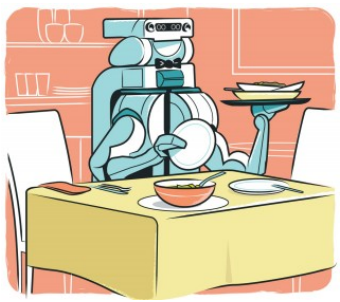
Situated Comprehension

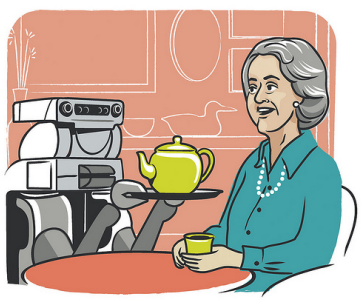


Mohan et al. (2013): Indexical Model for comprehension

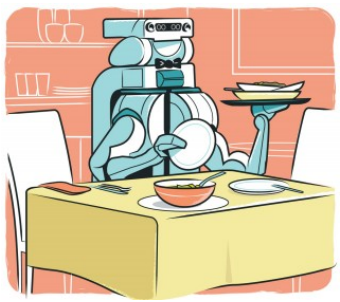
Mohan, S., Kirk, J., and Laird, J. A Computational Model for Situated Task Learning with Interactive Instruction. *International Conference on Cognitive Modeling*. 2013.

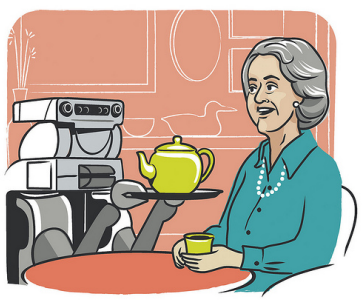
Interactive Task Learning Problem



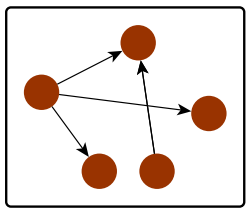


Interactive Task Learning Problem

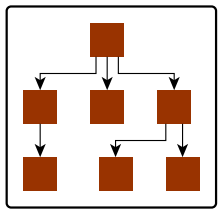




characteristics

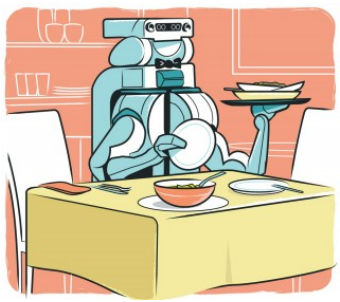


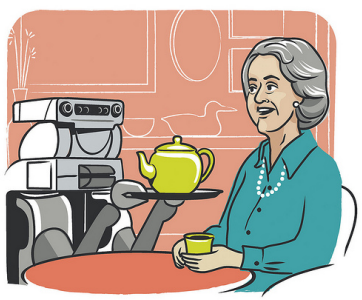
relational goal structure



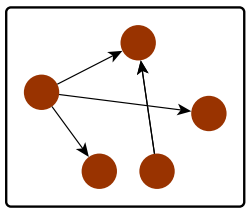
hierarchical decomposition

Interactive Task Learning Problem

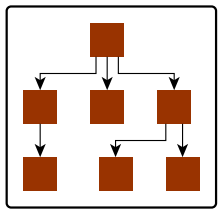




characteristics



relational goal structure

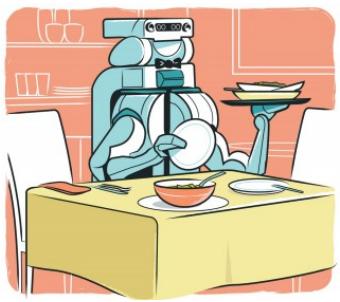


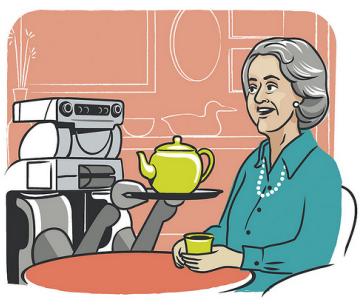
hierarchical decomposition

acquire

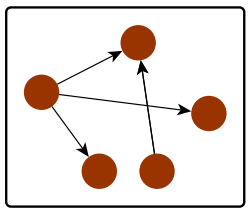
- what?
- how?
- when?

Interactive Task Learning Problem

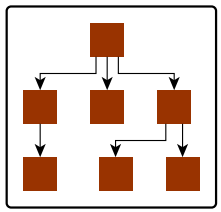




characteristics



relational goal structure



hierarchical decomposition

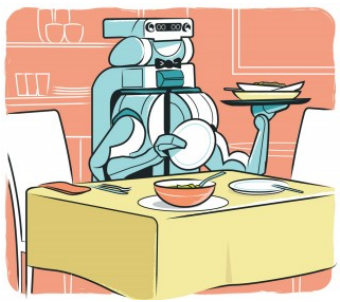
desiderata

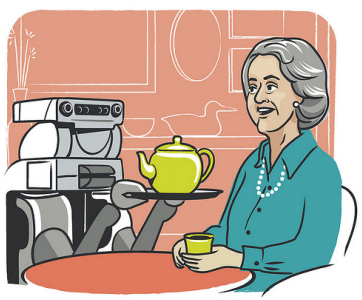
- multi-task learning
- assisted transfer
- fast generalization
- distributed initiative

acquire

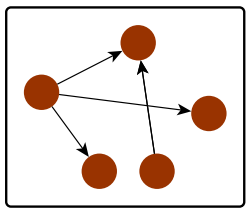
- what?
- how?
- when?

Interactive Task Learning Problem

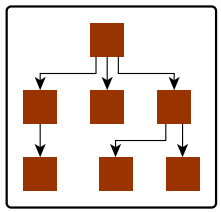




characteristics



relational goal structure



hierarchical decomposition

acquire

- what?
- how?
- when?

desiderata

- multi-task learning
- assisted transfer
- fast generalization
- distributed initiative

approach

- composable, hierarchical representations
- knowledge-rich machine learning - EBL

Task Representation

hierarchical, composable

For the task `store`:

What?

- parameters
  Store the green cylinder.
  `store([o], pantry, in([o],pantry))`
- subtasks
  `store`: `open`, `move [pick-up, put-down]`, `close`
- goal
  `in(O2,pantry)` $\wedge$ `closed(pantry)`

How?

- policy
  if `[state,task]` then `execute([subtask])`
- model
  if `[state,task]` then `[next-state]`

When?

- availability
  if `[state]` then `available(store)`
- termination
  if `[state]` then `terminate(store)`

Explanation-based Learning

Given: concept definition
Given: a positive example
Given: domain theory (prior beliefs, inference)

concept definition:
`is-cup(x)?`

example:
`OBJ1: color(OBJ1,red)` ∧
`is-concave(OBJ1)` ∧
`weight(OBJ1,light)`

domain theory:
[ ] `drinkable-from(x)` → `cup(x)`
[ ] `liftable(x)` ∧ `open(x)` → `drinkable-from(x)`
[ ] `has-part(x, flat-bottom)` → `stable(x)`
[ ] `is-concave(x)` → `open(x)`
[ ] `weight(x,light)` → `liftable(x)`
[ ] `color(x,red)` → `loves(Rosie,x)`
...

DeJong, G. and Mooney, R. Explanation-based Learning: An Alternative View. *Machine Learning* 1.2, 1986, pp. 145–176.

Explanation-based Learning

Given: concept definition
Given: a positive example
Given: domain theory (prior beliefs, inference)

1. construct explanations from examples
   - inference over domain theory
   - why does the example fit the concept definition?

concept definition:
`is-cup(x)?`

example:
`OBJ1: color(OBJ1,red)` $\wedge$
`is-concave(OBJ1)` $\wedge$
`weight(OBJ1,light)`

domain theory:
[ ] `drinkable-from(x)` $\rightarrow$ `cup(x)`
[ ] `liftable(x)` $\wedge$ `open(x)` $\rightarrow$ `drinkable-from(x)`
[ ] `has-part(x, flat-bottom)` $\rightarrow$ `stable(x)`
[ ] `is-concave(x)` $\rightarrow$ `open(x)`
[ ] `weight(x,light)` $\rightarrow$ `liftable(x)`
[ ] `color(x,red)` $\rightarrow$ `loves(Rosie,x)`
...

DeJong, G. and Mooney, R. Explanation-based Learning: An Alternative View. *Machine Learning* 1.2, 1986, pp. 145–176.

Explanation-based Learning

Given: concept definition
Given: a positive example
Given: domain theory (prior beliefs, inference)

1. construct explanations from examples
   - inference over domain theory
   - why does the example fit the concept definition?

concept definition:
`is-cup(x)?`

example:
`OBJ1: color(OBJ1,red)` ∧
`is-concave(OBJ1)` ∧
`weight(OBJ1,light)`

domain theory:
[✓] `drinkable-from(x)` → `cup(x)`
[✓] `liftable(x)` ∧ `open(x)` → `drinkable-from(x)`
[ ] `has-part(x, flat-bottom)` → `stable(x)`
[✓] `is-concave(x)` → `open(x)`
[✓] `weight(x,light)` → `liftable(x)`
[ ] `color(x,red)` → `loves(Rosie,x)`
...

DeJong, G. and Mooney, R. Explanation-based Learning: An Alternative View. *Machine Learning* 1.2, 1986, pp. 145–176.

Explanation-based Learning

Given: concept definition
Given: a positive example
Given: domain theory (prior beliefs, inference)

1. construct explanations from examples
   - inference over domain theory
   - why does the example fit the concept definition?
2. generalization
   - remove unused constraints & features
   - replace constants with variables

concept definition:
`is-cup(x)?`

example:
`OBJ1: color(OBJ1,red)` ∧
`is-concave(OBJ1)` ∧
`weight(OBJ1,light)`

domain theory:
[✓] `drinkable-from(x)` → `cup(x)`
[✓] `liftable(x)` ∧ `open(x)` → `drinkable-from(x)`
[ ] `has-part(x, flat-bottom)` → `stable(x)`
[✓] `is-concave(x)` → `open(x)`
[✓] `weight(x,light)` → `liftable(x)`
[ ] `color(x,red)` → `loves(Rosie,x)`
...

DeJong, G. and Mooney, R. Explanation-based Learning: An Alternative View. *Machine Learning* 1.2, 1986, pp. 145–176.

Explanation-based Learning

Given: concept definition
Given: a positive example
Given: domain theory (prior beliefs, inference)

1. construct explanations from examples
   - inference over domain theory
   - why does the example fit the concept definition?
2. generalization
   - remove unused constraints & features
   - replace constants with variables

concept definition:
`is-cup(x)?`

example:
`OBJ1:` ~~`color(OBJ1,red)`~~ $\wedge$
`is-concave(OBJ1)` $\wedge$
`weight(OBJ1,light)`

domain theory:
[✓] `drinkable-from(x)` $\rightarrow$ `cup(x)`
[✓] `liftable(x)` $\wedge$ `open(x)` $\rightarrow$ `drinkable-from(x)`
[ ] `has-part(x, flat-bottom)` $\rightarrow$ `stable(x)`
[✓] `is-concave(x)` $\rightarrow$ `open(x)`
[✓] `weight(x,light)` $\rightarrow$ `liftable(x)`
[ ] `color(x,red)` $\rightarrow$ `loves(Rosie,x)`
...

DeJong, G. and Mooney, R. Explanation-based Learning: An Alternative View. *Machine Learning* 1.2, 1986, pp. 145–176.

Explanation-based Learning

Given: concept definition
Given: a positive example
Given: domain theory (prior beliefs, inference)

1. construct explanations from examples
   - inference over domain theory
   - why does the example fit the concept definition?
2. generalization
   - remove unused constraints & features
   - replace constants with variables

concept definition:
`is-cup(x)?`

example:
`OBJ1:` ~~`color(OBJ1,red)`~~ $\wedge$
`is-concave([x1])` $\wedge$
`weight([x1],light)`

domain theory:
[✓] `drinkable-from(x)` $\rightarrow$ `cup(x)`
[✓] `liftable(x)` $\wedge$ `open(x)` $\rightarrow$ `drinkable-from(x)`
[ ] `has-part(x, flat-bottom)` $\rightarrow$ `stable(x)`
[✓] `is-concave(x)` $\rightarrow$ `open(x)`
[✓] `weight(x,light)` $\rightarrow$ `liftable(x)`
[ ] `color(x,red)` $\rightarrow$ `loves(Rosie,x)`
...

DeJong, G. and Mooney, R. Explanation-based Learning: An Alternative View. *Machine Learning* 1.2, 1986, pp. 145–176.

Explanation-based Learning

Given: concept definition
Given: a positive example
Given: domain theory (prior beliefs, inference)

1. construct explanations from examples
   - inference over domain theory
   - why does the example fit the concept definition?
2. generalization
   - remove unused constraints & features
   - replace constants with variables

concept definition:
`is-cup(x)?`

example:
`OBJ1:` ~~`color(OBJ1,red)`~~ $\wedge$
`is-concave([x1])` $\wedge$
`weight([x1],light)`

domain theory:
[✓] `drinkable-from(x)` $\rightarrow$ `cup(x)`
[✓] `liftable(x)` $\wedge$ `open(x)` $\rightarrow$ `drinkable-from(x)`
[ ] `has-part(x, flat-bottom)` $\rightarrow$ `stable(x)`
[✓] `is-concave(x)` $\rightarrow$ `open(x)`
[✓] `weight(x,light)` $\rightarrow$ `liftable(x)`
[ ] `color(x,red)` $\rightarrow$ `loves(Rosie,x)`
...

DeJong, G. and Mooney, R. Explanation-based Learning: An Alternative View. *Machine Learning* 1.2, 1986, pp. 145–176.

interactive EBL for Task Learning

Given: concept definition → task representation (policy, availability etc.)
~~*Given*: a positive example~~ ← acquired interactively or by exploration
Given: domain theory → primitive and learned task models

1. construct explanations from examples
   - inference over domain theory
   - why does the example fit the concept definition?
2. generalization
   - remove unused constraints & features ← guided by instruction, exploration
   - replace constants with variables ← guided by the structure of dialog

---

Mohan, S. and Laird, J. Learning Goal-Oriented Hierarchical Tasks from Situated Interactive Instruction. *to appear in the Proceedings of the Twenty Eighth AAAI conference on Artificial Intelligence.* 2014.

Implementation

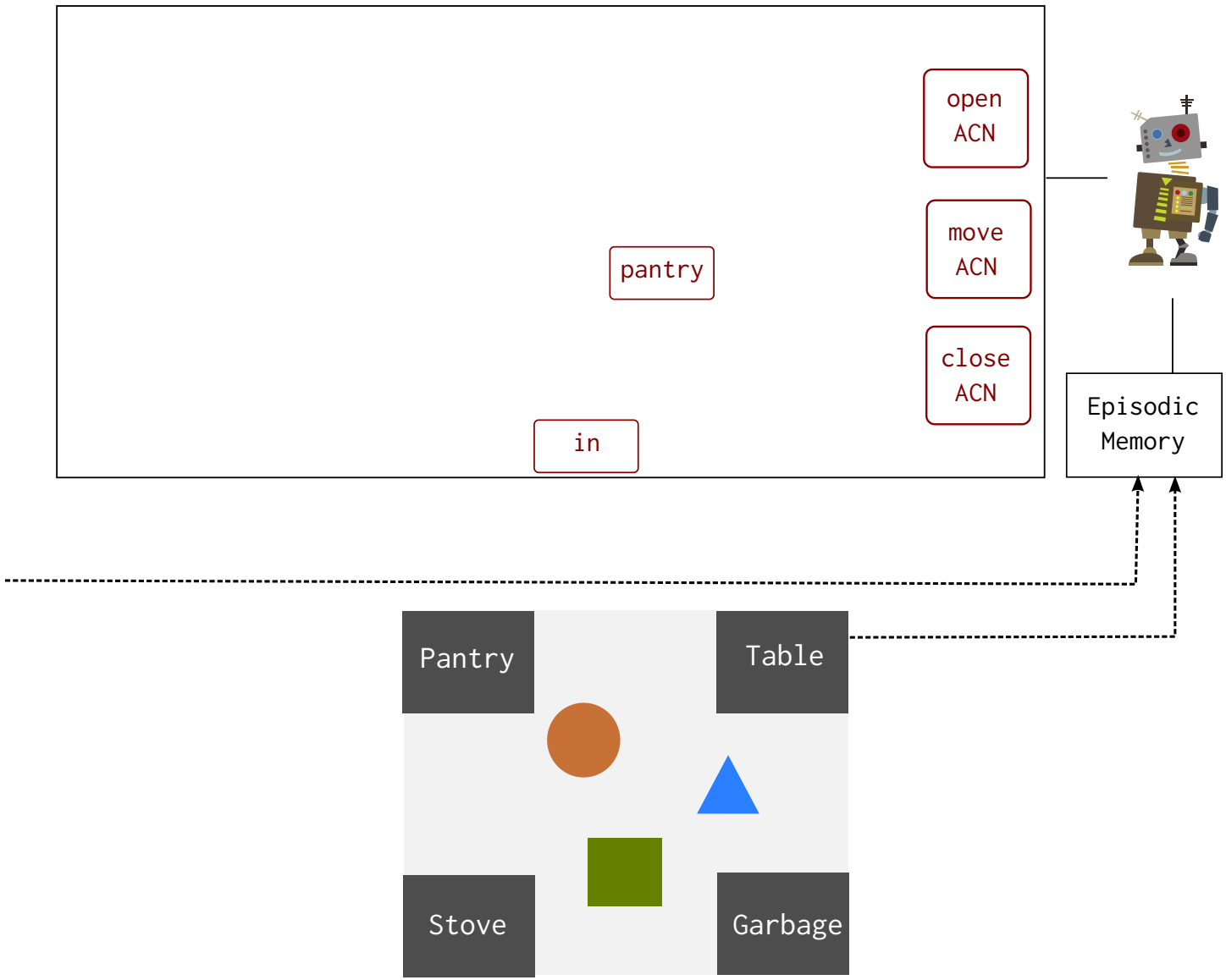
Specific to general learning

1. (interactive) Acquire a specific example of how to execute a task.
2. (EBL) Generalize the specific experience

uses several architectural components: semantic memory, episodic memory, procedural memory, decision processes, chunking

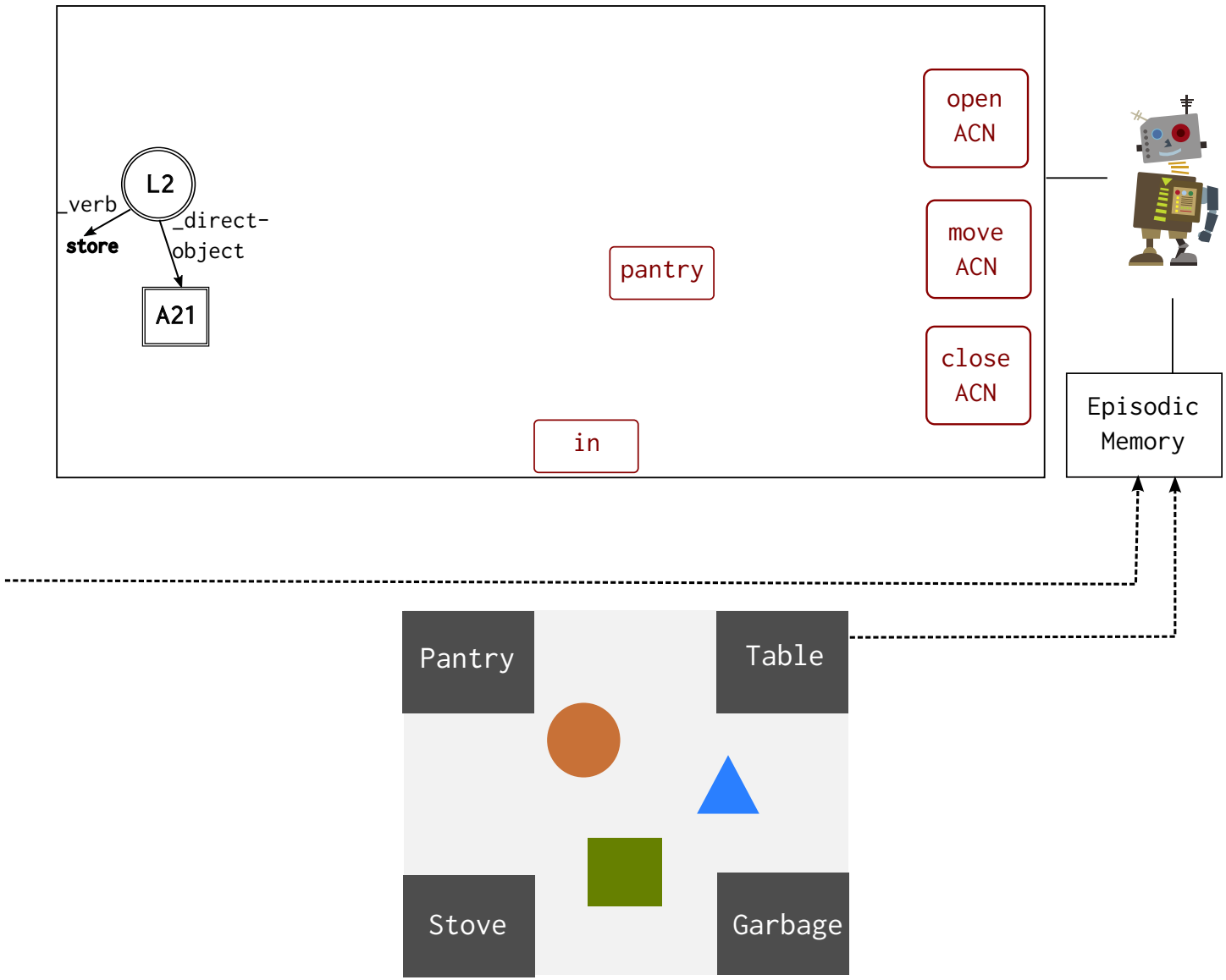
Interactive Example Execution

Interaction trace
Instructor: Store the green rectangle.



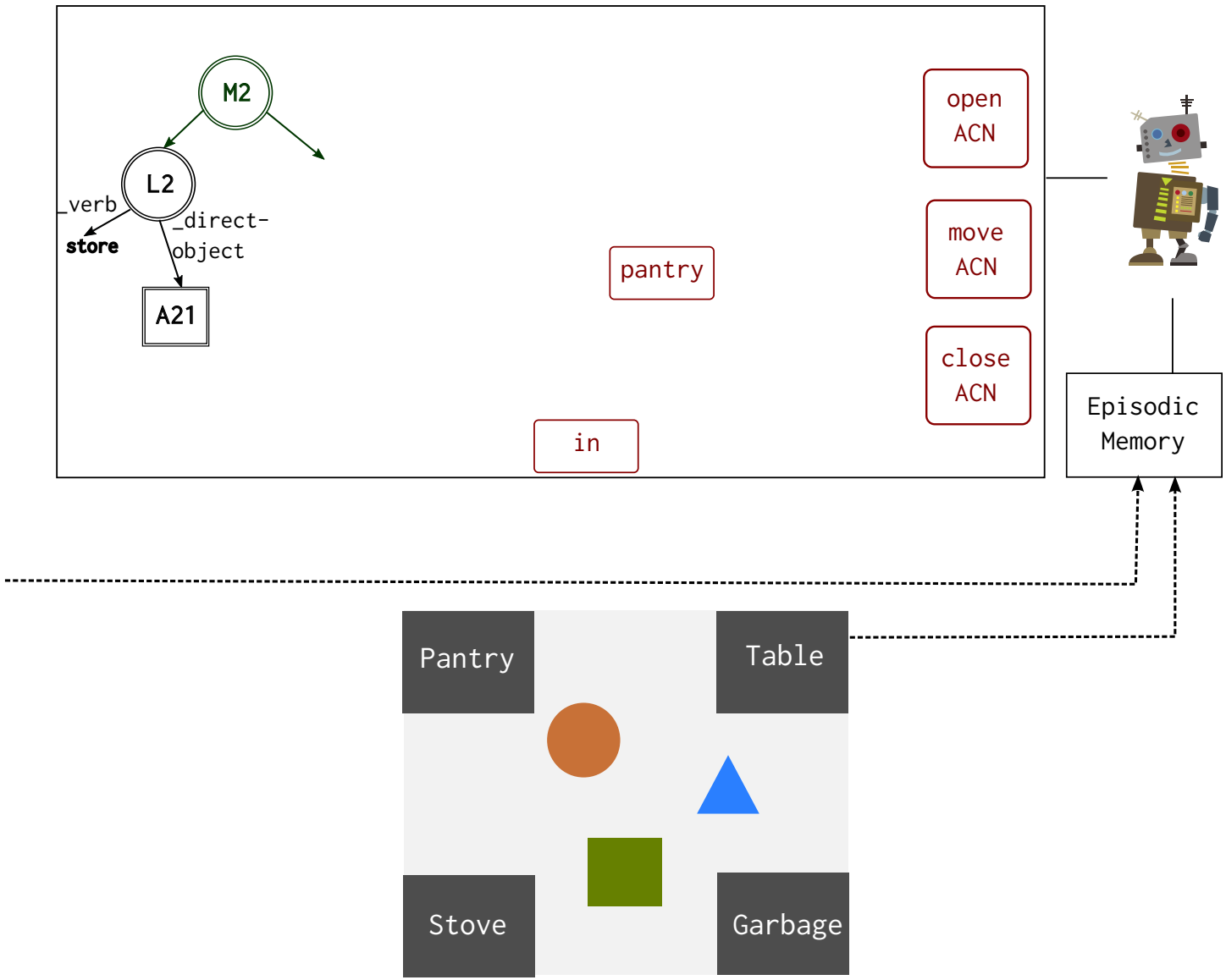
Interactive Example Execution

Interaction trace
Instructor: Store the green rectangle.



