Opening the pod bay doors

building intelligent agents that can interpret, generate and learn from natural language

Jacob Andreas

Language & Intelligence @ MIT CSAIL

LINGO.csail.mit.edu

Following natural language instructions

https://www.youtube.com/watch?v=-3m-Zu3qgM4

https://www.youtube.com/watch?v=G_v5B_gYceM

https://www.marcelvarallo.com/the-ballad-of-roomba-part2/

https://www.freep.com/story/money/cars/general-motors/2019/07/24/gms-self-driving-car-robot-taxi

Following natural language instructions

Instruction following: ingredients

Context

Environment

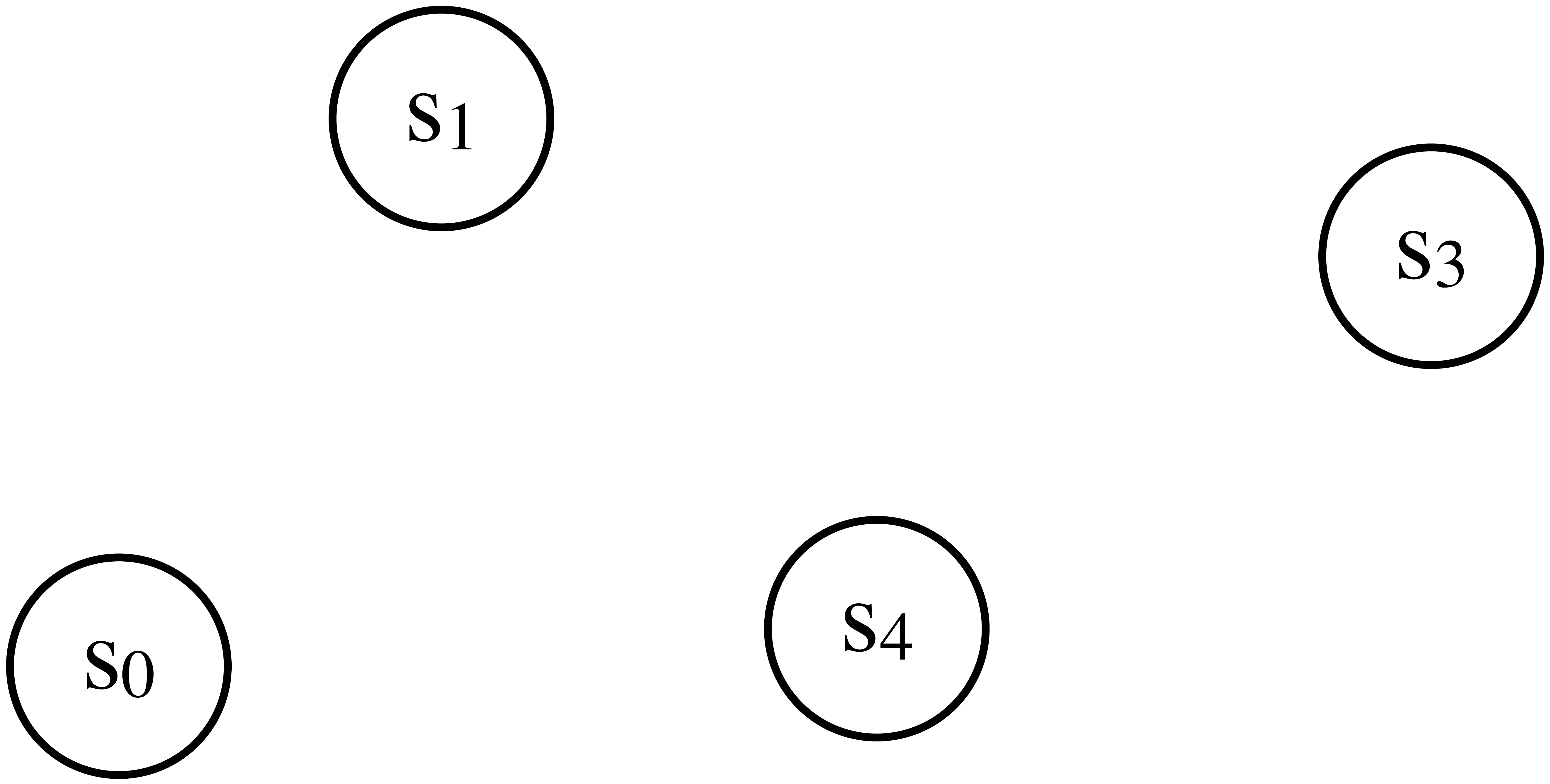
Actions

Data

Instructions

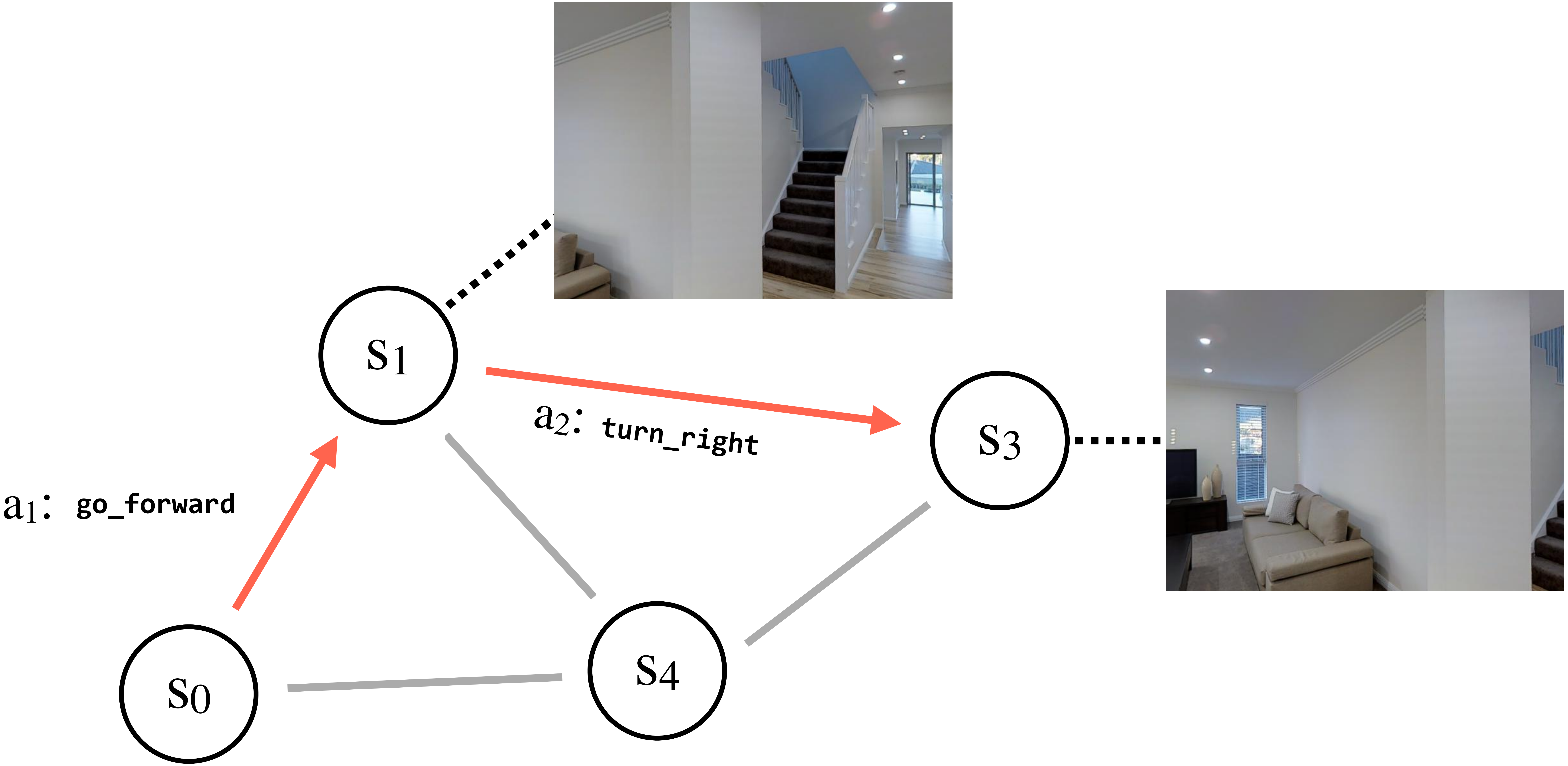
Supervision

Context: Environments & Actions

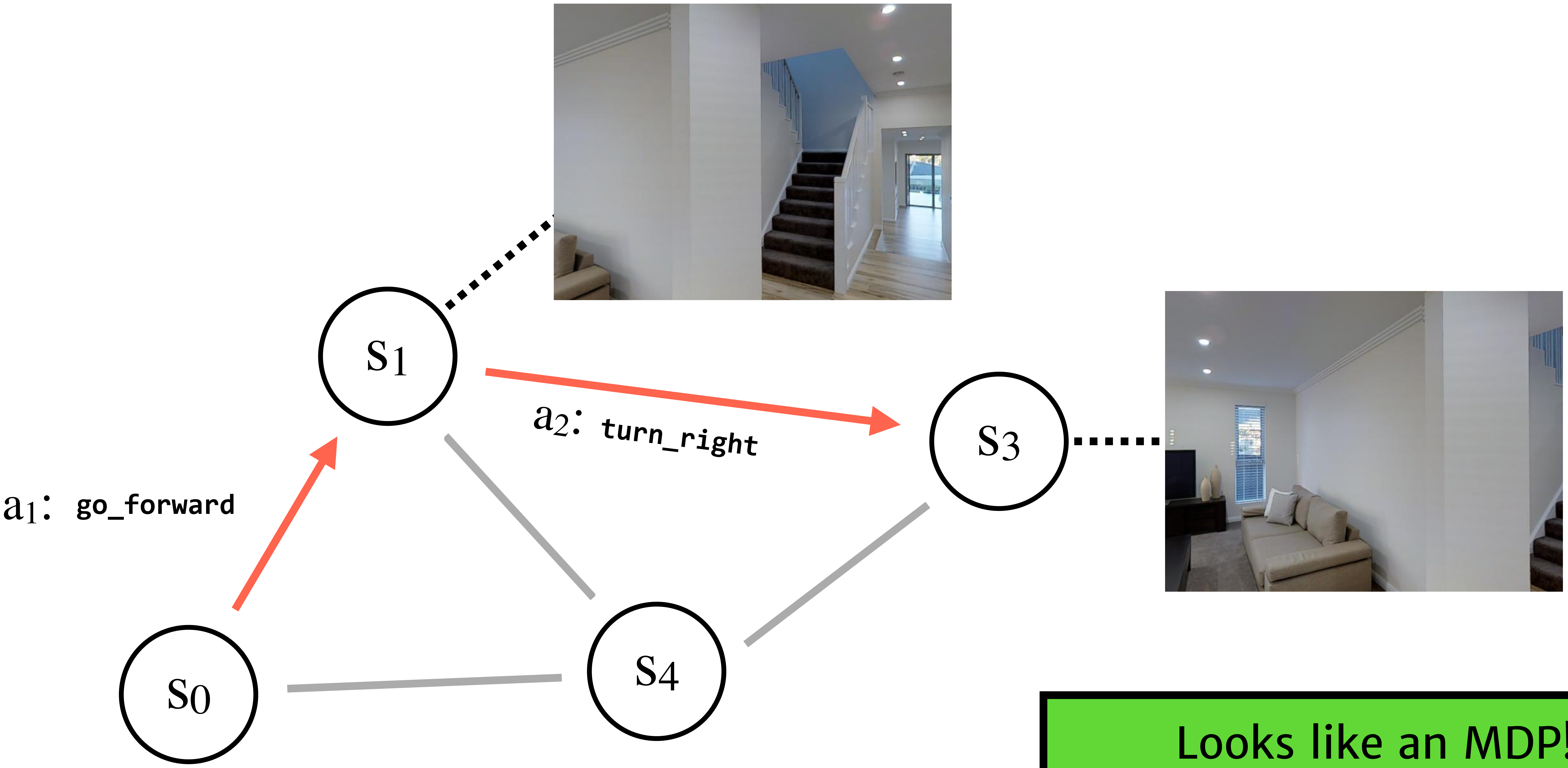


Context: Environments & Actions

Context: Environments & Actions

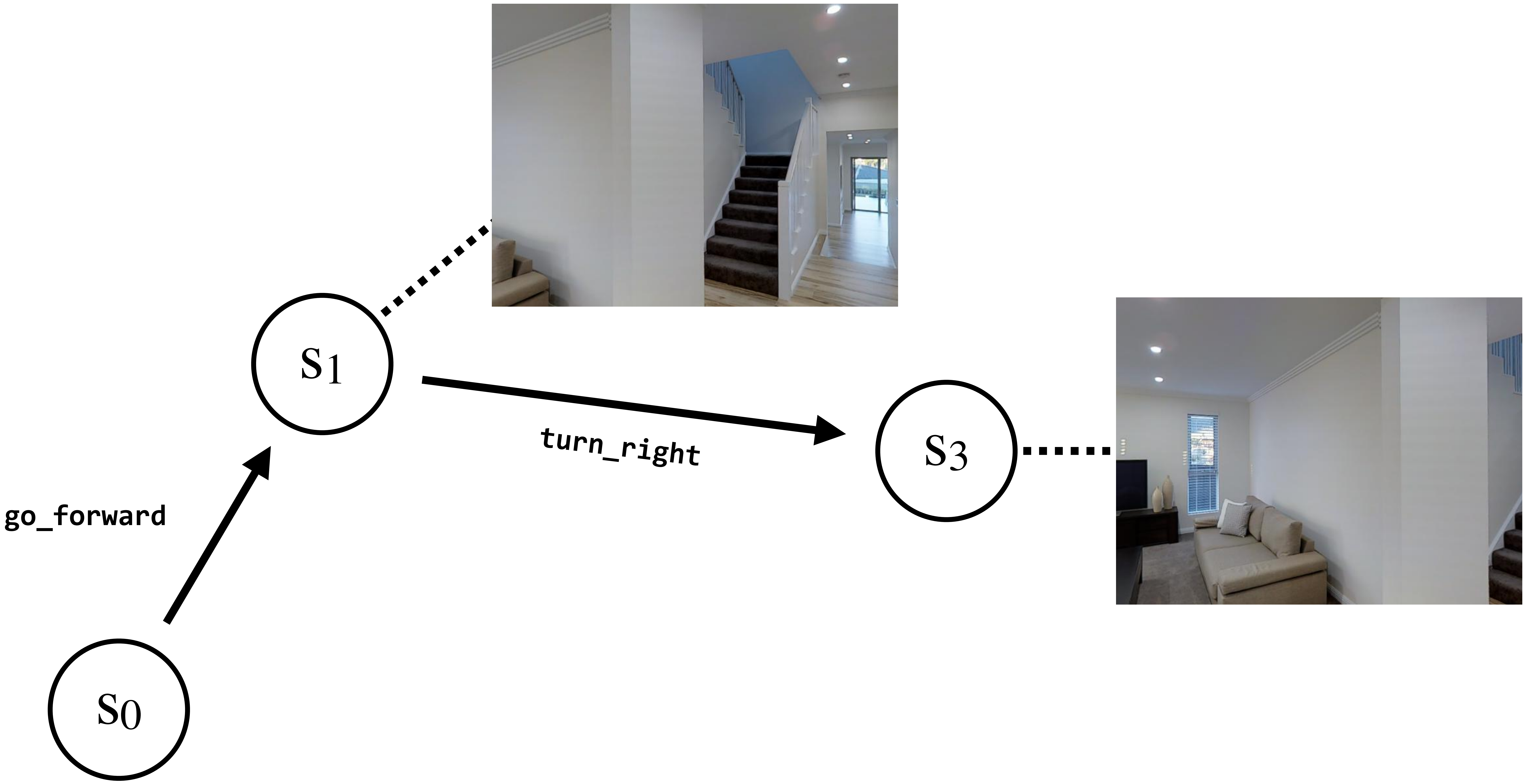


Context: Environments & Actions



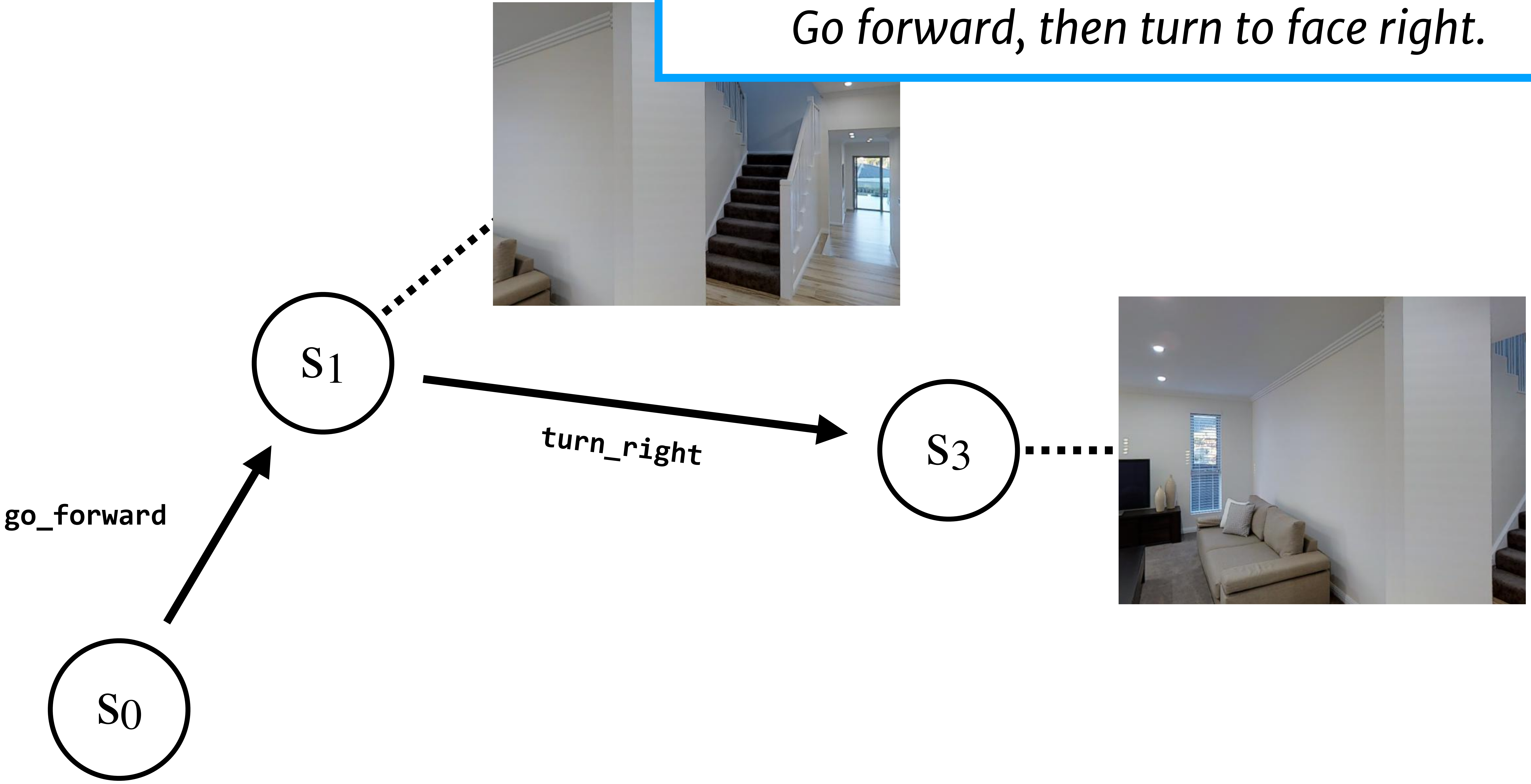
Looks like an MDP!

Instructions



Instructions

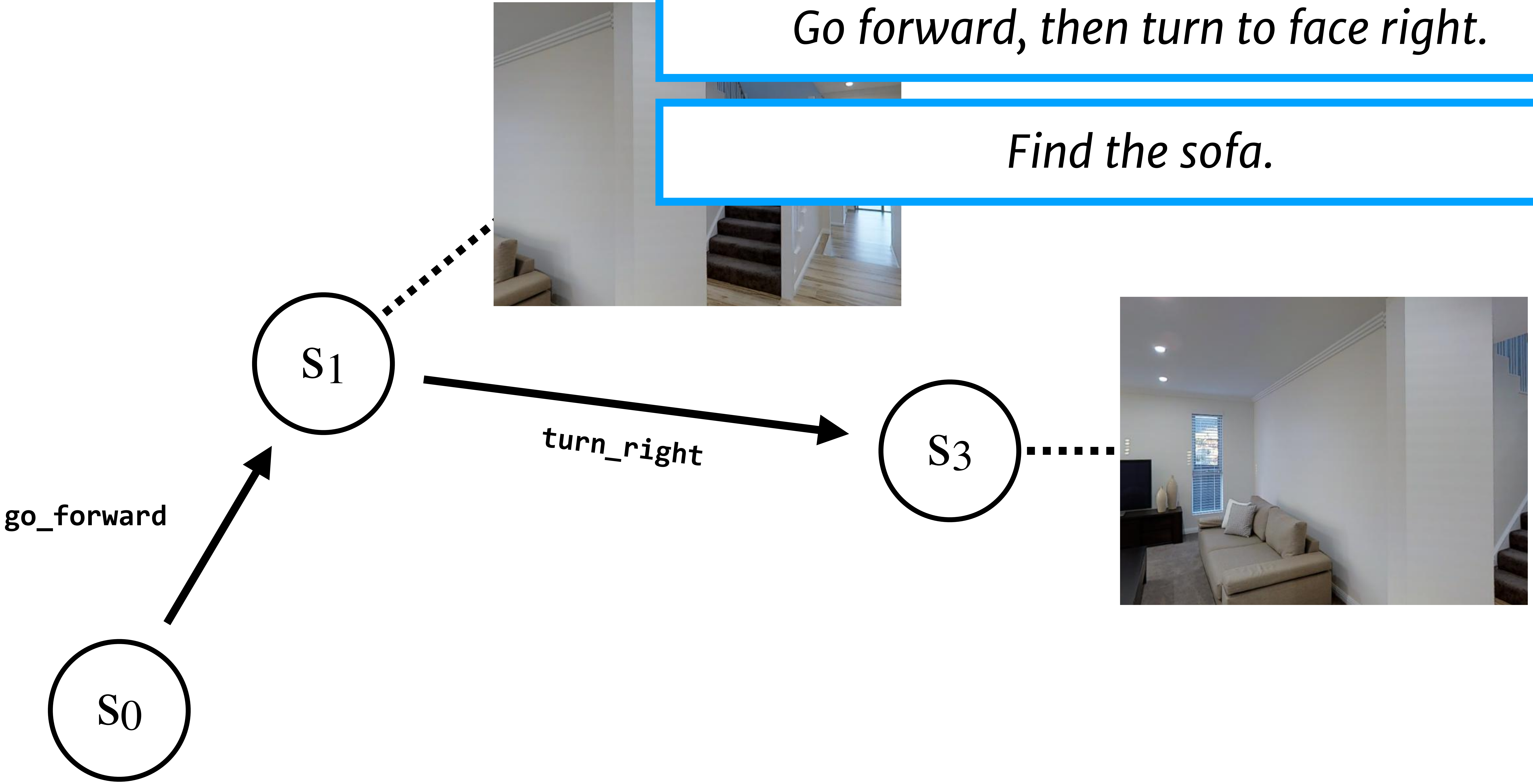
Go forward, then turn to face right.



Instructions

Go forward, then turn to face right.

Find the sofa.

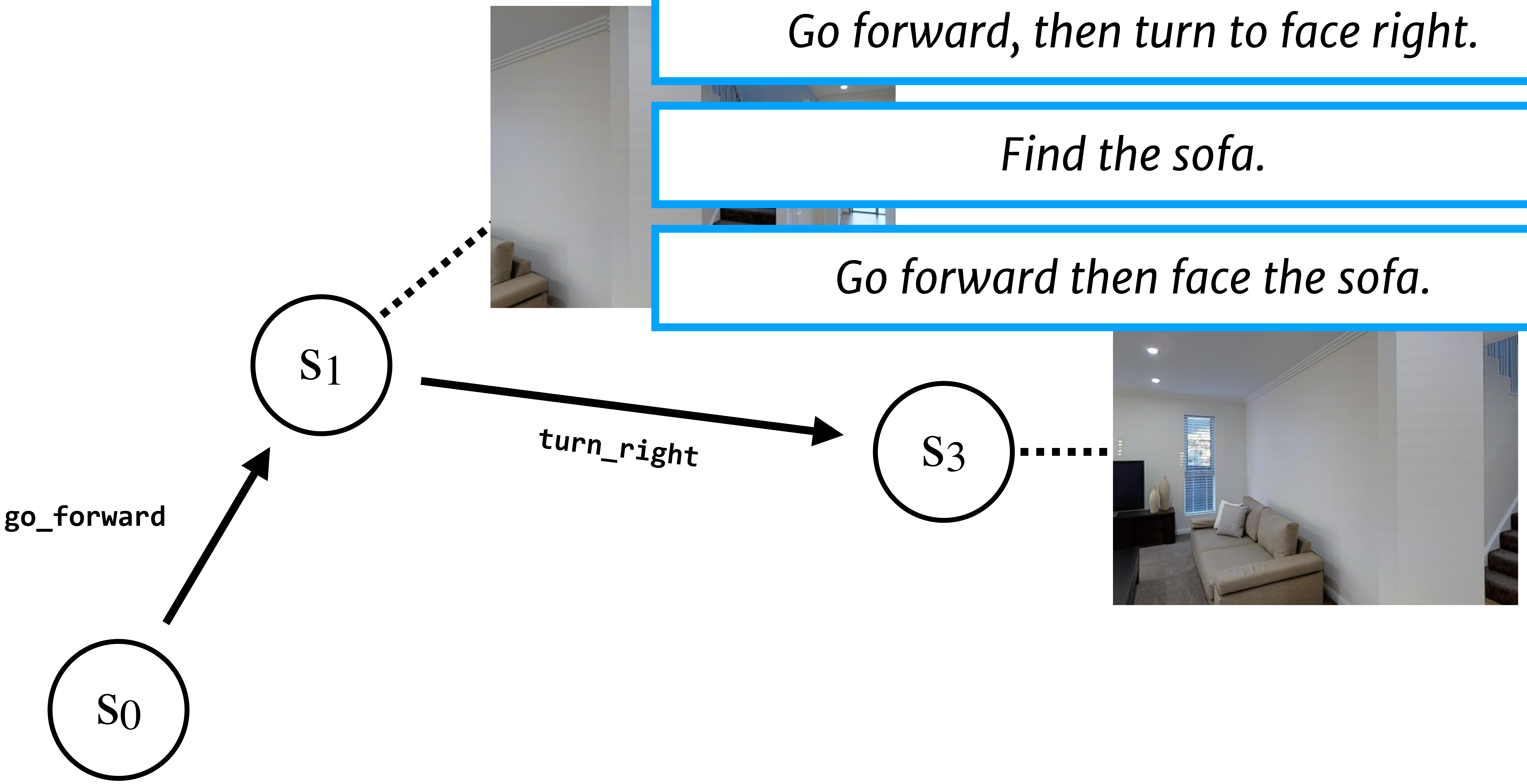


Instructions

Go forward, then turn to face right.