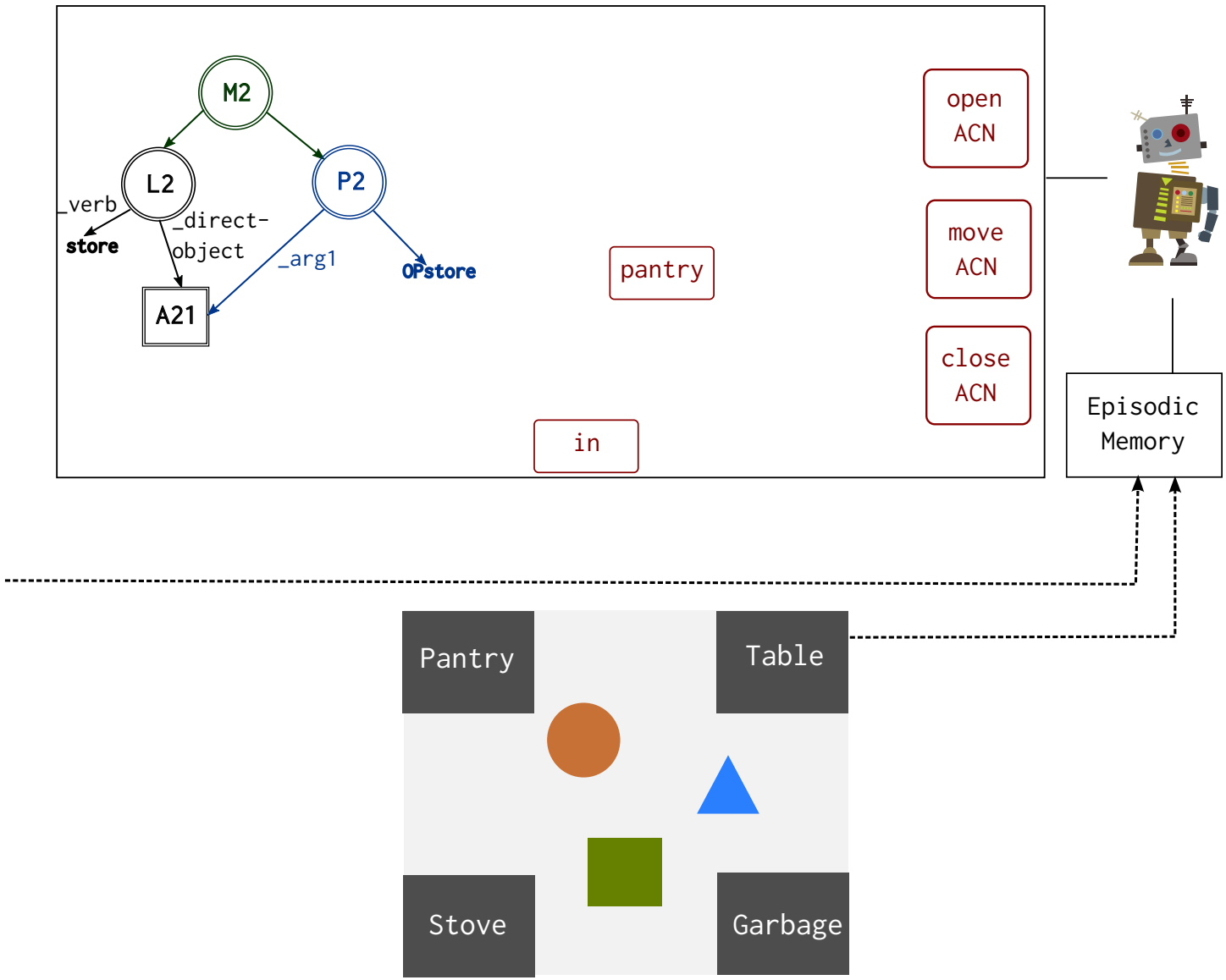
Interactive Example Execution

Interaction trace
Instructor: Store the green rectangle.



Interactive Example Execution

Interaction trace
Instructor: Store the green rectangle.

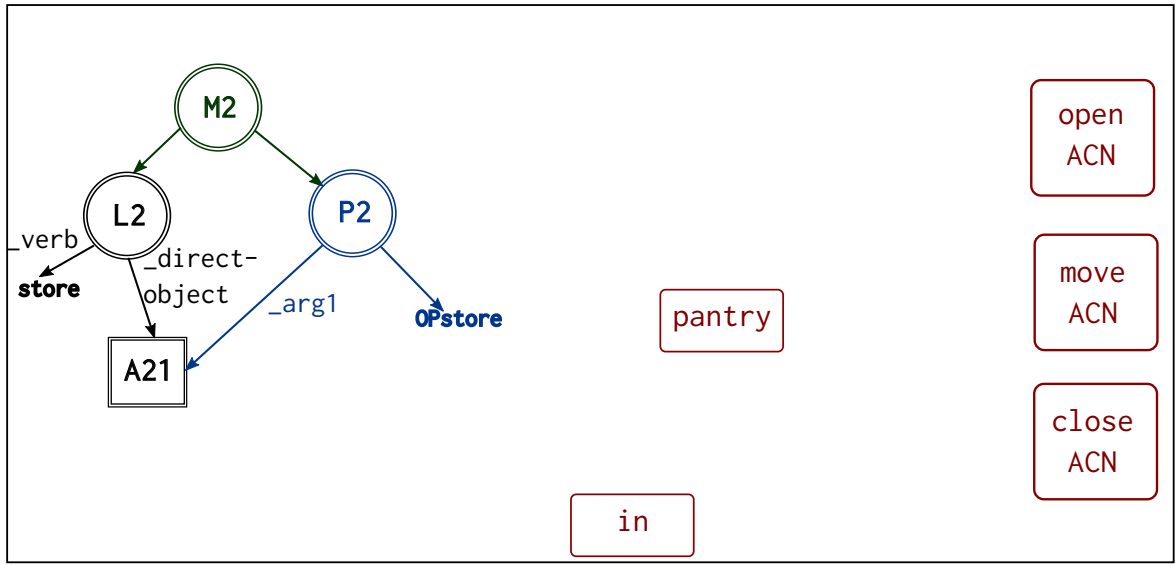


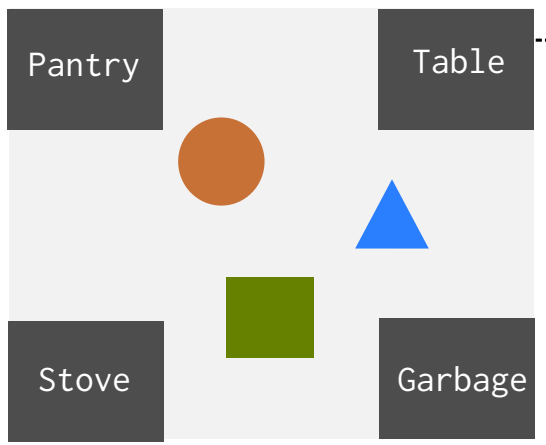
Interactive Example Execution

Interaction trace

Instructor: Store the green rectangle.

Agent: What is the goal of the action?





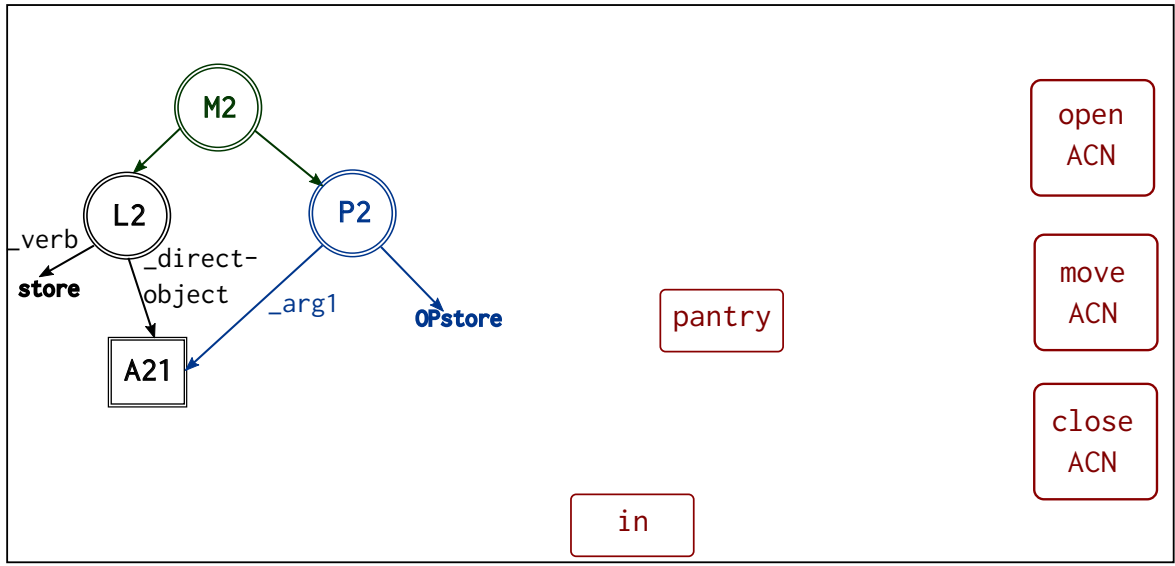
Interactive Example Execution

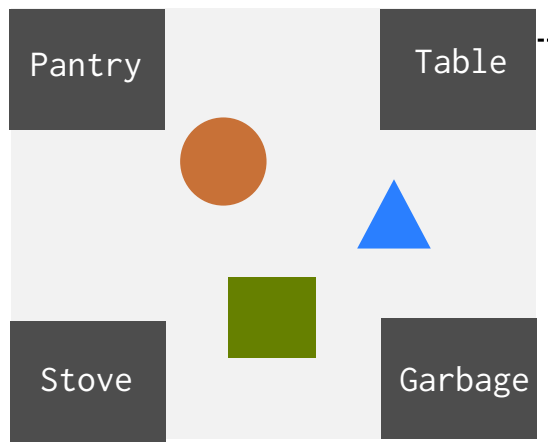
Interaction trace

Instructor: Store the green rectangle.

Agent: What is the goal of the action?

Instructor: The goal is the green rectangle in the pantry and the pantry closed.





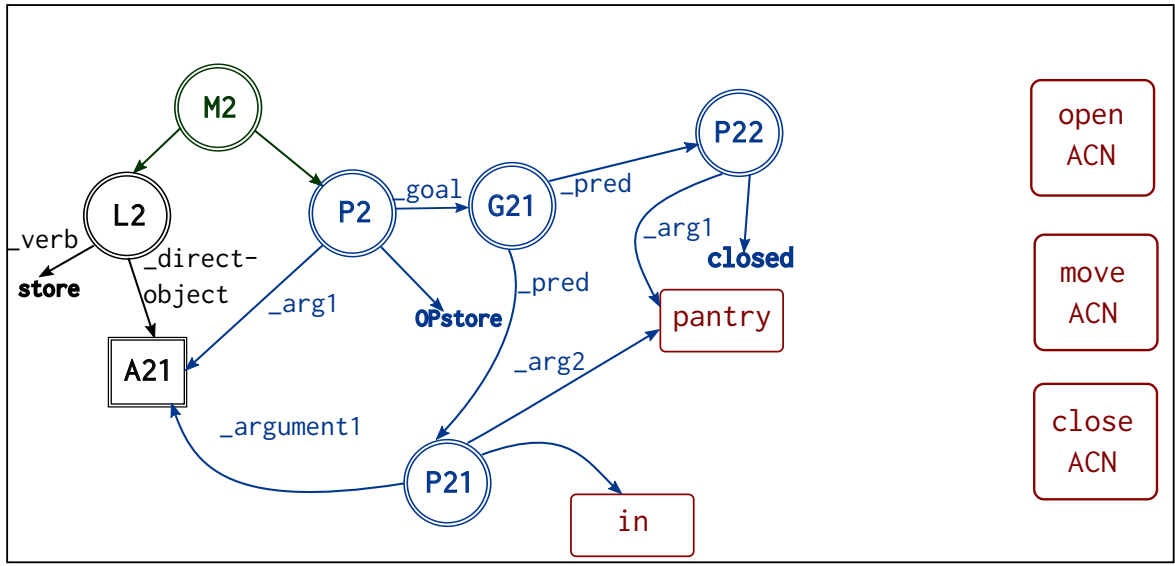
Interactive Example Execution

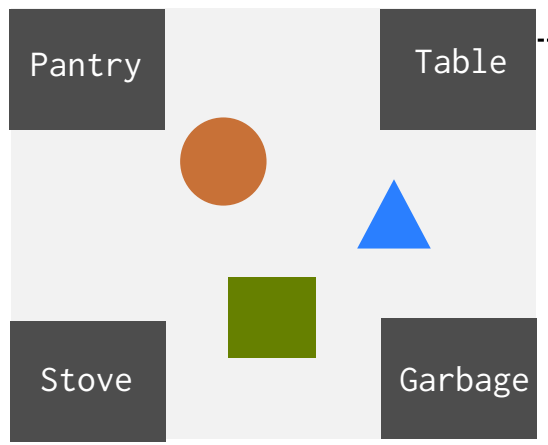
Interaction trace

Instructor: Store the green rectangle.

Agent: What is the goal of the action?

Instructor: The goal is the green rectangle in the pantry and the pantry closed.





Interactive Example Execution

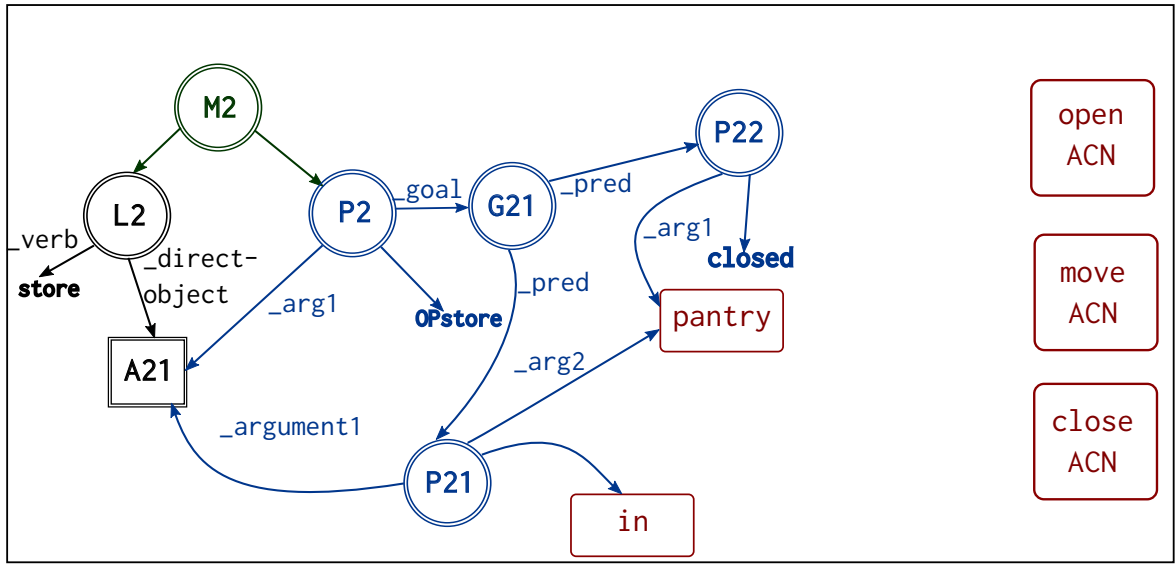
Interaction trace

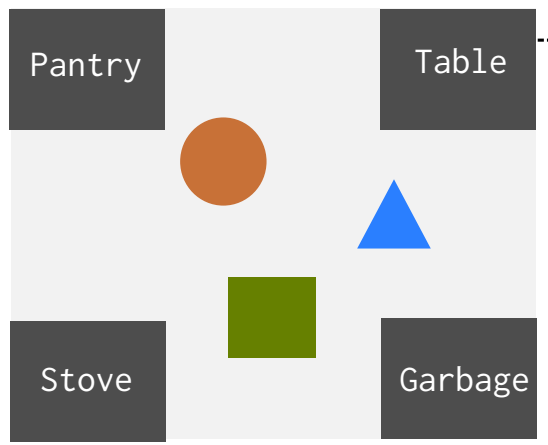
Instructor: Store the green rectangle.

Agent: What is the goal of the action?

Instructor: The goal is the green rectangle in the pantry and the pantry closed.

Agent: Which action should I take?





Interactive Example Execution

Interaction trace

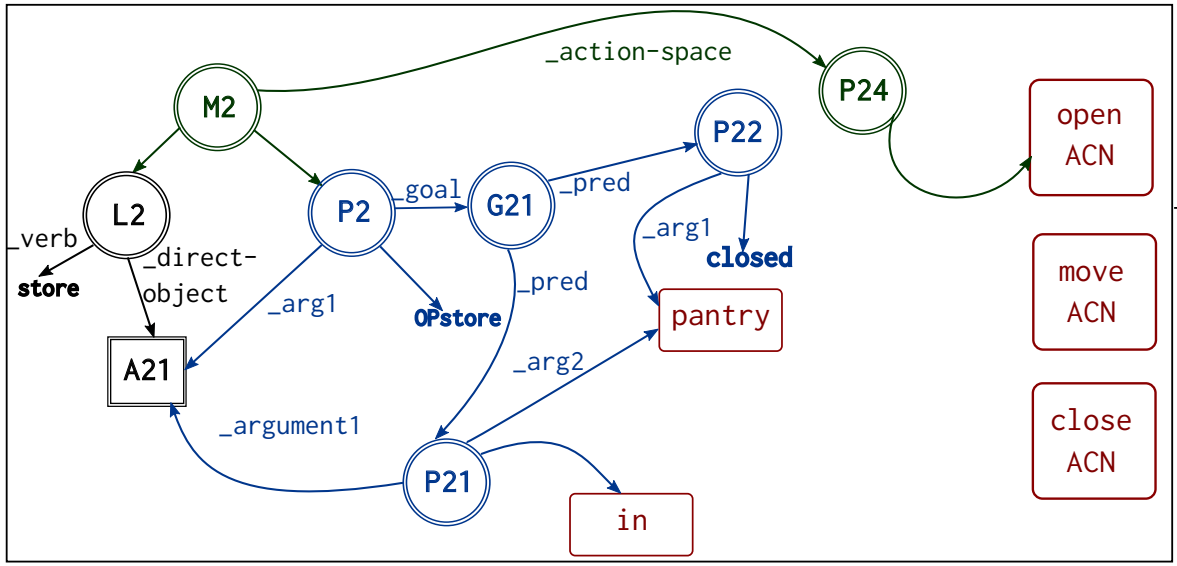
Instructor: Store the green rectangle.

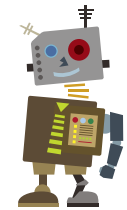
Agent: What is the goal of the action?

Instructor: The goal is the green rectangle in the pantry and the pantry closed.

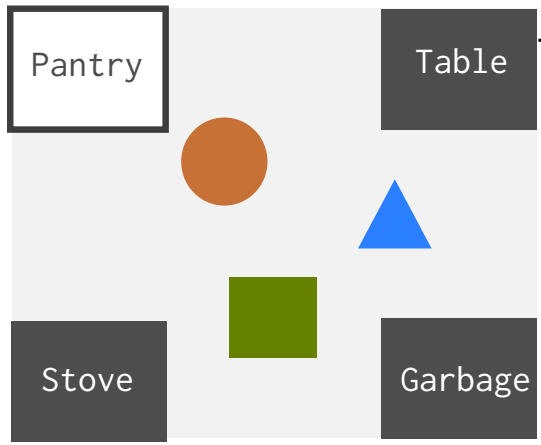
Agent: Which action should I take?

Instructor: Open the pantry.





Episodic Memory



Interactive Example Execution

Interaction trace

Instructor: Store the green rectangle.

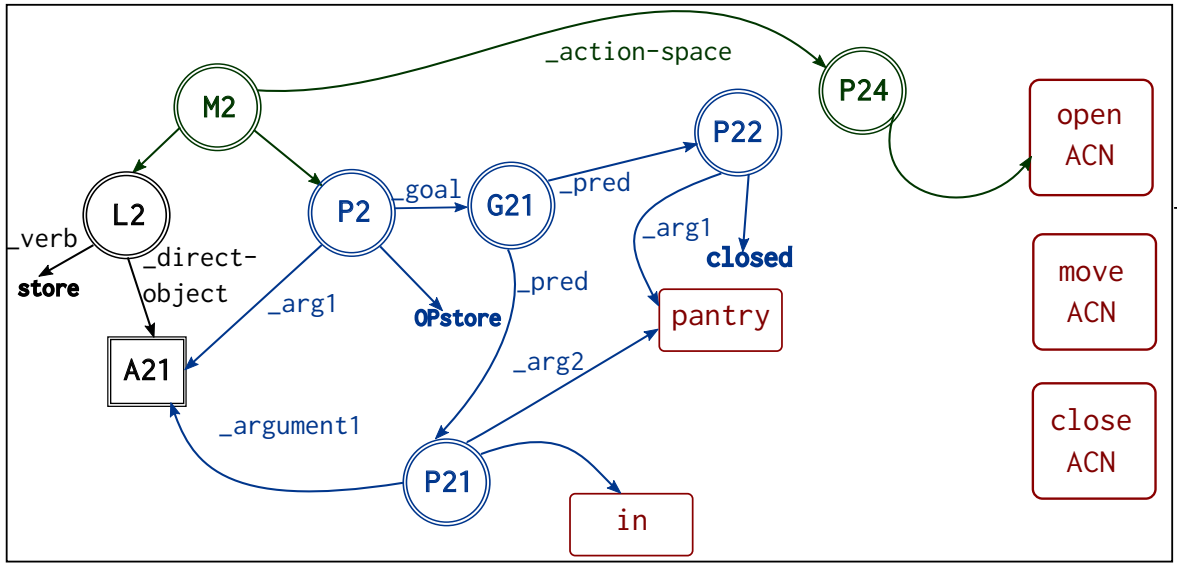
Agent: What is the goal of the action?

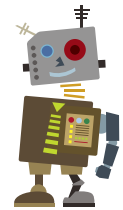
Instructor: The goal is the green rectangle in the pantry and the pantry closed.

Agent: Which action should I take?

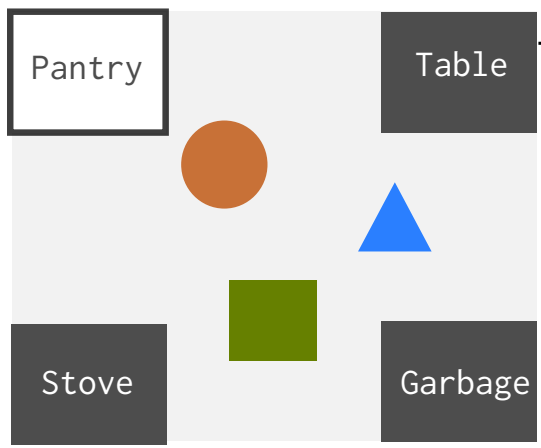
Instructor: Open the pantry.

Agent: Which action should I take?





Episodic Memory



Interactive Example Execution

Interaction trace

Instructor: Store the green rectangle.

Agent: What is the goal of the action?

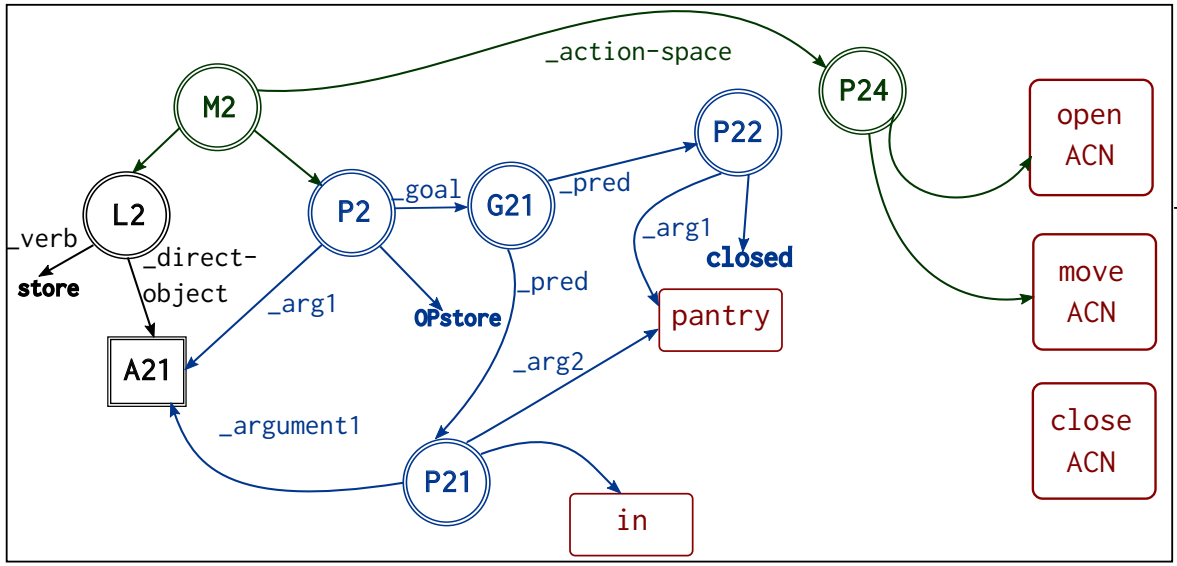
Instructor: The goal is the green rectangle in the pantry and the pantry closed.

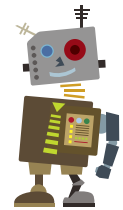
Agent: Which action should I take?

Instructor: Open the pantry.

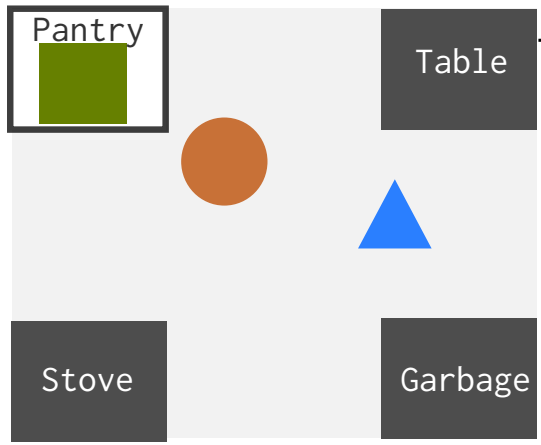
Agent: Which action should I take?

Instructor: Place the object in the pantry.





Episodic Memory



Interactive Example Execution

Interaction trace

Instructor: Store the green rectangle.

Agent: What is the goal of the action?

Instructor: The goal is the green rectangle in the pantry and the pantry closed.

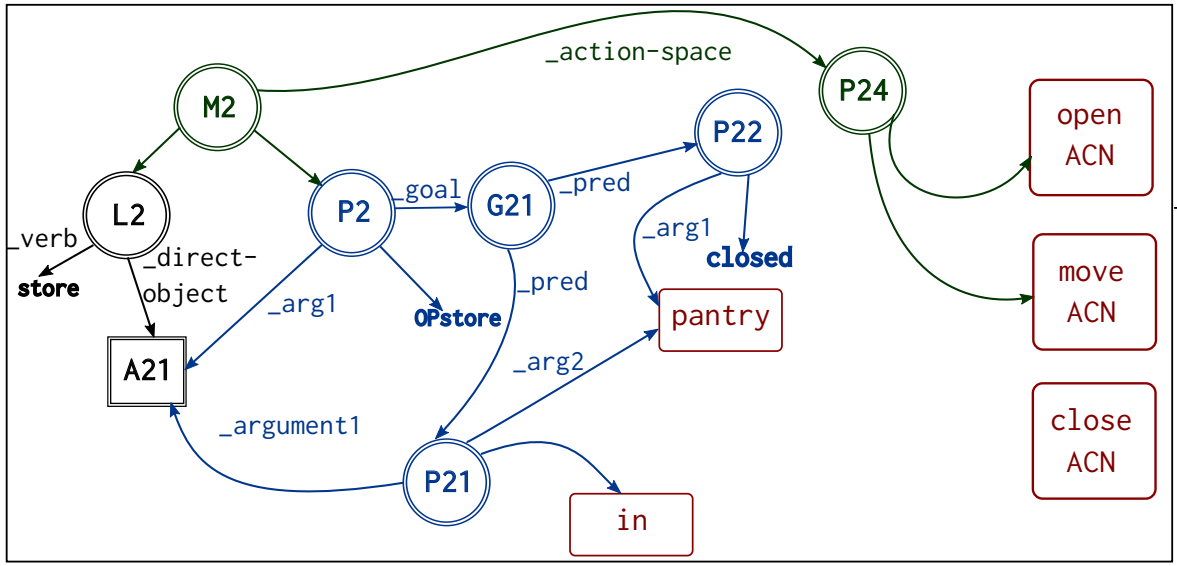
Agent: Which action should I take?

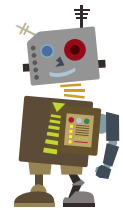
Instructor: Open the pantry.

Agent: Which action should I take?

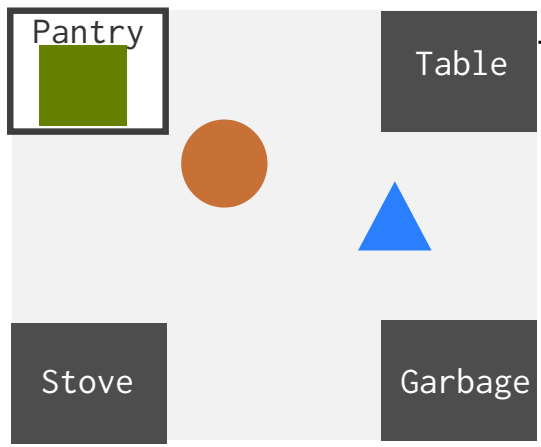
Instructor: Place the object in the pantry.

Agent: Which action should I take?





Episodic Memory



Interactive Example Execution

Interaction trace

Instructor: Store the green rectangle.

Agent: What is the goal of the action?

Instructor: The goal is the green rectangle in the pantry and the pantry closed.

Agent: Which action should I take?

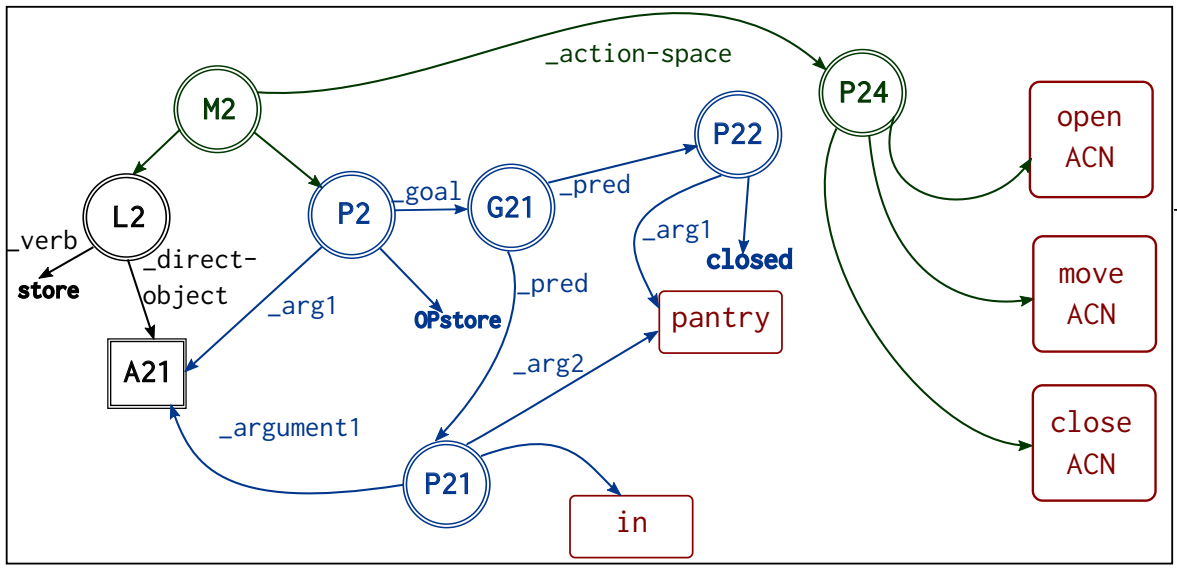
Instructor: Open the pantry.

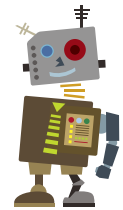
Agent: Which action should I take?

Instructor: Place the object in the pantry.

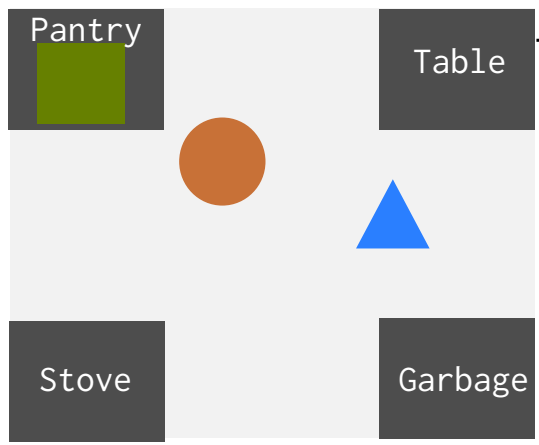
Agent: Which action should I take?

Instructor: Close the pantry.





Episodic Memory



Interactive Example Execution

Interaction trace

Instructor: Store the green rectangle.

Agent: What is the goal of the action?

Instructor: The goal is the green rectangle in the pantry and the pantry closed.

Agent: Which action should I take?

Instructor: Open the pantry.

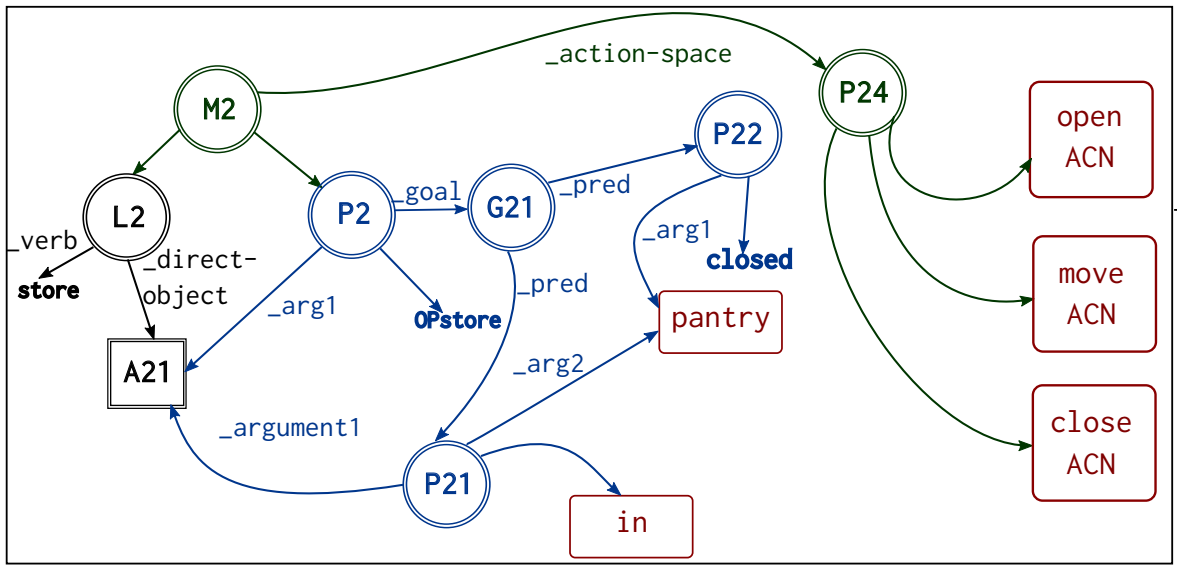
Agent: Which action should I take?

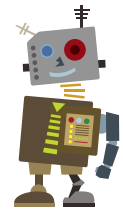
Instructor: Place the object in the pantry.

Agent: Which action should I take?

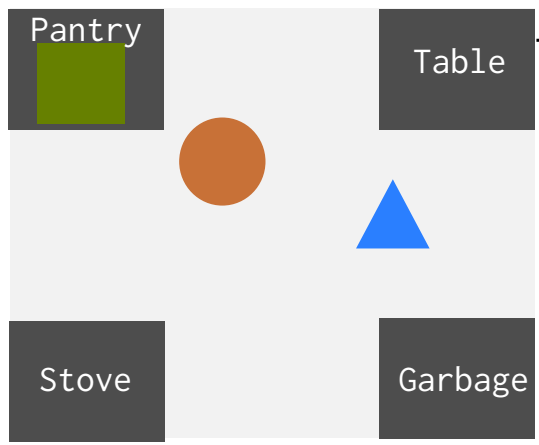
Instructor: Close the pantry.

Agent: Which action should I take?





Episodic Memory



Interactive Example Execution

Interaction trace

Instructor: Store the green rectangle.

Agent: What is the goal of the action?

Instructor: The goal is the green rectangle in the pantry and the pantry closed.

Agent: Which action should I take?

Instructor: Open the pantry.

Agent: Which action should I take?

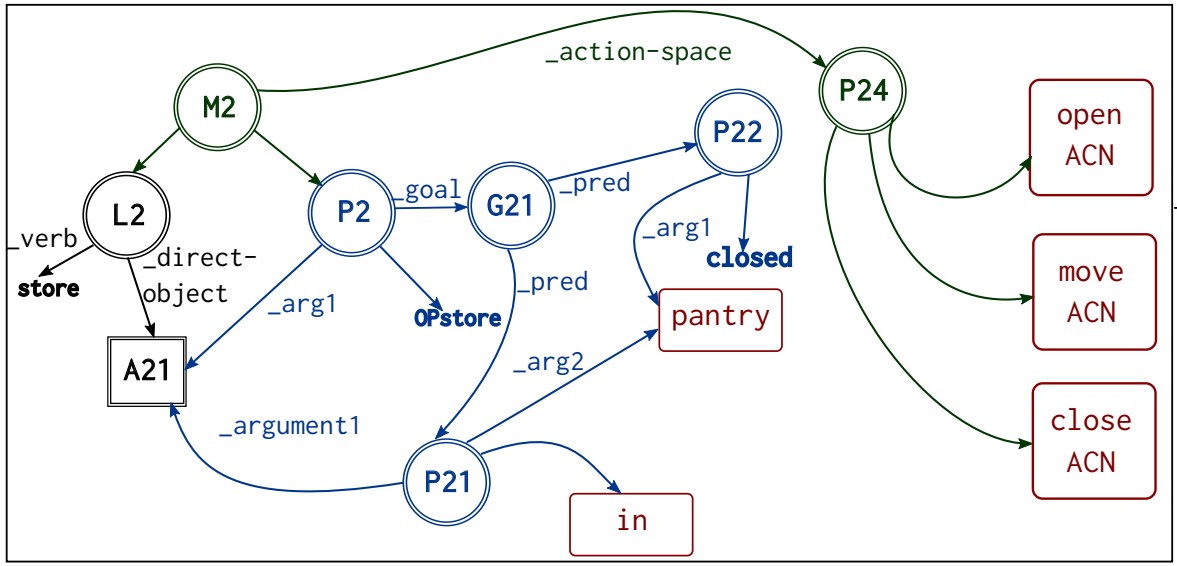
Instructor: Place the object in the pantry.

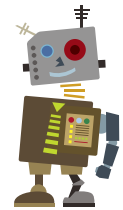
Agent: Which action should I take?

Instructor: Close the pantry.

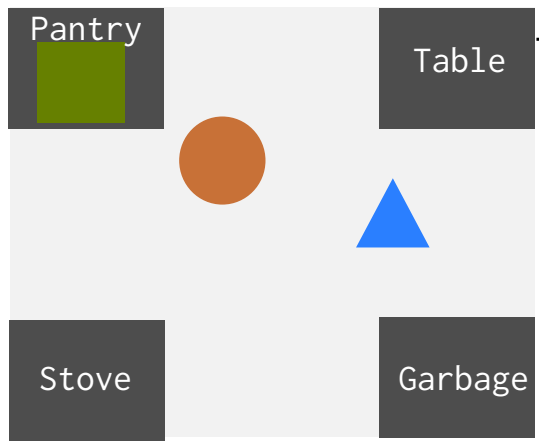
Agent: Which action should I take?

Instructor: You are done.