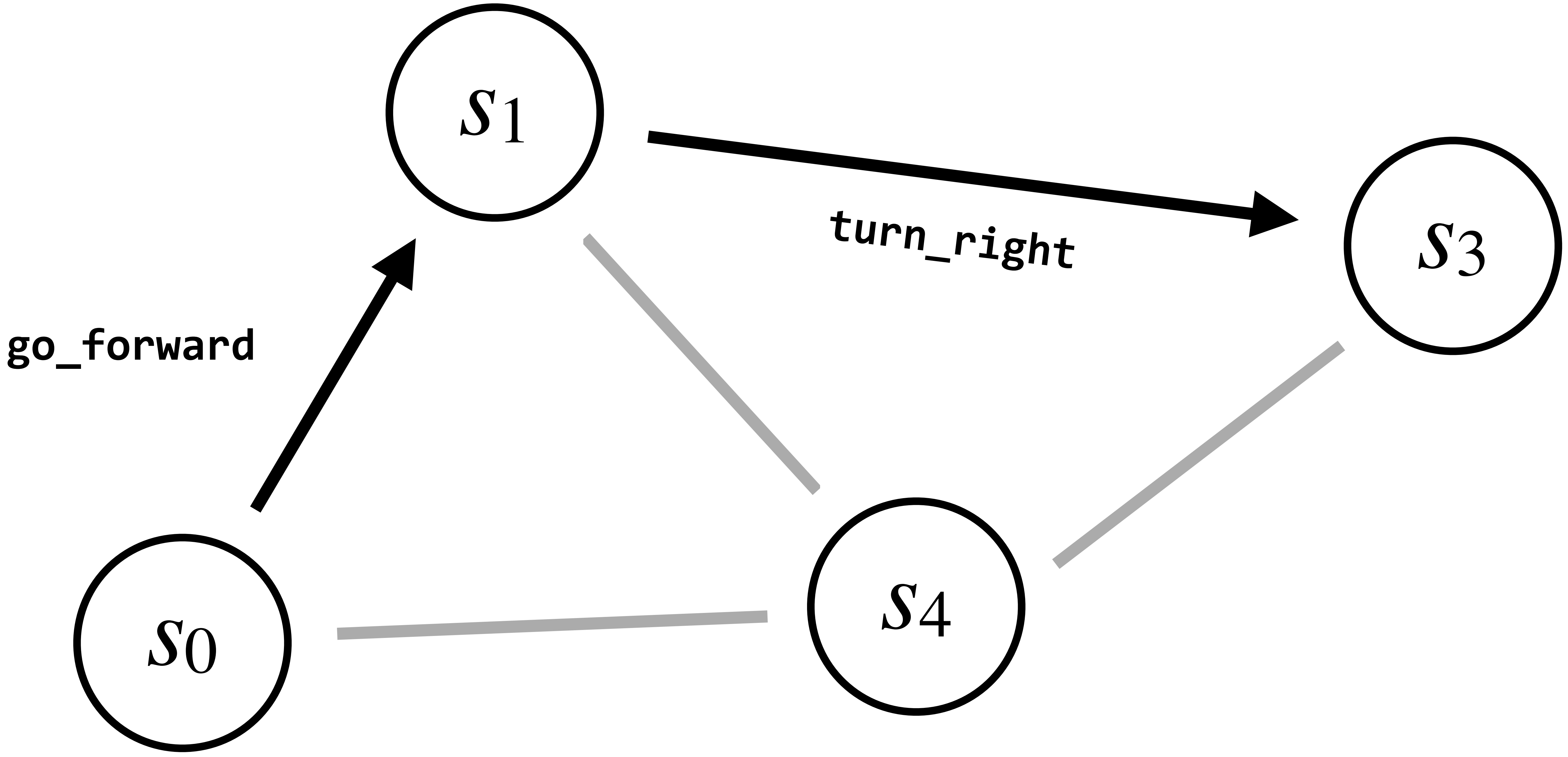
Find the sofa.

Go forward then face the sofa.



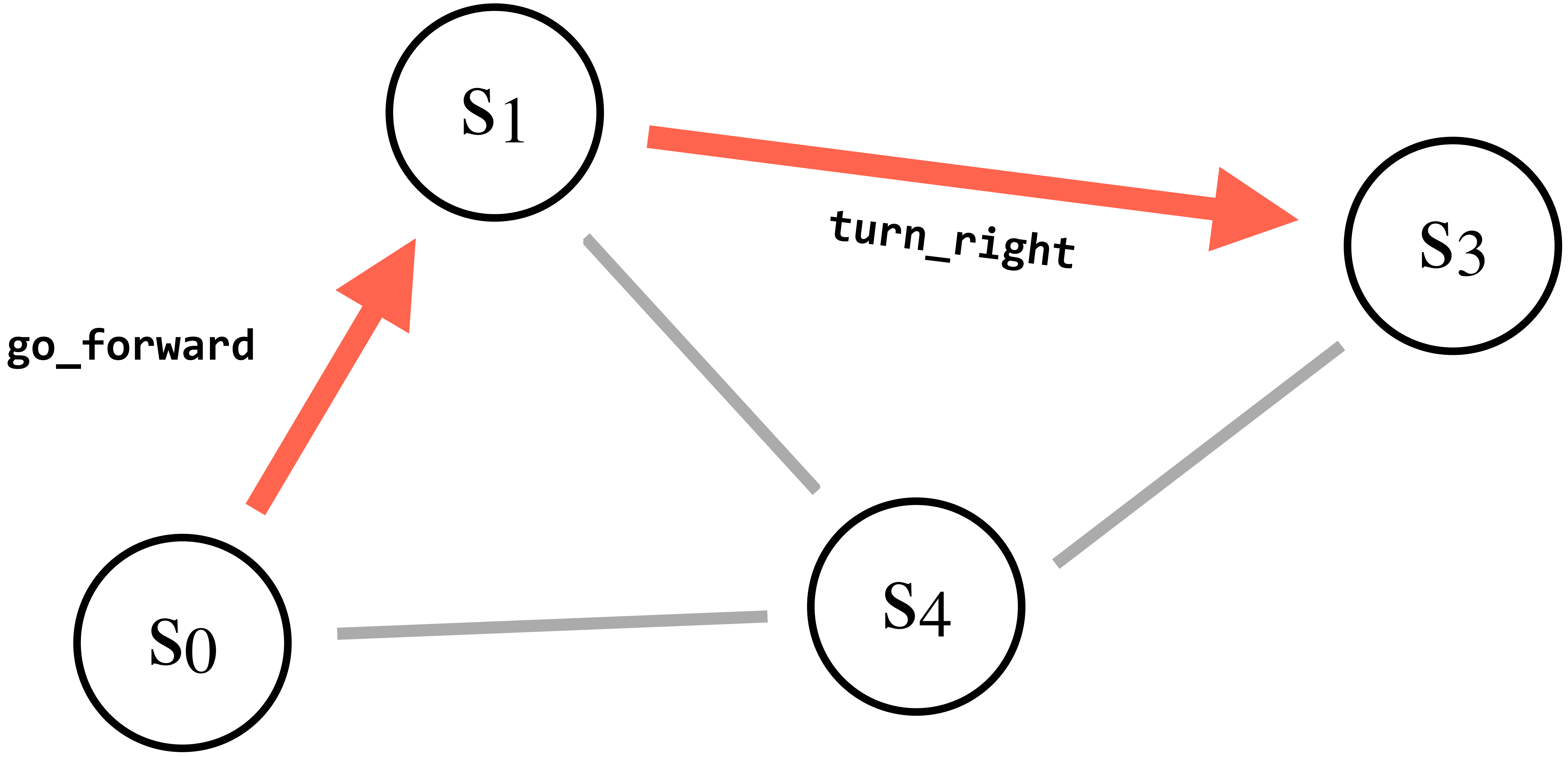
Supervision

Find the sofa.



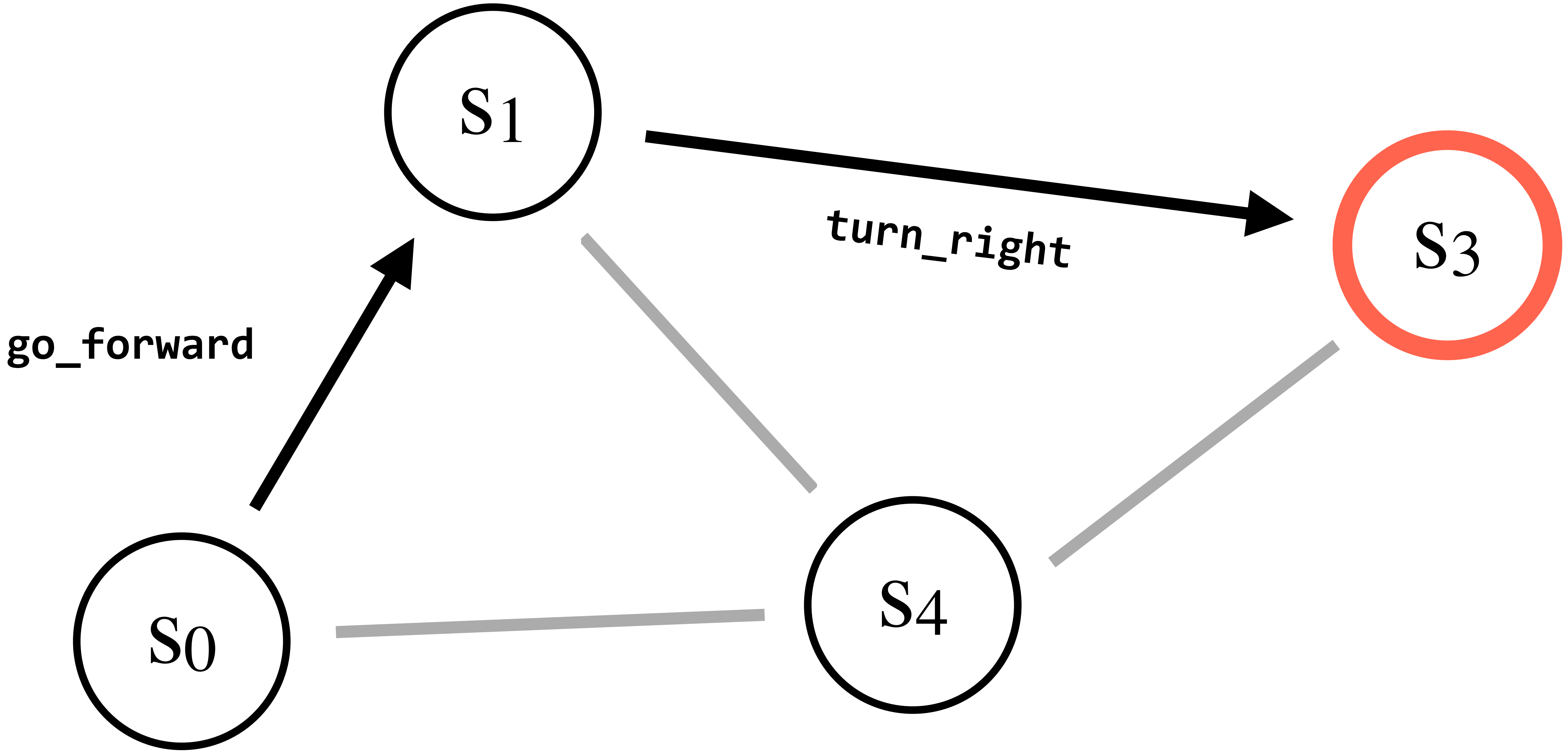
Supervision

Find the sofa.



Supervision

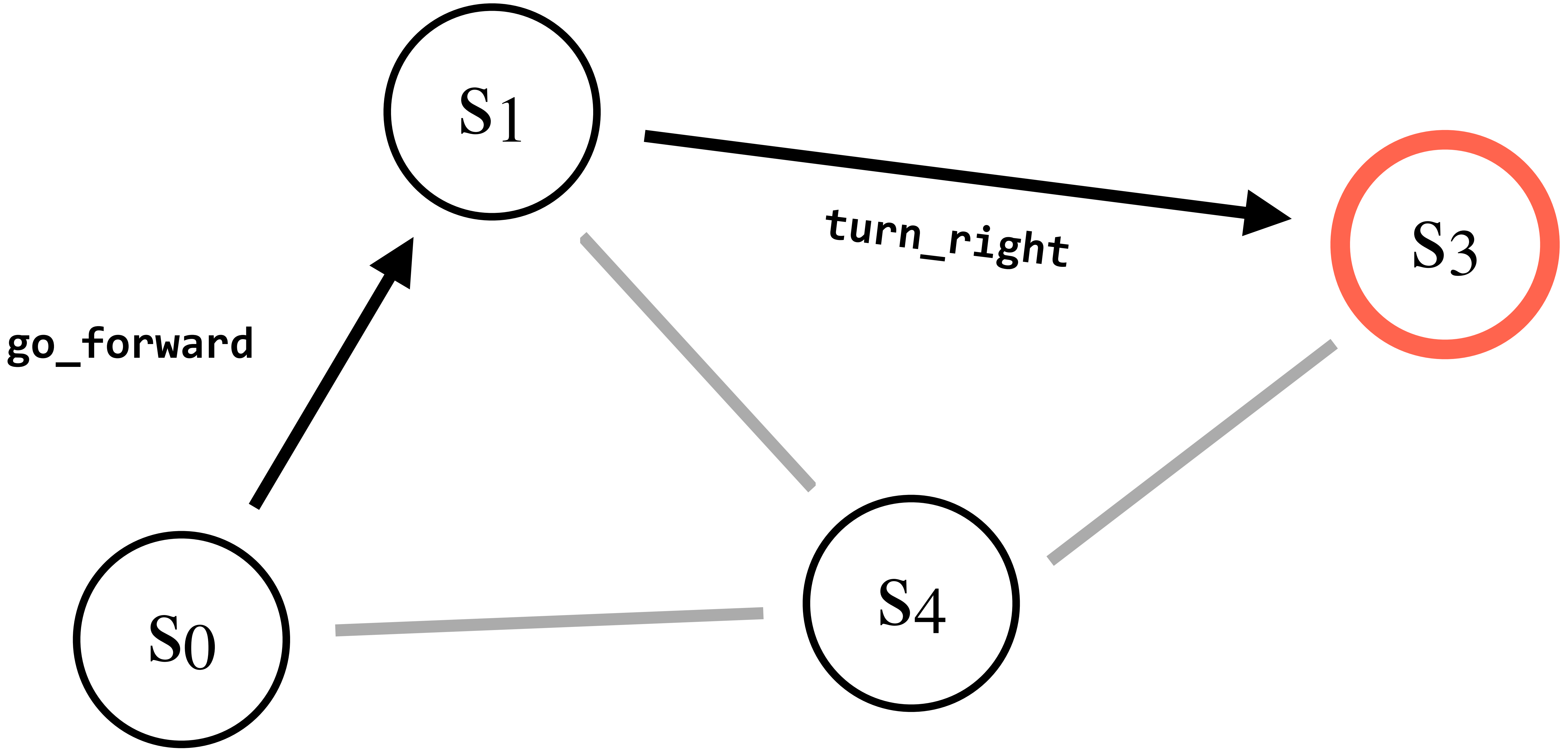
Find the sofa.



Supervision

Go forward, then turn to face right.

Find the sofa.



Instruction following: formally

Context

States S

Actions A

Transitions T: $S \times A \to S$

Data

Instruction X

Demo Y

Reward R

Goal: find a policy $S \times X \to A$

As machine translation

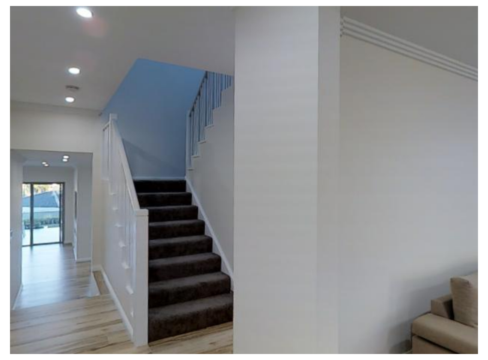
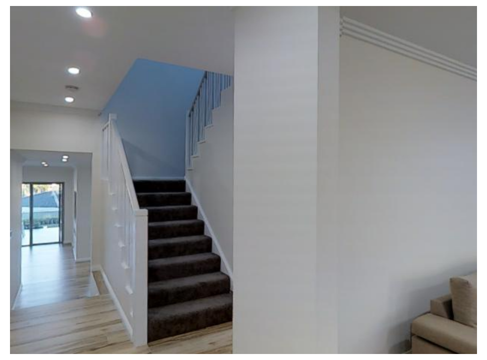
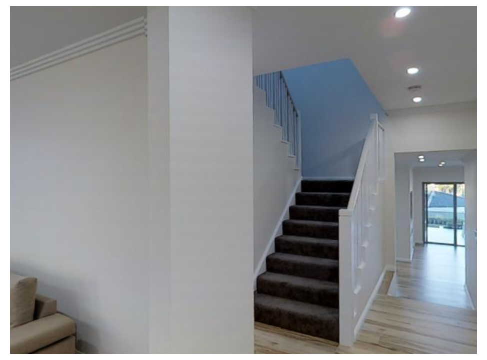
Move into the living room. Go forward then face the sofa.

```
go_forward turn_left turn_left go_forward turn_right
```

As machine translation

Move into the living room. Go forward then face the sofa.

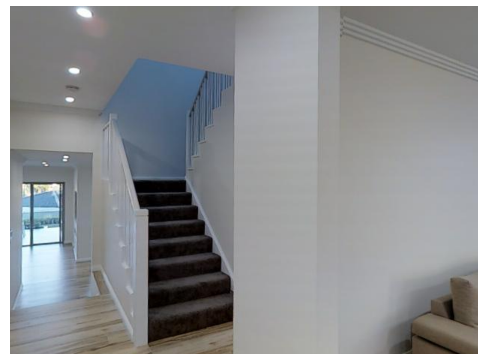
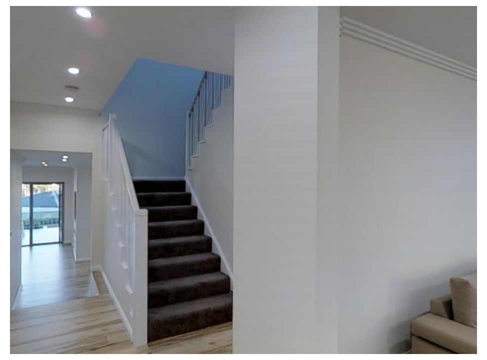
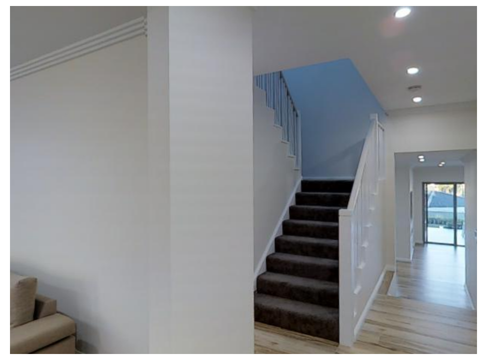
```
go_forward turn_left turn_left go_forward turn_right
```

As machine translation

Move into the living room. Go forward then face the sofa.

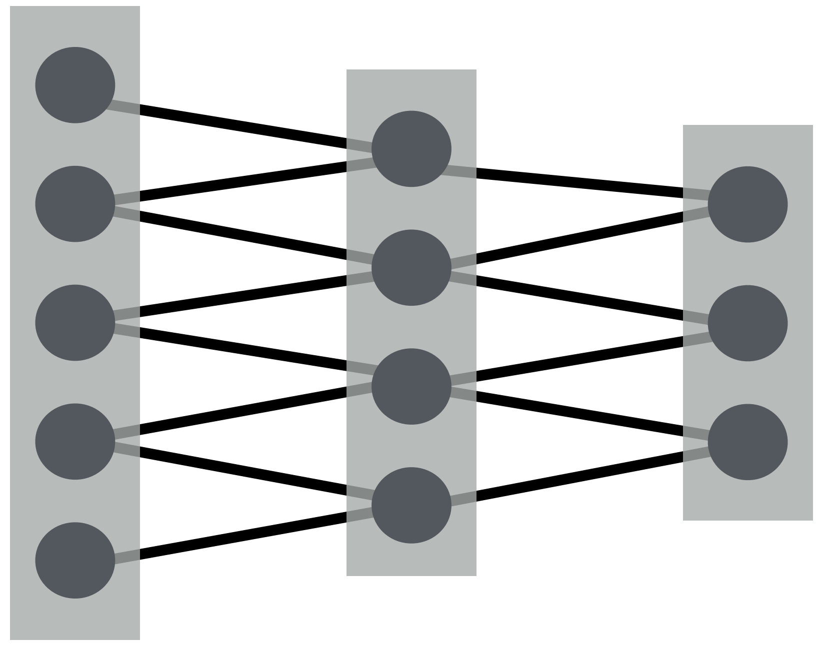
`go_forward turn_left turn_left go_forward turn_right`

Approach 1: predicting action sequences

From instructions to actions

Go through the door and end facing into the next room.



turn_right

turn_left

go_forward

stop

From instructions to actions

Go through the door and end facing into the next room.

From instructions to actions

Key idea: solve this like a normal MDP, with the instruction as part of the state observation.

From instructions to actions

Training

$$\max_{\theta} \ \mathrm{p}(action \mid text, state; \theta)$$

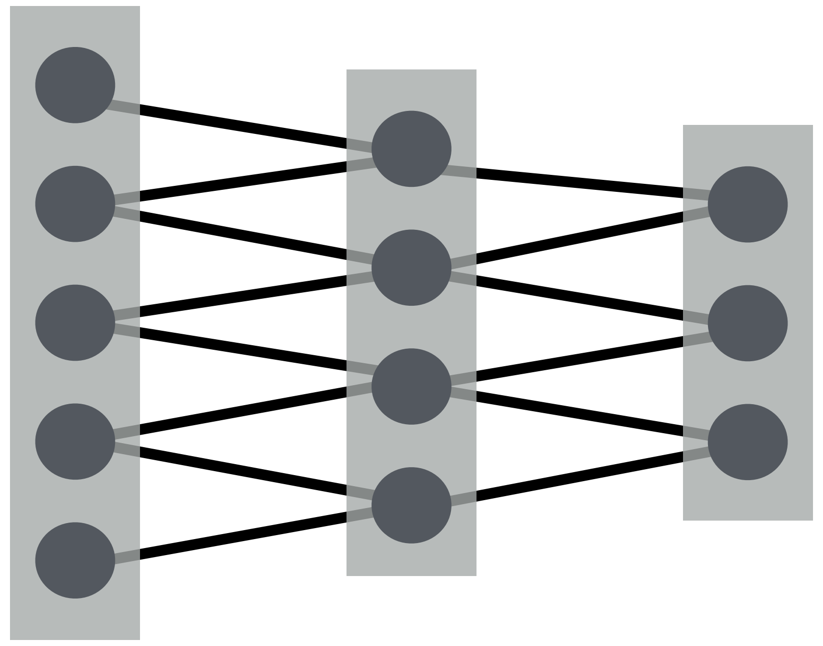
$$\max_{\theta} \ \mathbf{E}_{state \mid \theta} \mathrm{R}(action \mid state)$$

Evaluation

$$\max_{action} \ \mathrm{p}(action \mid text, state; \theta)$$

Are we there yet?

Go through the door and end facing into the next room.



turn_right

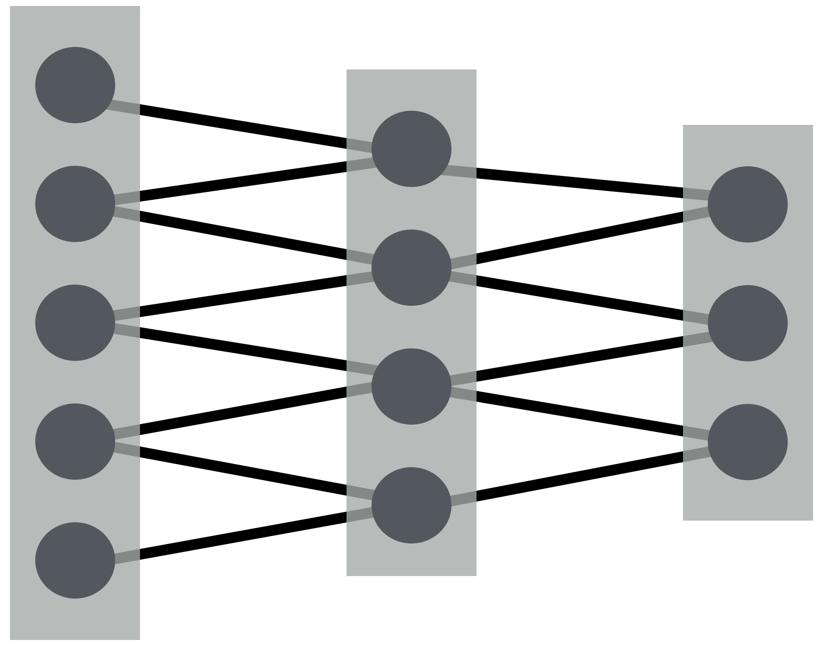
turn_left

go_forward

stop

Are we there yet?

Go through the door and end facing into the next room.



turn_right

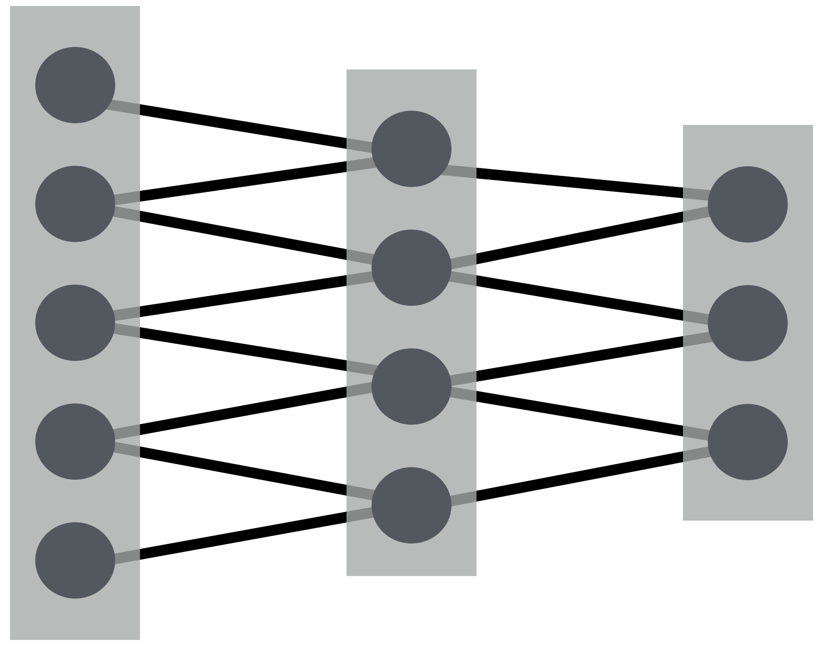
turn_left

go_forward

stop

Are we there yet?

Go through the door and end facing into the next room.



turn_right

turn_left

go_forward

stop

Are we there yet?

Key idea: make the state space track both "reading state" and physical state.