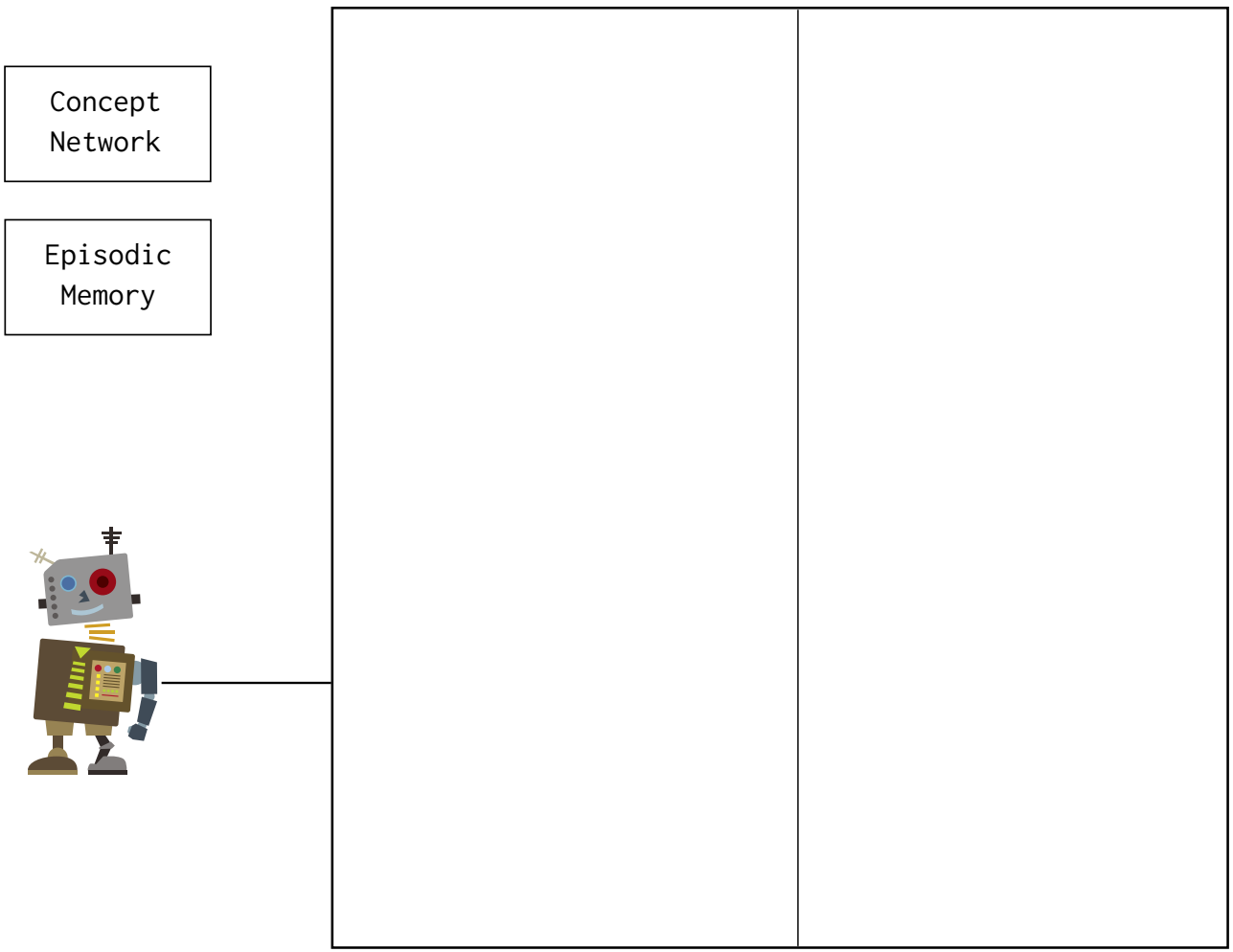




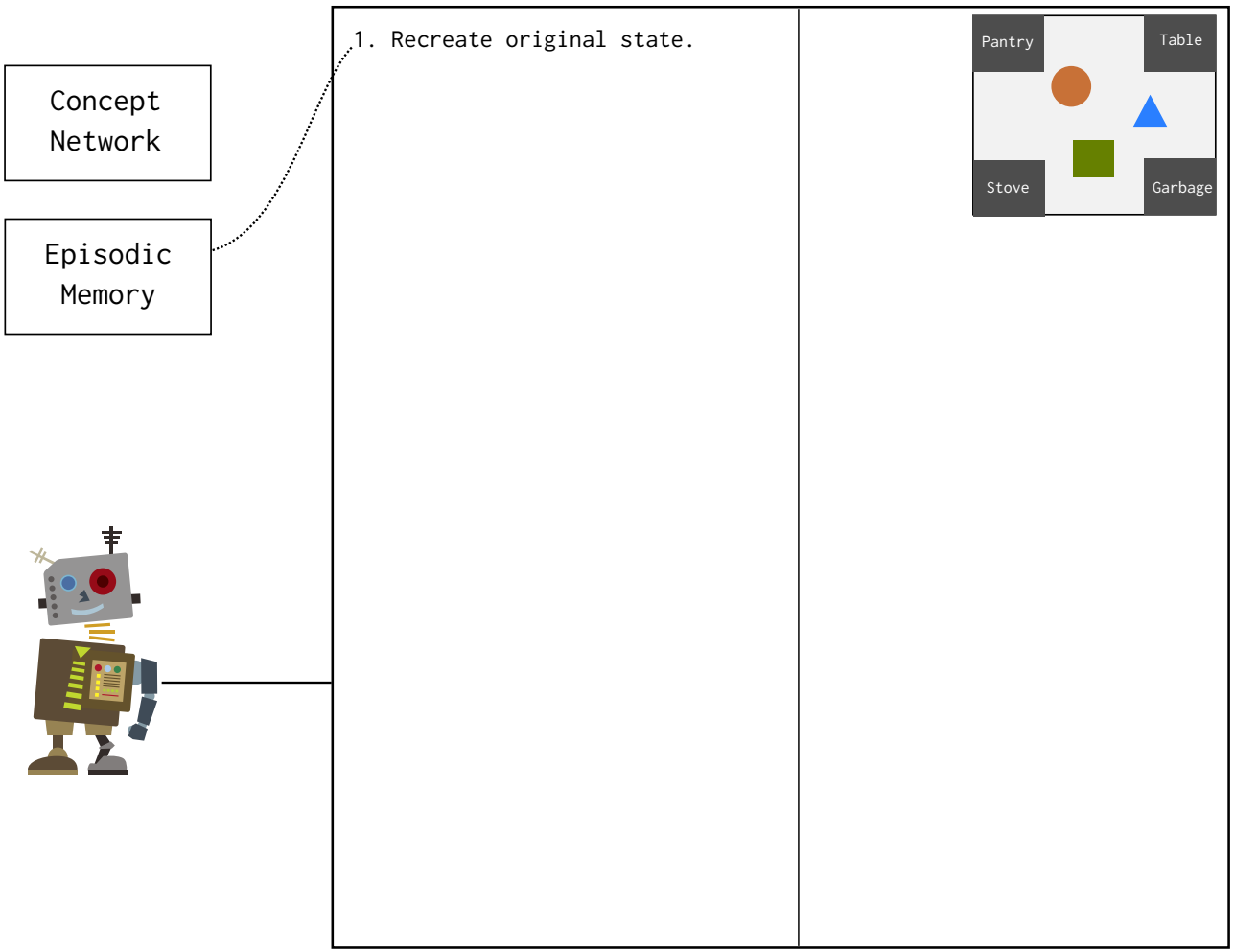
Episodic Memory



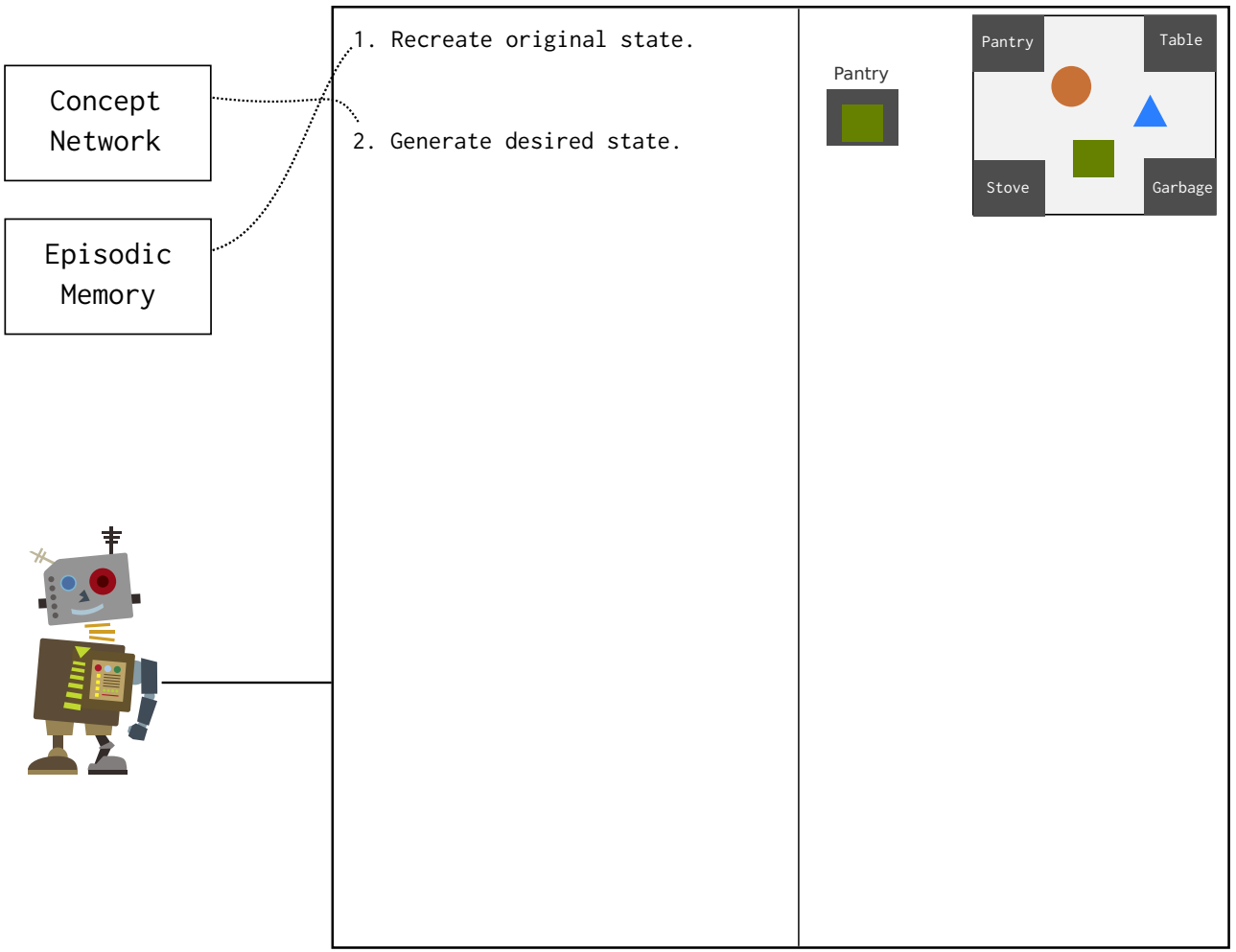
Retrospective EBL



Retrospective EBL

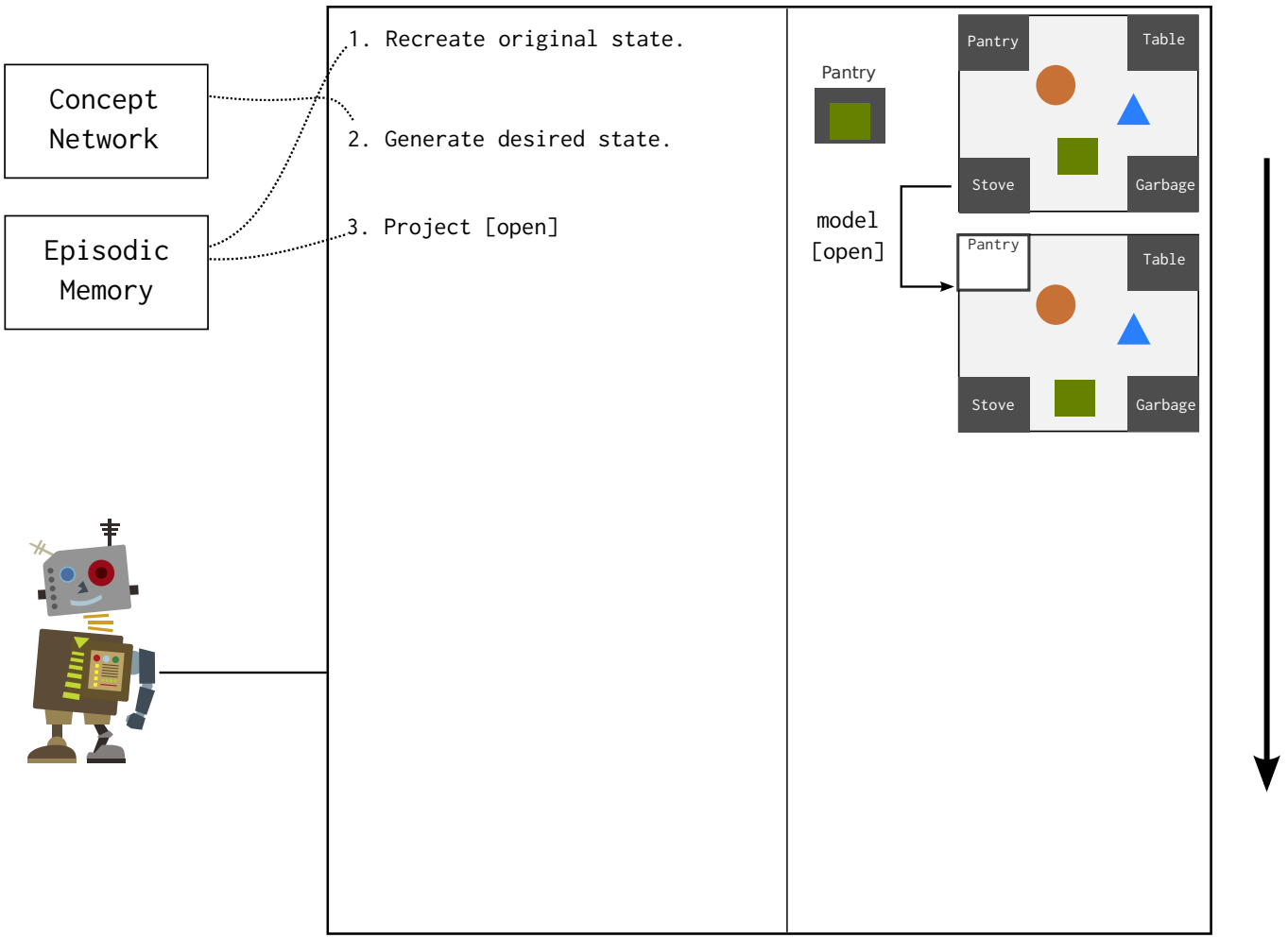


Retrospective EBL



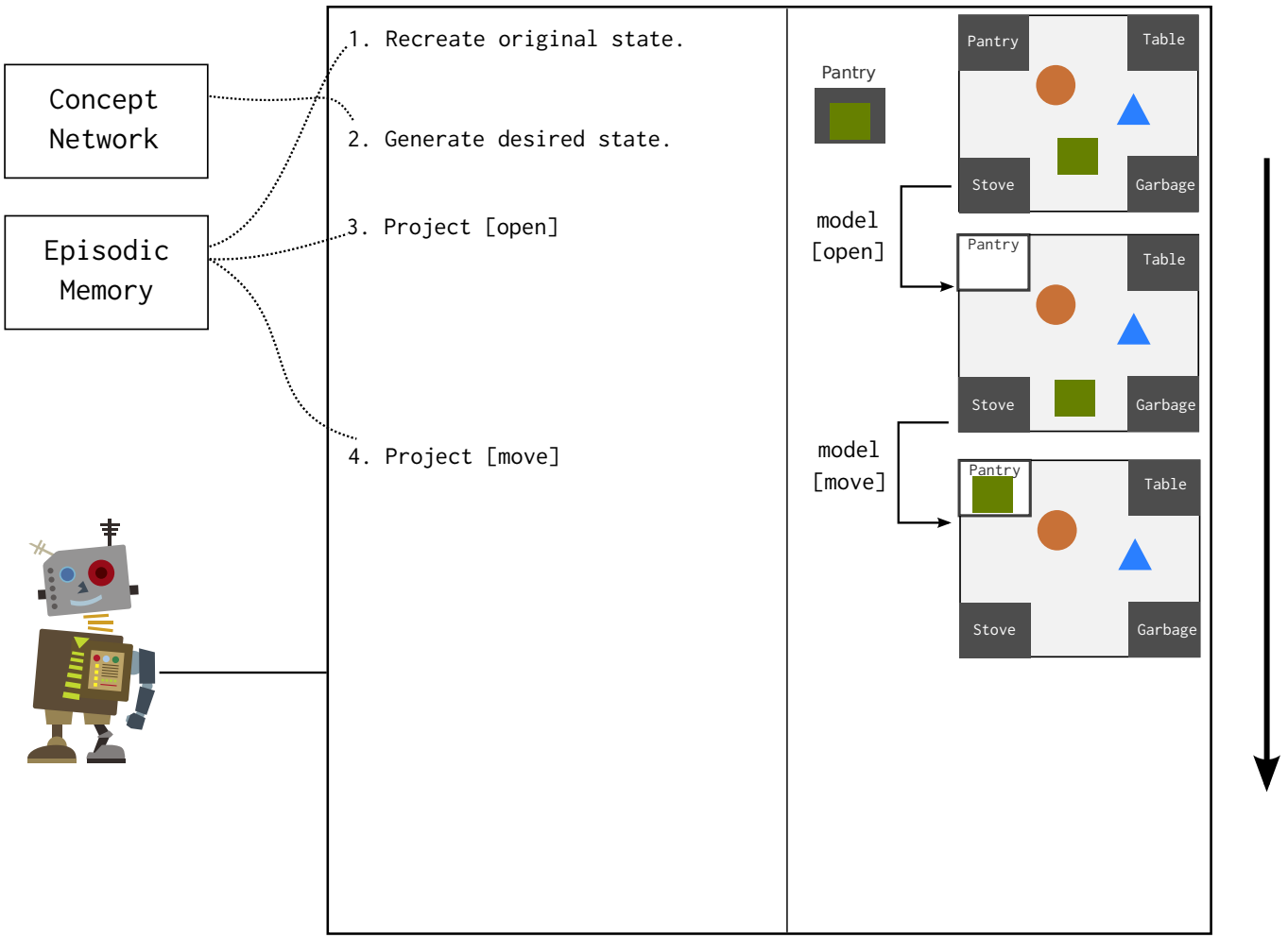
terminate store:
If `store([x])` and
`IN([x],PANTRY)` and
`CLOSED(PANTRY)`
-->
terminate `store[x]`

Retrospective EBL



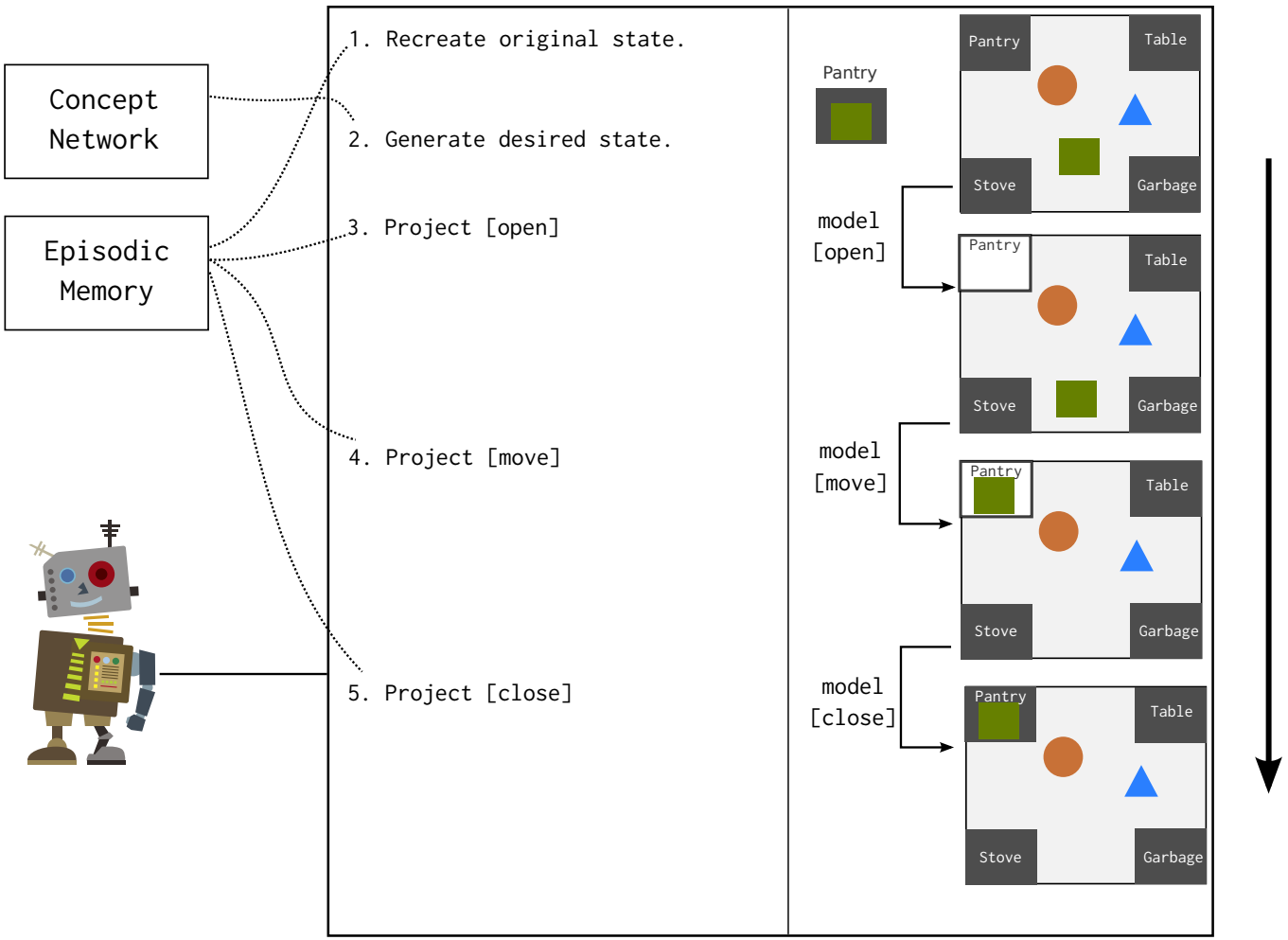
terminate store:
If `store([x])` and
`IN([x],PANTRY)` and
`CLOSED(PANTRY)`
-->
terminate `store[x]`

Retrospective EBL



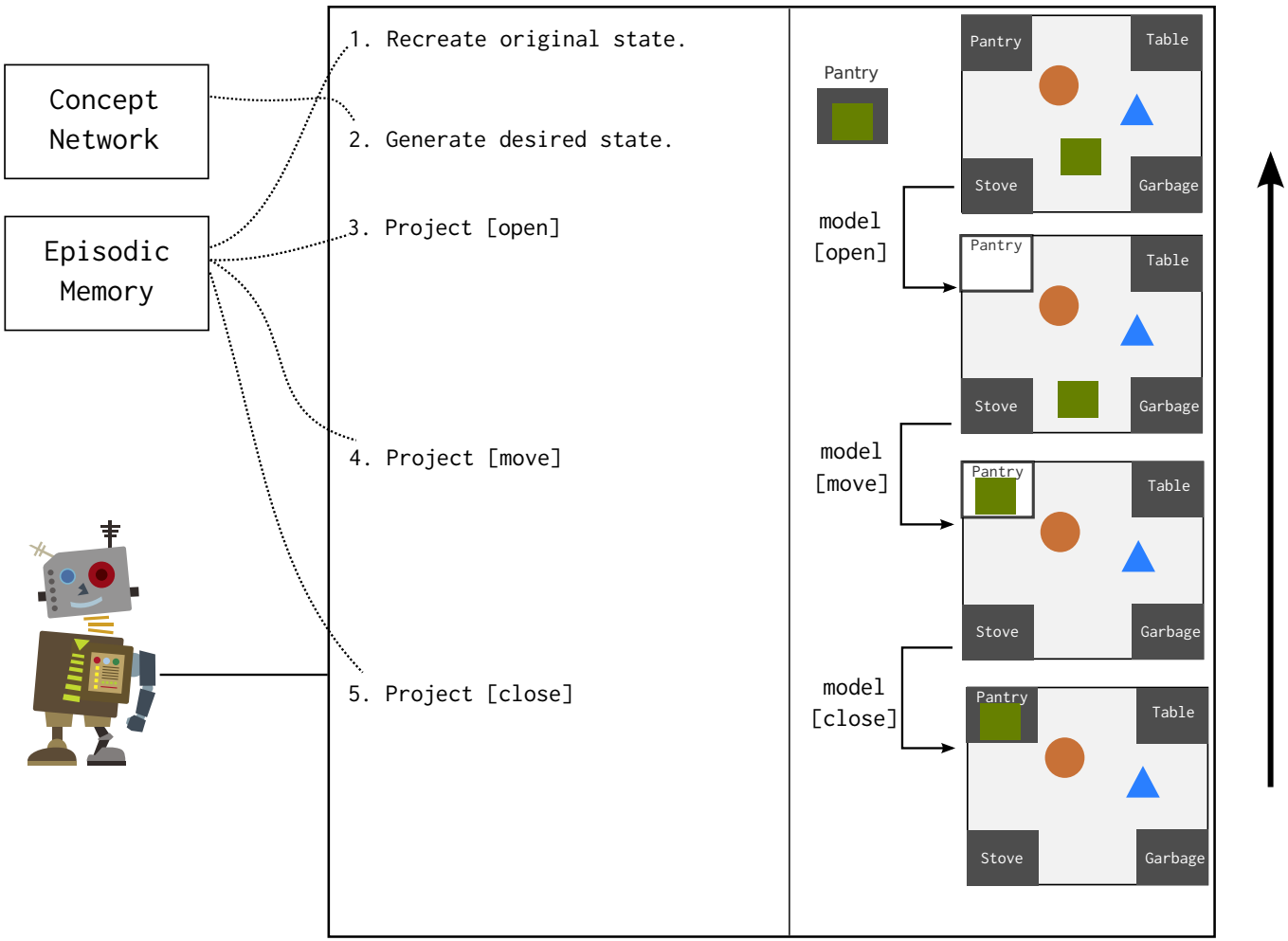
terminate store:
If `store([x])` and
`IN([x],PANTRY)` and
`CLOSED(PANTRY)`
-->
terminate `store[x]`

Retrospective EBL



terminate store:
If `store([x])` and
`IN([x],PANTRY)` and
`CLOSED(PANTRY)`
-->
terminate `store[x]`

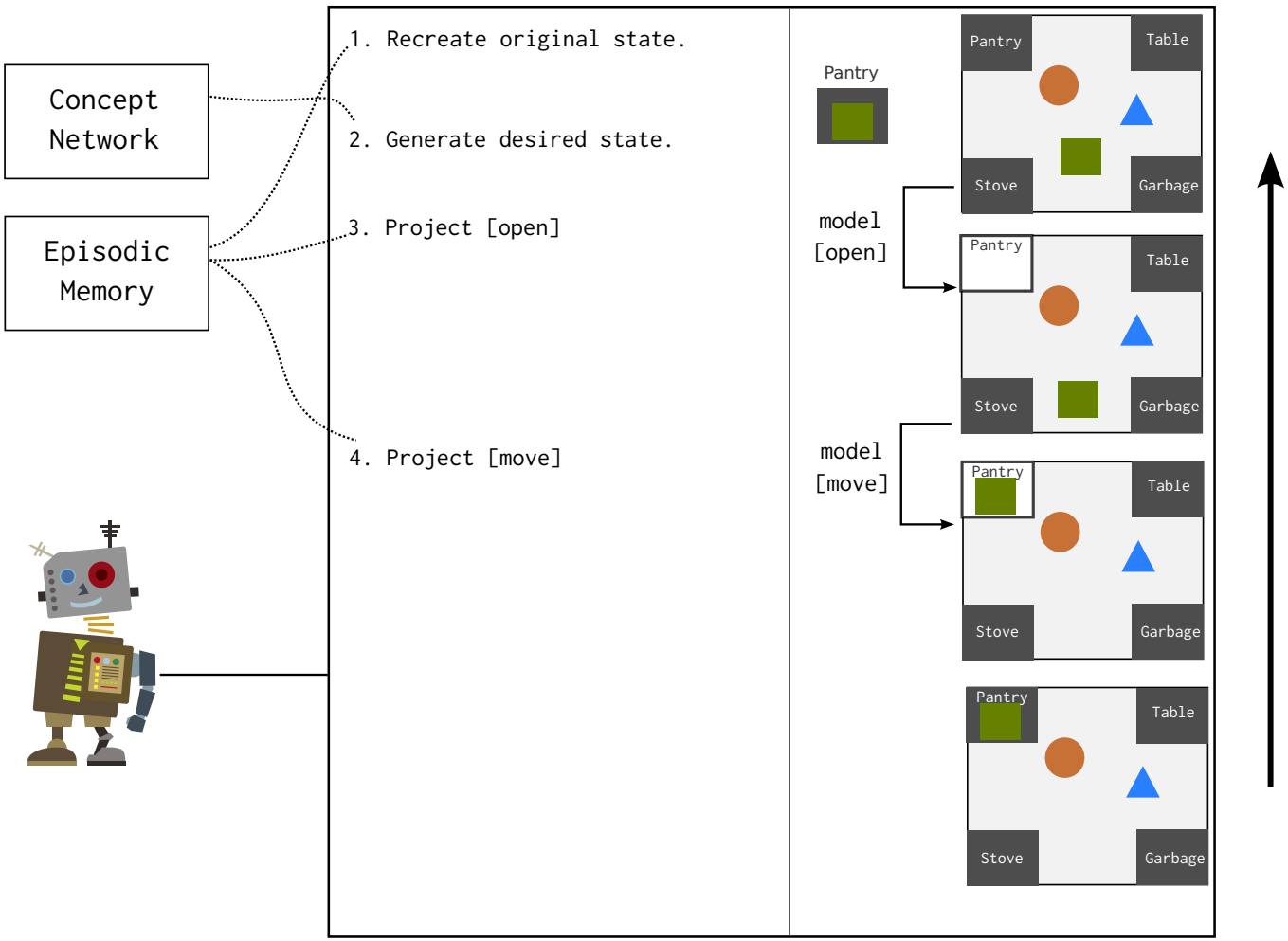
Retrospective EBL



terminate store:
If `store([x])` and `IN([x],PANTRY)` and `CLOSED(PANTRY)`
-->
terminate `store[x]`

select close:
If `store([x])` and `IN([x],PANTRY)` and `OPEN(PANTRY)`
-->
select `close(PANTRY)`

Retrospective EBL



terminate store:
If `store([x])` and `IN([x],PANTRY)` and `CLOSED(PANTRY)`
`-->`
terminate `store[x]`

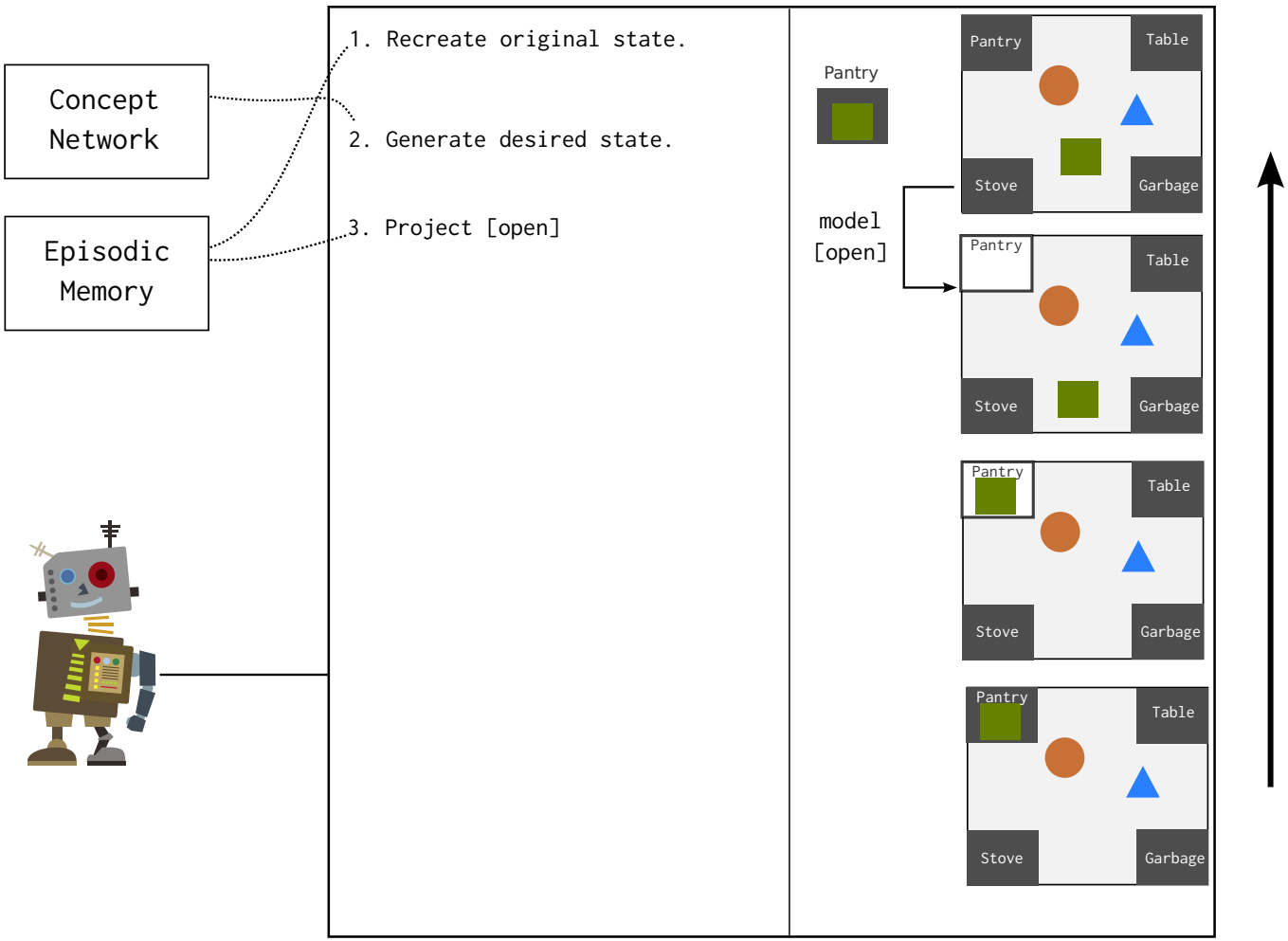
---

select close:
If `store([x])` and `IN([x],PANTRY)` and `OPEN(PANTRY)`
`-->`
select `close(PANTRY)`

---

select place:
If `store([x])` and `-IN([x],PANTRY)` and `OPEN(PANTRY)`
`-->`
select `place([x],IN,PANTRY)`

Retrospective EBL



terminate store:
If `store([x])` and `IN([x],PANTRY)` and `CLOSED(PANTRY)`
-->
terminate `store[x]`

---

select close:
If `store([x])` and `IN([x],PANTRY)` and `OPEN(PANTRY)`
-->
select `close(PANTRY)`

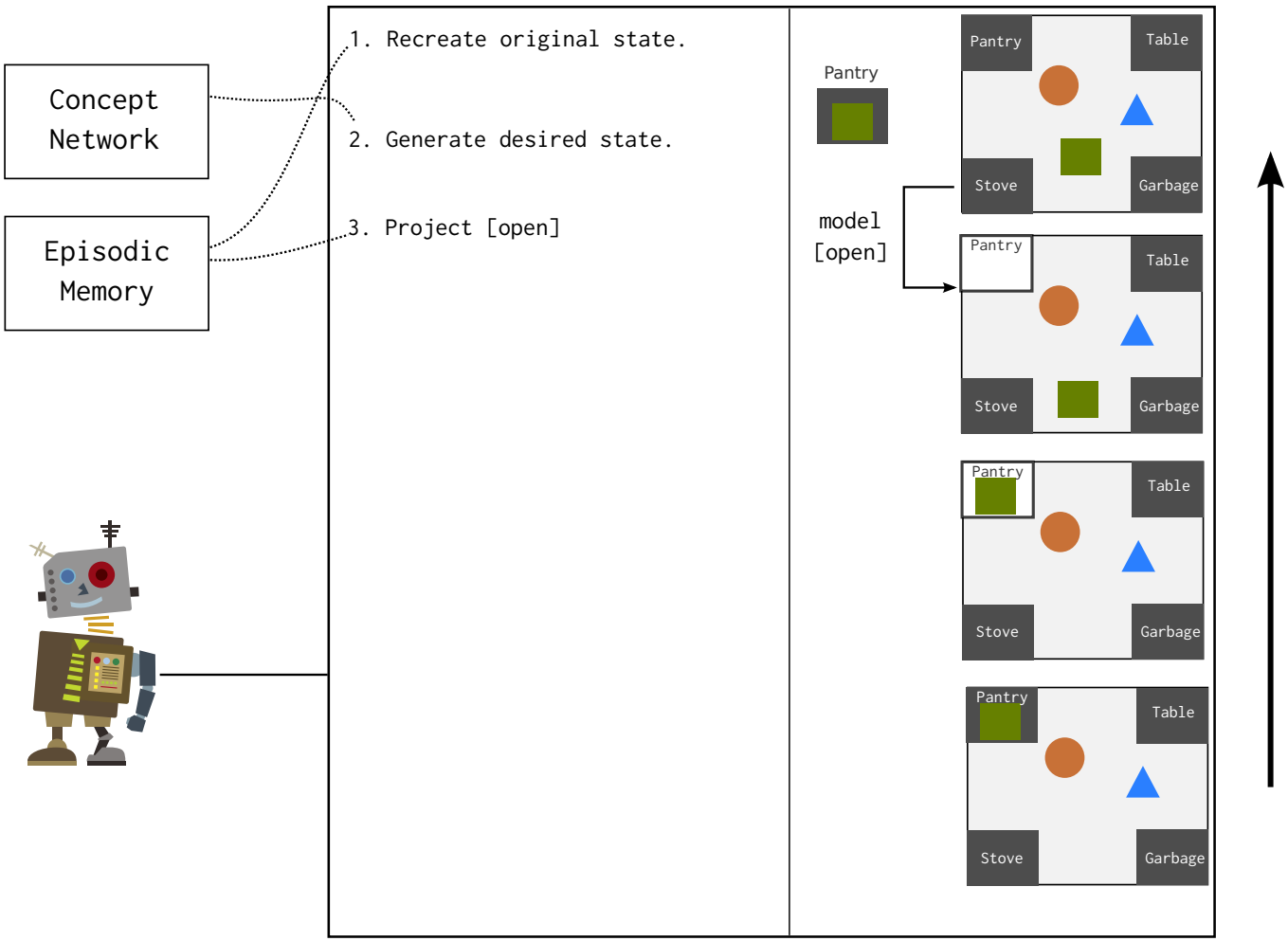
---

select place:
If `store([x])` and `-IN([x],PANTRY)` and `OPEN(PANTRY)`
-->
select `place([x],IN,PANTRY)`

---

select open:
If `store([x])` and `-IN([x],PANTRY)` and `CLOSED(PANTRY)`
-->
select `open(PANTRY)`

Retrospective EBL



terminate store:
If `store([x])` and `IN([x],PANTRY)` and `CLOSED(PANTRY)`
-->
terminate `store[x]`

---

select close:
If `store([x])` and `IN([x],PANTRY)` and `OPEN(PANTRY)`
-->
select `close(PANTRY)`

---

select place:
If `store([x])` and `-IN([x],PANTRY)` and `OPEN(PANTRY)`
-->
select `place([x],IN,PANTRY)`

---

select open:
If `store([x])` and `-IN([x],PANTRY)` and `CLOSED(PANTRY)`
-->
select `open(PANTRY)`

---

available store:
If `-IN([x],PANTRY)` or `OPEN(PANTRY)`
-->
available `store([x])`

Evaluation Overview

Desiderata

- multi-task learning
- fast generalization
- assisted transfer
- distributed initiative in learning

Multi-task Learning

pick-up-&-place:
`place([x],[rel],[y])`, `move([x],[y])`, `discard([x])`, `store([x])`
functional:
`cook([x])`, `serve([x])`
organizational:
`stack-3([x],[y],[z])`, `stack-4([x],[y],[z],[w])`

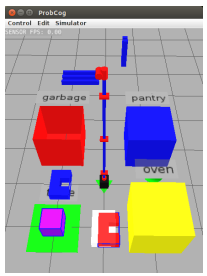
Generality in Learning

Learns general representations of tasks from few (~2-3) instances

Generality in Learning

Learns general representations of tasks from few (~2-3) instances

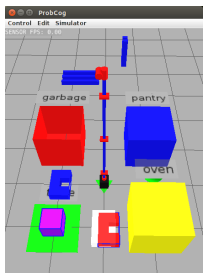
abstraction



Generality in Learning

Learns general representations of tasks from few (~2-3) instances

abstraction



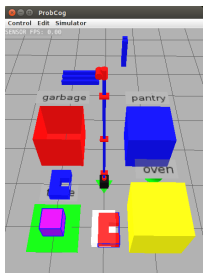
predicate selection

select open:
If `store(O1)` and `-IN(O1,PANTRY)` and `CLOSED(PANTRY)` and `CLOSED(STOVE)` and `OFF(STOVE)` and `-ON(O2,STOVE)` and ...
-->
select `open(PANTRY)`

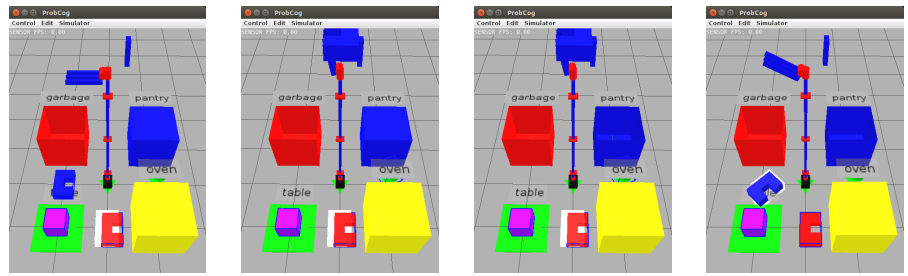
Generality in Learning

Learns general representations of tasks from few (~2-3) instances

abstraction



causal analysis



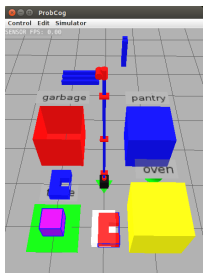
predicate selection

select open:
If `store(O1)` and `-IN(O1,PANTRY)` and `CLOSED(PANTRY)` and `CLOSED(STOVE)` and `OFF(STOVE)` and `-ON(O2,STOVE)` and ...
-->
select `open(PANTRY)`

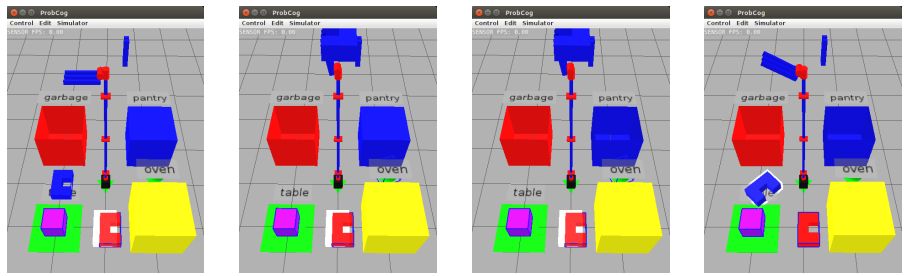
Generality in Learning

Learns general representations of tasks from few (~2-3) instances

abstraction



causal analysis



predicate selection

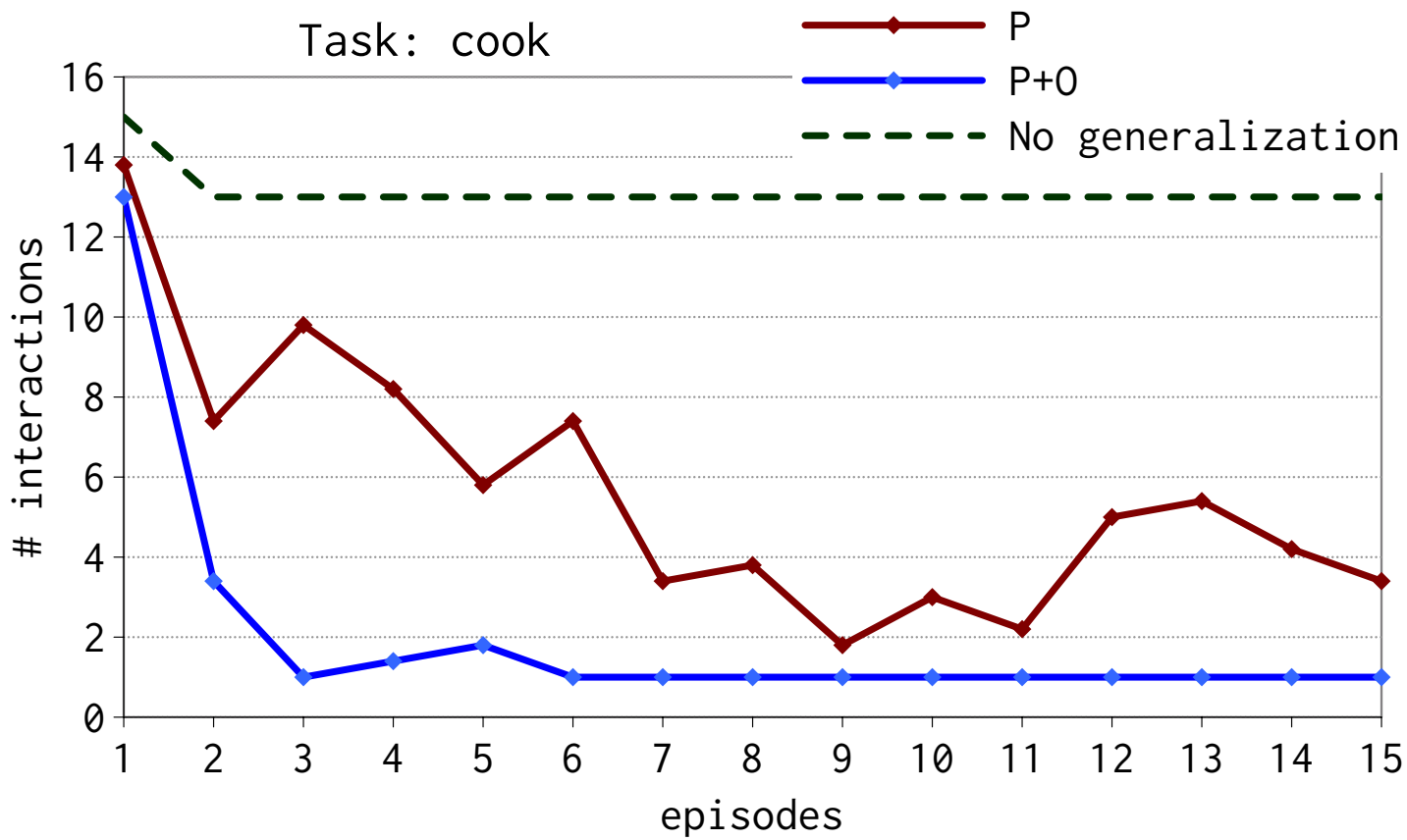
select open:
If `store(O1)` and `-IN(O1,PANTRY)` and `CLOSED(PANTRY)` and `CLOSED(STOVE)` and `OFF(STOVE)` and `-ON(O2,STOVE)` and ...
-->
select `open(PANTRY)`

variablization

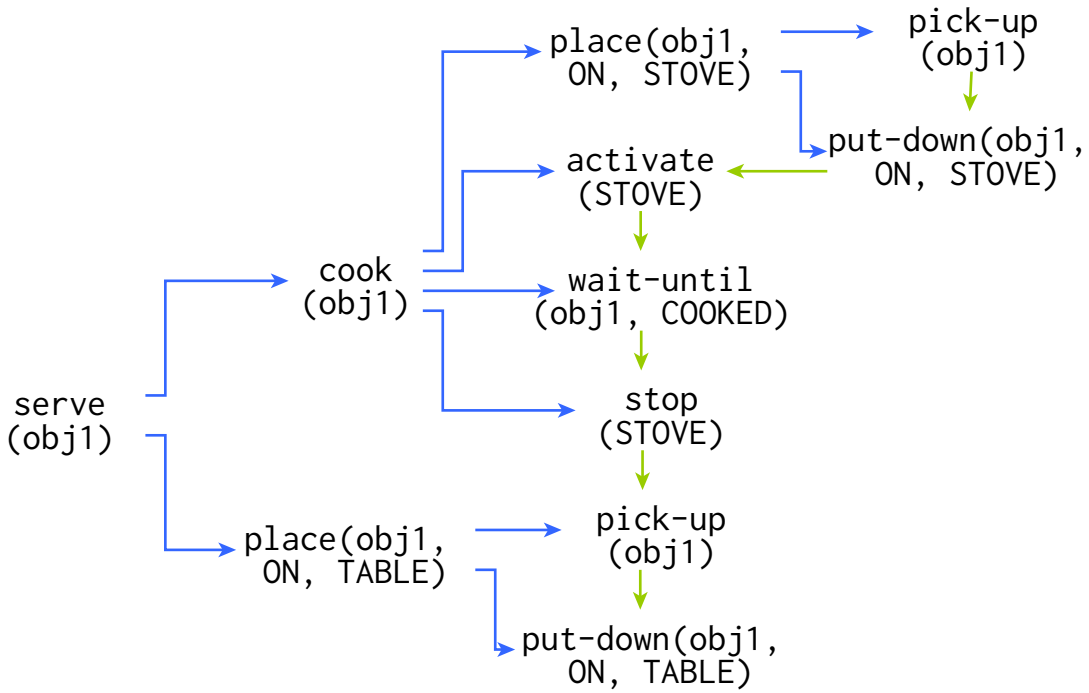
Store the green rectangle.
The goal is the green rectangle in the pantry and the pantry is closed.
Open the pantry.
Move the green rectangle to the pantry.
...

Generality in Learning

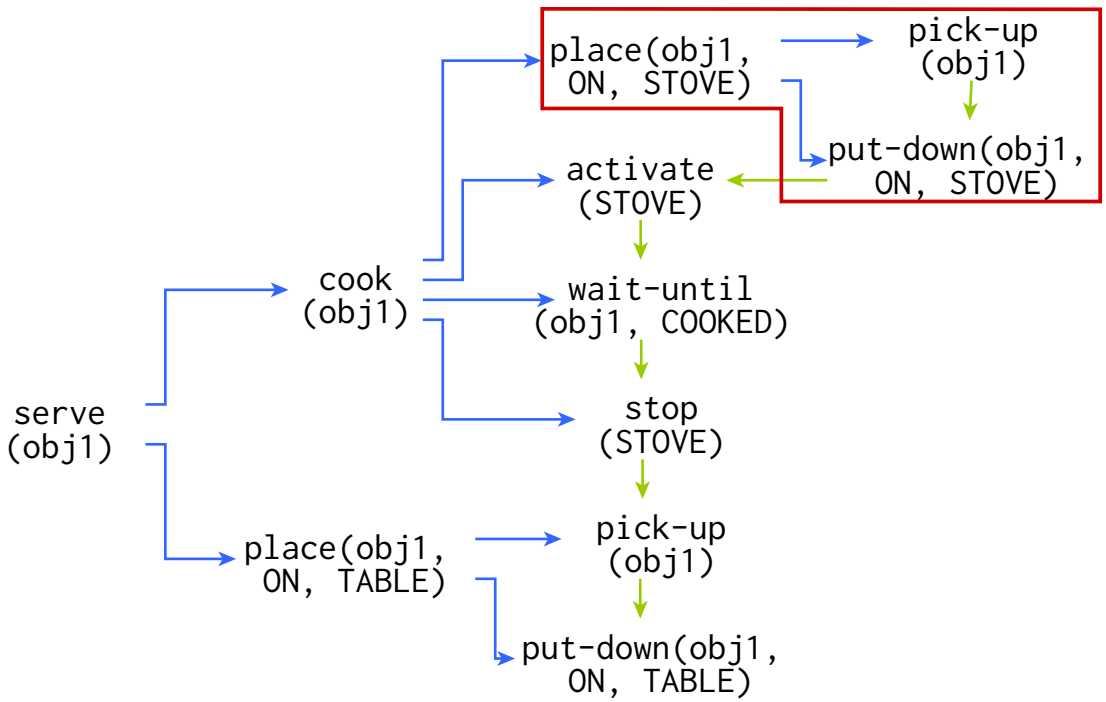
P: predicate selection only; P+O predicate selection + object variablization



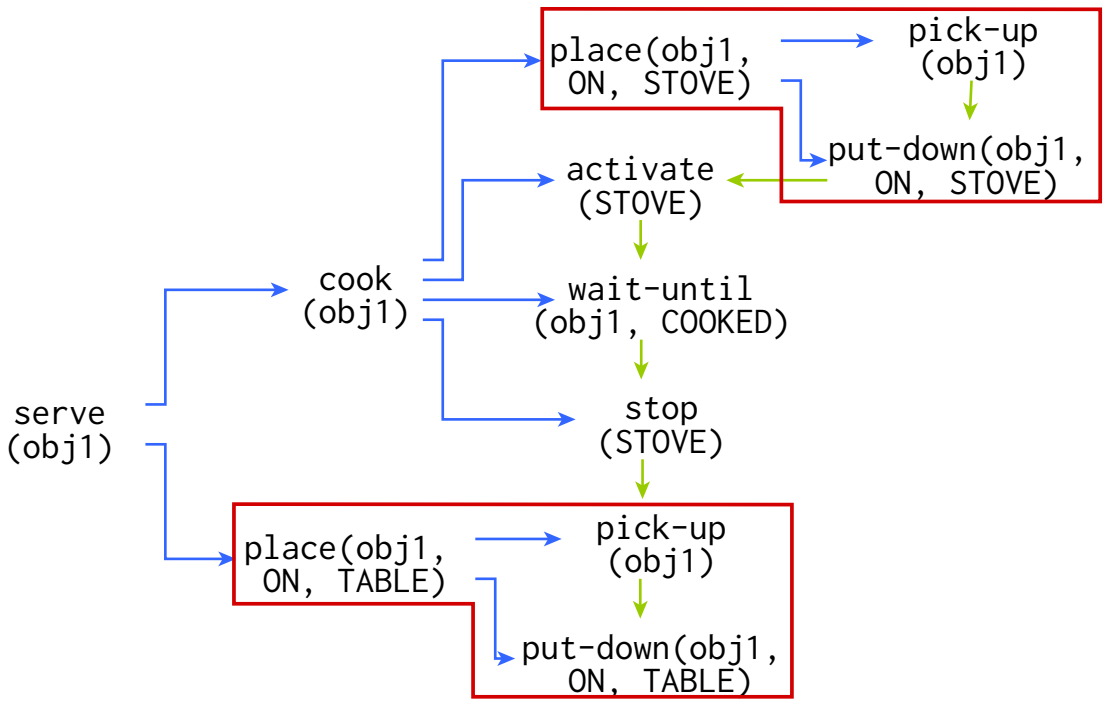
Hierarchical Learning



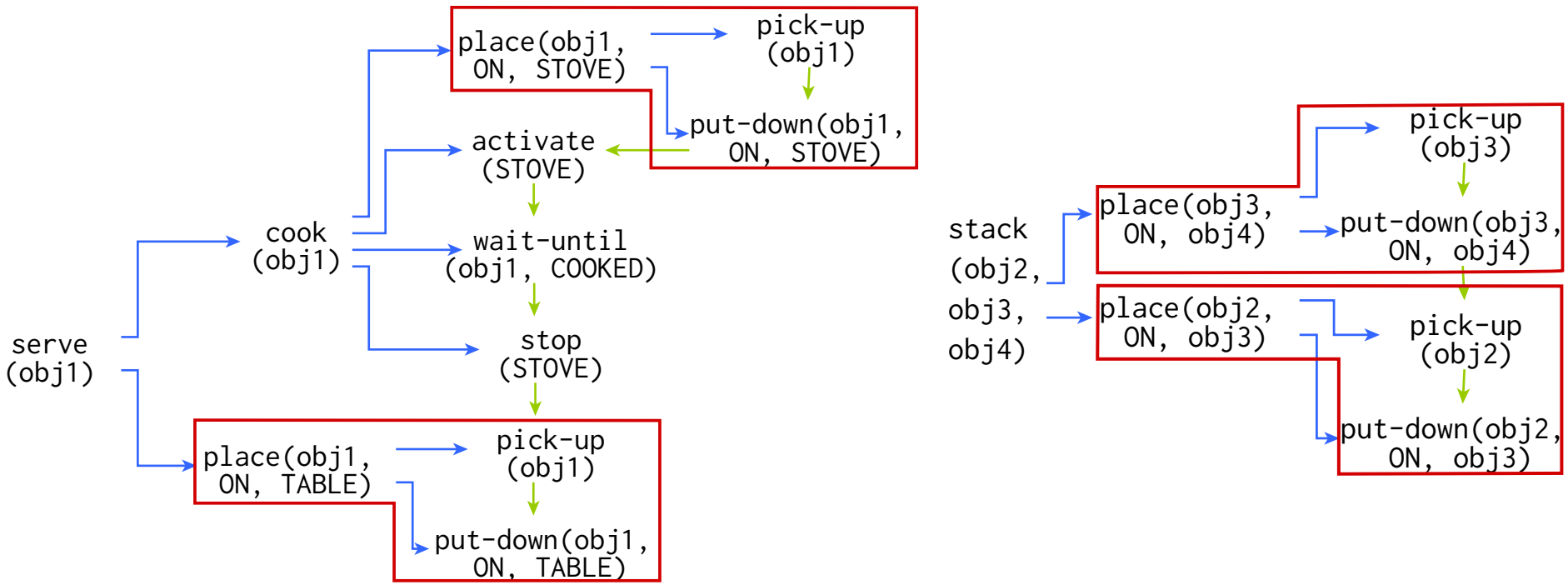
Hierarchical Learning



Hierarchical Learning

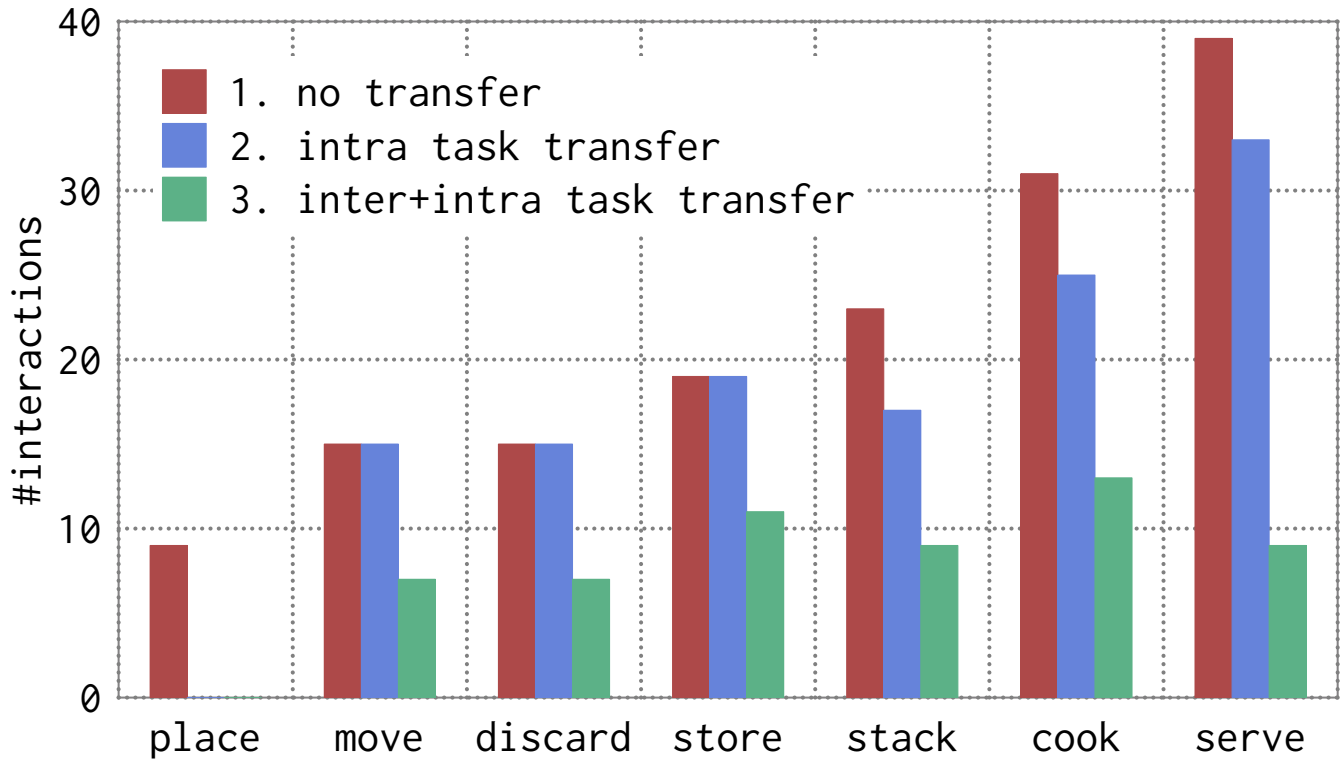


Hierarchical Learning



Transfer

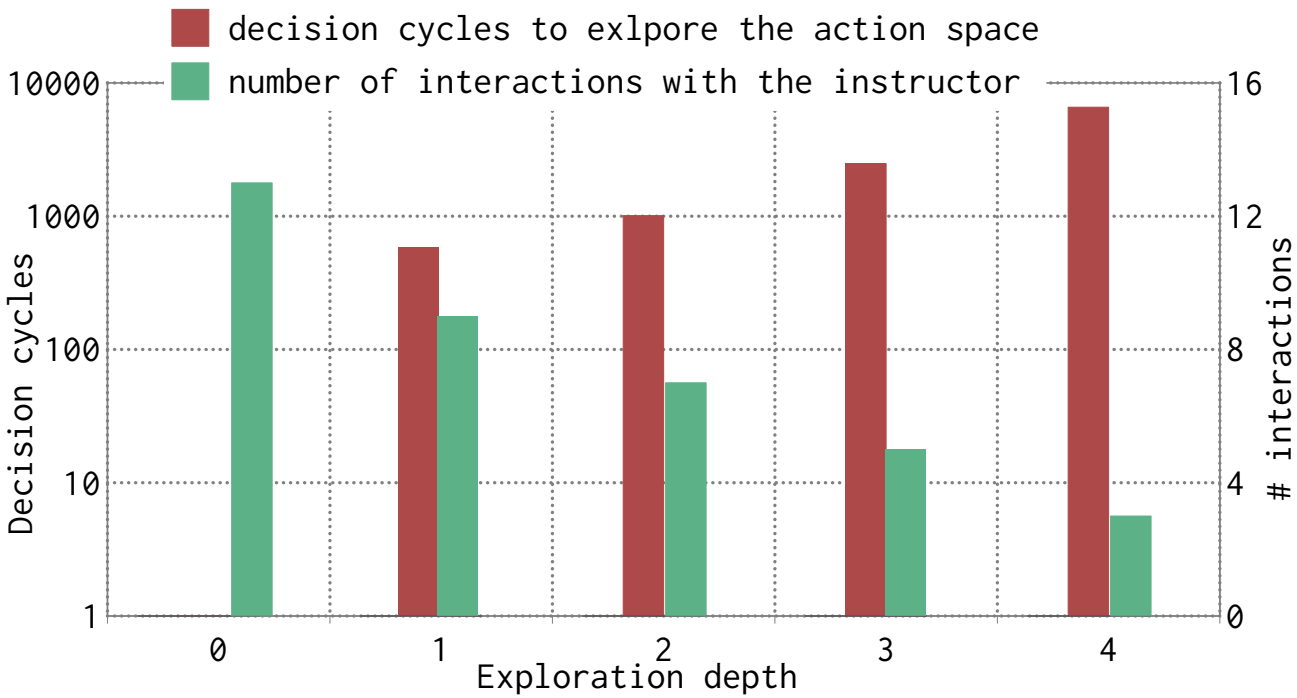
Exploits the common policy space for instruction-aided transfer.



Distributed Initiative

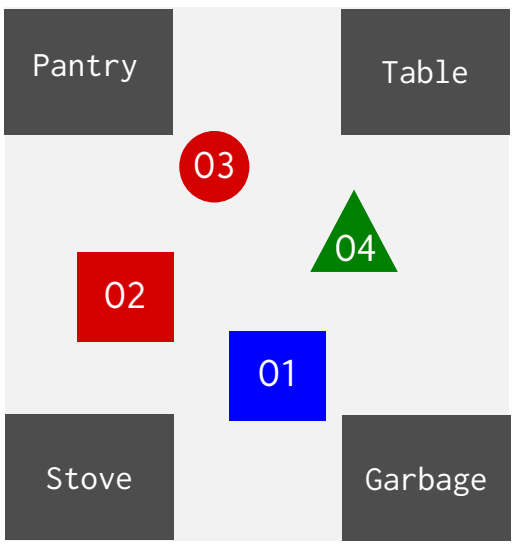
Integrates agent-driven exploration and instruction-guided exploitation

Learning the task *store*.