Augmented state spaces

Environment states S_e

Environment actions A_e

Reading states S_r

Reading actions A_r

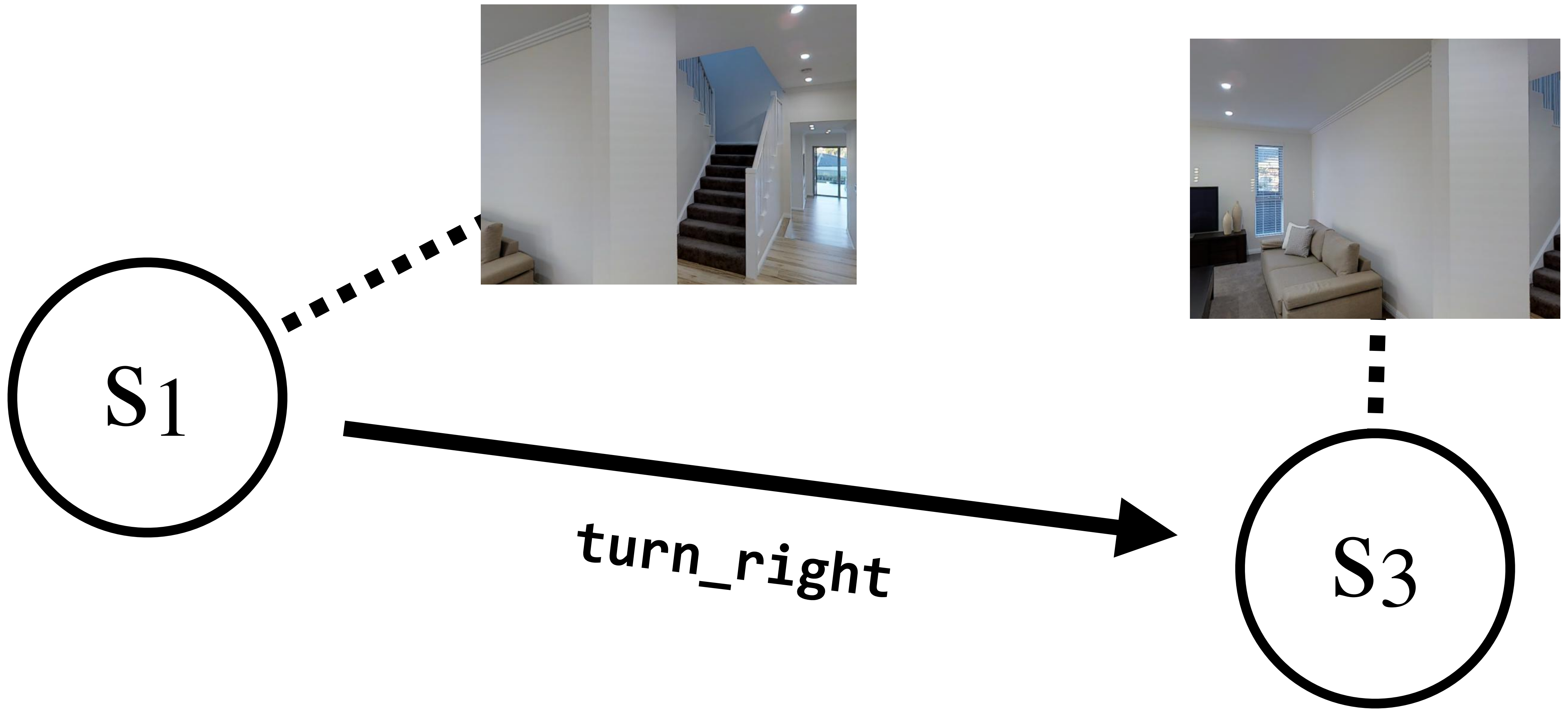
Augmented state spaces

Environment states S_e

Environment actions A_e

Reading states S_e

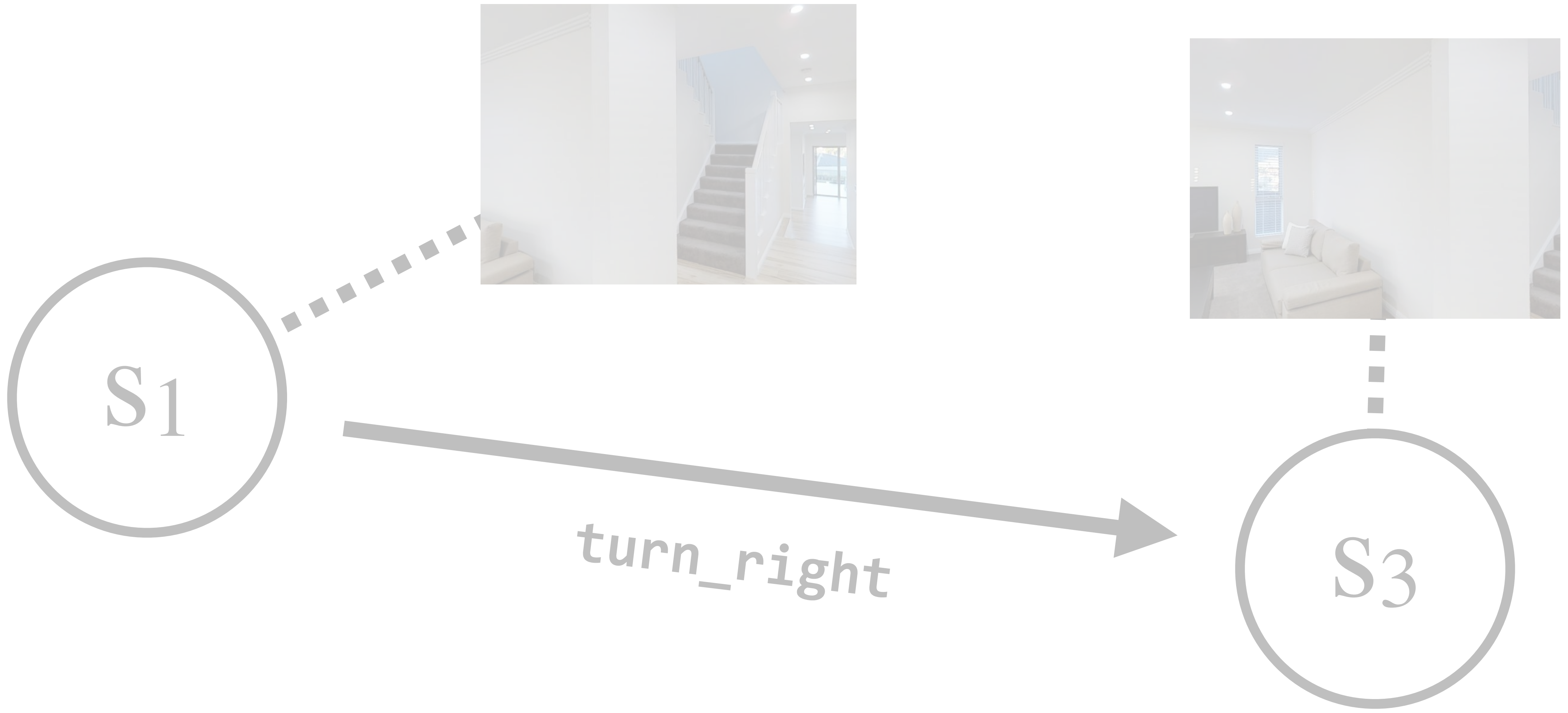
Reading actions A_e



Augmented state spaces

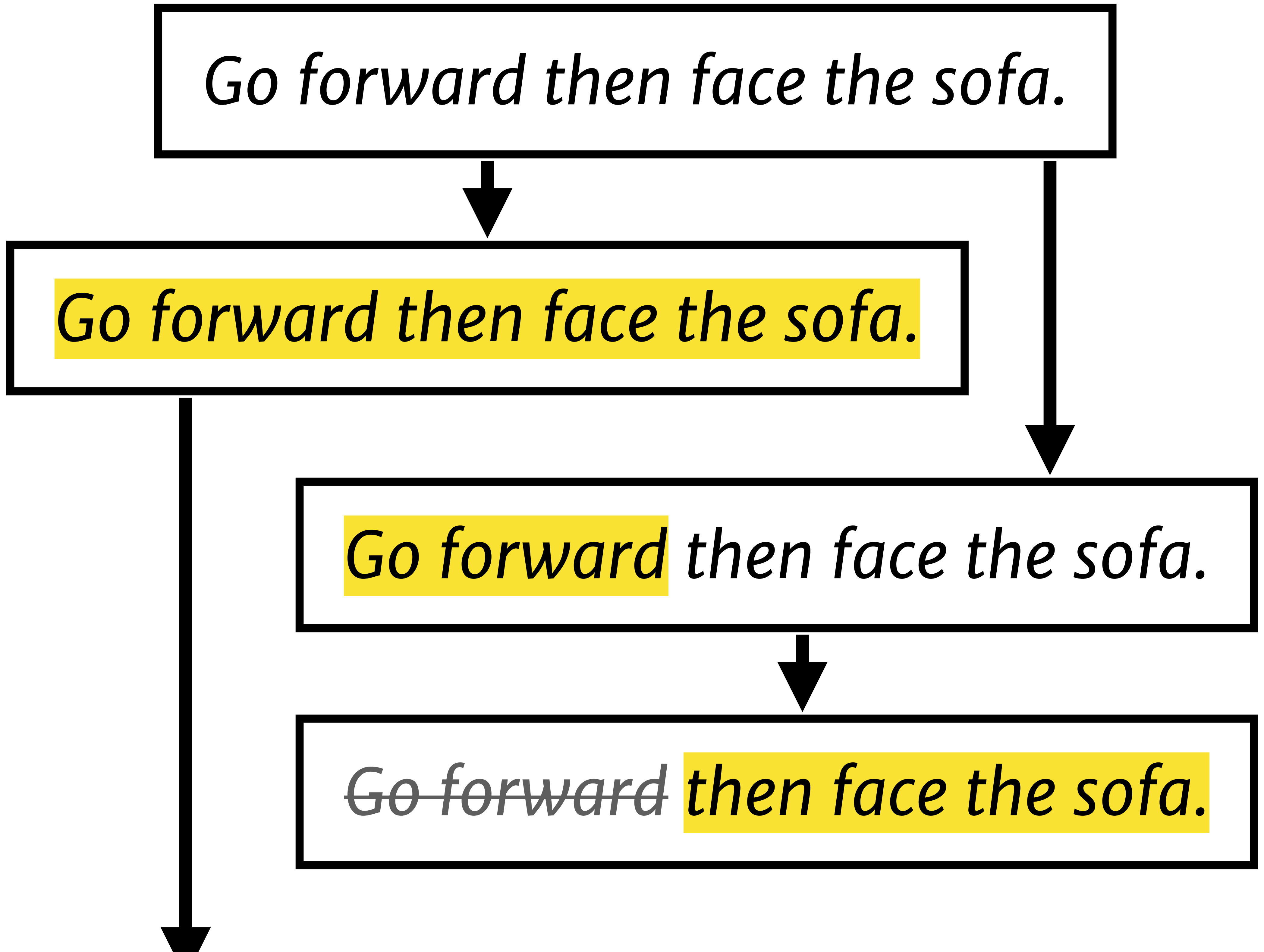
Environment states S_e

Environment actions A_e



Reading states S_e

Reading actions A_e



Augmented state spaces

$\boxed{\begin{array}{l}\text{States } S = S_e \times S_r \\ \text{Actions } A = A_e \cup A_r \\ \text{Transitions } T\colon S \times A \to S\end{array}}$

Goal: find a policy $S \times X \to A$

Augmented state spaces: training

Training

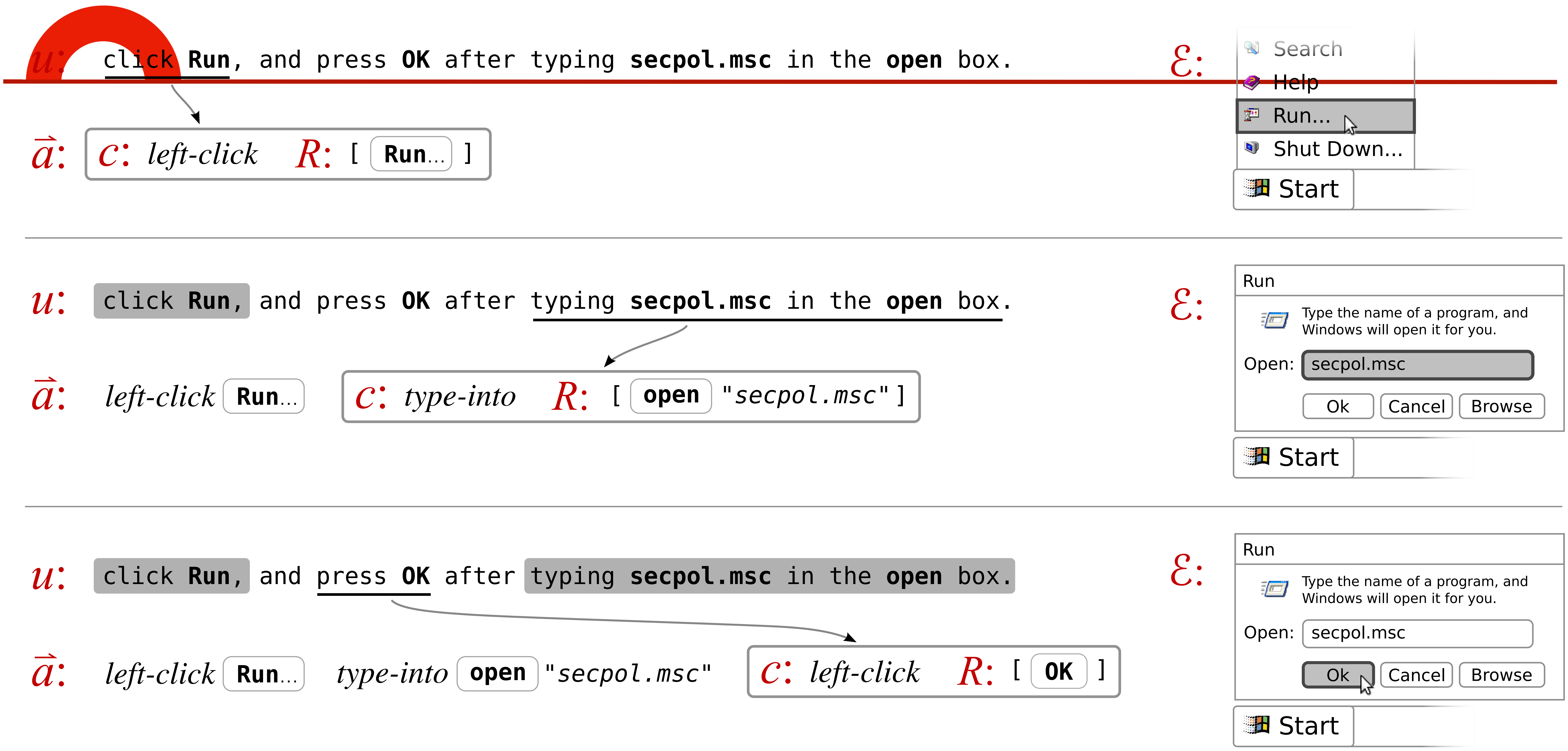
$$\cancel{\max_{action}\ \mathrm{p}(action \mid text, state; \theta)}$$

$$\max\ \mathbf{E}_{state \mid \theta}\, \mathrm{R}(action \mid state)$$

Evaluation

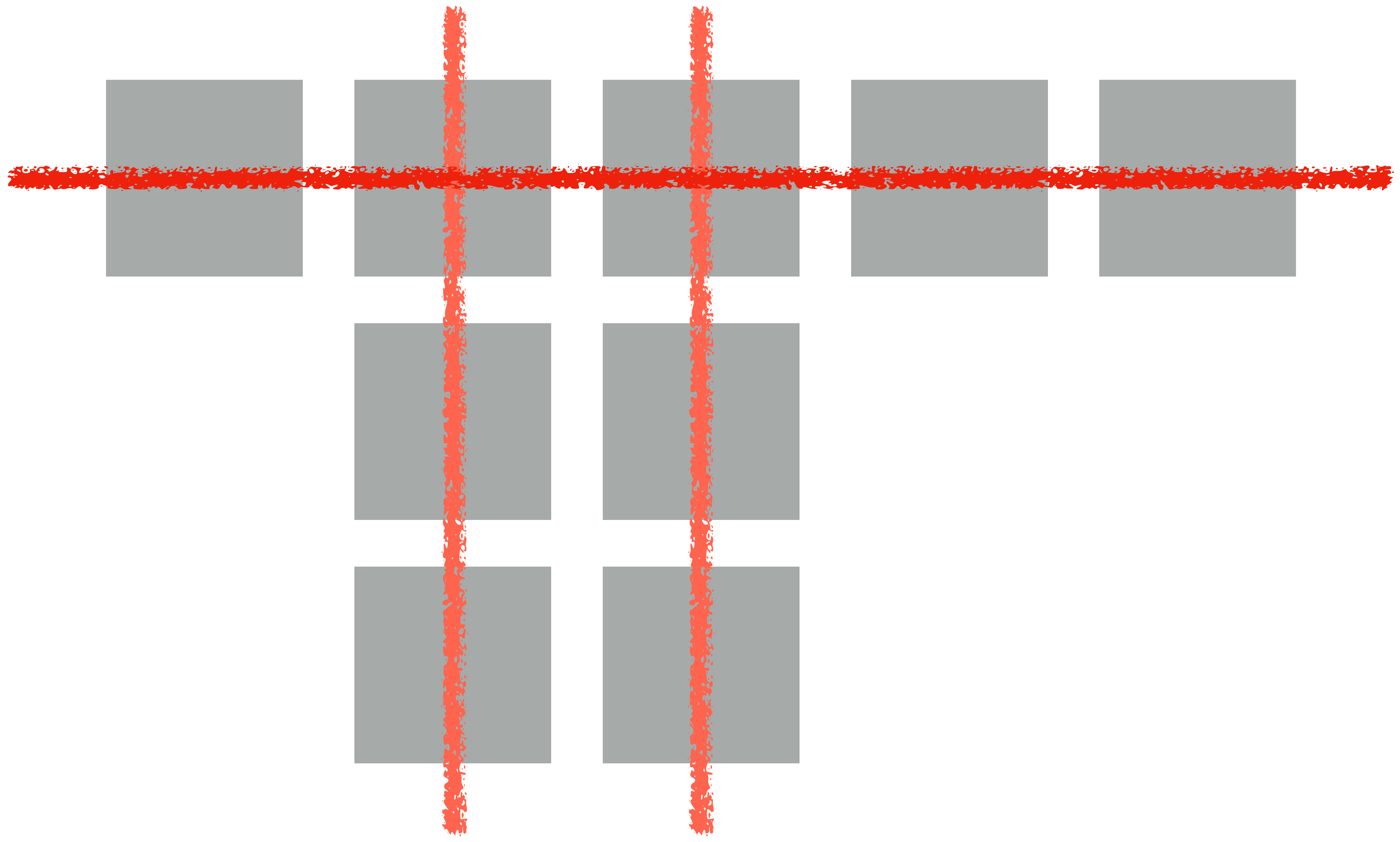
$$\max_{action}\ \mathrm{p}(action \mid text, state; \theta)$$

[Branavan et al., ACL ’09]



u: click Run, and press OK after typing secpol.msc in the open box.
ε:
Search
Help
Run...
Shut Down...
Start
a⃗: c: left-click R: [Run...]
u: click Run, and press OK after typing secpol.msc in the open box.
ε:
Run
Type the name of a program, and Windows will open it for you.
Open: secpol.msc
Ok Cancel Browse
Start
a⃗: left-click Run... c: type-into R: [open "secpol.msc"]
u: click Run, and press OK after typing secpol.msc in the open box.
ε:
Run
Type the name of a program, and Windows will open it for you.
Open: secpol.msc
Ok Cancel Browse
Start
a⃗: left-click Run... type-into open "secpol.msc" c: left-click R: [OK]

~~clear the two long columns,~~ and then the row



Augmented state spaces: better training

Training

$$\cancel{\max \; \mathrm{p}(\mathit{action} \mid \mathit{text}, \mathit{state}; \theta)}$$

$$\max \; \mathbf{E}_{\mathit{state} \mid \theta} \, \mathrm{R}(\mathit{action} \mid \mathit{state})$$

Evaluation

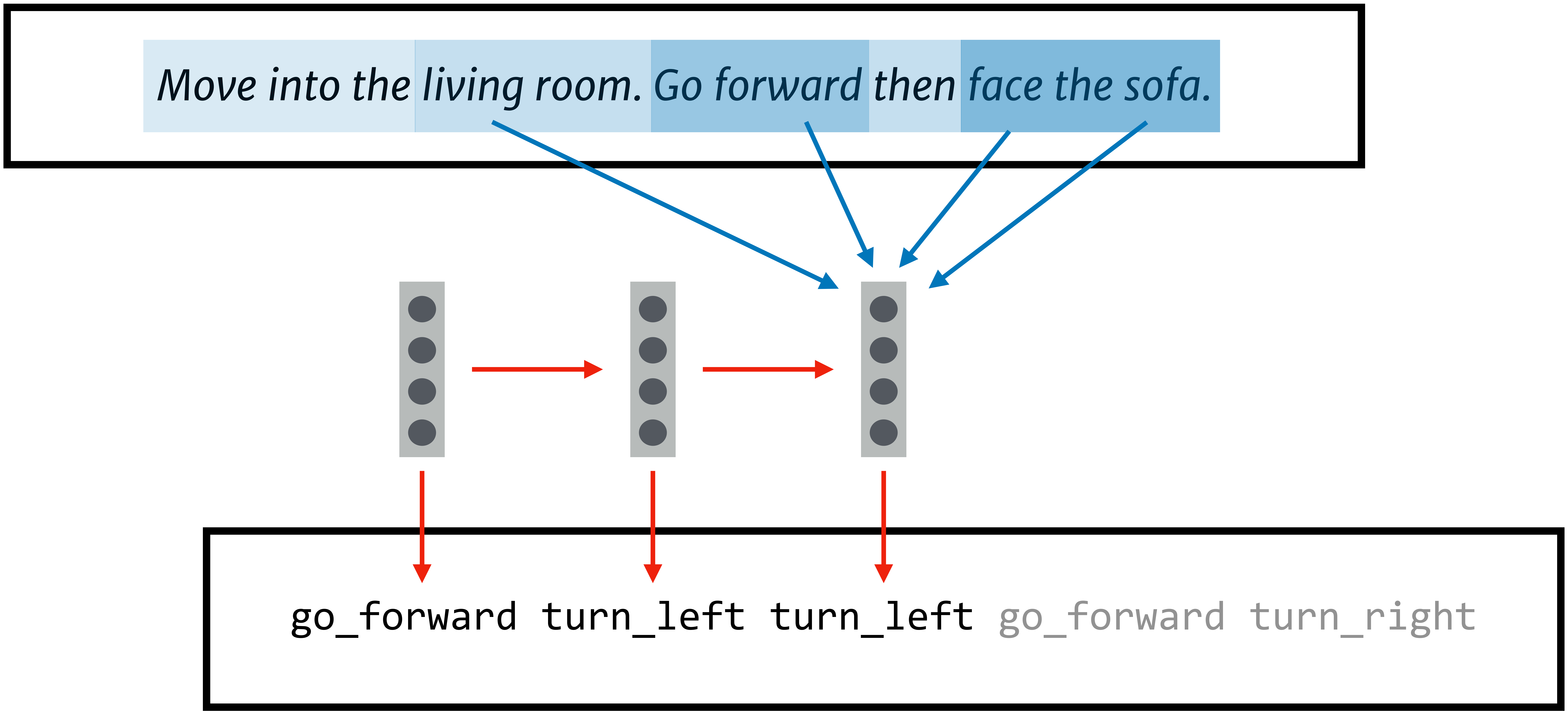
$$\max_{\mathit{action}} \; \mathrm{p}(\mathit{action} \mid \mathit{text}, \mathit{state}; \theta)$$

Learning the reading state

Move into the living room. Go forward then face the sofa.

`go_forward turn_left turn_left go_forward turn_right`

Learning the reading state



Learning the reading state

Key idea: move "reading state" into the hidden state of an RNN.

[Mei et al., AAAI '16]

Learning the reading state

Training

$$\max \ \mathrm{p}(action \mid text, state; \theta)$$

$$\max \ \mathbf{E}_{state \mid \theta} \, \mathrm{R}(action \mid state)$$

Evaluation

$$\max_{action} \ \mathrm{p}(action \mid text, state; \theta)$$

***human**: Walk past hall table. Walk into bedroom. Make left at table clock. Wait at bathroom door threshold.*

Approach 2: predicting constraints

Actions, goals, constraints

Find a table next to a chair.

```
go_forward go_forward turn_left go_forward turn_left
```

Actions, goals, constraints

[Find] [a table] [next to] [a chair].

```
go_forward go_forward turn_left go_forward turn_left
```

Actions, goals, constraints

[Find] [a table] [next to] [a chair].

Actions, goals, constraints

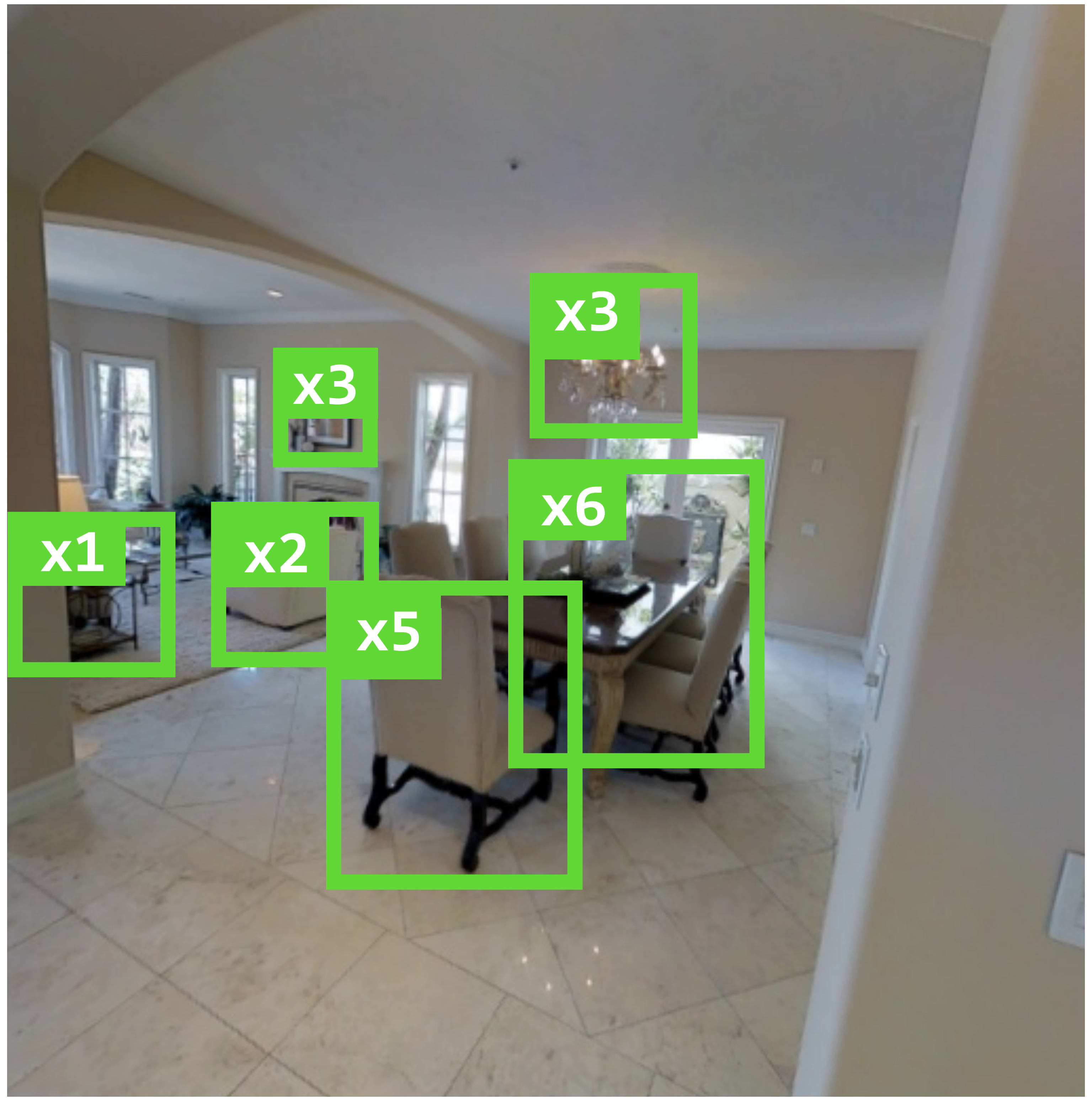
[Find] [a table] [next to] [a chair].

Actions, goals, constraints

Key idea: predict constraints rather than action sequences, and let a planner do the rest of the work.

Predicting constraints

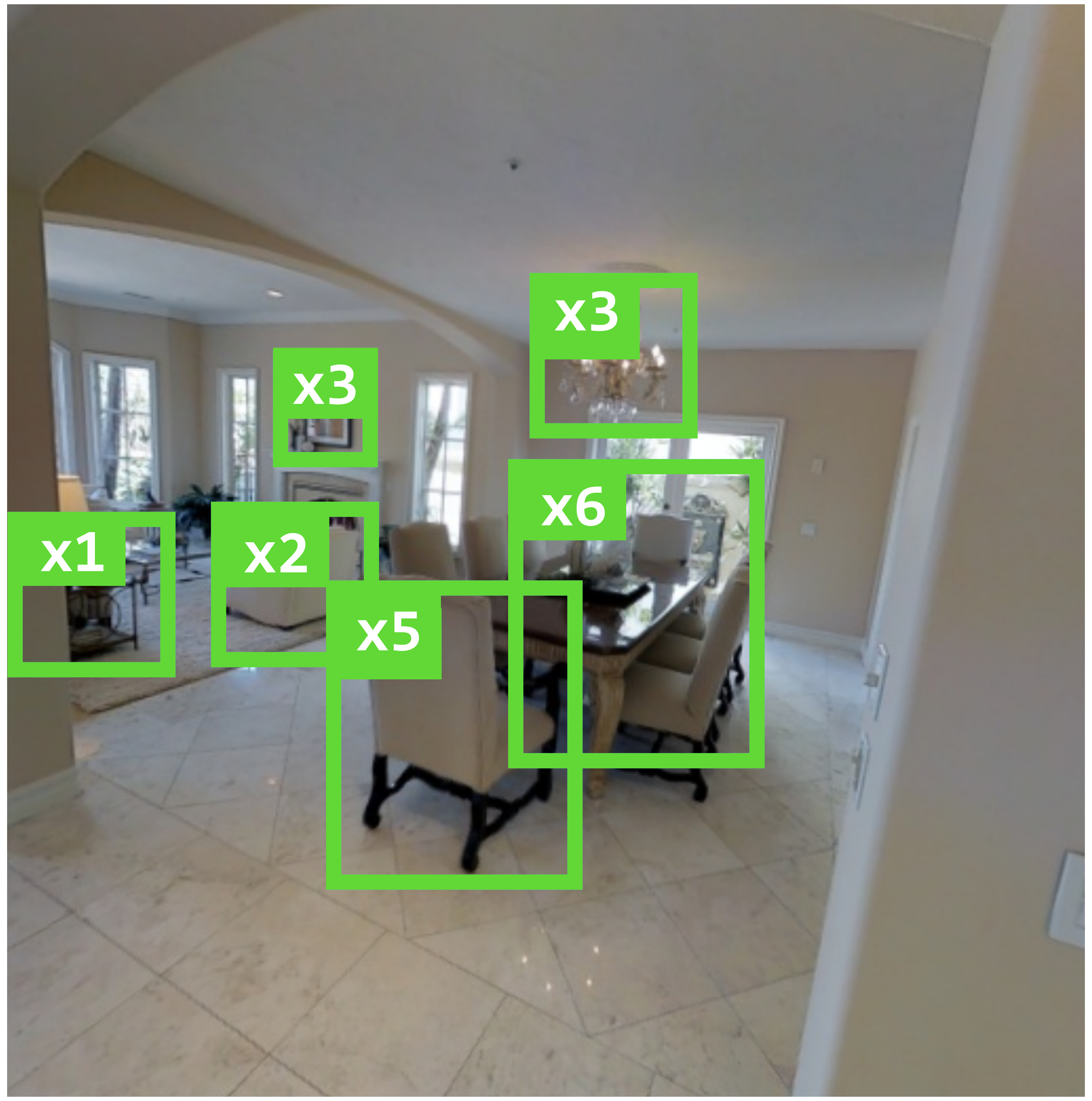
[Find] [a table] [next to] [a chair].