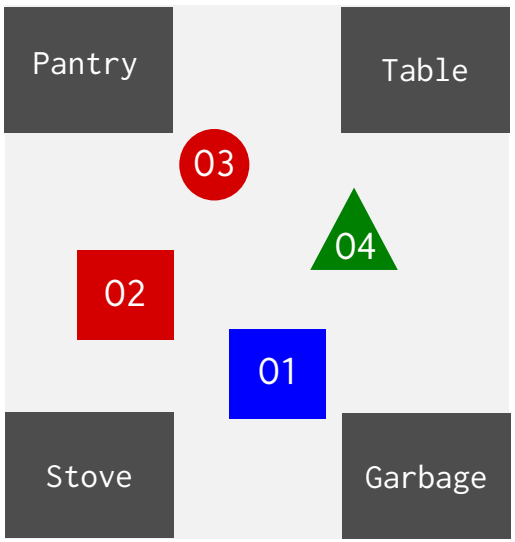
Situated Comprehension Problem

Store the blue rectangle.
The goal is the rectangle in the pantry.
Open that
Pick it up.
Put it in the pantry.
...



Situated Comprehension Problem

Store the blue rectangle.
The goal is the rectangle in the pantry.
Open that
Pick it up.
Put it in the pantry.
...

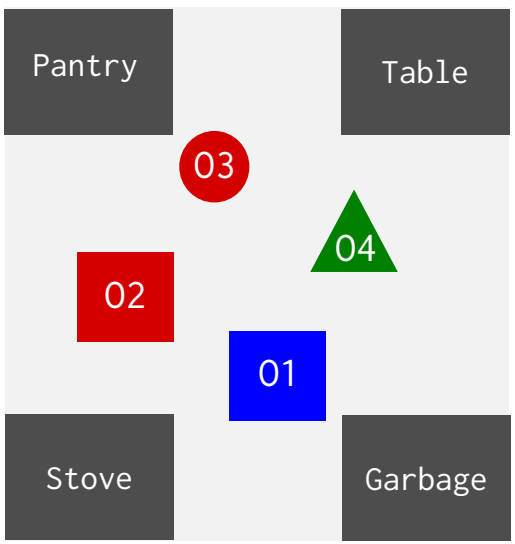


characteristics

- situated
- contextual
- embedded in interaction

Situated Comprehension Problem

Store the blue rectangle.
The goal is the rectangle in the pantry.
Open that
Pick it up.
Put it in the pantry.
...



characteristics

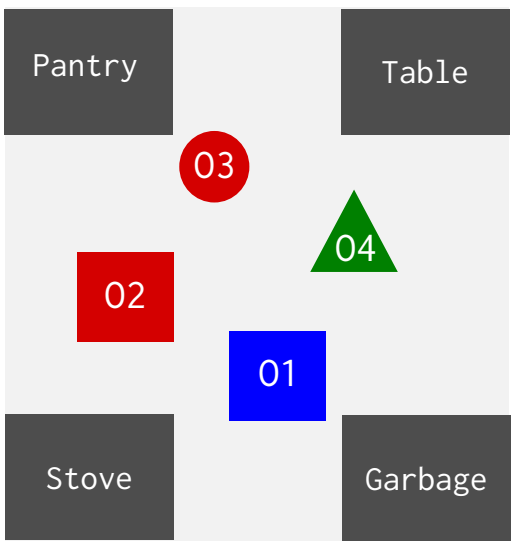
- situated
- contextual
- embedded in interaction

challenges

- diverse, mixed representations
- continual knowledge acquisition

Situated Comprehension Problem

Store the blue rectangle.
The goal is the rectangle in the pantry.
Open that
Pick it up.
Put it in the pantry.
...



characteristics

- situated
- contextual
- embedded in interaction

challenges

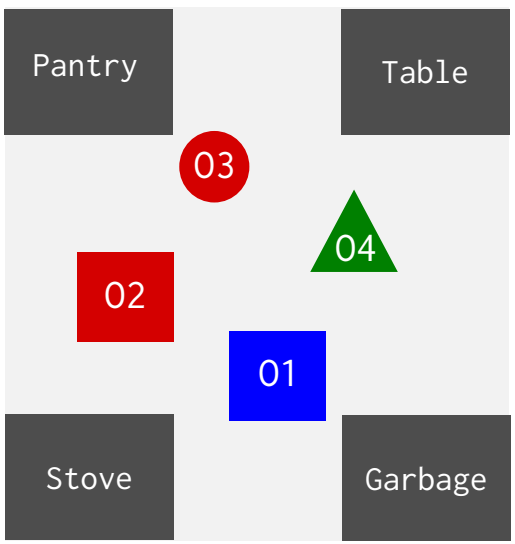
- diverse, mixed representations
- continual knowledge acquisition

desiderata

- amodal symbols → modal knowledge
- exploit non-linguistic context, knowledge
- inform interaction

Situated Comprehension Problem

Store the blue rectangle.
The goal is the rectangle in the pantry.
Open that
Pick it up.
Put it in the pantry.
...



characteristics

- situated
- contextual
- embedded in interaction

challenges

- diverse, mixed representations
- continual knowledge acquisition

desiderata

- amodal symbols → modal knowledge
- exploit non-linguistic context, knowledge
- inform interaction

approach

- comprehension as search
- use context to guide/constrain search

The Indexical Hypothesis

Glenberg and Robertson (1999), Barsalou (1999), Zwaan (2003)

- Linguistic communication is reference
- Speaker/hearer have a common ground
  - shared perceptions
  - common-sense knowledge
  - similar experiences
- Linguistic features are cues to search common ground
- Language specifies scene, knowledge fills up details

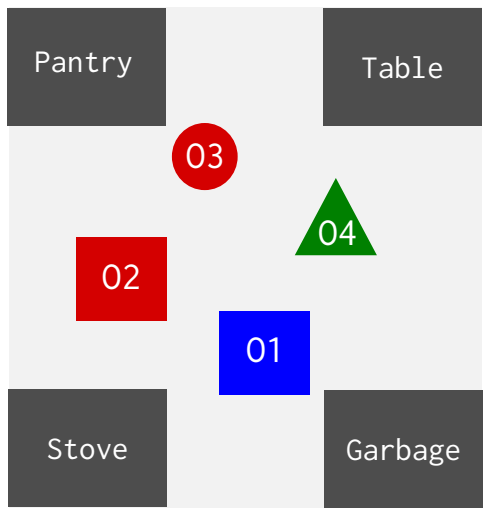
The Indexical Model

Step 1: Index components

Move the blue object to the right of the pantry.

indexical maps

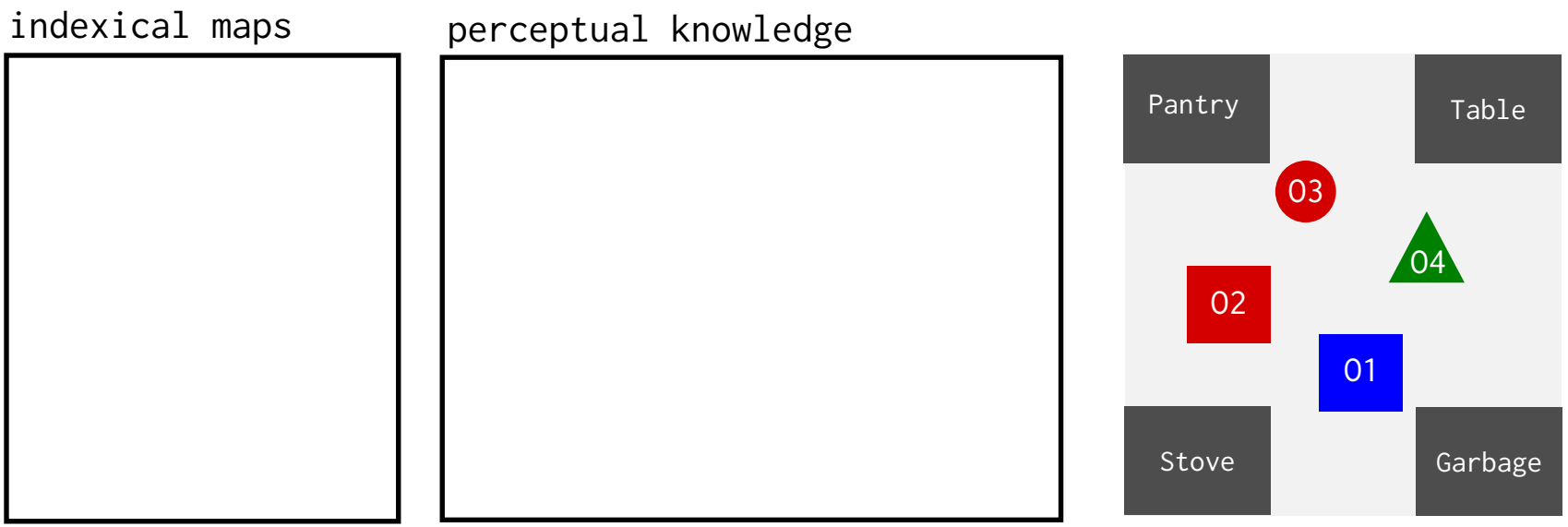
perceptual knowledge



The Indexical Model

Step 1: Index components

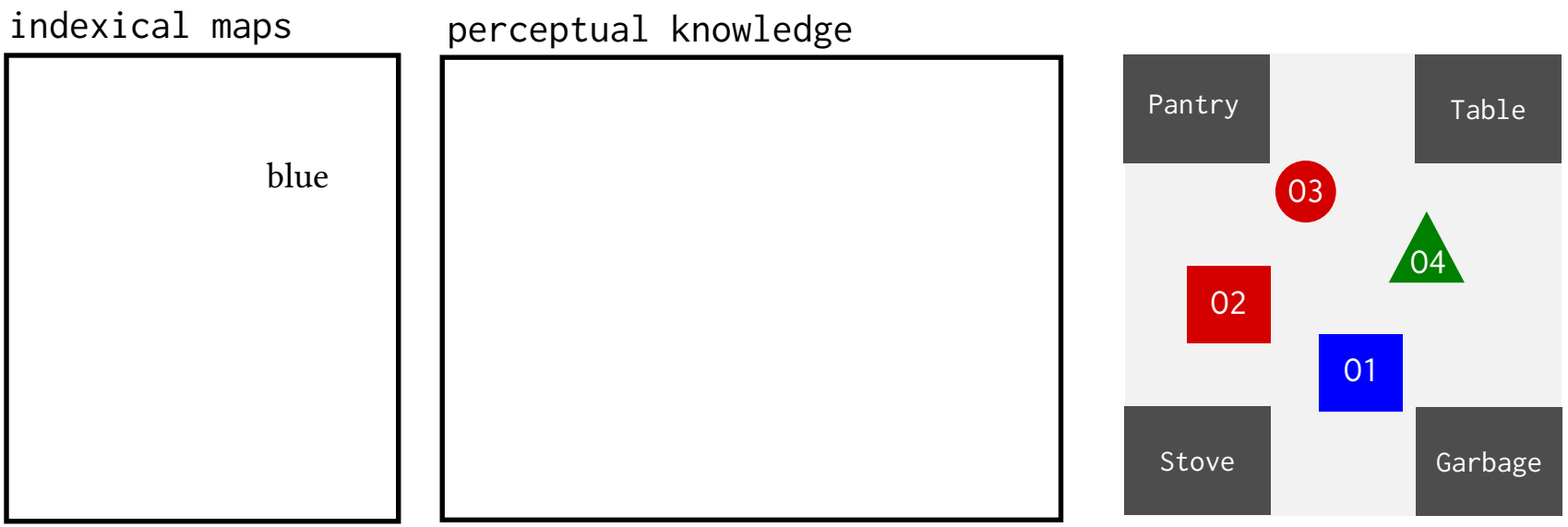
Move the blue object to the right of the pantry.



The Indexical Model

Step 1: Index components

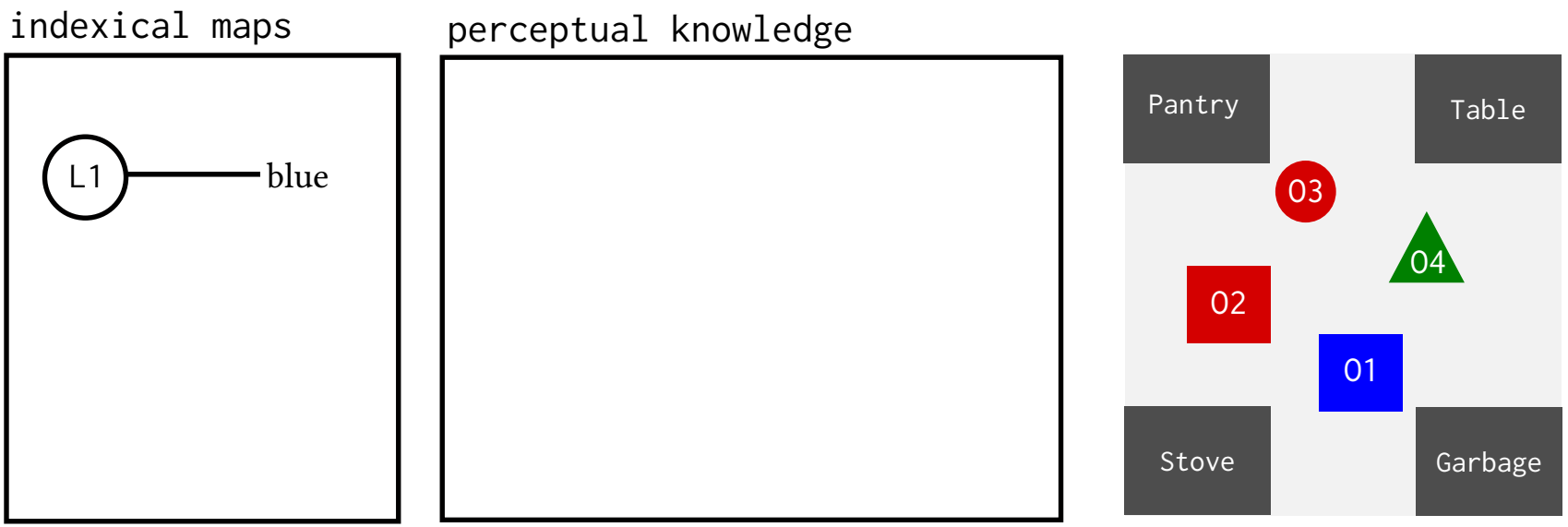
Move the blue object to the right of the pantry.



The Indexical Model

Step 1: Index components

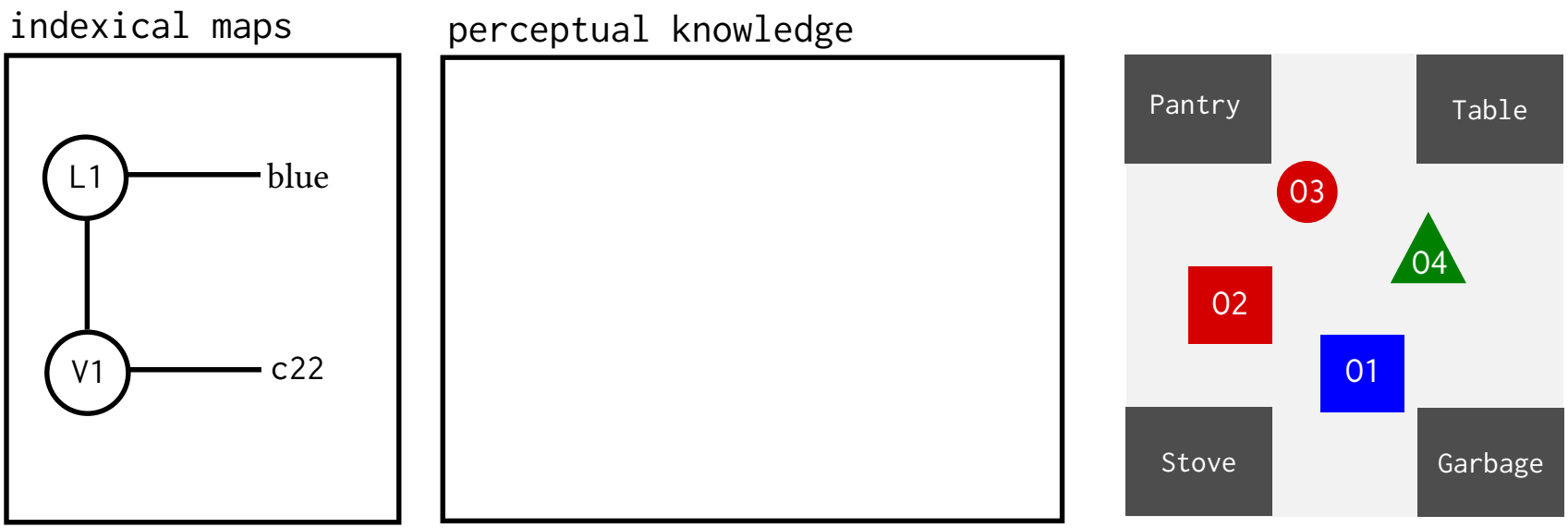
Move the blue object to the right of the pantry.



The Indexical Model

Step 1: Index components

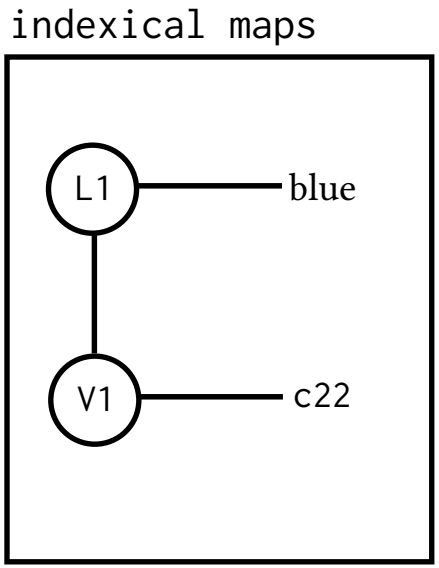
Move the blue object to the right of the pantry.

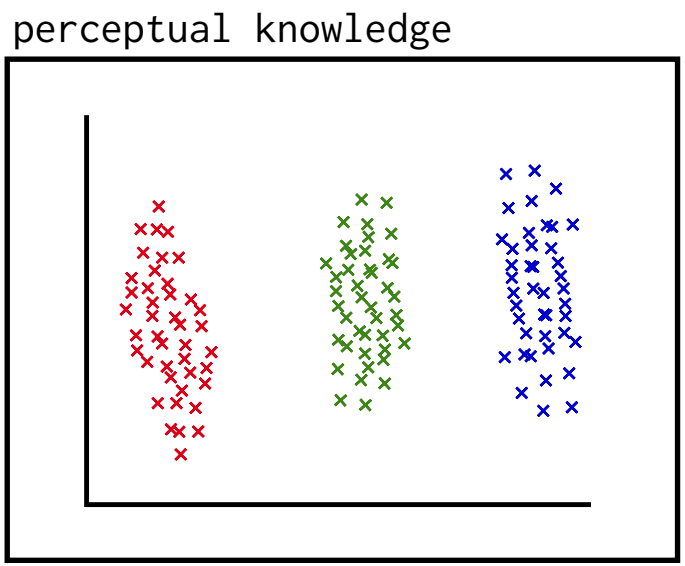


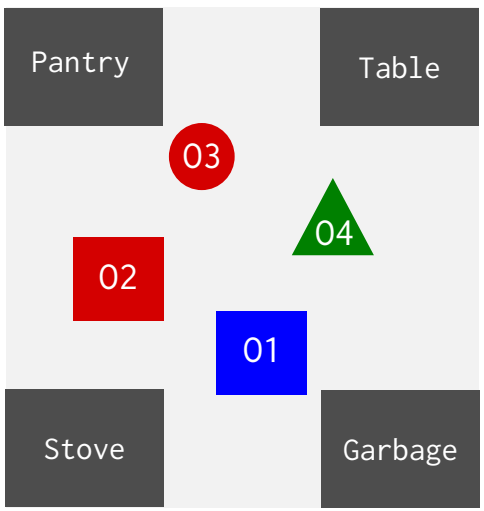
The Indexical Model

Step 1: Index components

Move the blue object to the right of the pantry.



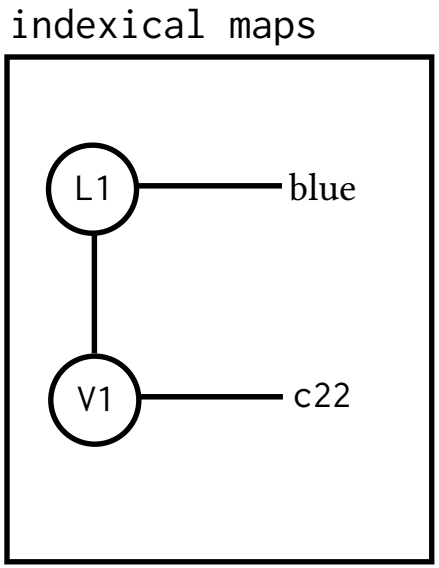


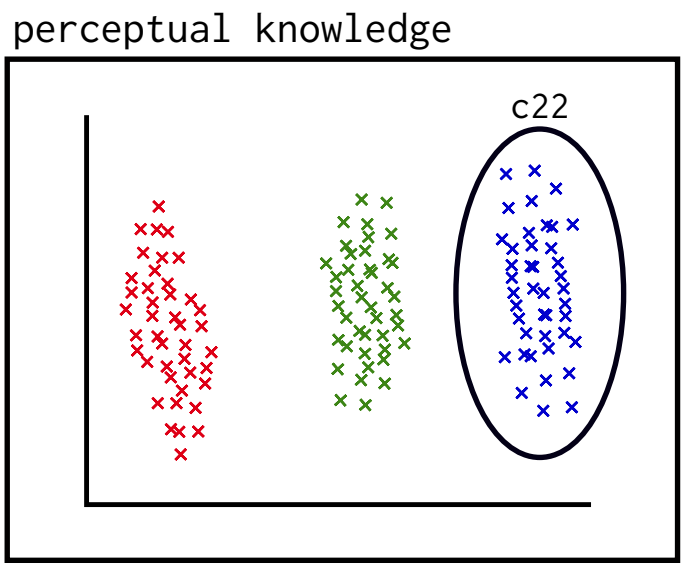


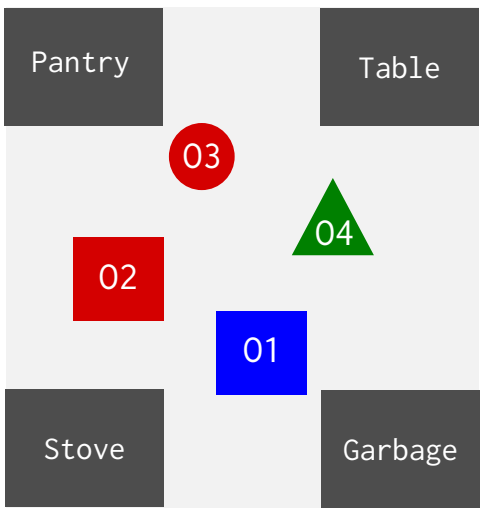
The Indexical Model

Step 1: Index components

Move the blue object to the right of the pantry.



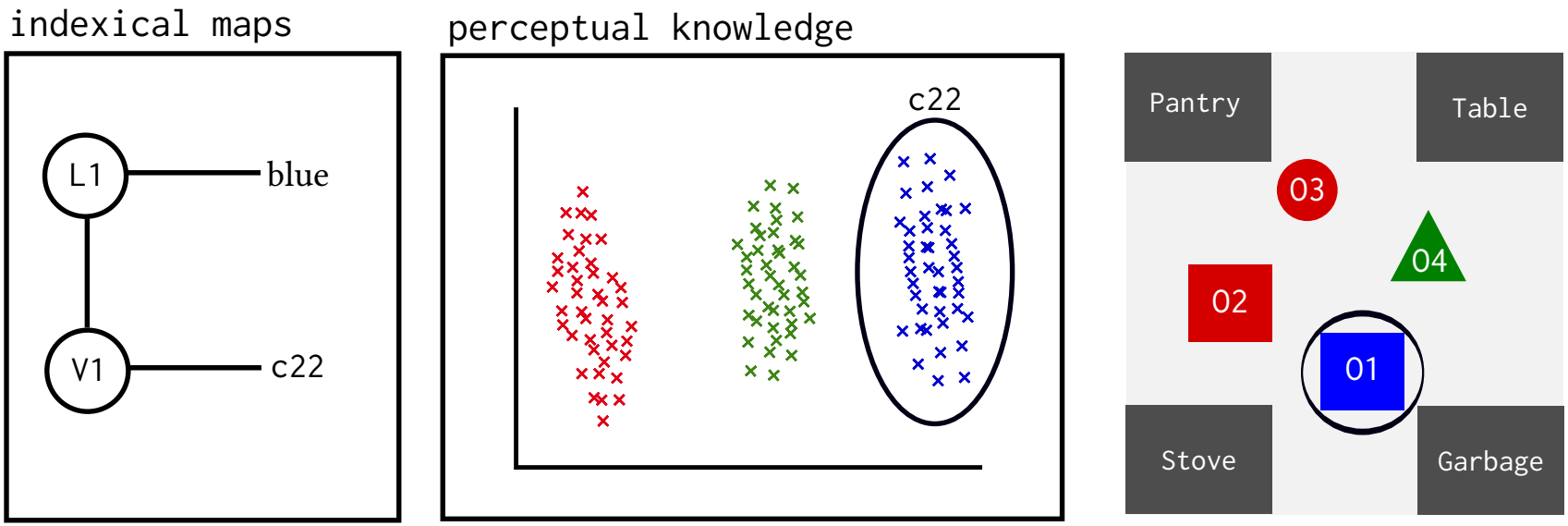




The Indexical Model

Step 1: Index components

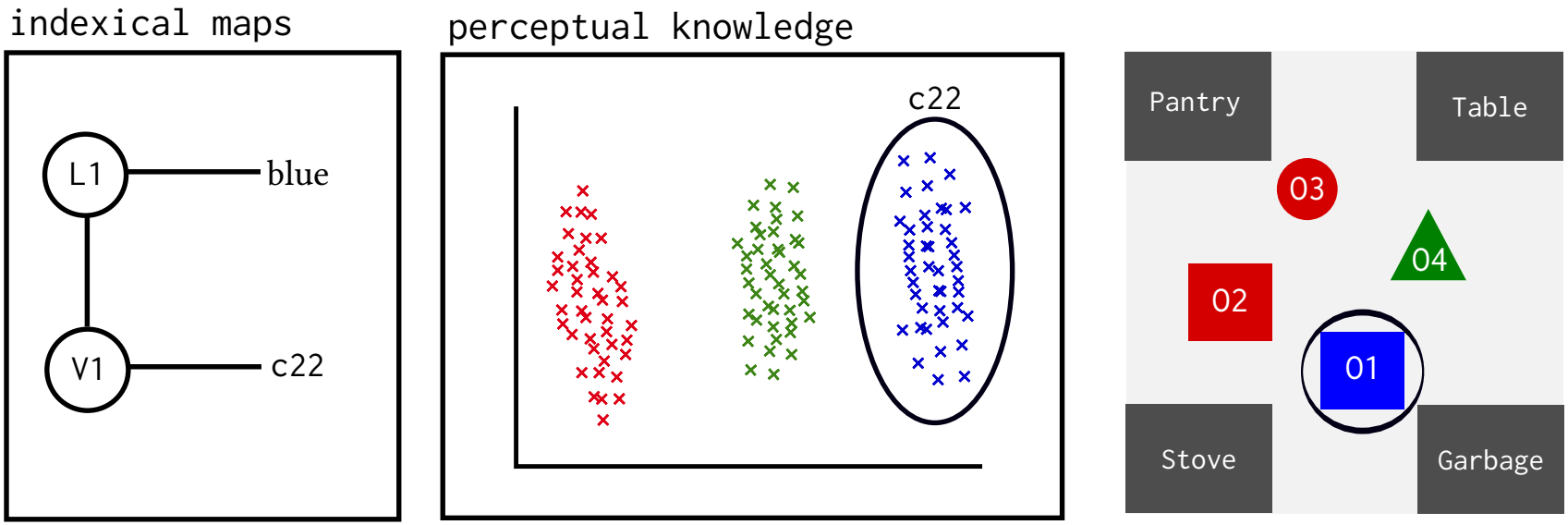
Move the blue object to the right of the pantry.