Predicting constraints

[Find] [a table] [next to] [a chair].

x1? x3? x4?

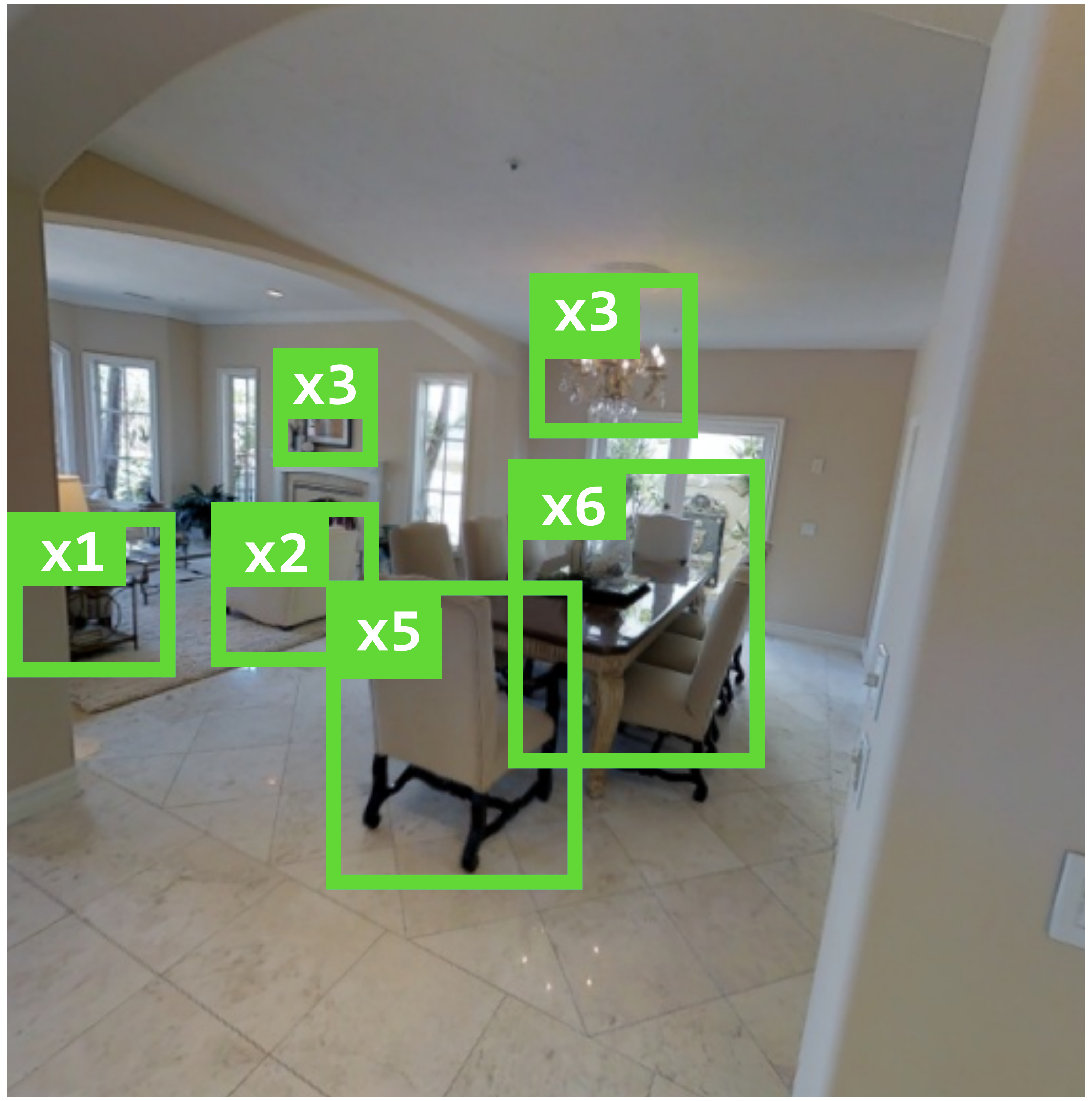


Predicting constraints

[Find] [a table] [next to] [a chair].

x6?

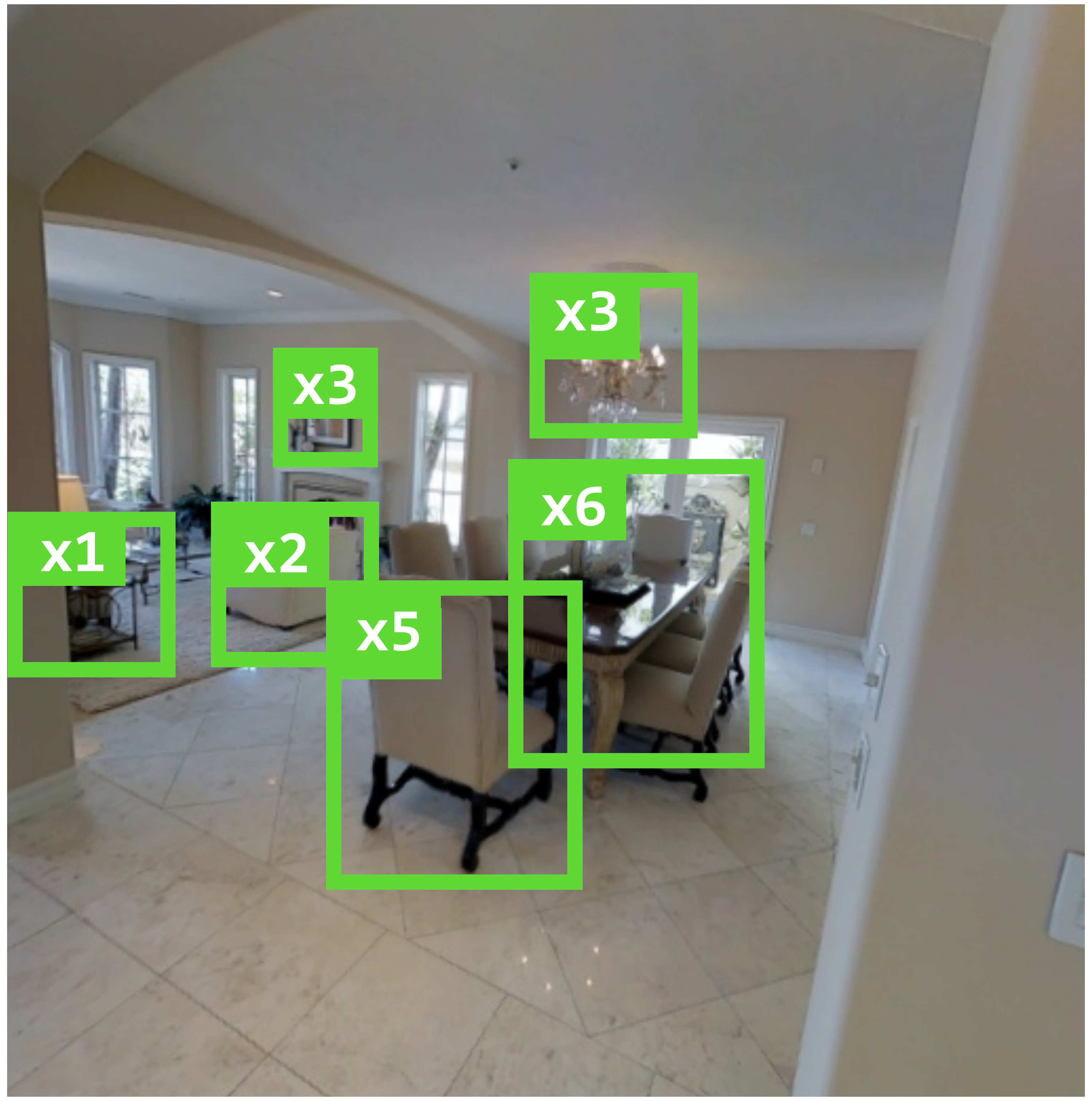
x5?



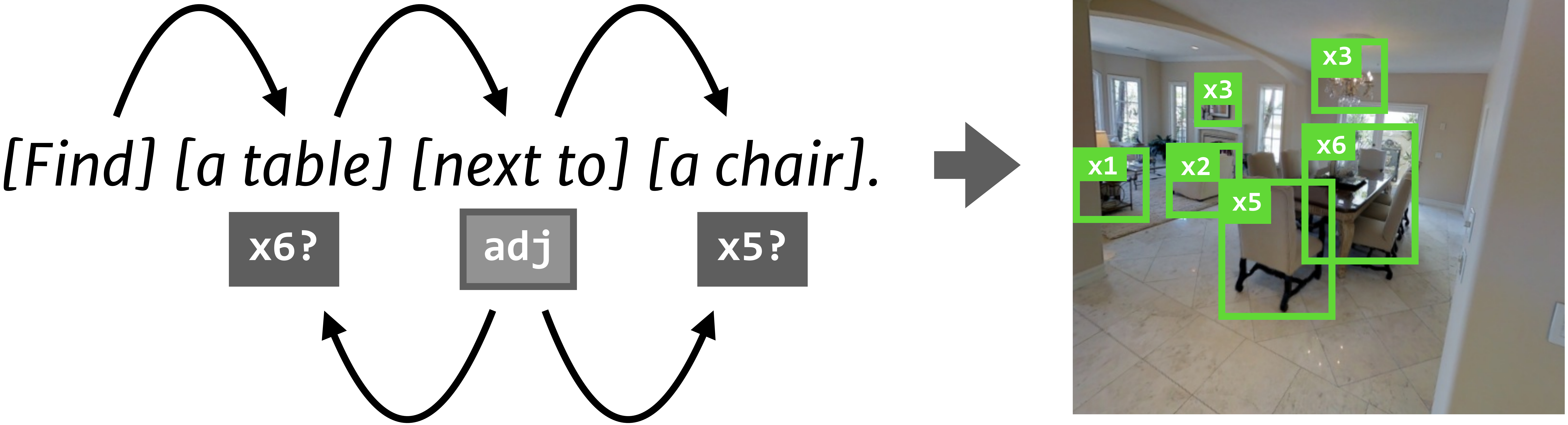
Predicting constraints

[Find] [a table] [next to] [a chair].

x6? adj x5?



Predicting constraints



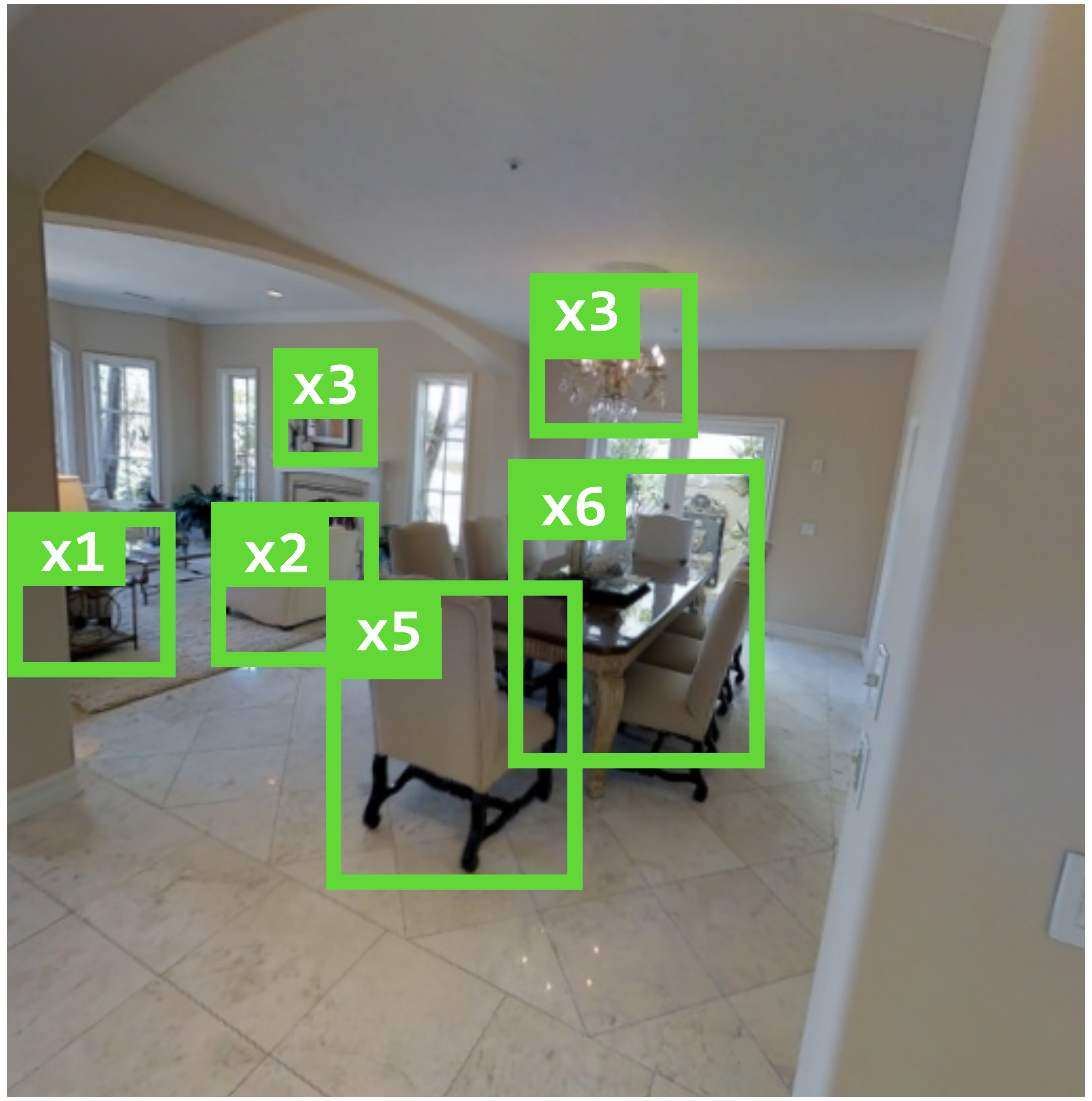
Predicting constraints

[Find] [a table] [next to] [a chair].

?

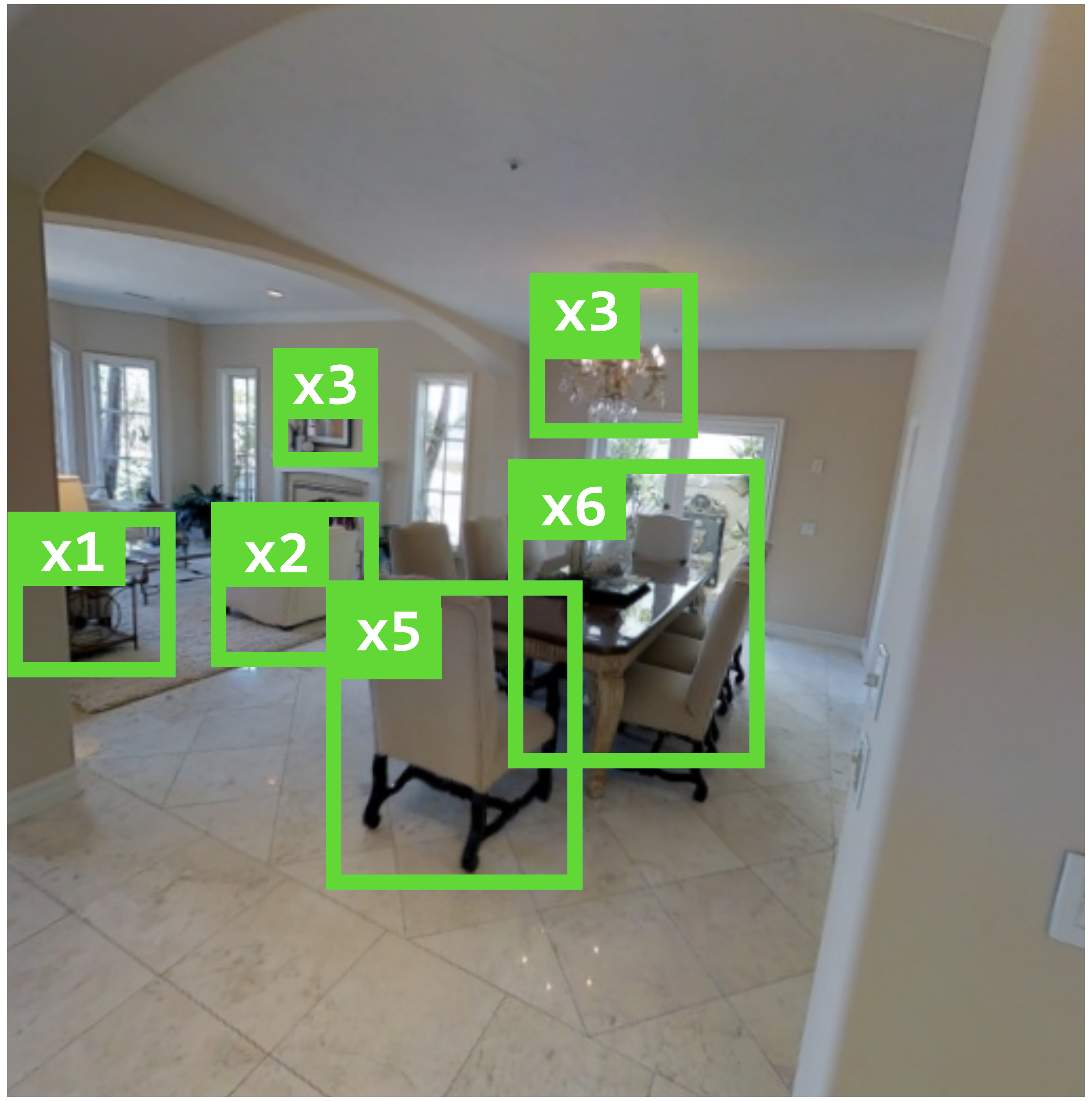
?

?



Predicting constraints

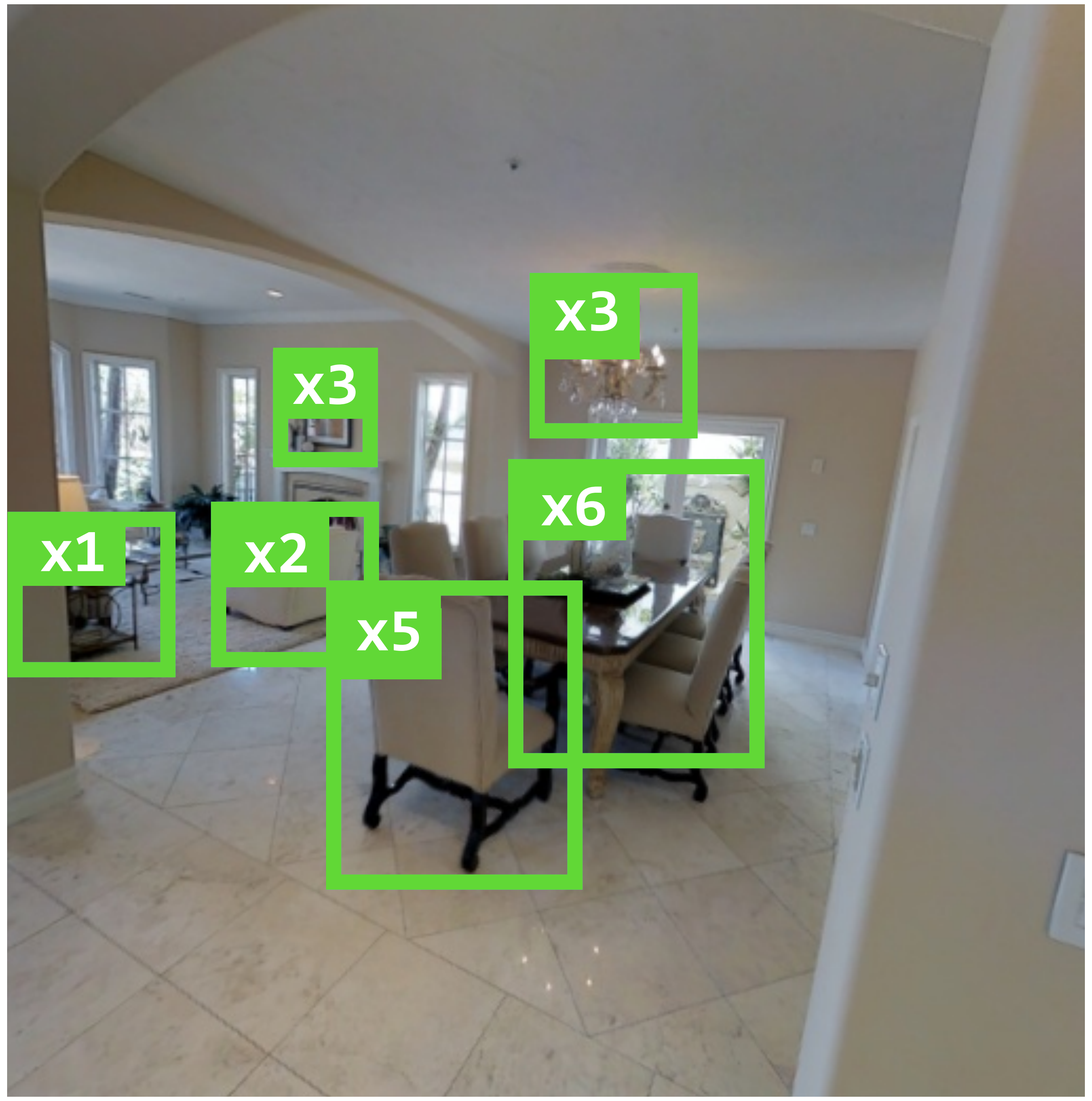
[Find] [a table] [next to] [a chair].



Predicting constraints

[Find] [a table] [next to] [a chair].

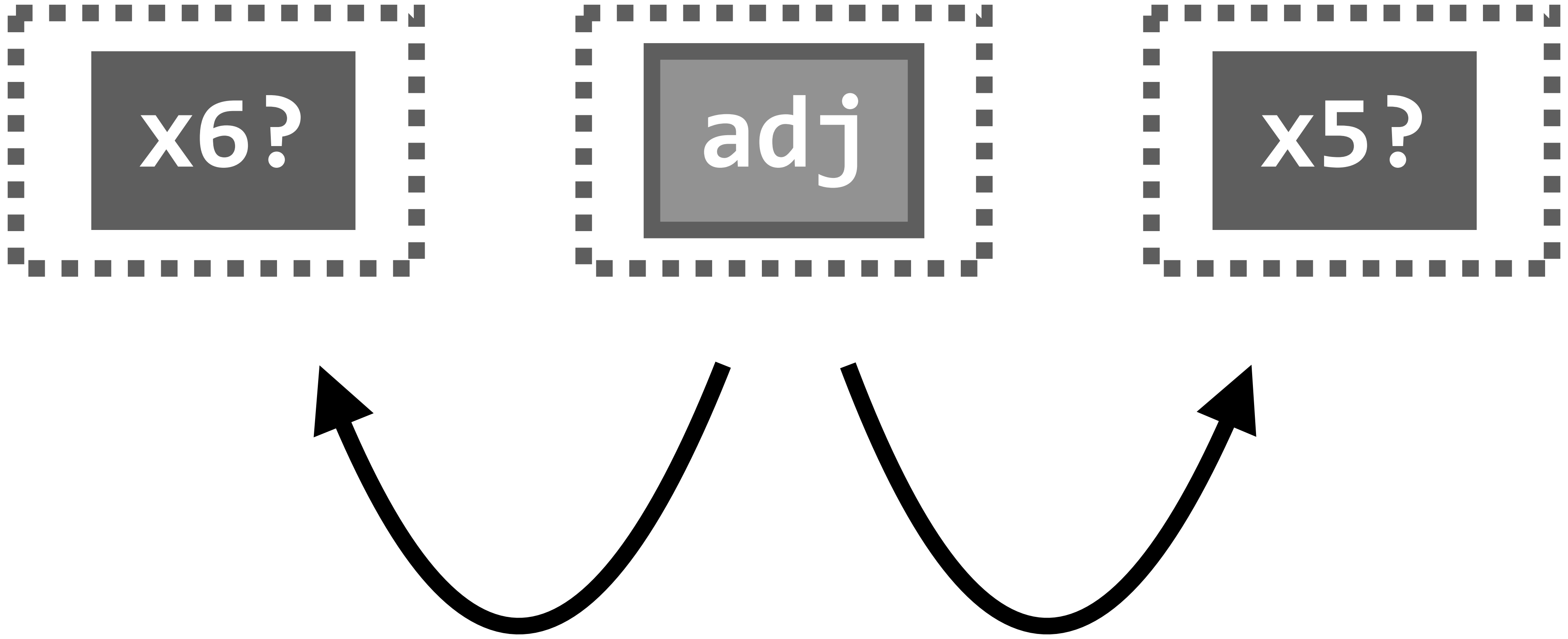
`obj?` `rel?` `obj?`

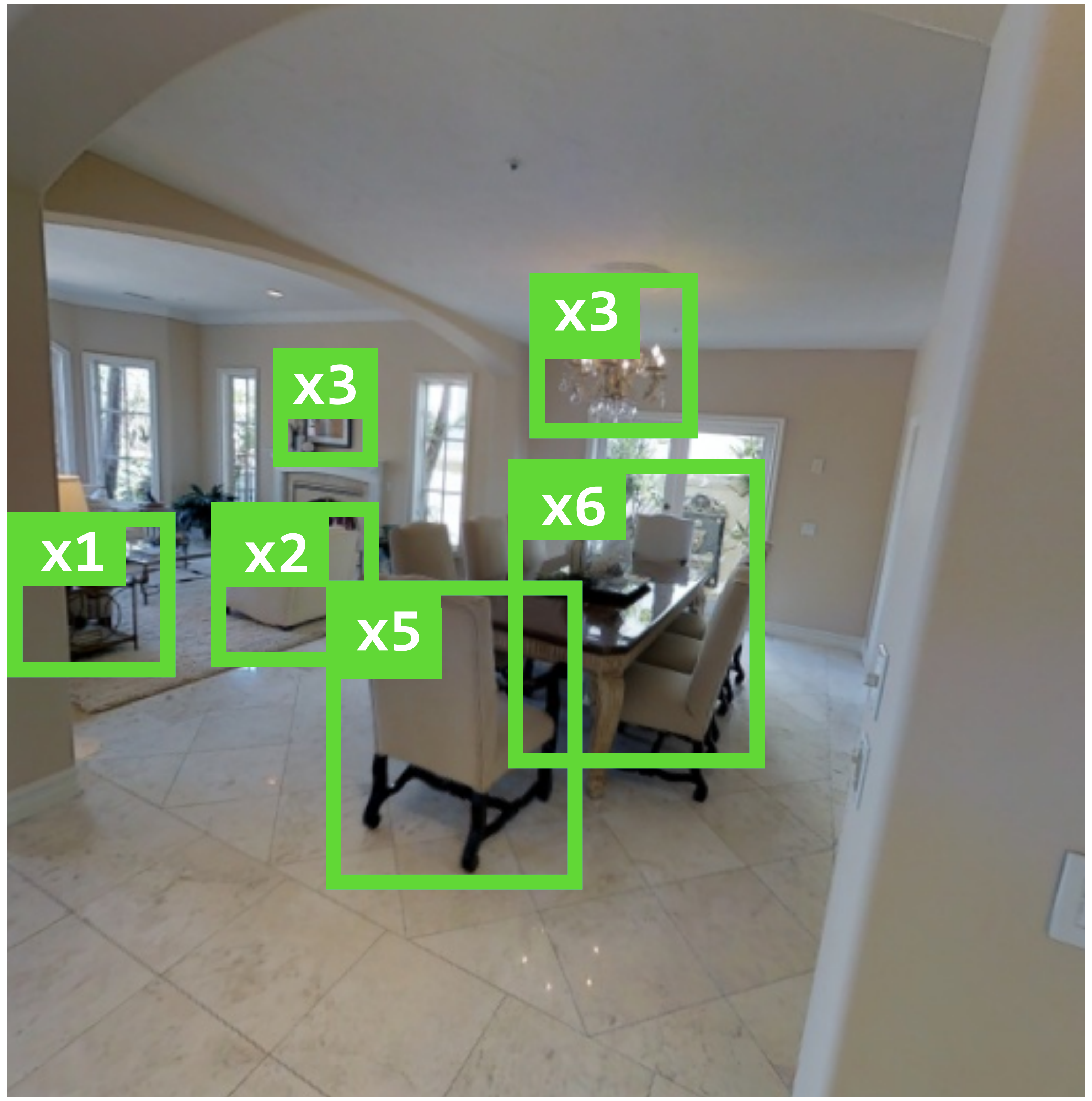


Learning a constraint parser

$$\max_{\theta} \ \mathrm{p}(\mathit{labels} \mid \mathit{text}, \mathit{graph};\, \theta)$$

[Find] [a table] [next to] [a chair].



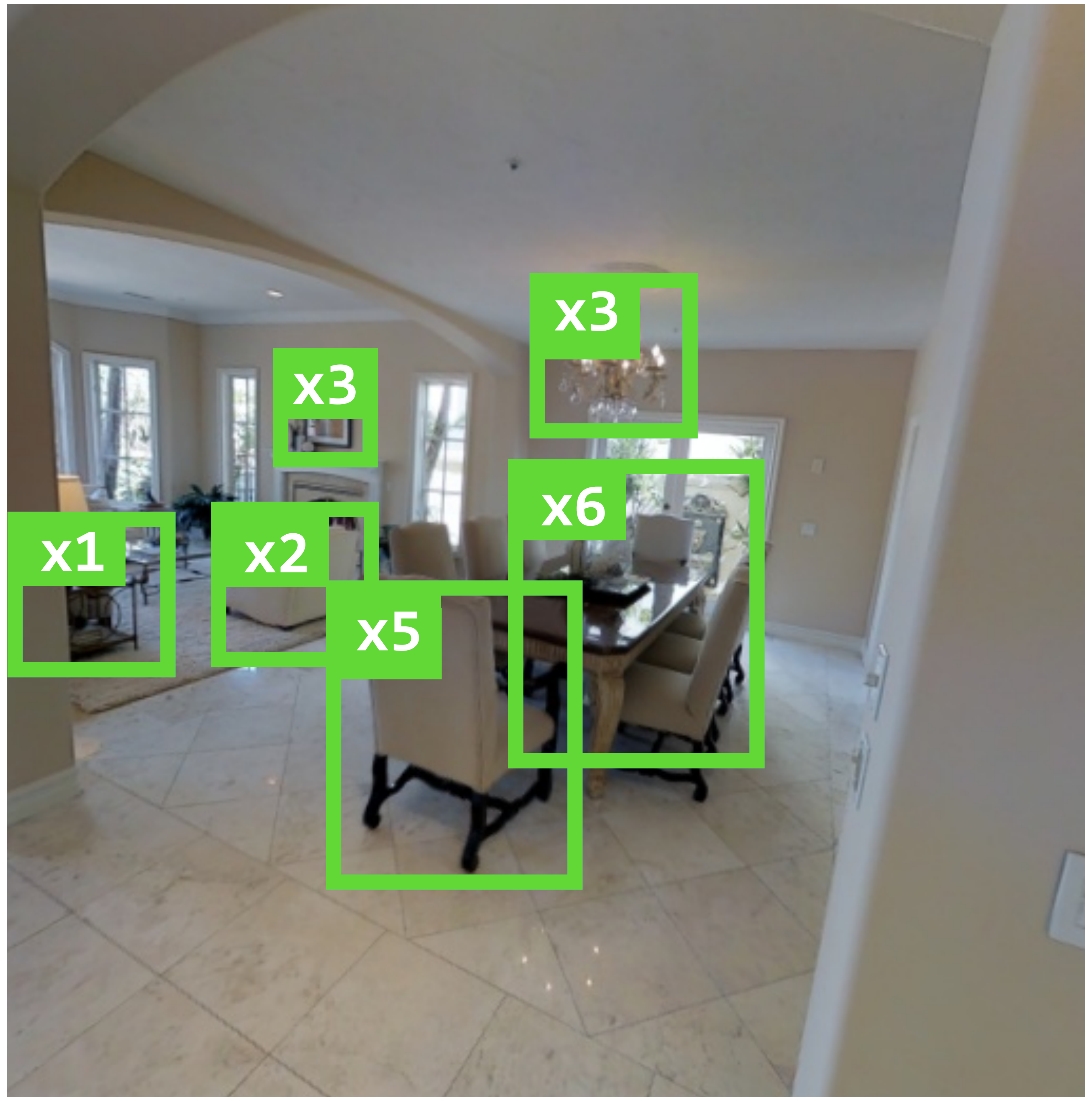


Inferring constraints

$$\max_{labels} \; p(labels \mid text, graph; \theta)$$

[Find] [a table] [next to] [a chair].

x6? adj x5?

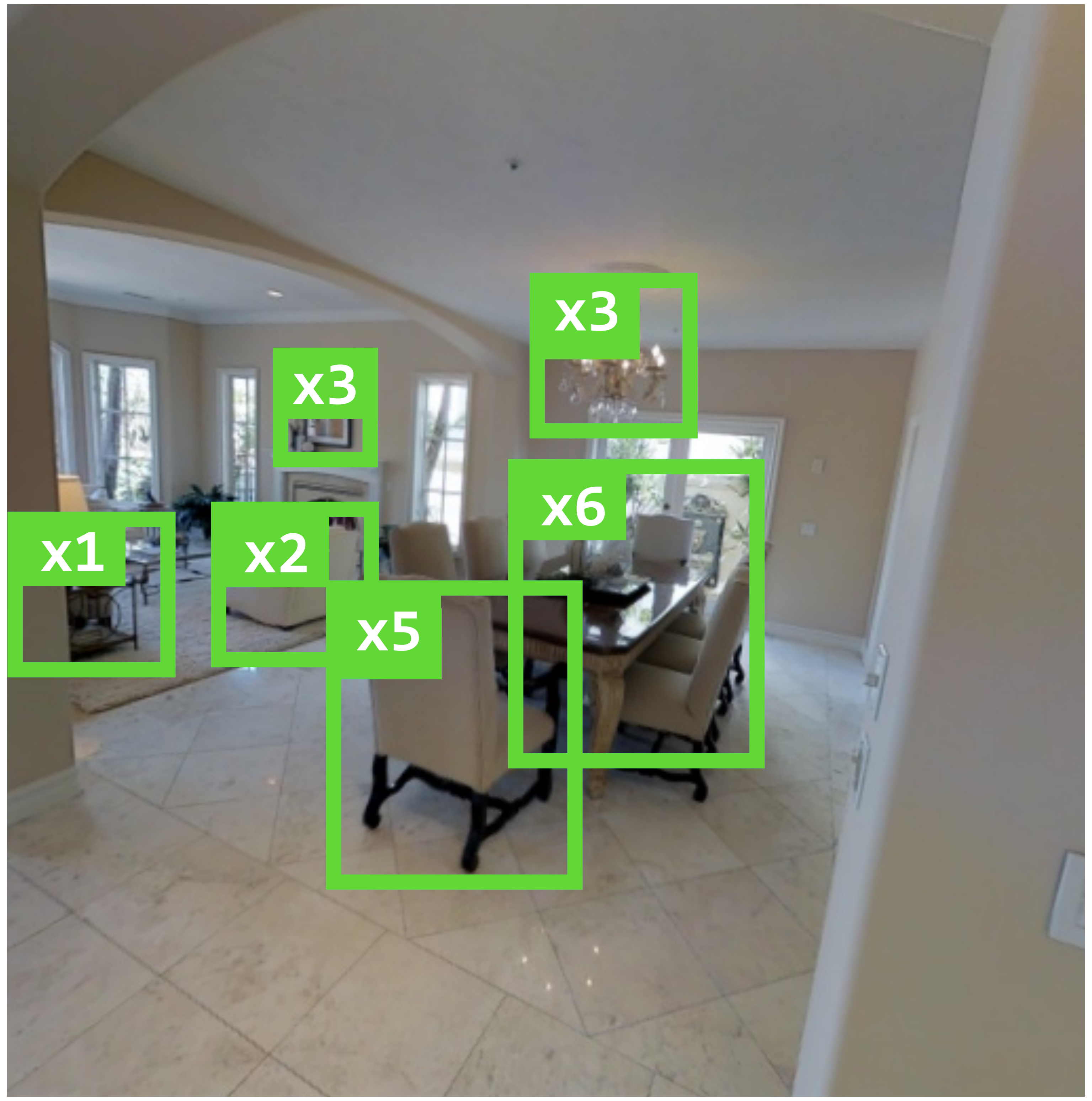


Inferring constraints

$$\max_{labels} \; p(labels \mid text, graph; \theta)$$

[Put] [the cup] [on] [the table].

x6? adj x5?



[Tellex et al., NCAI ’11]

Logical constraint languages

$$\max_{\theta} \; \mathrm{p}(\mathit{constraint} \mid \mathit{text};\, \theta) \qquad \max_{\mathit{constraint}} \; \mathrm{p}(\mathit{constraint} \mid \mathit{text};\, \theta)$$

Find a table next to a chair.

↓

```
at( x1 ) table( x1 ) next_to( x1 , x2 ) chair( x2 )
```

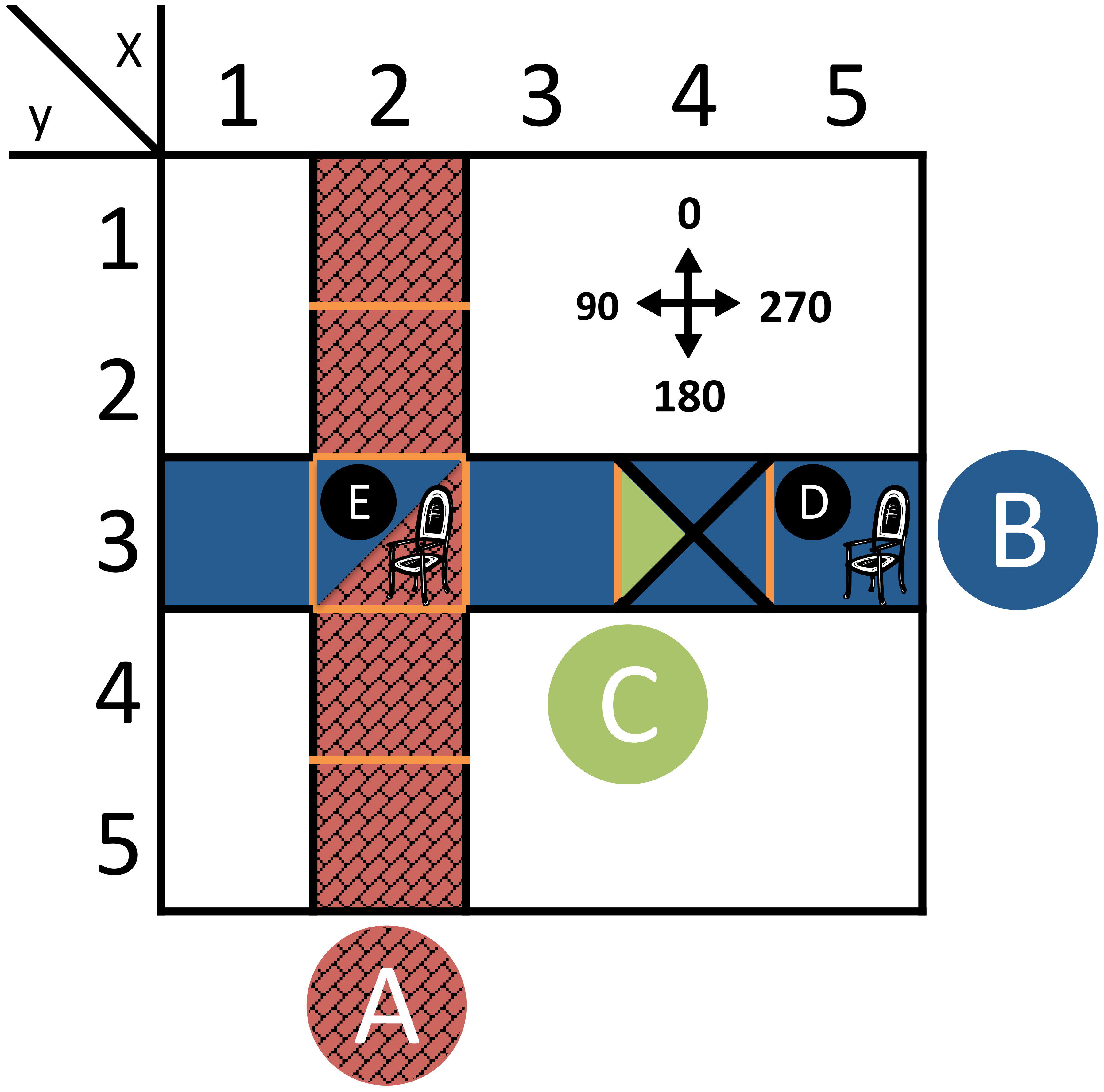
Logical constraint languages

$$\max_{\theta} \ p(\mathit{constraint} \mid \mathit{text};\ \theta) \qquad \max_{\mathit{constraint}} \ p(\mathit{constraint} \mid \mathit{text};\ \theta)$$

Find a table next to a chair.

```
at( x1 ) table( x1 ) next_to( x1 , x2 ) chair( x2 )
```

Logical constraint languages



{D E}	(a) chair $\lambda x.chair(x)$
{A B}	(b) hall $\lambda x.hall(x)$
E	(c) the chair $\iota x.chair(x)$
C	(d) you you
{B}	(e) blue hall $\lambda x.hall(x) \wedge blue(x)$
{E}	(f) chair in the intersection $\lambda x.chair(x) \wedge$ $intersect(\iota y.junction(y), x)$
{A B E}	(g) in front of you $\lambda x.in\_front\_of(you, x)$

[Artzi et al., TACL '13]

Constraints without logic

Find a table next to a chair.

`go_forward turn_left turn_left go_forward turn_right`

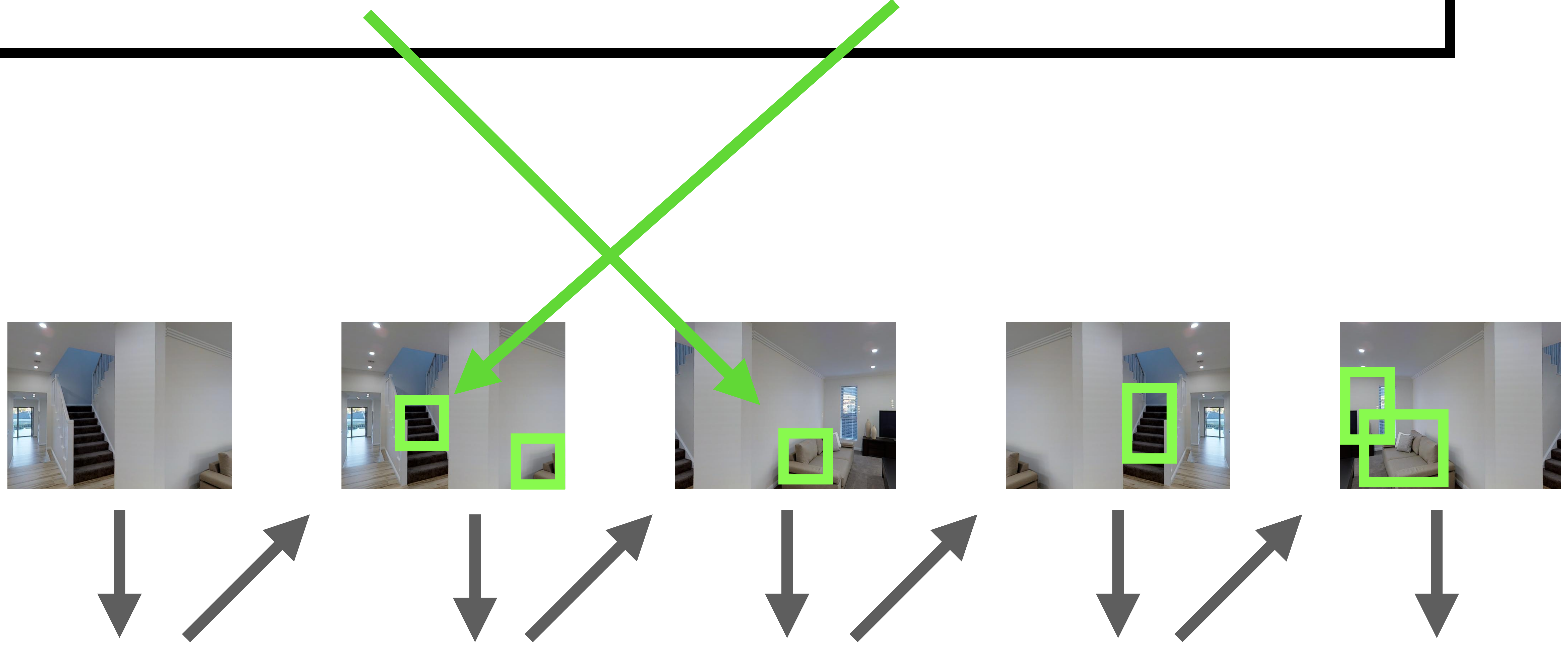
Constraints without logic

Key idea: use freeform learned potential functions rather than symbolic constraints

[Andreas & Klein, EMNLP ’16]

Constraints without logic

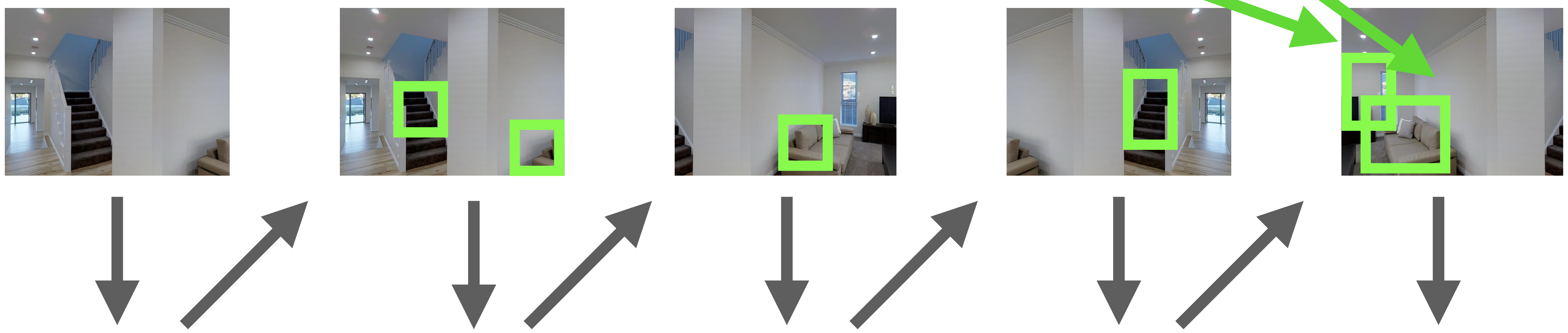
Find a table next to a chair.



`go_forward turn_left turn_left go_forward turn_right`

Constraints without logic

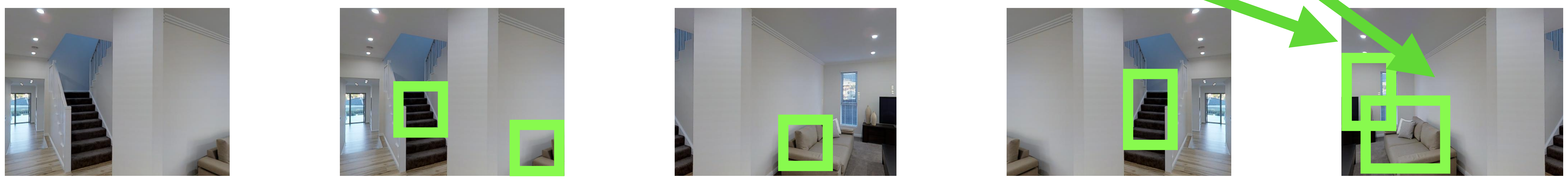
Find a table next to a chair.



`go_forward turn_left turn_left go_forward turn_right`

Constraints without logic

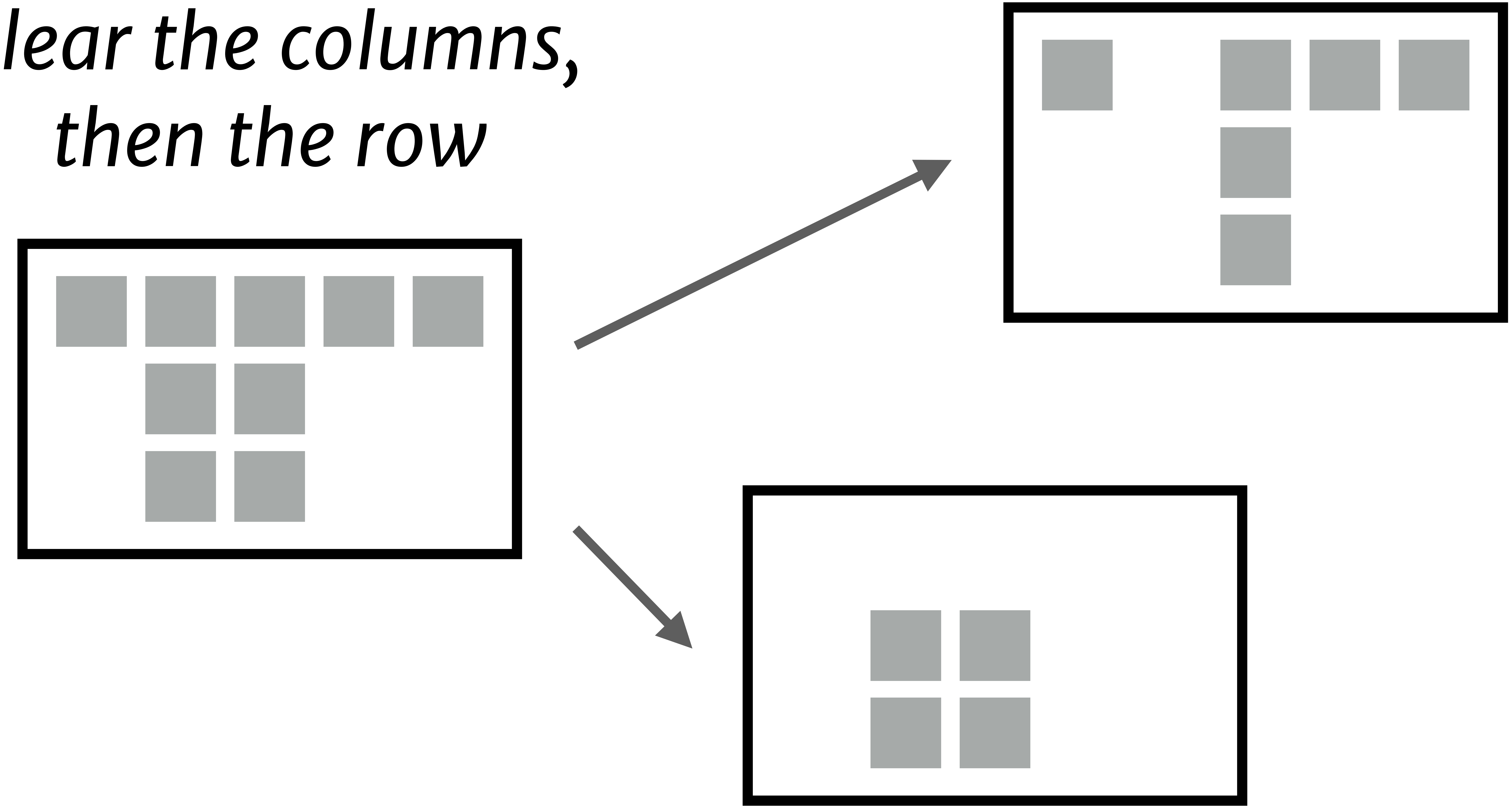
Find a table next to a chair.



$$\max_{\theta,\ alignment} \frac{f(plan, alignment \mid text; \theta)}{\sum f(plan', alignment' \mid text; \theta)} \qquad \max_{\substack{plan,\\ alignment}} f(plan, alignment \mid text; \theta)$$

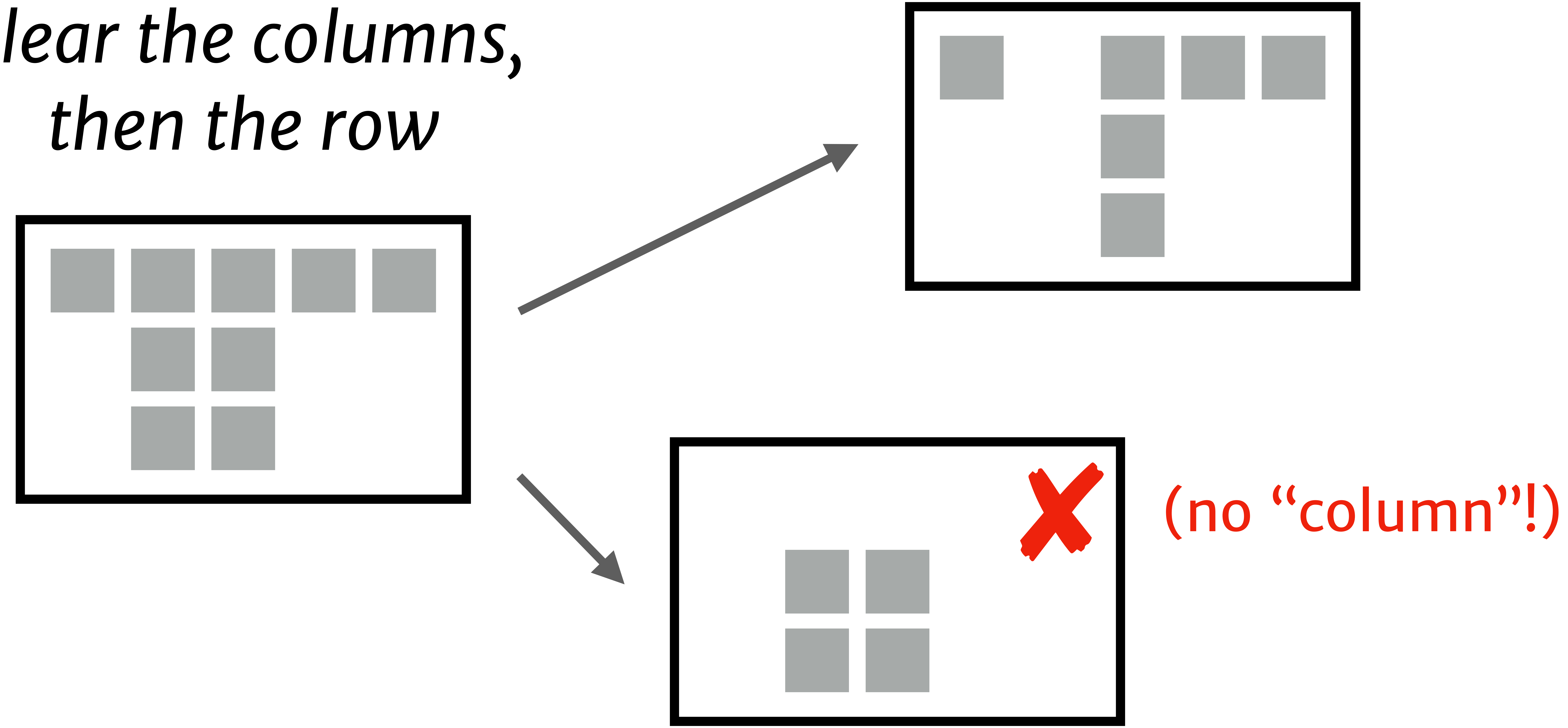
Constraints without logic

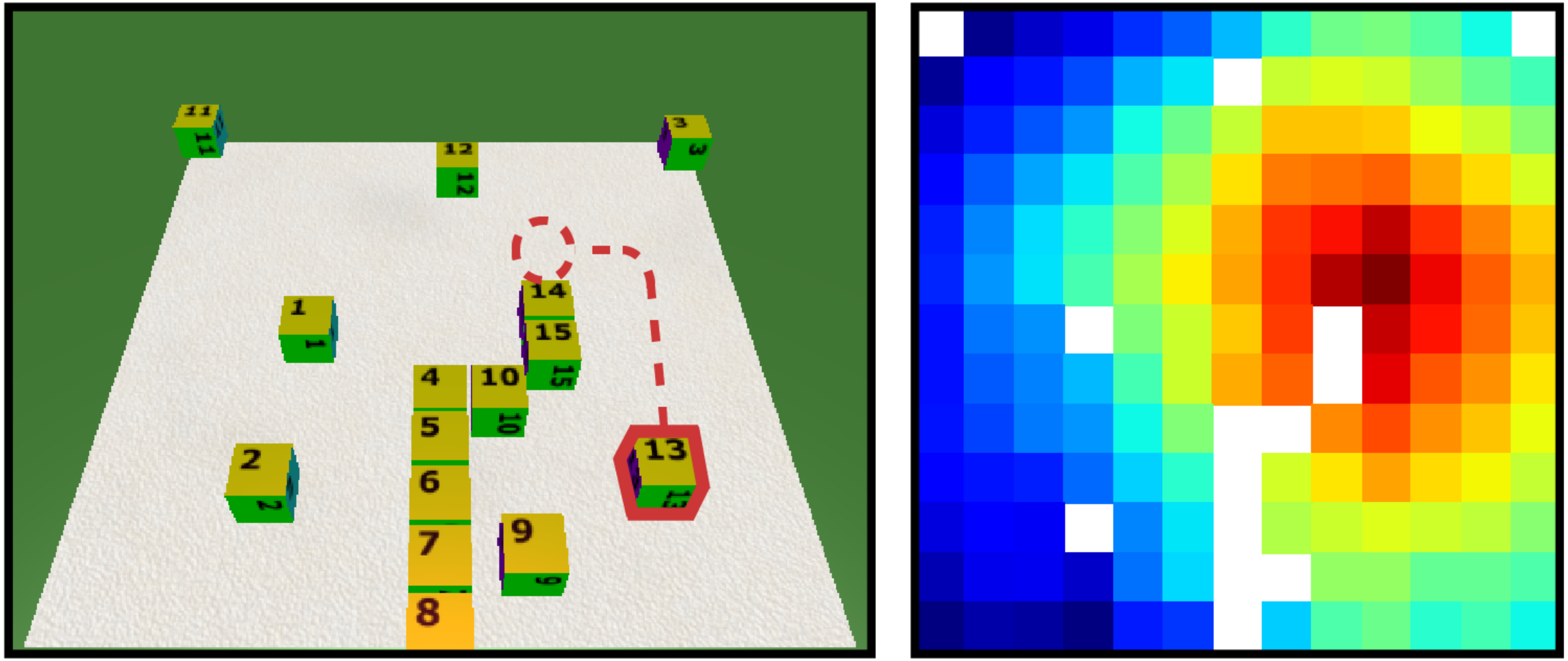
Clear the columns,
then the row



Constraints without logic

Clear the columns,
then the row





Take block 13 and place it directly above block 14 so they are almost touching.