The Indexical Model

Step 1: Index components

Move the blue object to the right of the pantry.
01



The Indexical Model

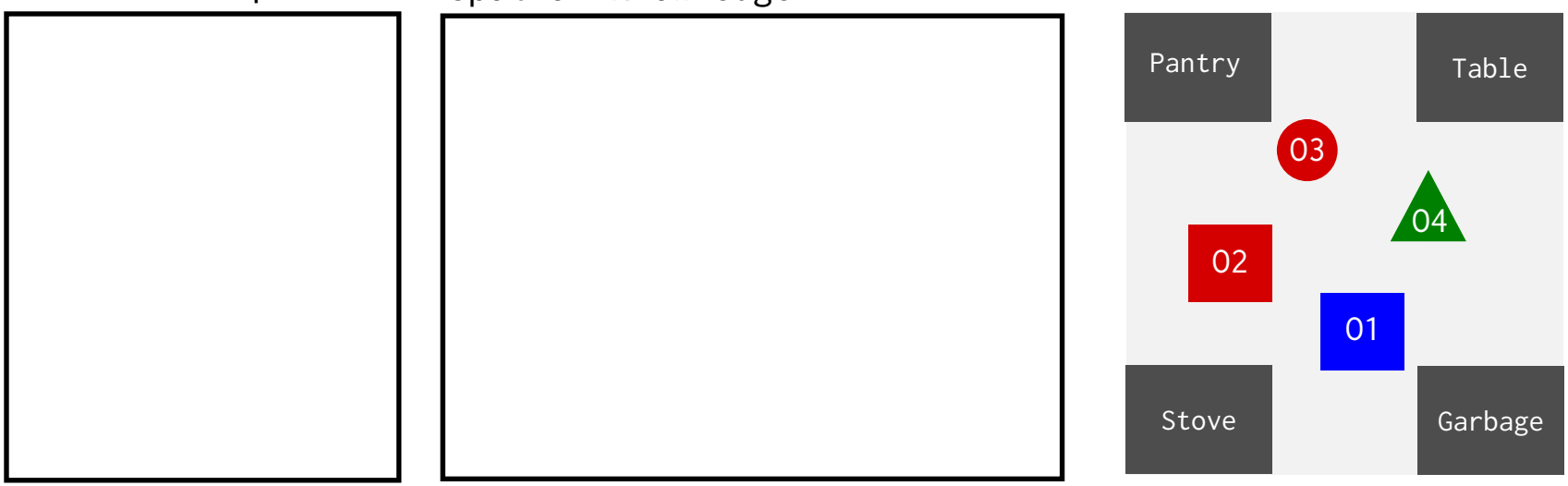
Step 1: Index components

Move the blue object to the right of the pantry.

01

indexical maps

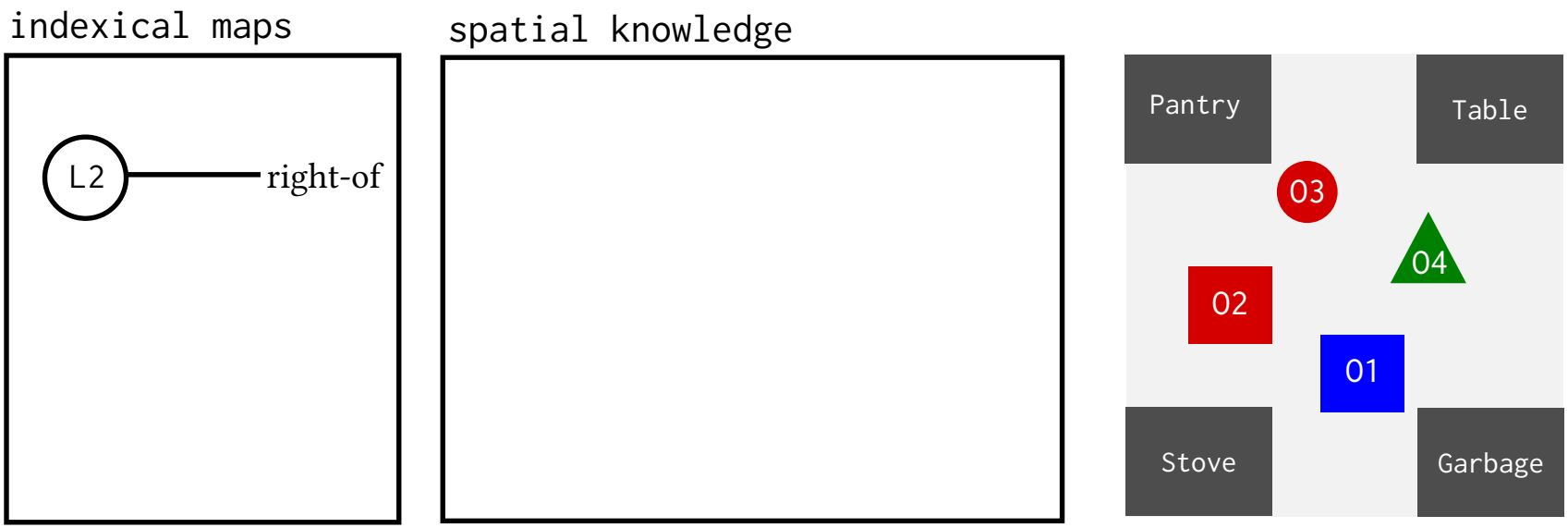
spatial knowledge



The Indexical Model

Step 1: Index components

Move the blue object to the right of the pantry.
01

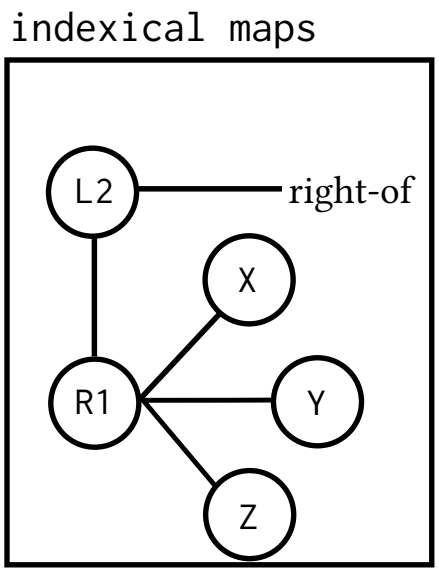


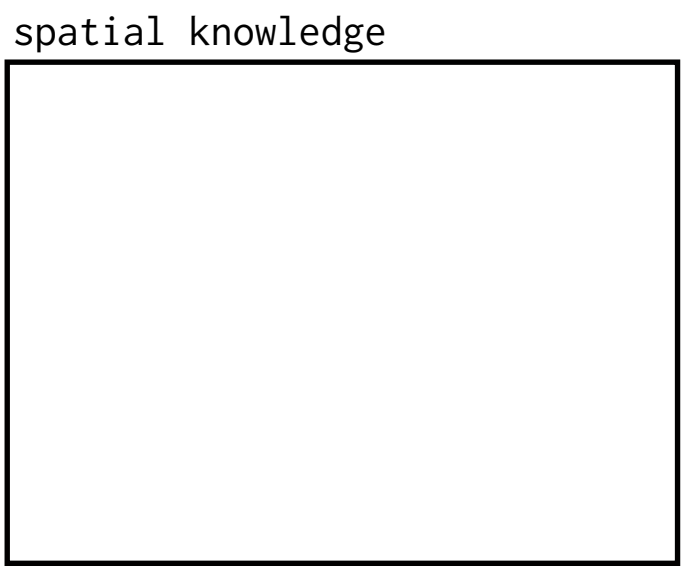
The Indexical Model

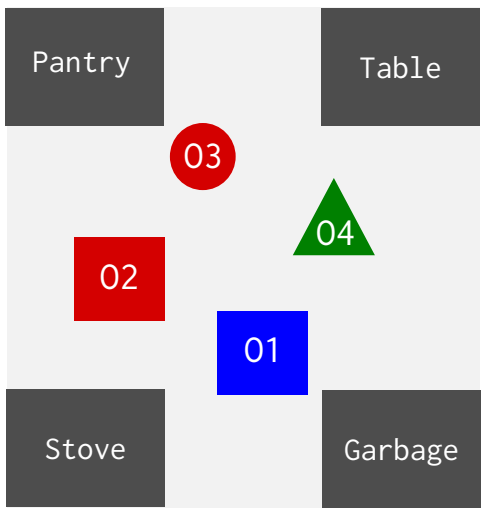
Step 1: Index components

Move the blue object to the right of the pantry.

01





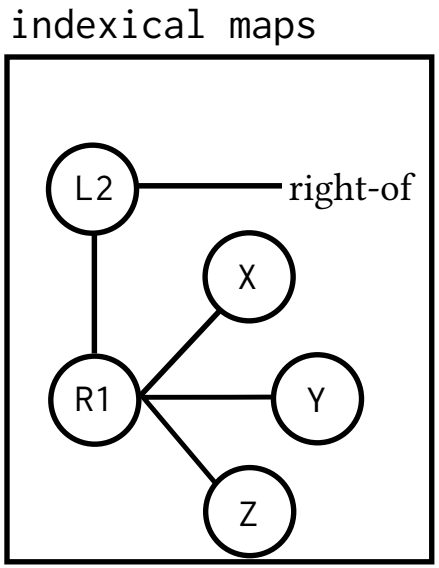


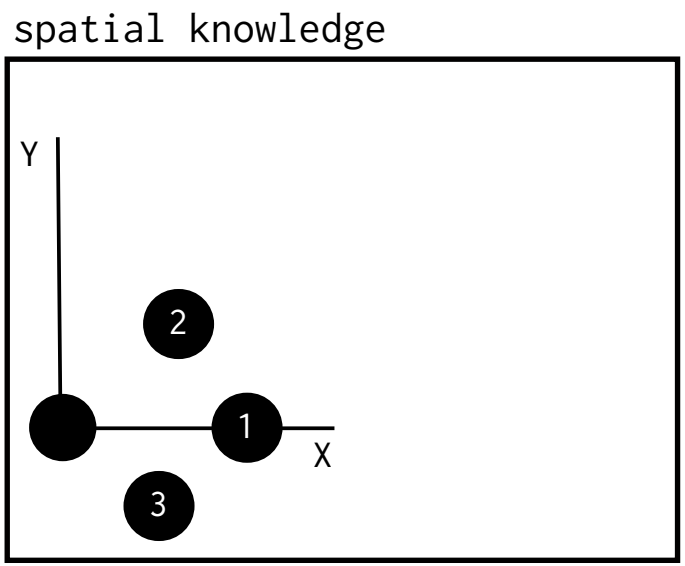
The Indexical Model

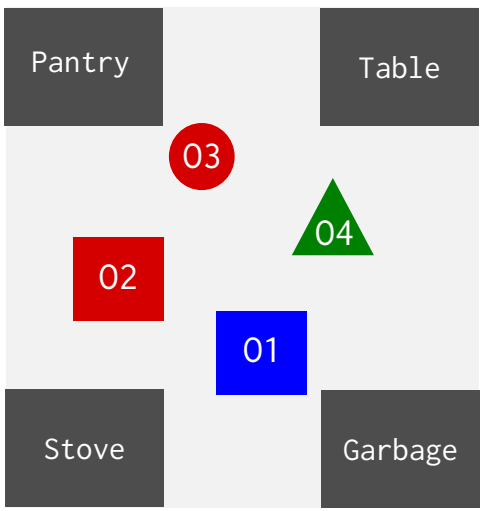
Step 1: Index components

Move the blue object to the right of the pantry.

O1



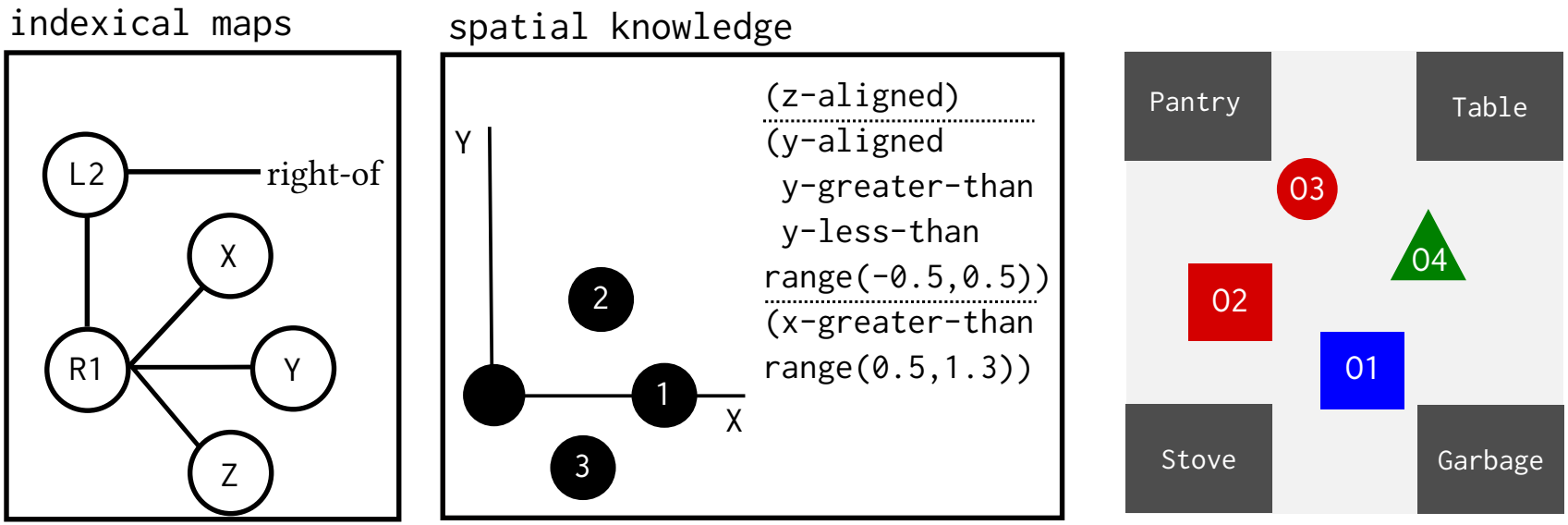




The Indexical Model

Step 1: Index components

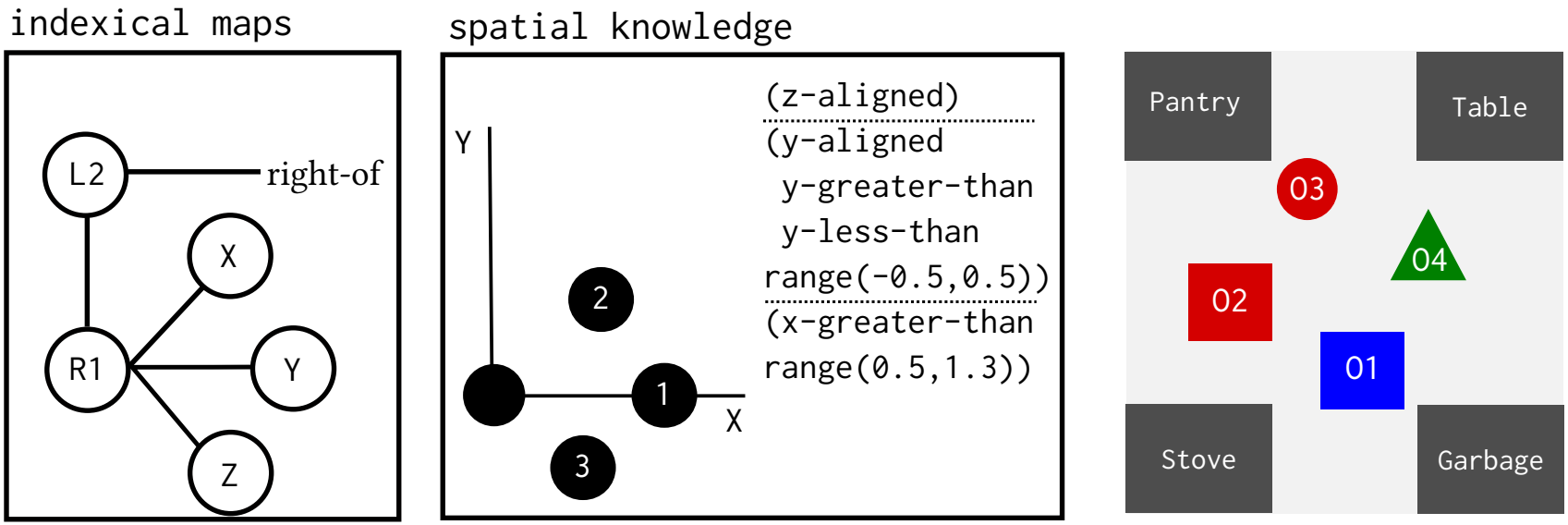
Move the blue object to the right of the pantry.
01



The Indexical Model

Step 1: Index components

Move the blue object to the right of the pantry.
O1 R1



The Indexical Model

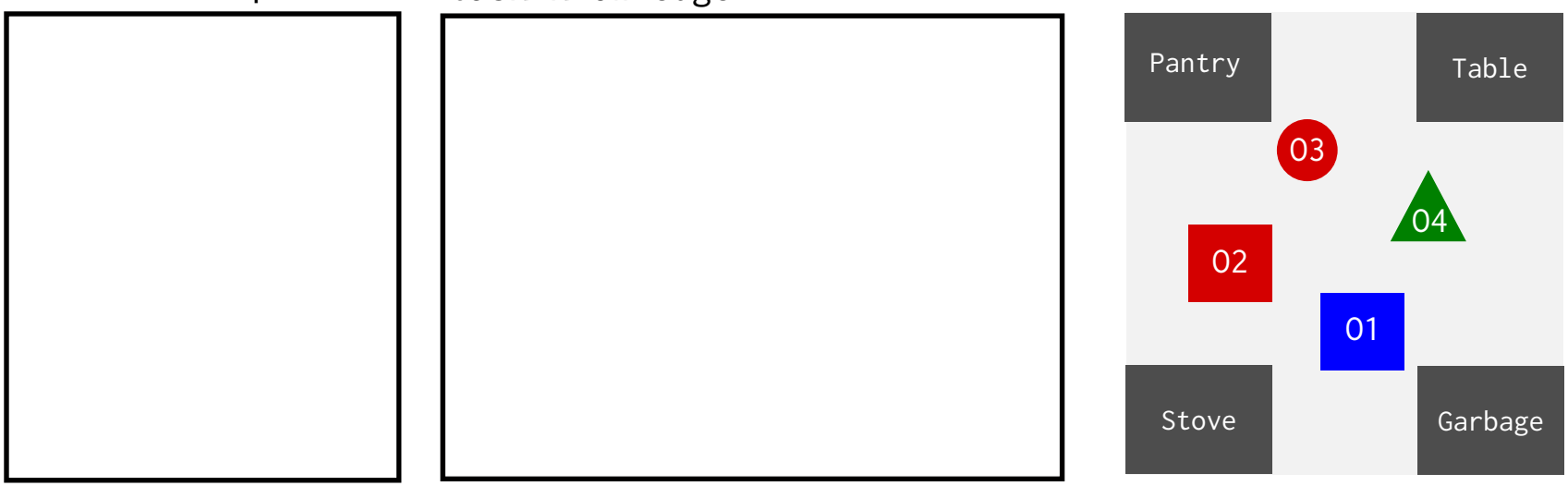
Step 1: Index components

Move the blue object to the right of the pantry.

O1 R1

indexical maps

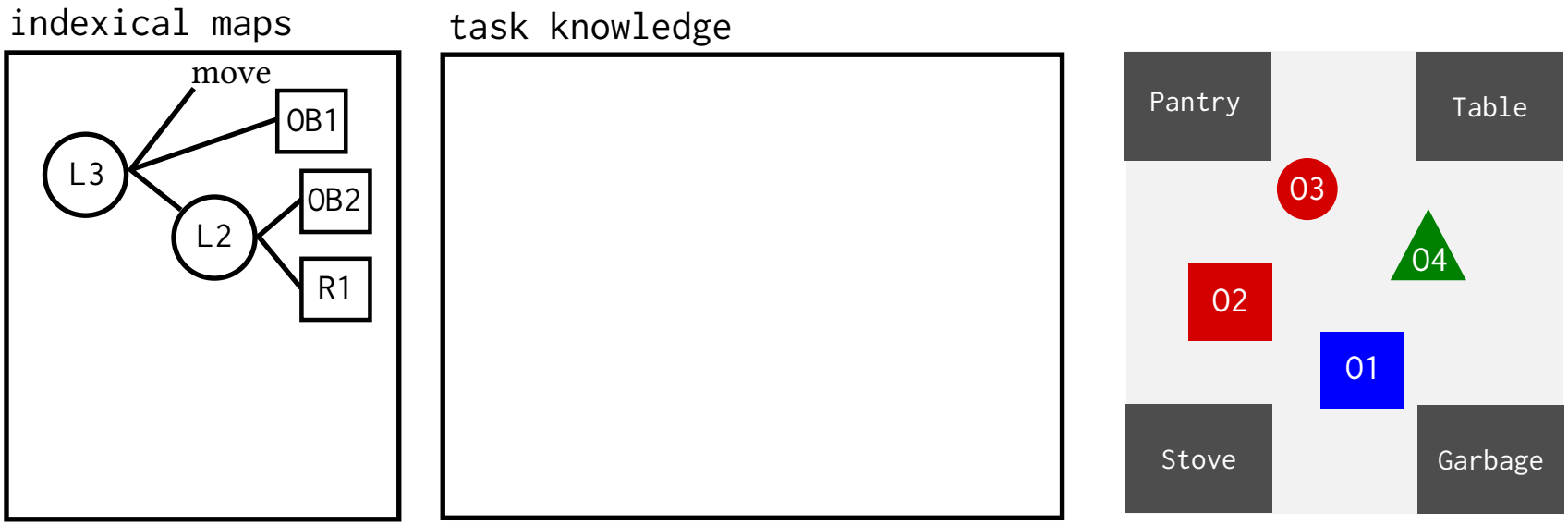
task knowledge



The Indexical Model

Step 1: Index components

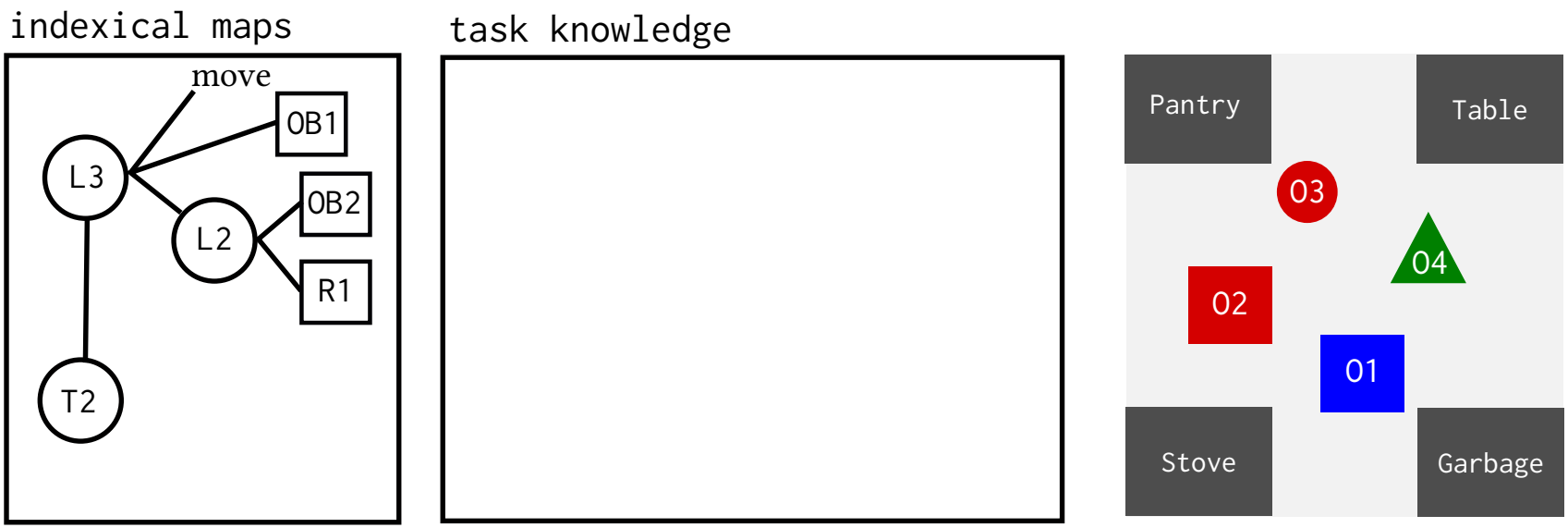
Move the blue object to the right of the pantry.
O1 R1



The Indexical Model

Step 1: Index components

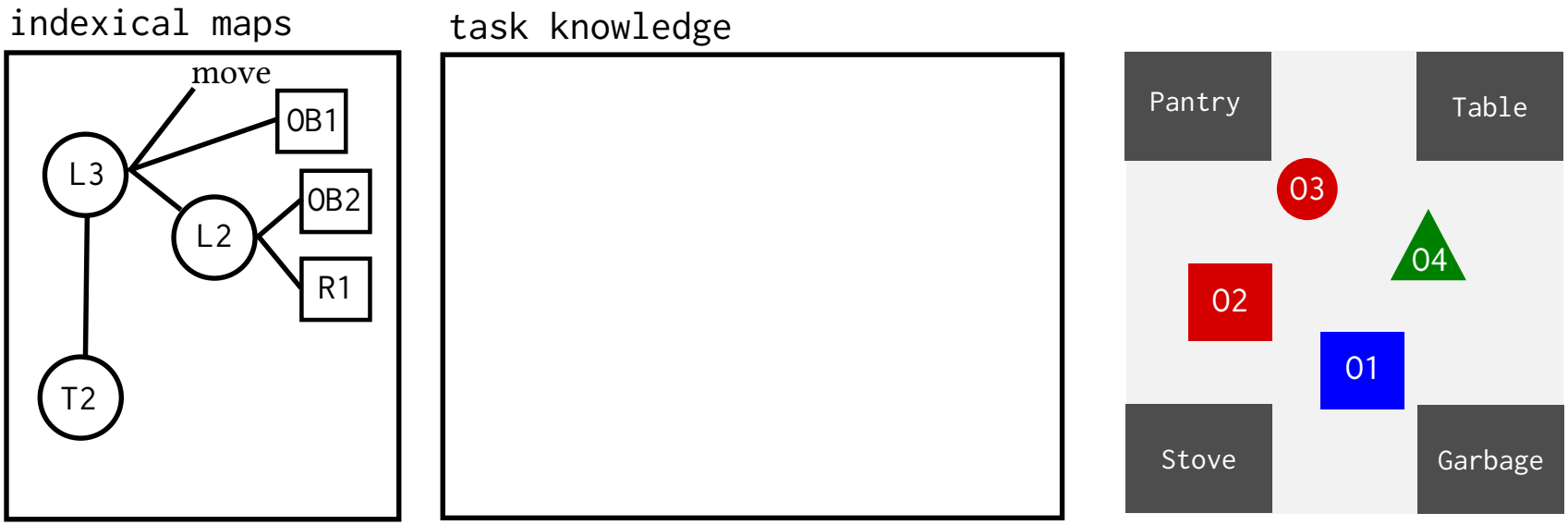
Move the blue object to the right of the pantry.
O1 R1



The Indexical Model

Step 1: Index components

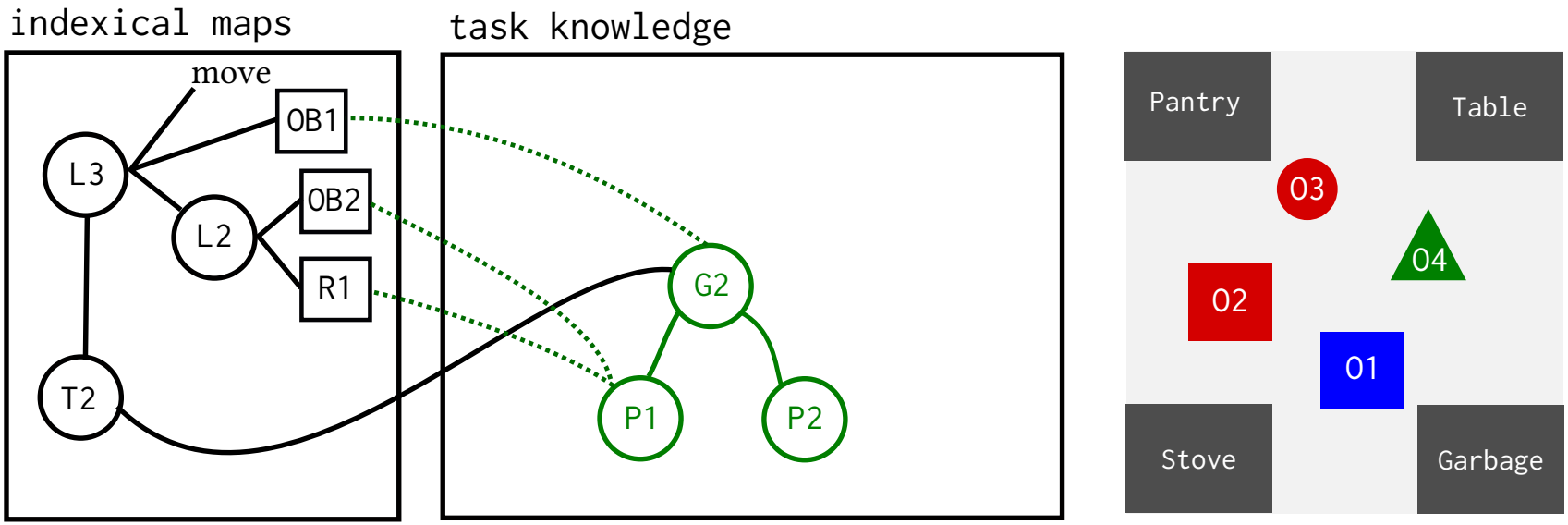
Move the blue object to the right of the pantry.
T2 O1 R1



The Indexical Model

Step 1: Index components

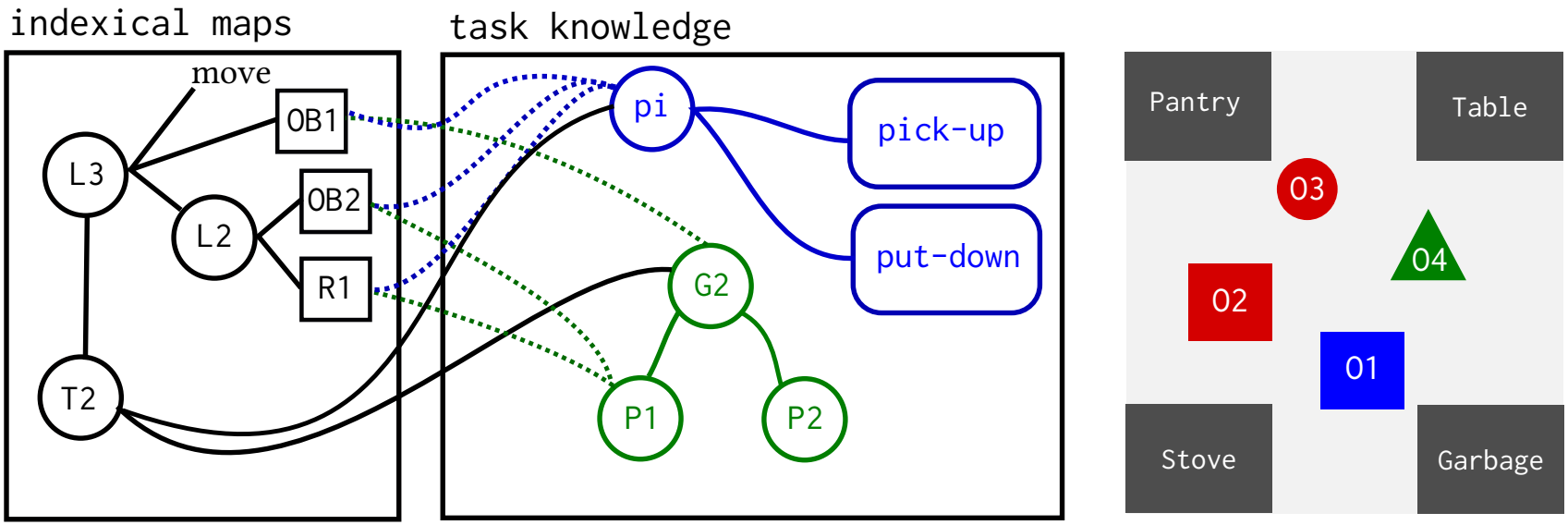
Move the blue object to the right of the pantry.
T2 O1 R1



The Indexical Model

Step 1: Index components

Move the blue object to the right of the pantry.
T2 O1 R1



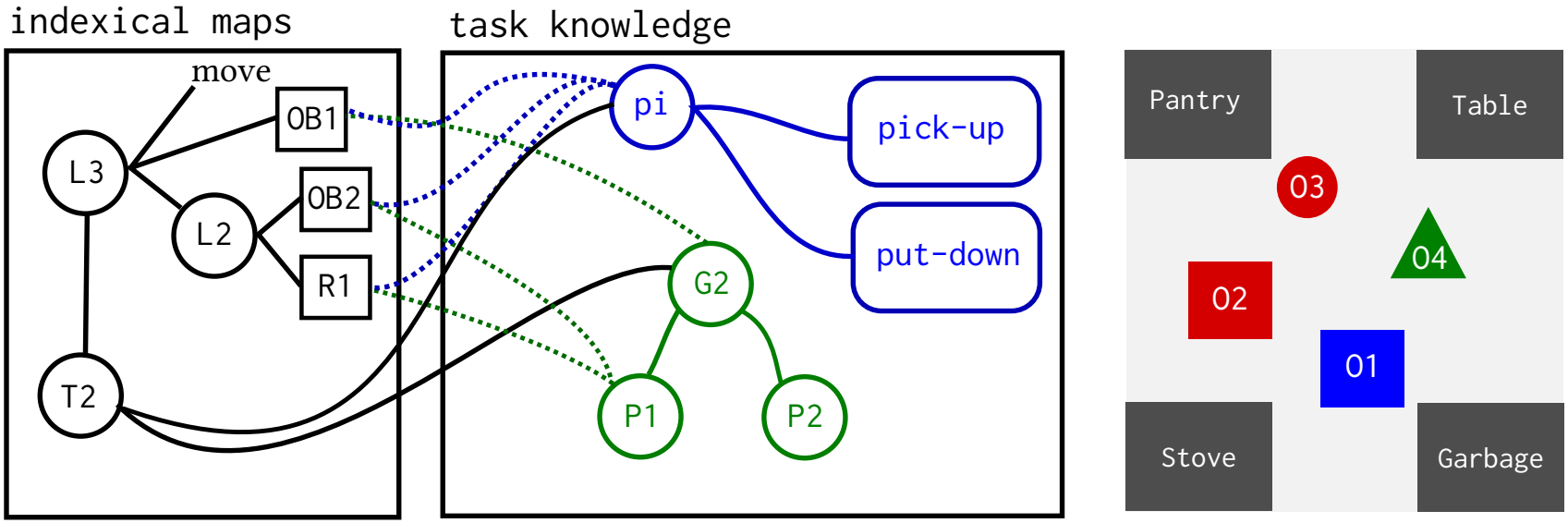
The Indexical Model

Step 2: Extract and instantiate domain knowledge

Move the blue object to the right of the pantry.

T2 O1 R1

```
task: T2(O1,(R1,pantry); G2(O1,(R1,pantry)); pi(O1,R1,pantry))
```



The Indexical Model

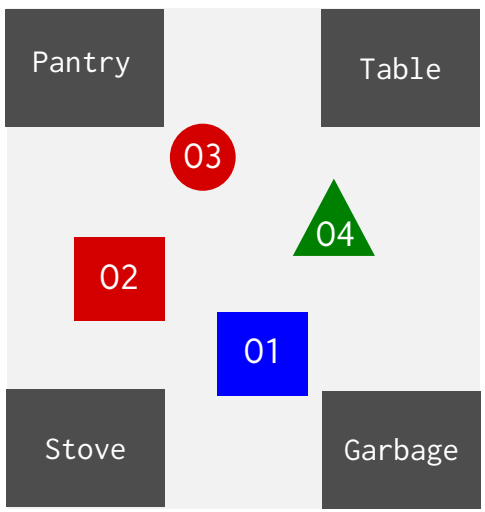
Step 3: Mesh constraints

Move the blue object to the right of the pantry.

T2 O1 R1

```
I  : T2(O1,(R1,pantry); G2(O1,(R1,pantry)); pi(O1,R1,pantry))
```

```
available [A]:
T2(O1,(R1,pantry); G2(O1,(R1,pantry)); pi(O1,R1,pantry))
T3(O1; G3(O1,(IN,pantry)); pi(O1,IN,pantry))
T4 ...
...
```



The Indexical Model

Step 3: Mesh constraints

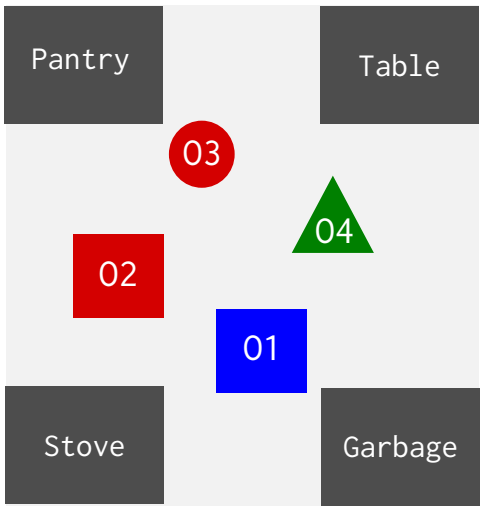
Move the blue object to the right of the pantry.

T2 O1 R1

```
I  : T2(O1,(R1,pantry); G2(O1,(R1,pantry)); pi(O1,R1,pantry))
```

```
available [A]:
T2(O1,(R1,pantry); G2(O1,(R1,pantry)); pi(O1,R1,pantry))
T3(O1; G3(O1,(IN,pantry)); pi(O1,IN,pantry))
T4 ...
...


execute [I ∩ A]
T2(O1,(R1,pantry); G2(O1,(R1,pantry)); pi(O1,R1,pantry))
```



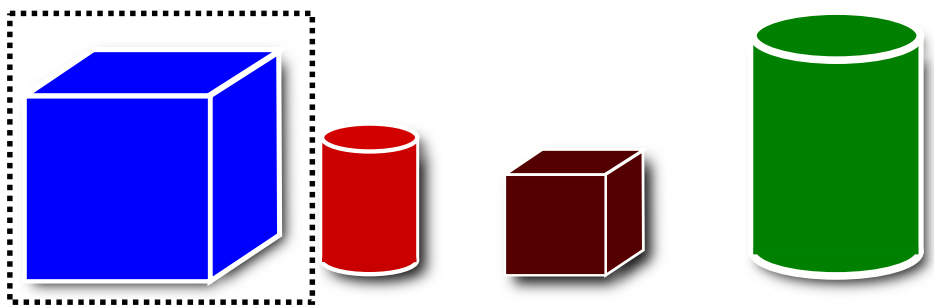
Referring Expression Resolution

Referring expressions are situational {*it*, *this cube*, *that*, *the large cube*}

Referring Expression Resolution

Referring expressions are situational {*it*, *this cube*, *that*, *the large cube*}

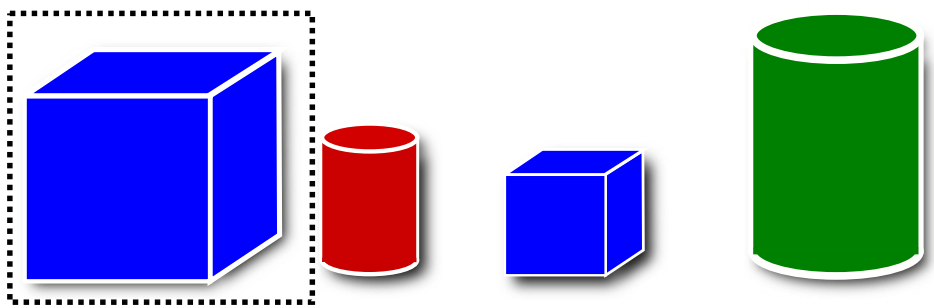
Pick up the blue cube.



Referring Expression Resolution

Referring expressions are situational {*it*, *this cube*, *that*, *the large cube*}

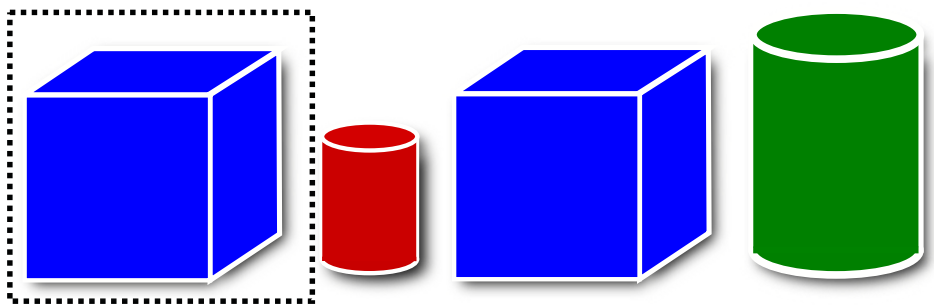
Pick up the large, blue cube.



Referring Expression Resolution

Referring expressions are situational {*it*, *this cube*, *that*, *the large cube*}

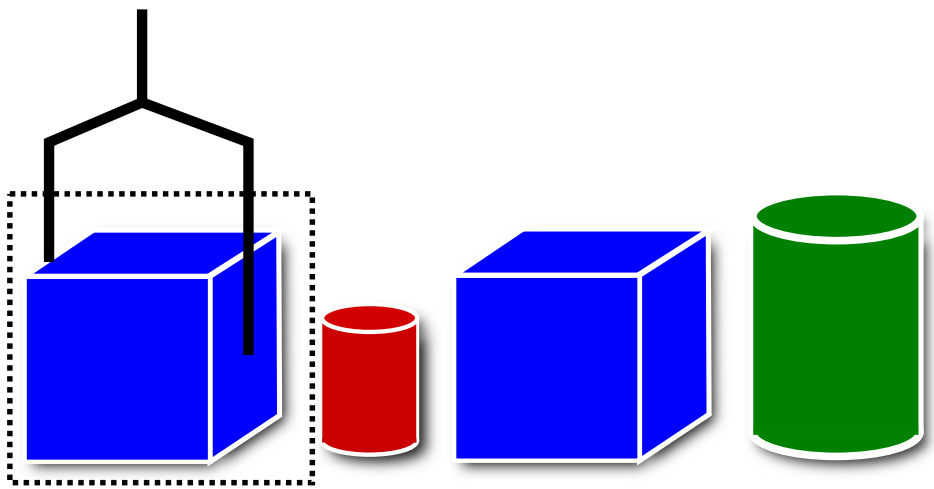
Store the cube on the left of the red cylinder. Pick it up.



Referring Expression Resolution

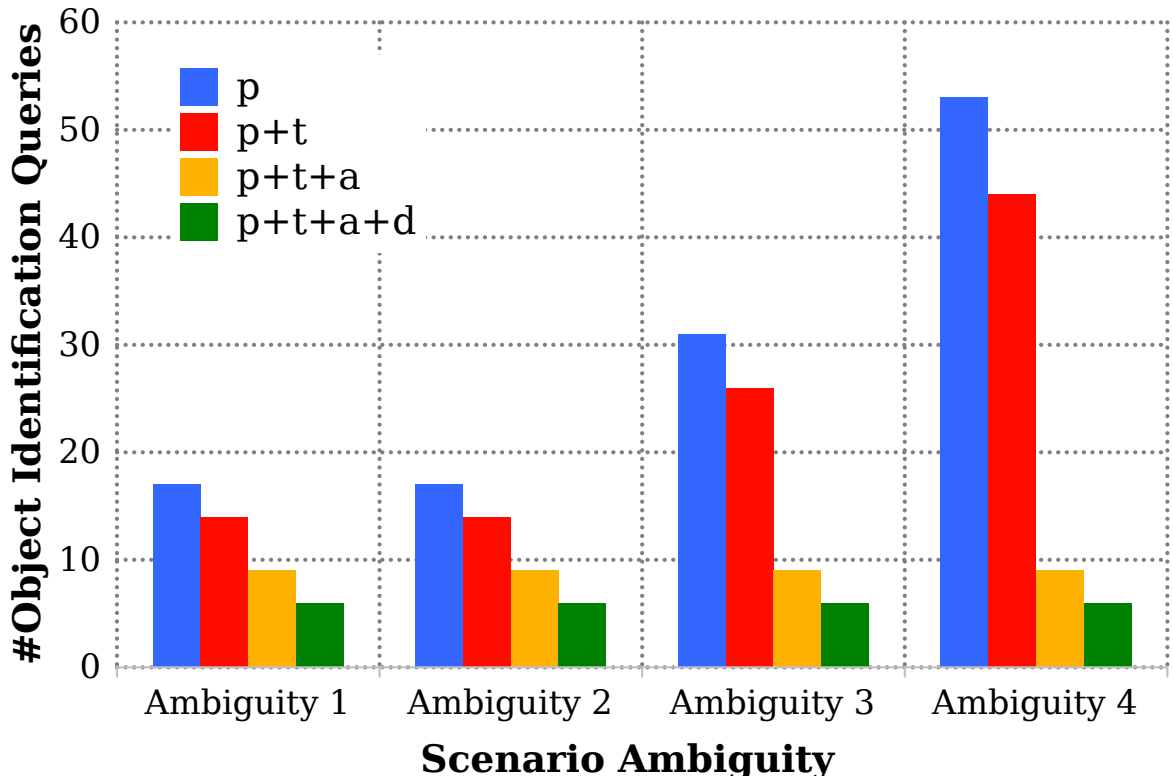
Referring expressions are situational {*it*, *this cube*, *that*, *the large cube*}

Put it down.



RE Resolution Performance

- Corpus: instructional dialogs,
  12 *it*,
  4 *this*,
  3 *that cylinder*
  14 *the red cylinder*
- Queries
  Instructor: Pick it.
  Agent: Which object?
- Linguistic context only
  Stanford CoreNLP fails at 28.6% of references.



Other Results

Unexpressed arguments

- Variability in how tasks are described.
  a. *Take the trash out to the curb.*
  b. *Take the trash out.*
- Human speakers and hearers rely on shared experience/knowledge
  - agreement on where the trash is usually deposited (common ground)
  - missing information is filled in from knowledge
- Rosie can rely on instructional experience

Comprehension scales elegantly, online with learning.

Summary

Situated Interactive Instruction

- interaction, comprehension, learning

Interactive task learning

- EBL based comprehensive task learning
- fast generalization, learning
- distributed onus of learning

Situated comprehension

- language features are cues to search the common ground
- non-linguistic context is useful for resolving ambiguities

Future Work

Interactive task learning

- other types of tasks: maintenance, performance
- handling instructional errors
- integration with *LfD*

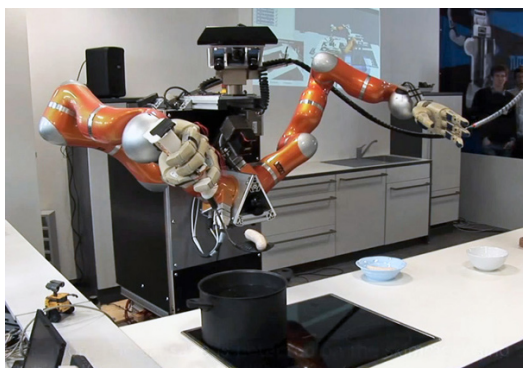
Comprehension

- ambiguity: pp-phrase, verb-task polysemy
- complexity: partially-observable environments

HRI experiments

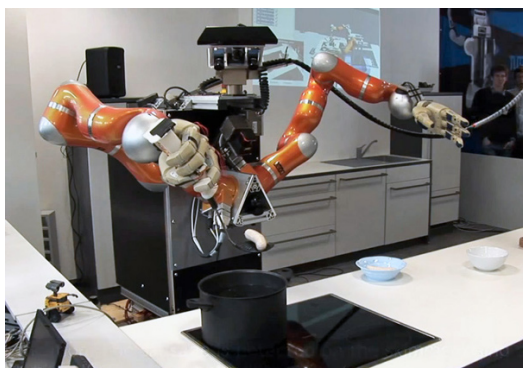
Artificial Autonomous Collaborators

intelligent co-operative behavior in novel environments



Artificial Autonomous Collaborators

intelligent co-operative behavior in novel environments



interaction, agent design, architecture

Experience

Experience

interaction

vector space language models

Chatterjee and Mohan
(ICTAI 2007; CICLING 2008)

Experience

interaction

vector space language models

Chatterjee and Mohan
(ICTAI 2007; CICLING 2008)

agent design

RL in complex domains

Mohan and Laird
(AIIDE 2010; AAAI 2009)

Joshi et al.
(STAIRS 2012; ICTAI 2012)

Experience

interaction

vector space language models

Chatterjee and Mohan
(ICTAI 2007; CICLING 2008)

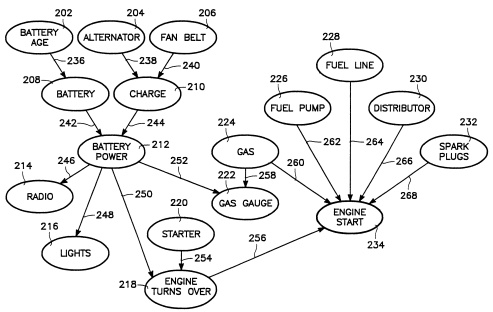
agent design

RL in complex domains

Mohan and Laird
(AIIDE 2010; AAAI 2009)

Joshi et al.
(STAIRS 2012; ICTAI 2012)

architecture



rational long-term memory

Li and Mohan (ongoing)

Interaction

Interaction

Addressing a variety of social constructs

- situated interactions are limiting
- human communication is largely non-situated
- learning by reading, from the web and media, from story-telling
- challenges: accumulate common ground through simulations

Interaction

Addressing a variety of social constructs

- situated interactions are limiting
- human communication is largely non-situated
- learning by reading, from the web and media, from story-telling
- challenges: accumulate common ground through simulations

Multi-modal interaction

- language is not always efficient
- human interaction involves secondary signals
- interactions through diagrams, expression (linguistic and non)

Interaction

Addressing a variety of social constructs

- situated interactions are limiting
- human communication is largely non-situated
- learning by reading, from the web and media, from story-telling
- challenges: accumulate common ground through simulations

Multi-modal interaction

- language is not always efficient
- human interaction involves secondary signals
- interactions through diagrams, expression (linguistic and non)

Deployment in real scenarios

Agent Design

Agent Design

Modeling human collaborators

- a good collaborator adapts to
  - task participants
  - varying situations
- requires robust models
- cognitive: personality, communication style, task preferences
- situational: time constraints, expectations, cognitive bandwidth
- predictive interaction

Agent Design

Modeling human collaborators

- a good collaborator adapts to
  - task participants
  - varying situations
- requires robust models
- cognitive: personality, communication style, task preferences
- situational: time constraints, expectations, cognitive bandwidth
- predictive interaction

Integrated learning

- guided knowledge acquisition
- autonomous exploration
- situational adaptation

Architecture

Architecture

Robustness to noise

- interaction is extremely noisy
- mental states are partially-observable at best
- dealing with incorrect learning and adaptation
- reliable long-term behavior

Architecture

Robustness to noise

- interaction is extremely noisy
- mental states are partially-observable at best
- dealing with incorrect learning and adaptation
- reliable long-term behavior

Handling real interactions in real scenarios

- limited computational resources
- complex, dynamic situations