[Janner et al., TACL ’18]

Our toolkit so far

Instruction following

Act in complex environments

With expressive policies that condition on instructions and observations

Track progress over time

In the underlying state space or RNN state

Plan ahead and reason about outcomes

With a symbolic planner or learned cost function

What else can we do?

Application: instruction generation

Instruction following

Move into the living room. Go forward then face the sofa.

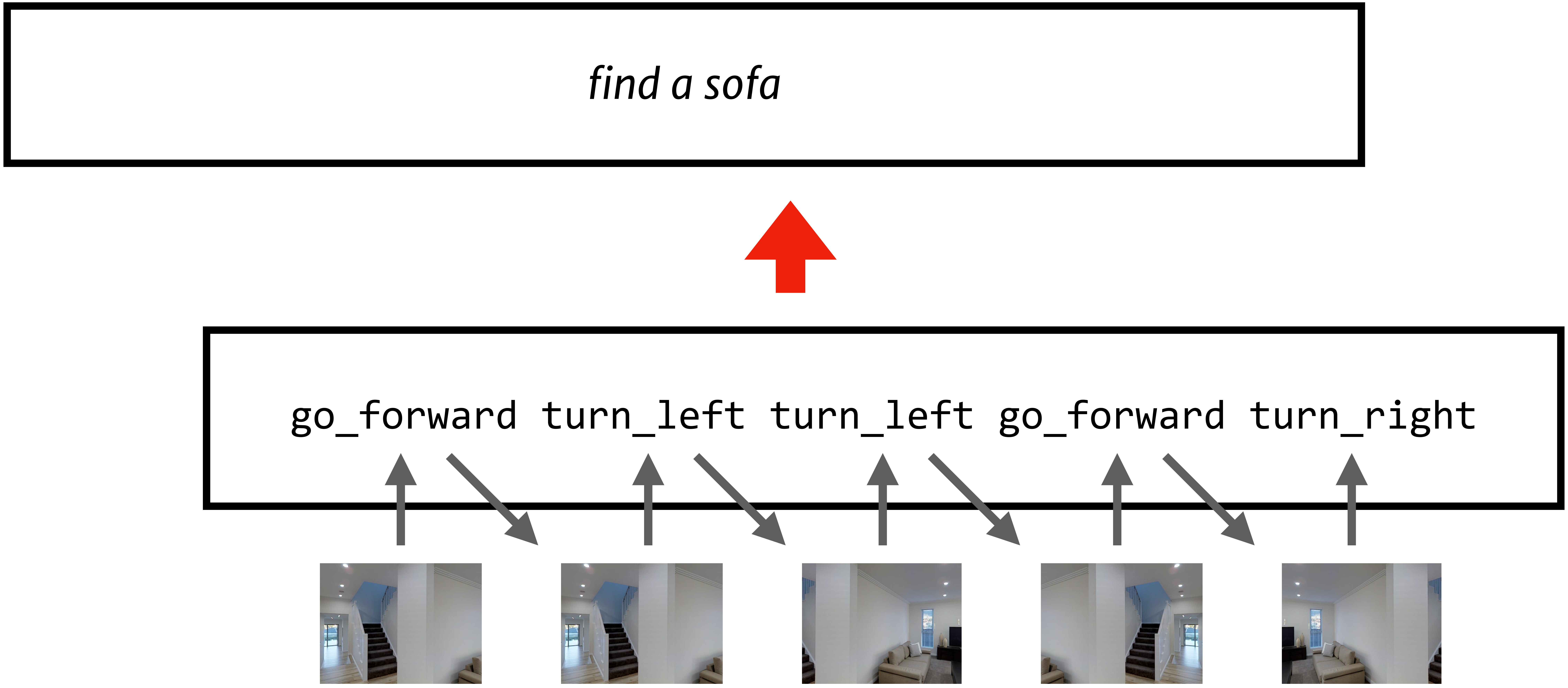
`go_forward turn_left turn_left go_forward turn_right`

Instruction ~~following~~ generation

Move into the living room. Go forward then face the sofa.

`go_forward turn_left turn_left go_forward turn_right`

Prediction action sequences



Instruction generation

Key idea: a good instruction gets readers to their goal with high probability (whatever the training data says!)

Instruction generation

Max posterior probability

$$\max_{text} \ p(text \mid plan; \theta)$$

("how do people describe this?")

Instruction generation

Max posterior probability

$$\max_{text} \; p(text \mid plan; \theta)$$

("how do people describe this?")

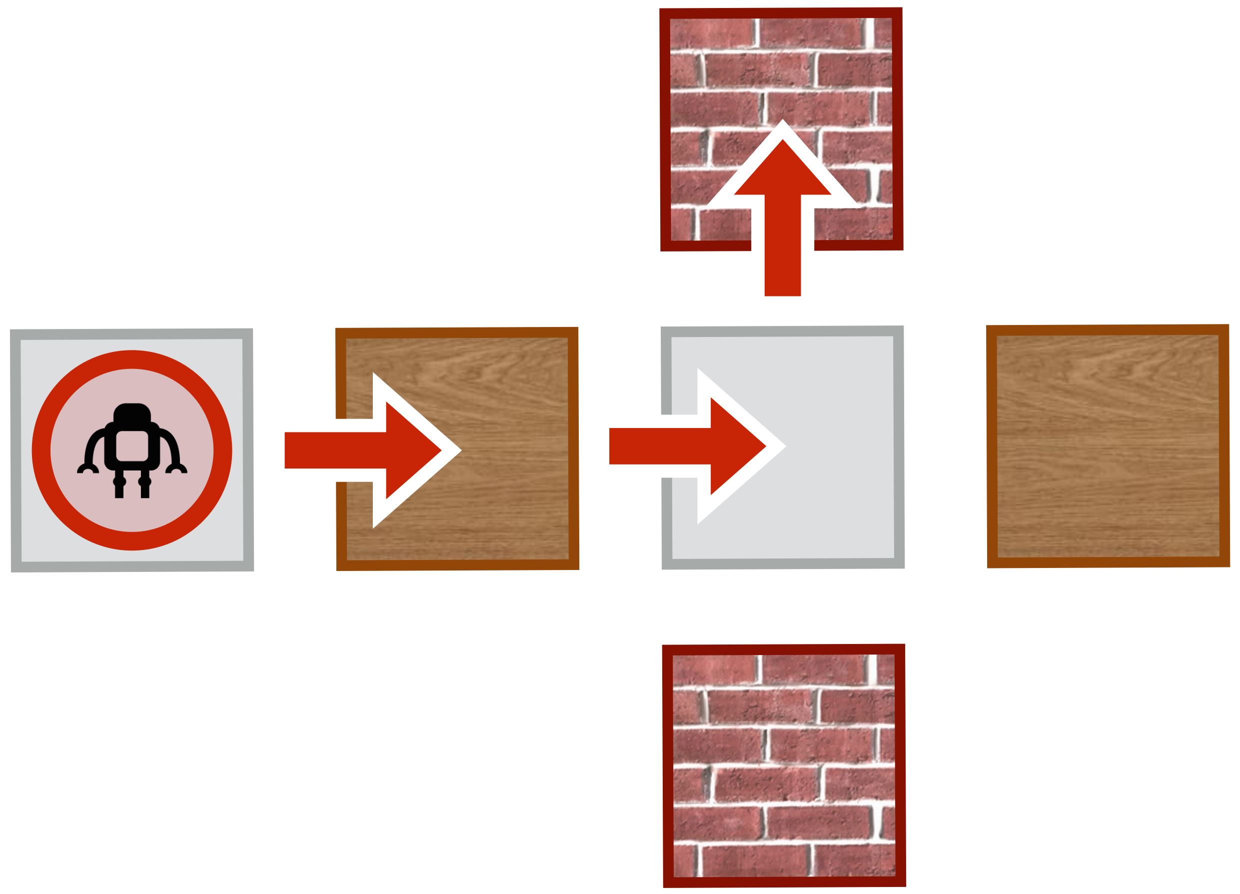
min Bayes risk

$$\max_{text} \; p(plan \mid text; \theta)$$

("how do I make people do this?")

Reasoning about outcomes

$$\max_{text} \; p(plan \mid text; \theta)$$

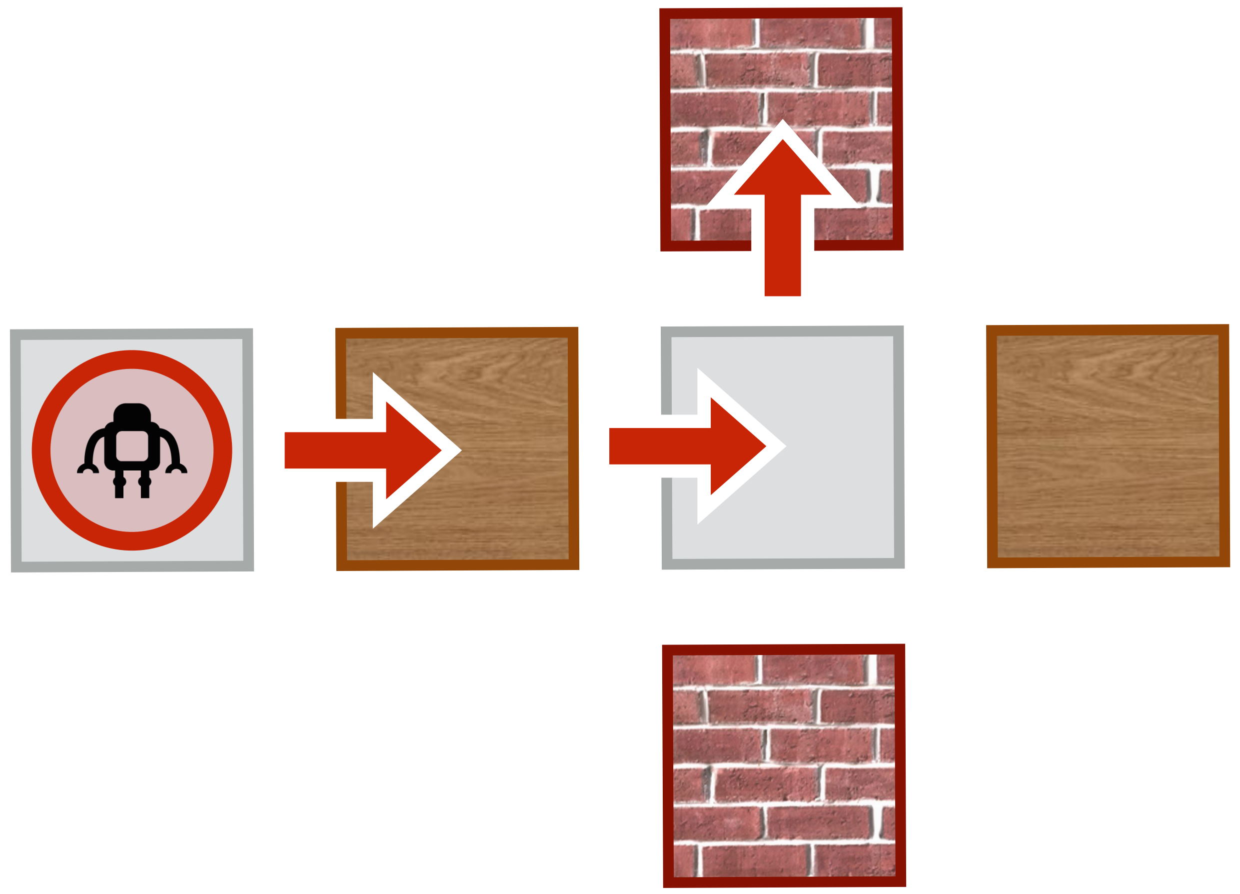


I will make a turn.

Instruction follower

Reasoning about outcomes

$$\max_{\textit{text}} \; p(\textit{plan} \mid \textit{text}; \theta)$$

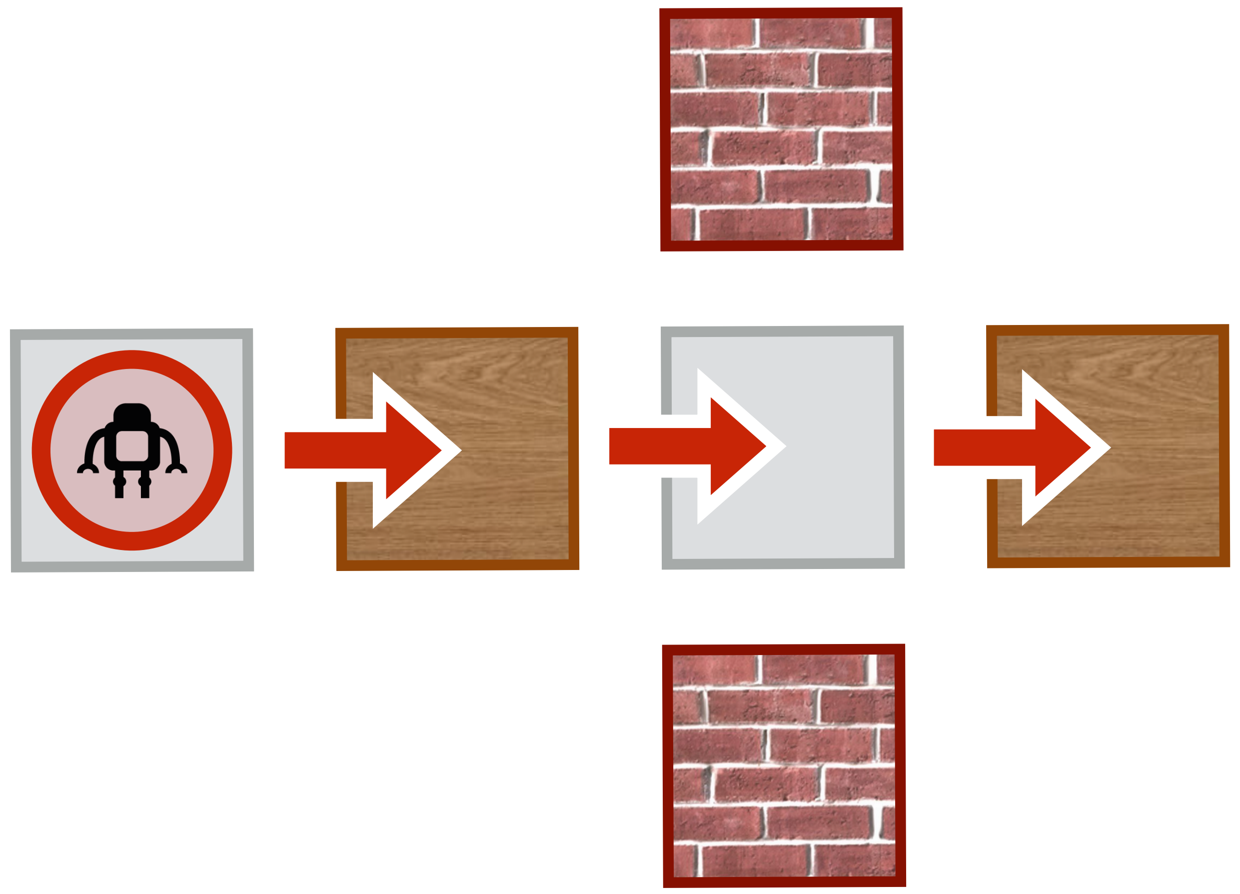


I will make a turn.

Listener

Reasoning about outcomes

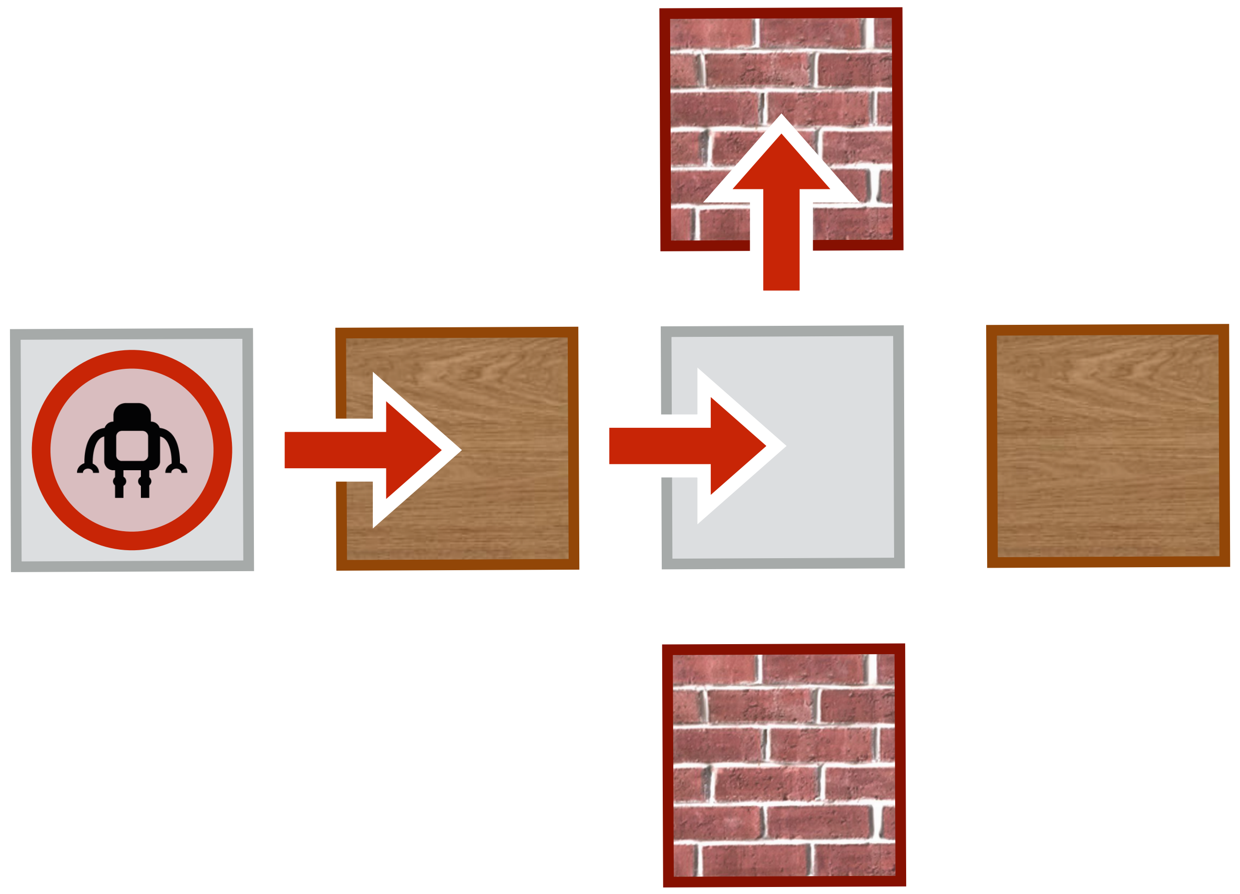
$$\max_{\textit{text}} \; p(\textit{plan} \mid \textit{text}; \theta)$$



I will go straight through.

Reasoning about outcomes

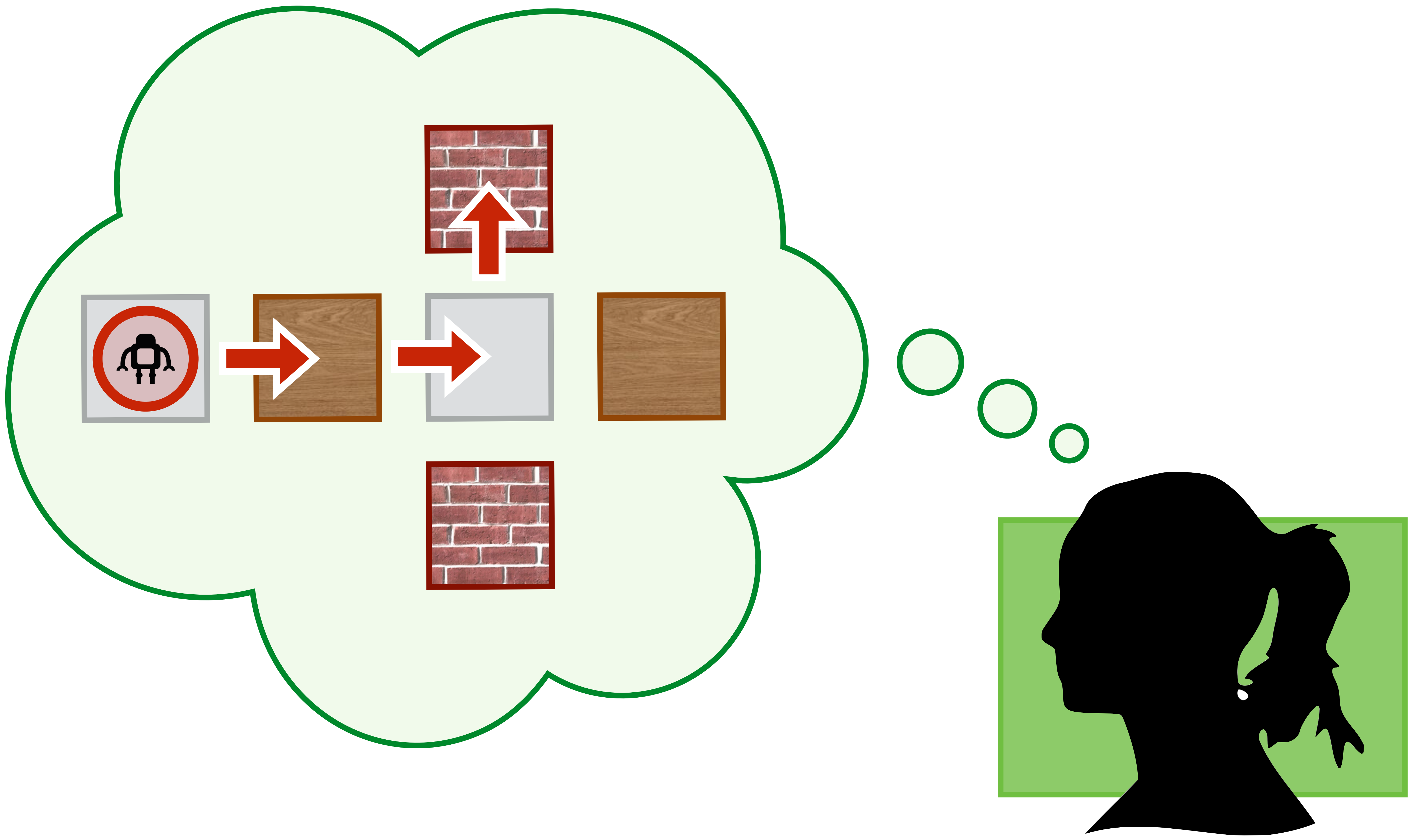
$$\max_{\color{deepskyblue}{text}} \; p(\color{salmon}{plan} \mid \color{deepskyblue}{text}; \theta)$$



I will turn left at the brick intersection.

[Fried et al., NAACL ’18]

Reasoning about belief



I will turn left at the brick intersection.

[Frank & Goodman, Trends in Cog. Sci. '12]

speaker: *Walk past the dining room table and chairs and wait there.*

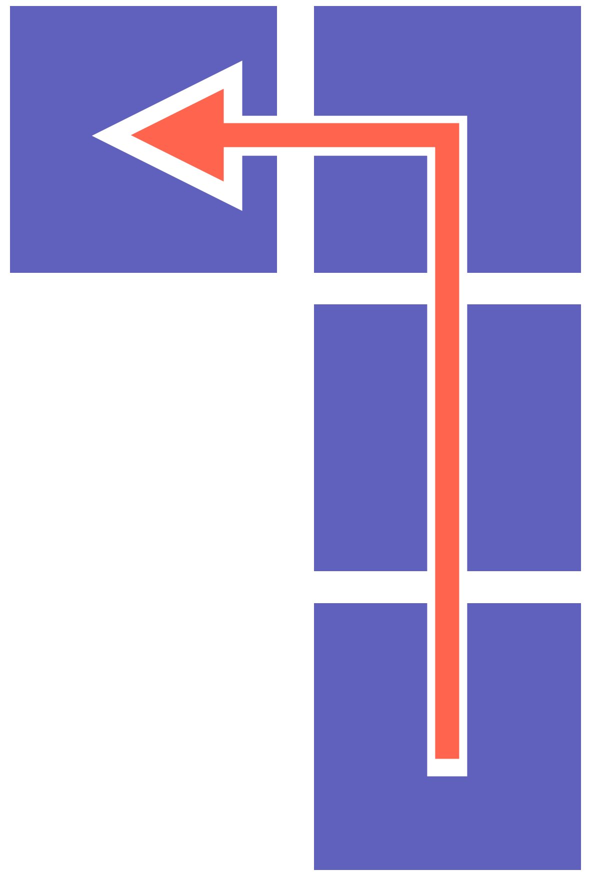
listener: *Walk past the dining room table and chairs and take a right into the living room. Stop once you are on the rug.*

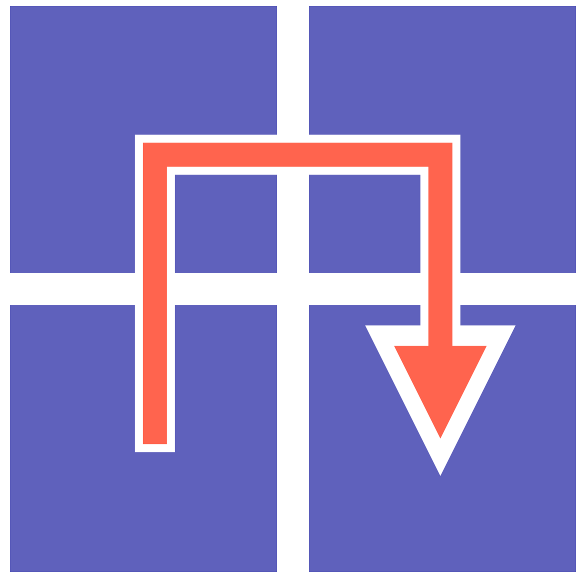
human: *Turn right and walk through the kitchen. Go right into the living room and stop by the rug.*

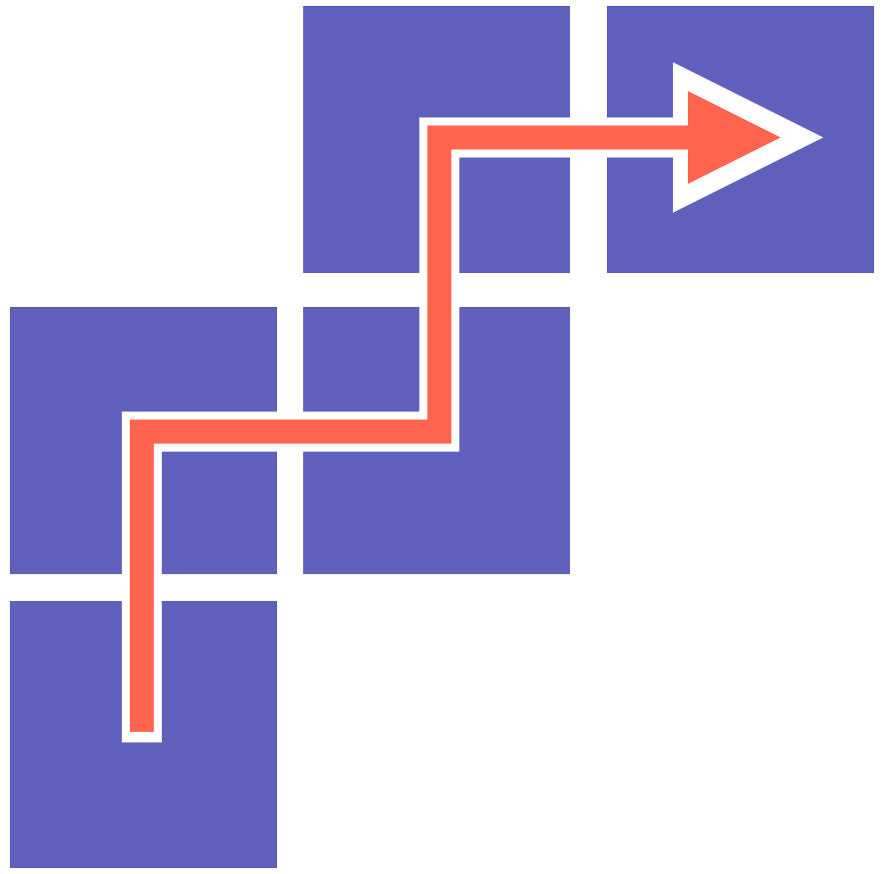
Application: machine teaching

Instructions as scaffolds for RL

Instructions as parameter-tying schemes







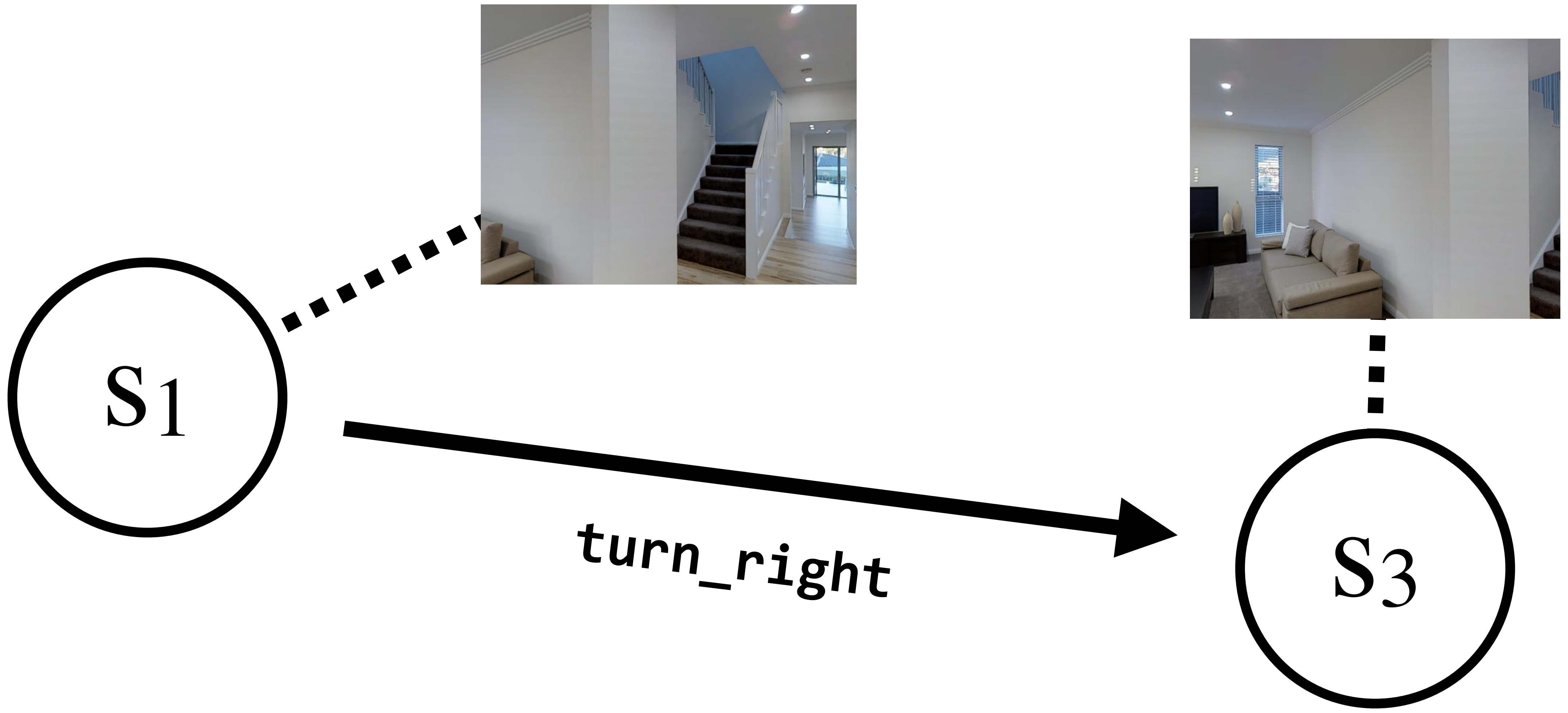
Instructions as parameter-tying schemes

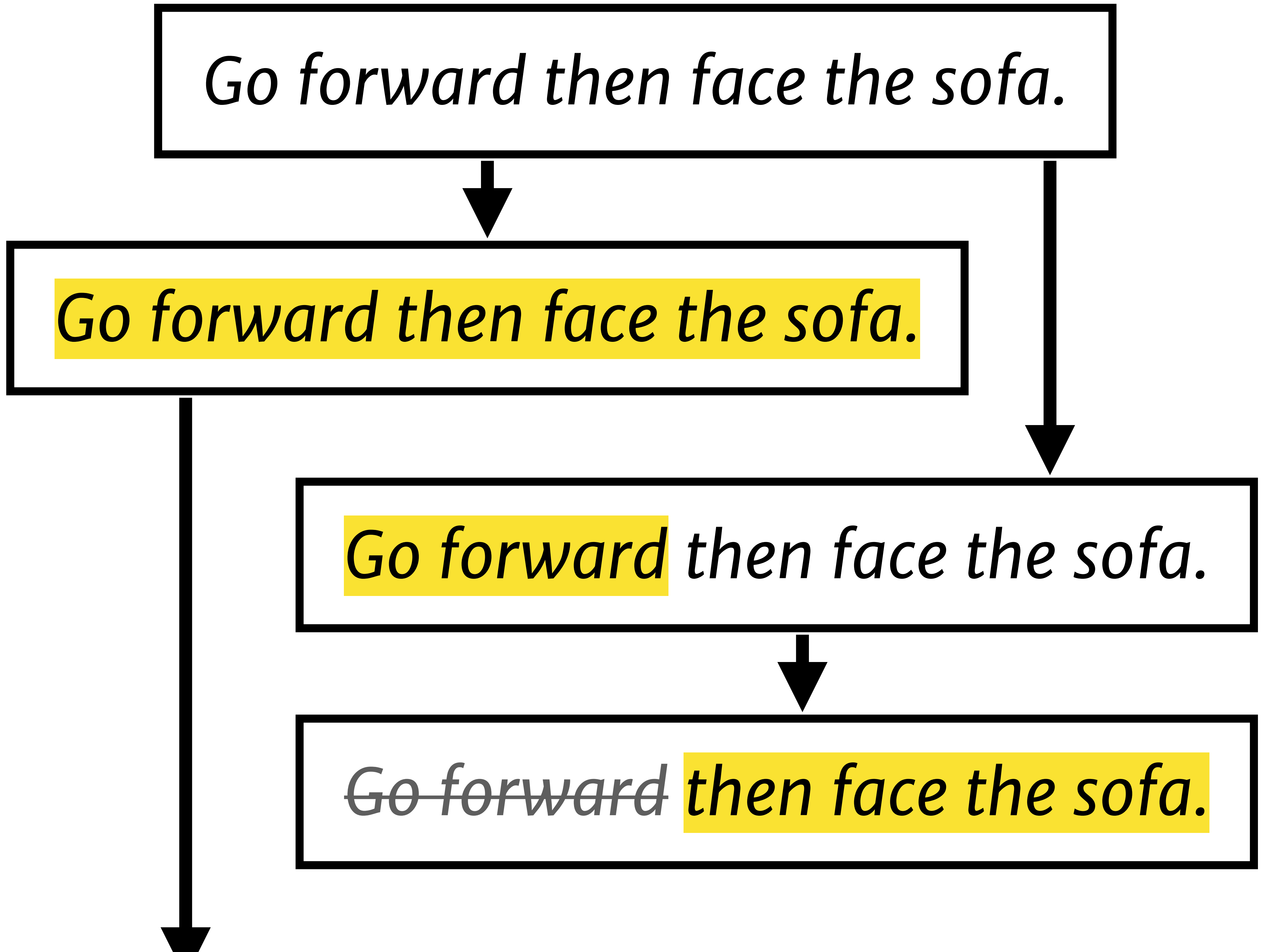
Environment states S_e

Environment actions A_e

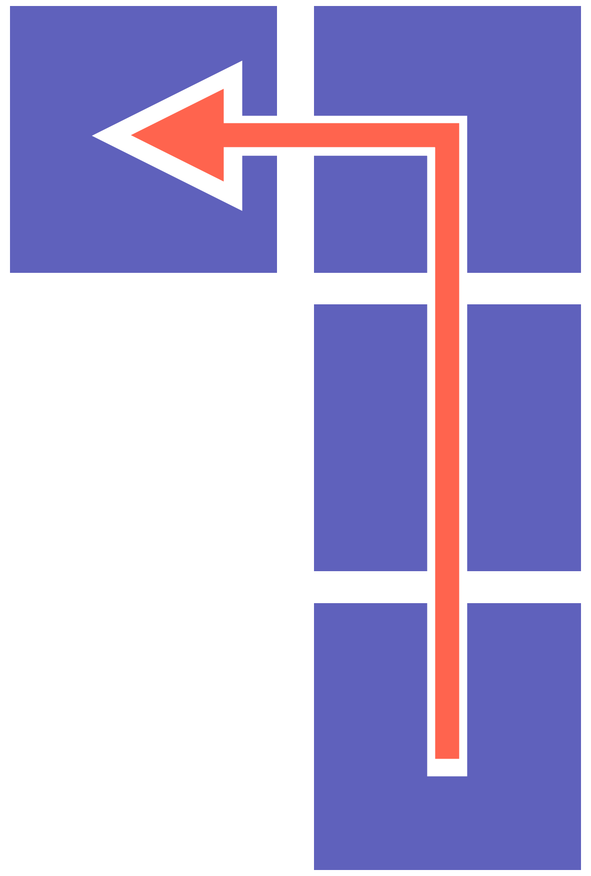
Reading states S_e

Reading actions A_e

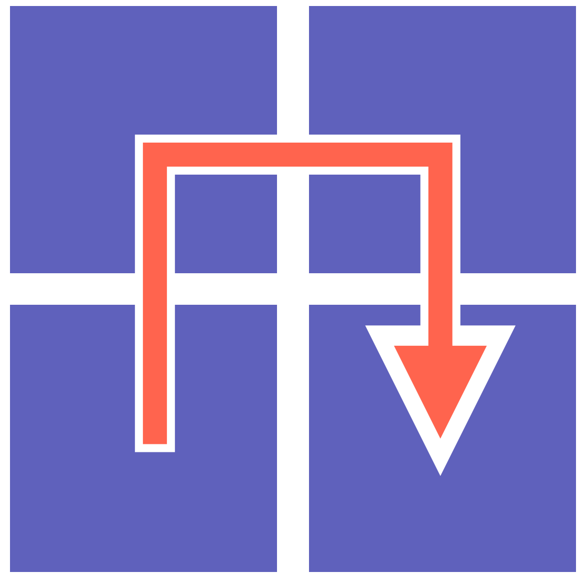




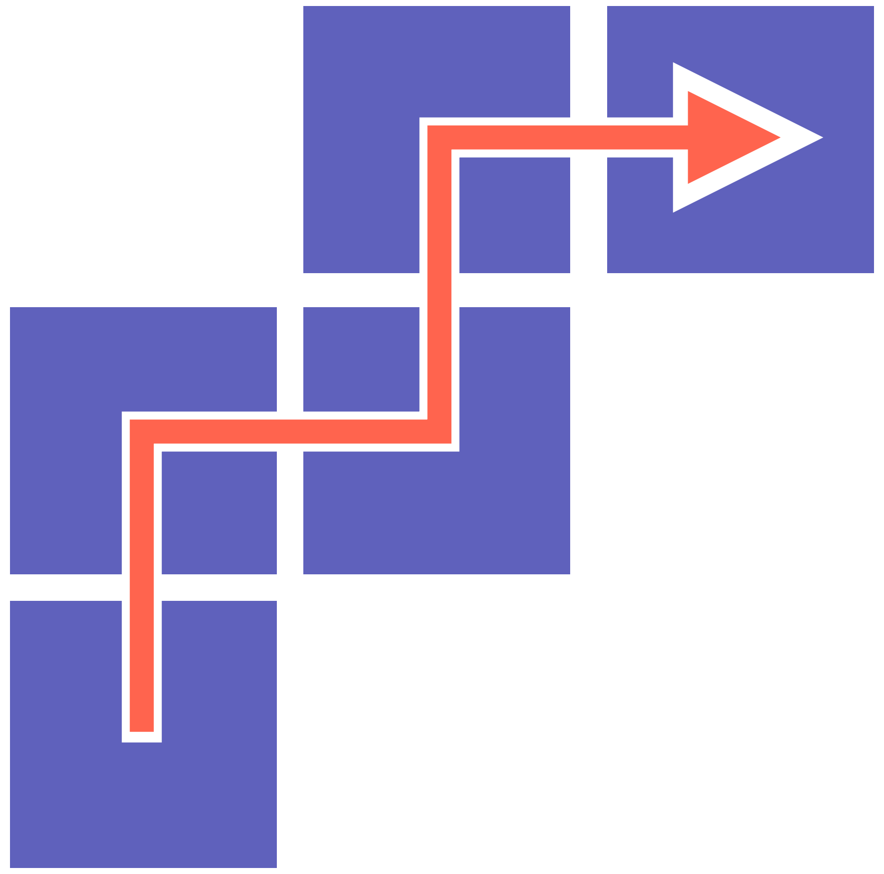
Instructions as parameter-tying schemes



go north, go north, go west



go north, go east, go south

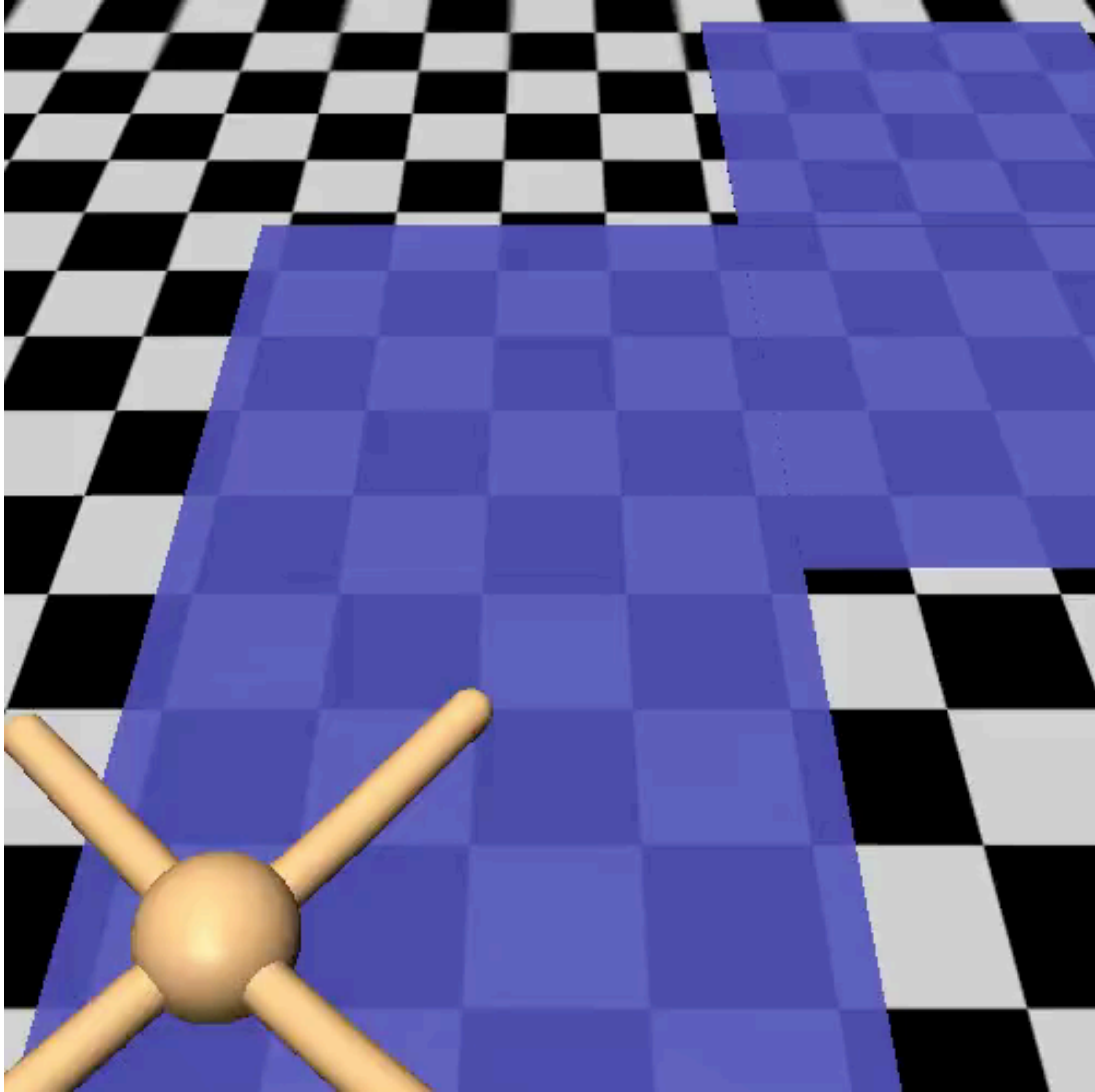


go north, go east, go north, ...

Go north.

Go east.

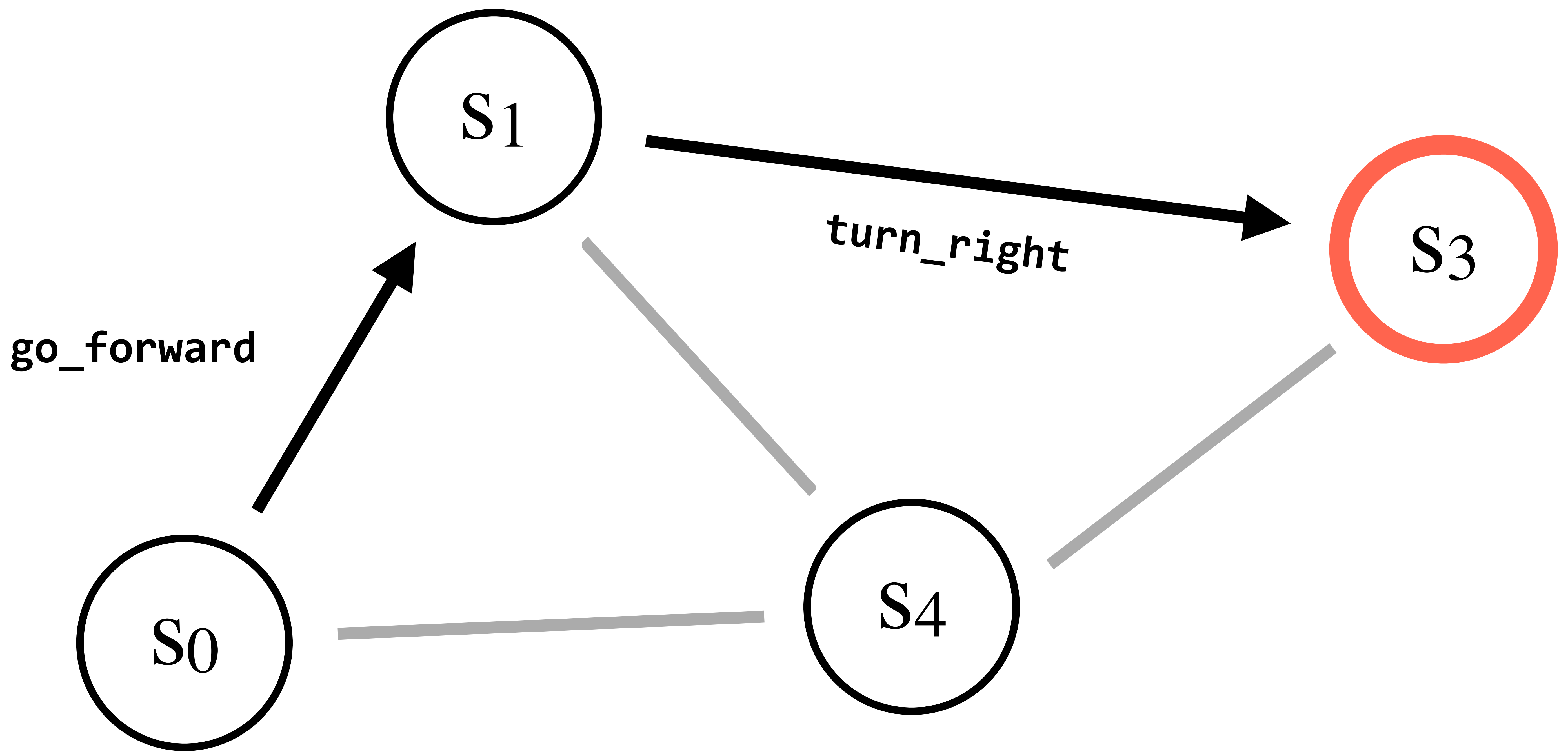
Go north.



[Andreas et al., ICML '17]

Learning interactively from corrections

Supervision



Conditioning on the past

Push the chair against the wall.

```
go_forward grasp turn_left go_forward release
```

Conditioning on the past

Push the chair against the wall.

`go_forward grasp turn_left go_forward release`

No, the red chair.

`turn_left grasp go_forward go_forward release`

Conditioning on the past

Push the chair against the wall.

`go_forward grasp turn_left go_forward release`

No, the red chair.

`turn_left grasp go_forward go_forward release`

Now a little to the left.

`turn_left grasp go_forward turn_left release`

Conditioning on the past

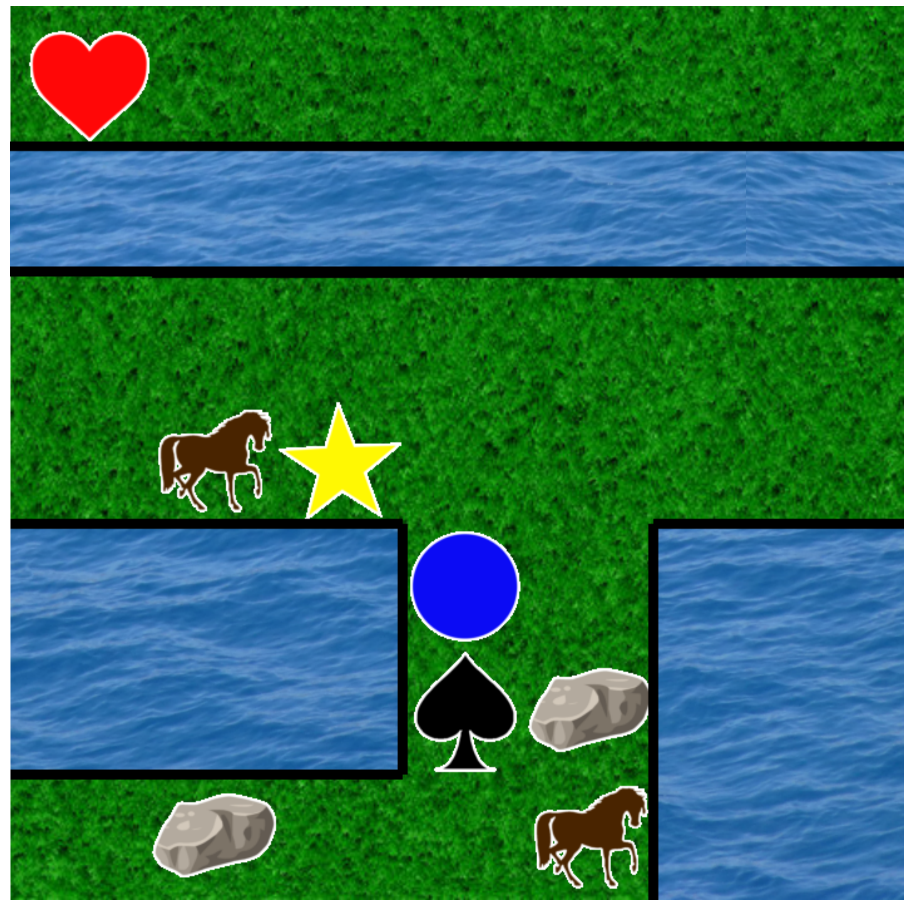
Key idea: learn to solve problems interactively by conditioning on the whole history of instructions.

[Co-Reyes et al., ICLR '19]


Touch cyan block.
Move closer to magenta block.
Move a lot up.
Move a little up.

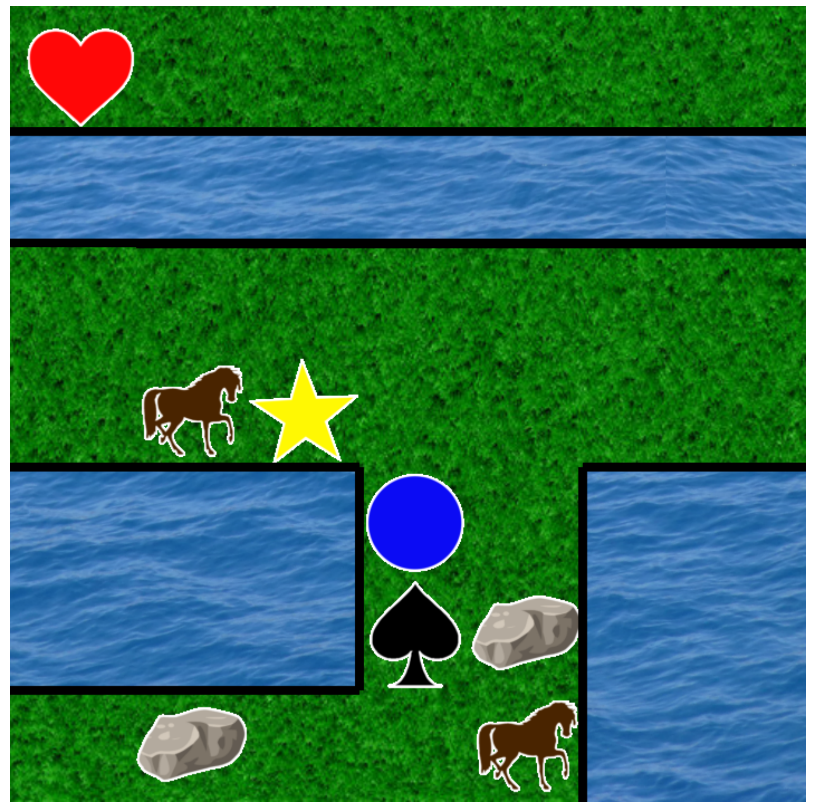
Learning with latent language

Language learning as pretraining

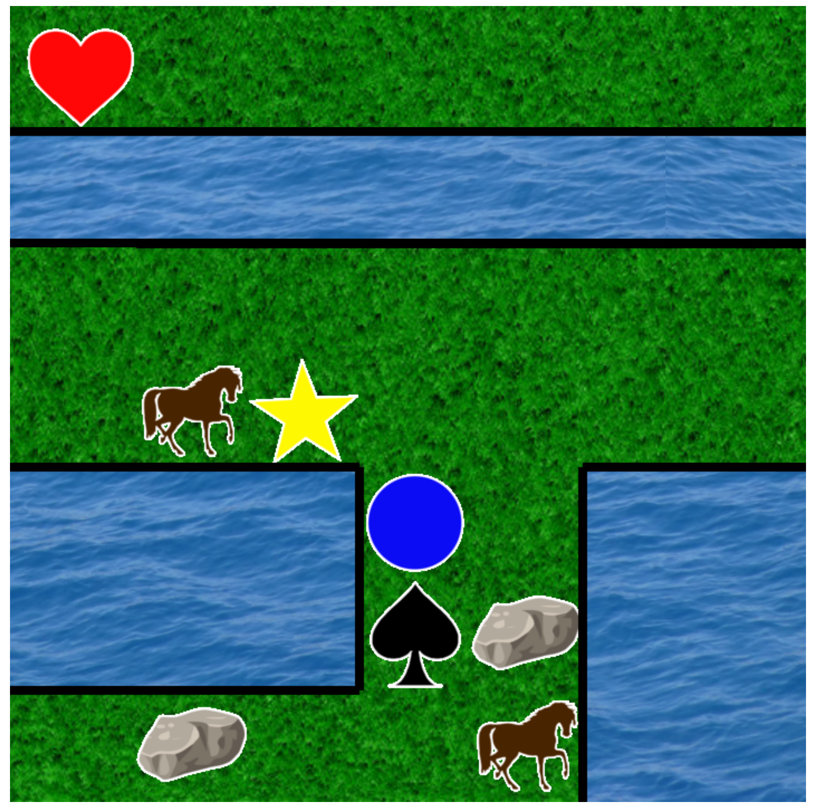


→ `reach the heart` → **FORWARD**

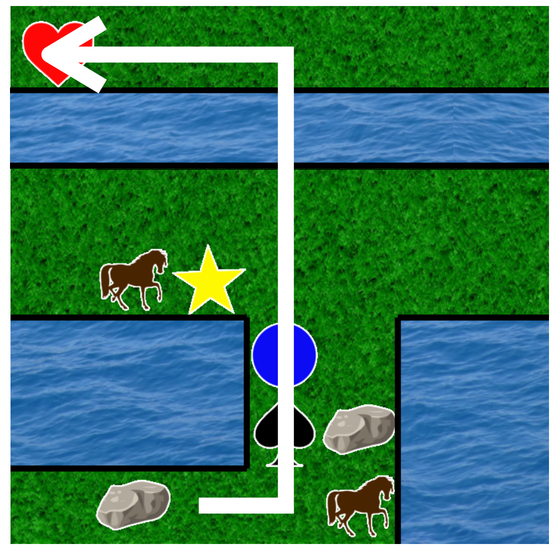
Structured exploration



Structured exploration

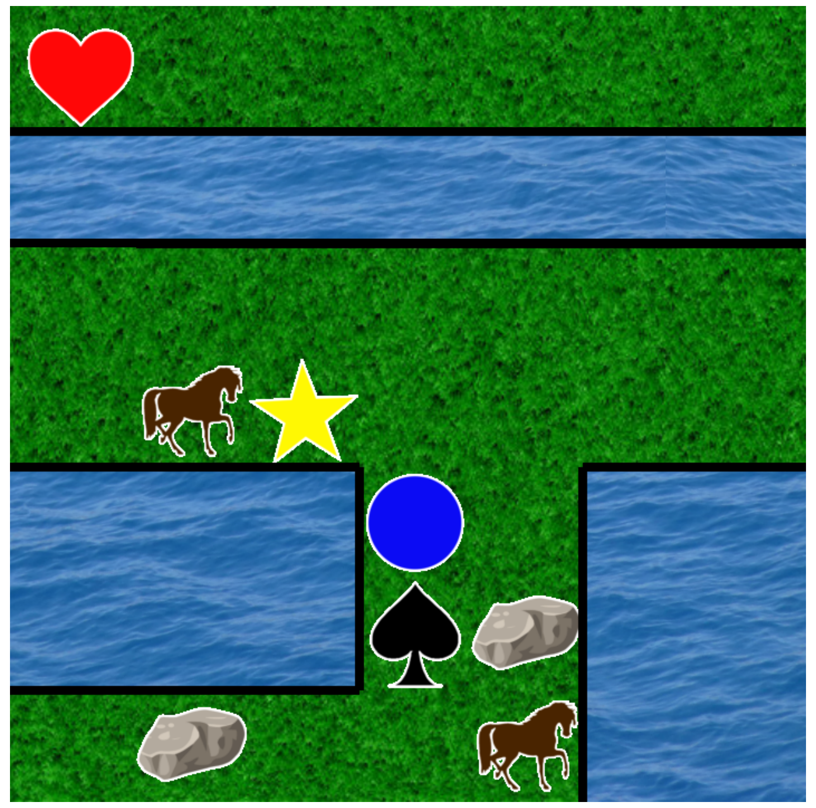


reach the heart

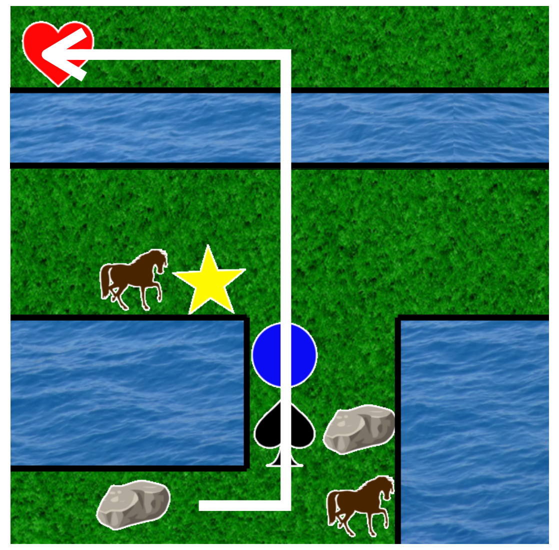


R = −1

Structured exploration

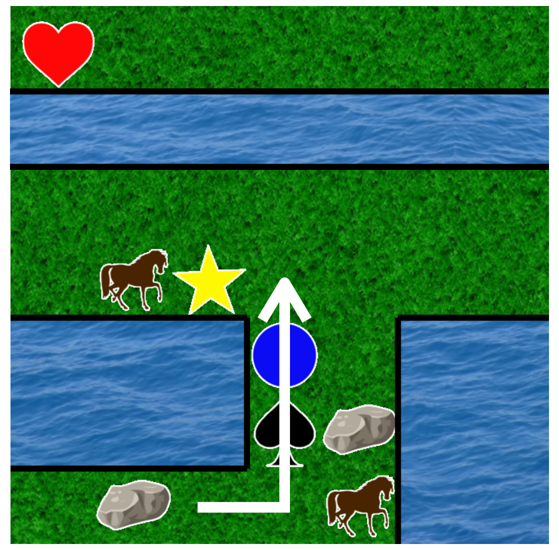


reach the heart



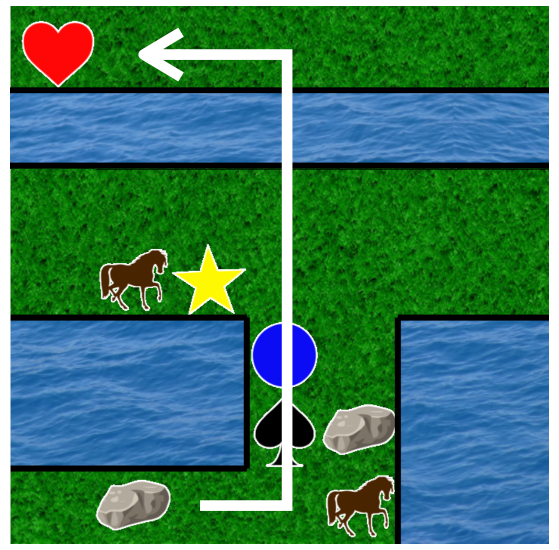
R = −1

east of the gold star



R = 0

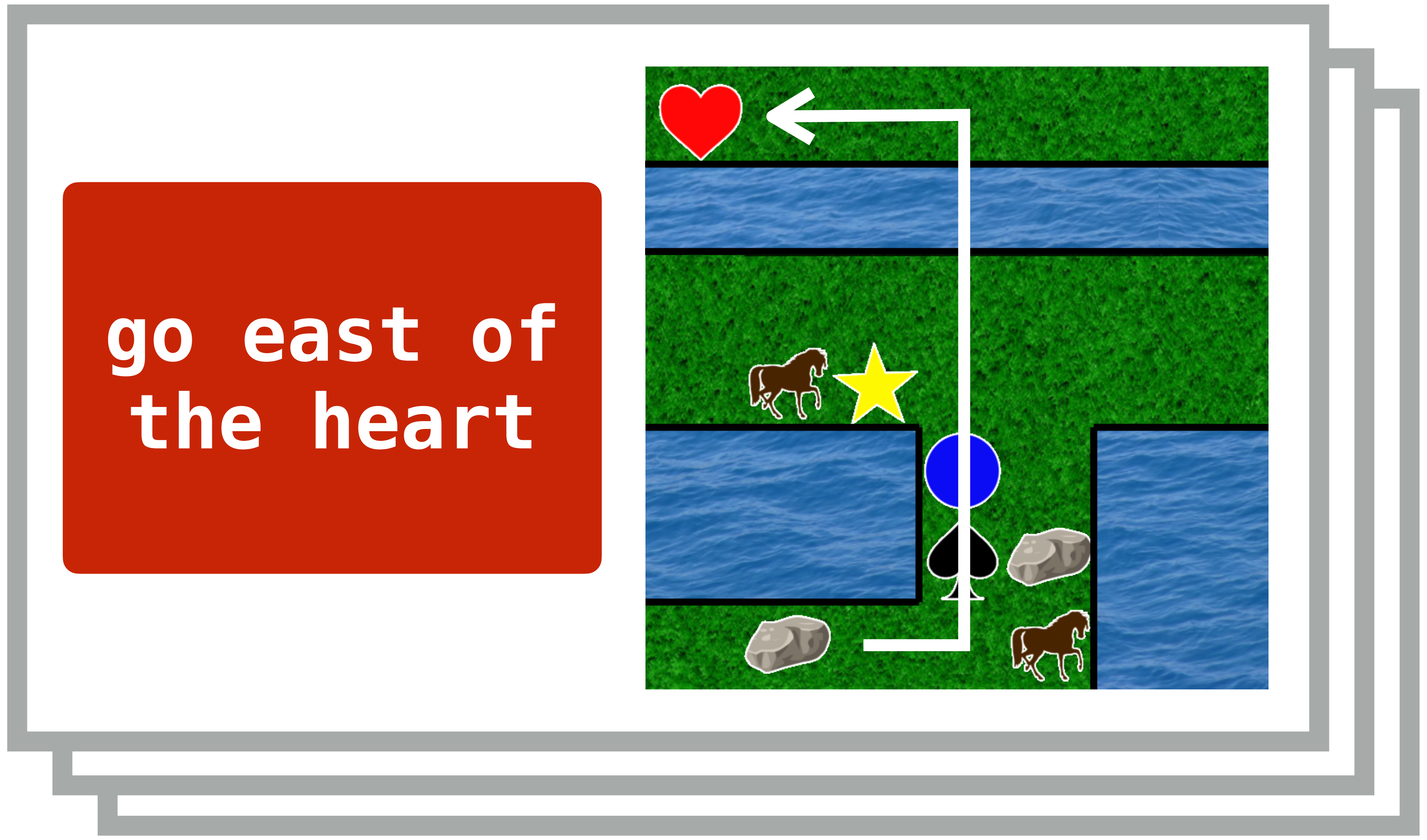
go to the east of the heart



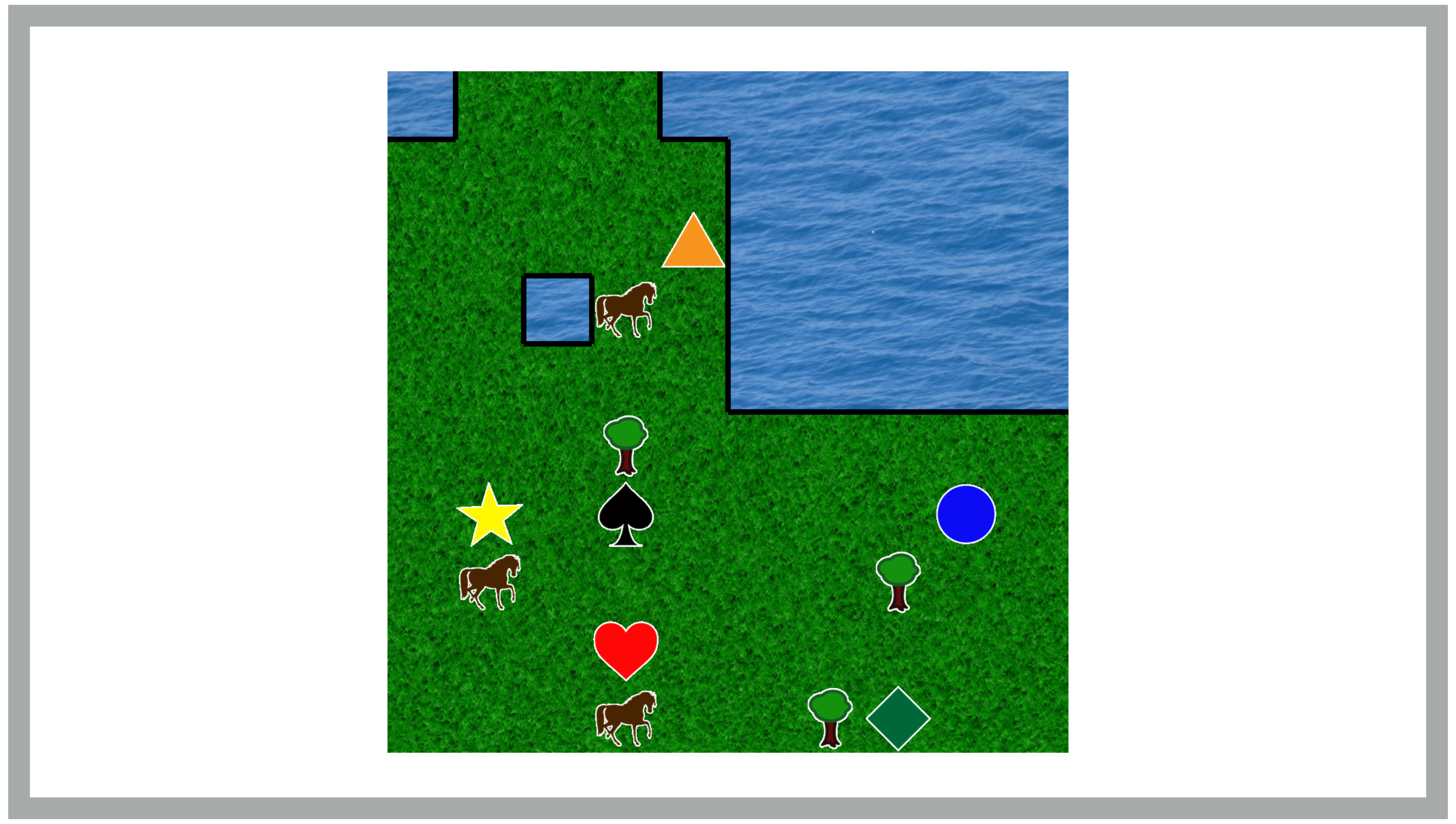
R = 3

Structured exploration

Language learning



Reinforcement learning



[Andreas et al., NAACL ’19]

Structured few-shot learning

examples

emboldens		emboldecs
kisses		kisses
loneliness	➡	locelicess
vein		veic
dogtrot		dogtrot

change any n
to a c

pred. description

Structured few-shot learning

examples

emboldens	emboldecs
kisses	kisses
loneliness →	locelicess
vein	veic
dogtrot	dogtrot

true description

replace all n s with c

change any n to a c

pred. description

true output

loocies

↑

loonies

↓

loocies

pred. output

TICK: 47

MINERALS: 300

Faster

Current FPS is 29

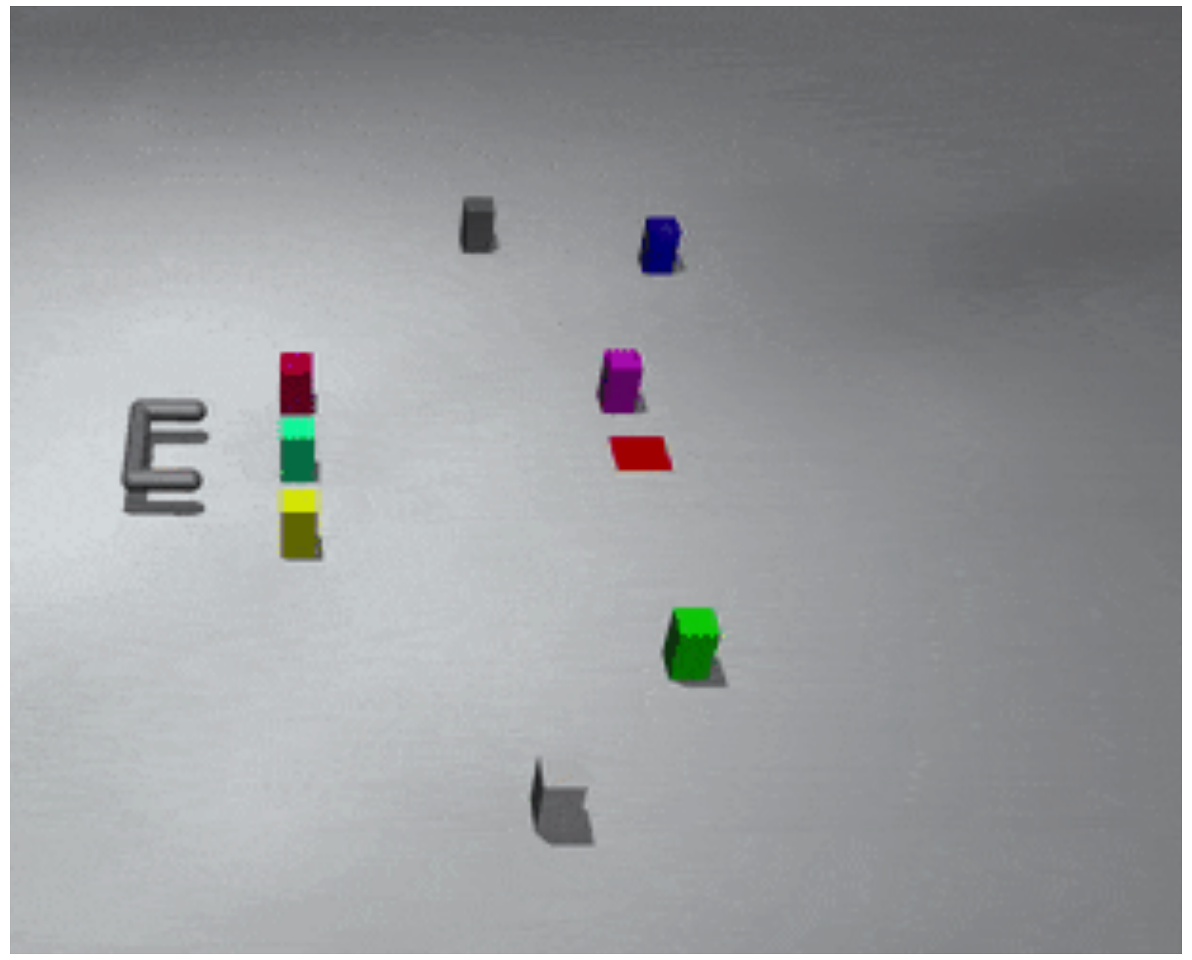
Current progress_pe

Current order to execute on:

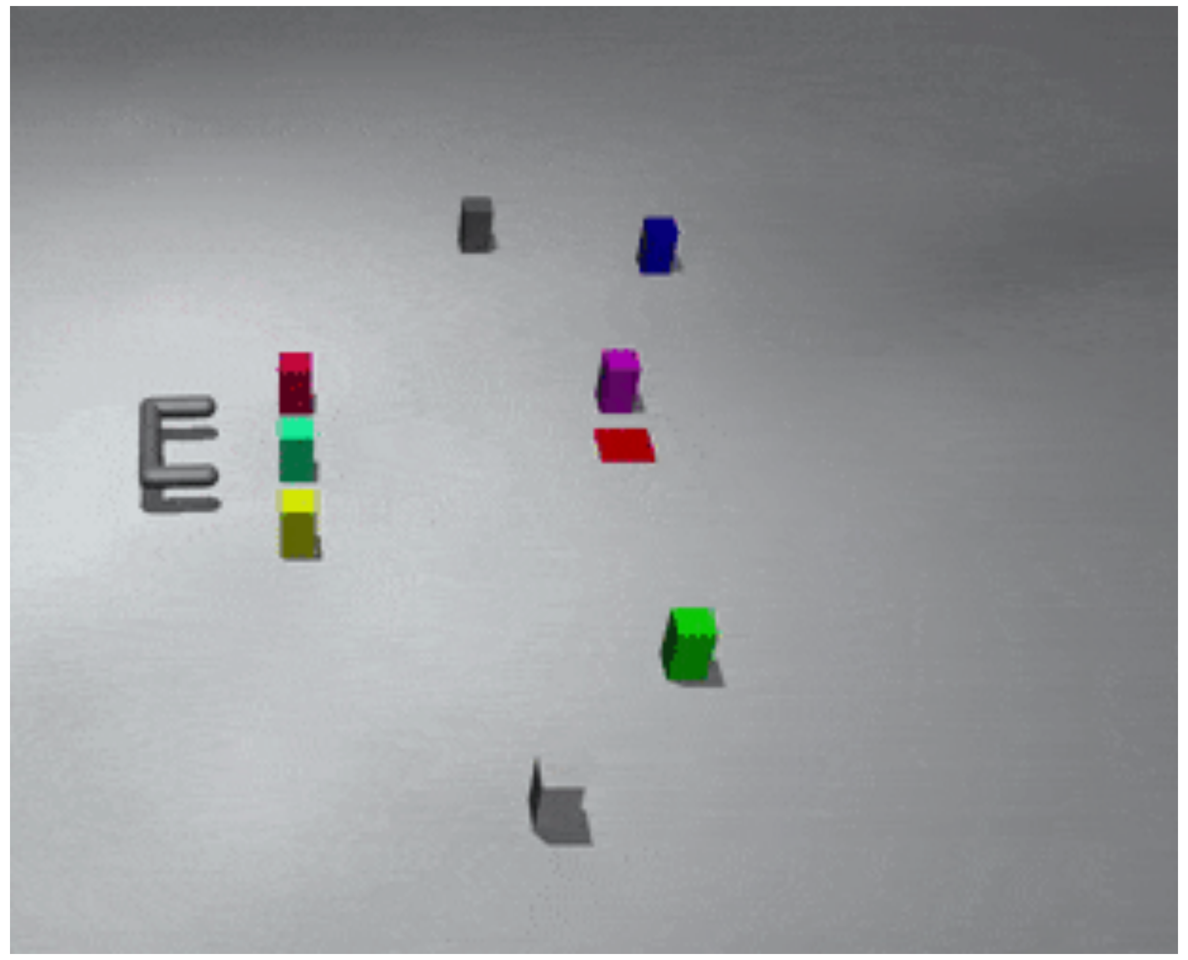
Send 2 peasants to mine upper ore

Future challenges

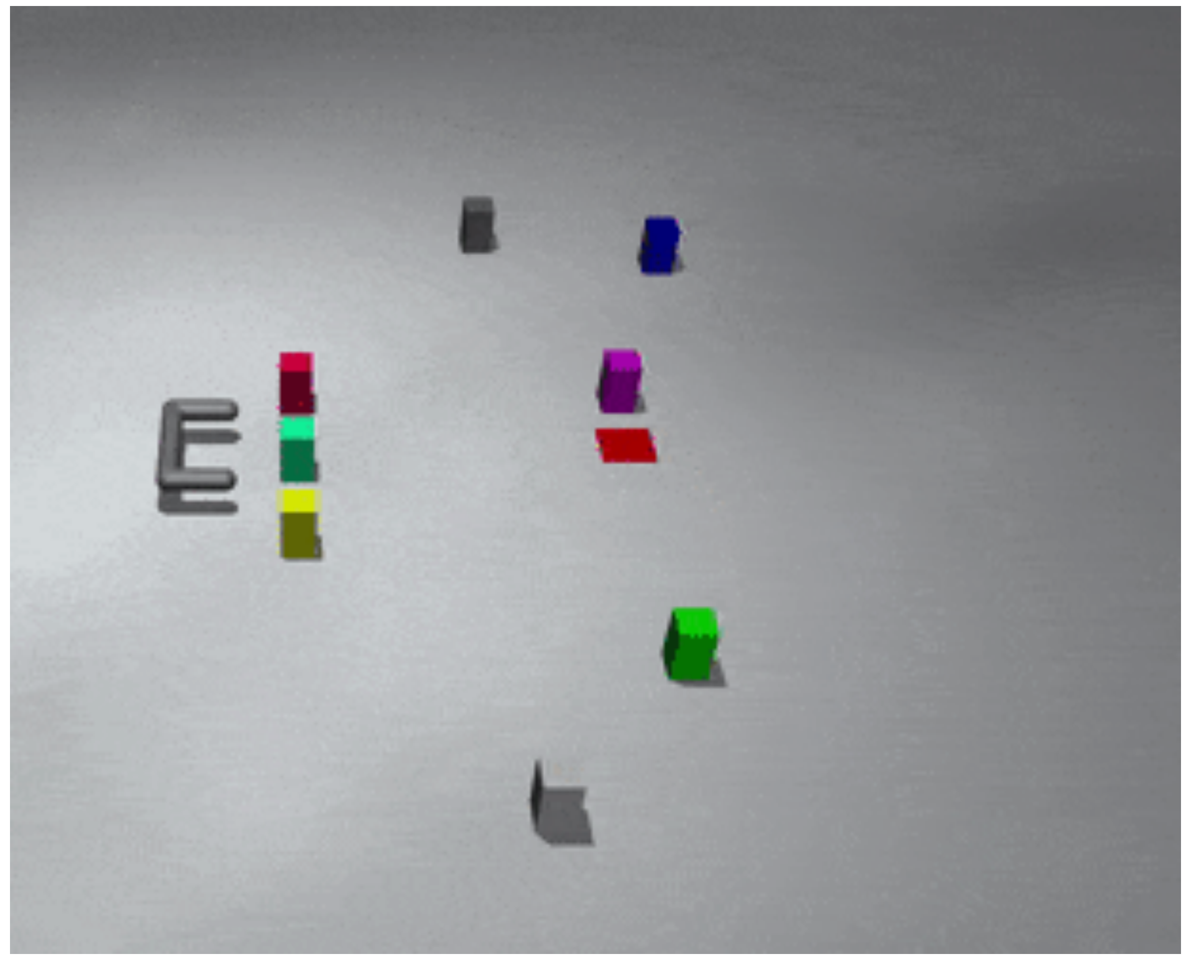
Fake data



Touch cyan block.

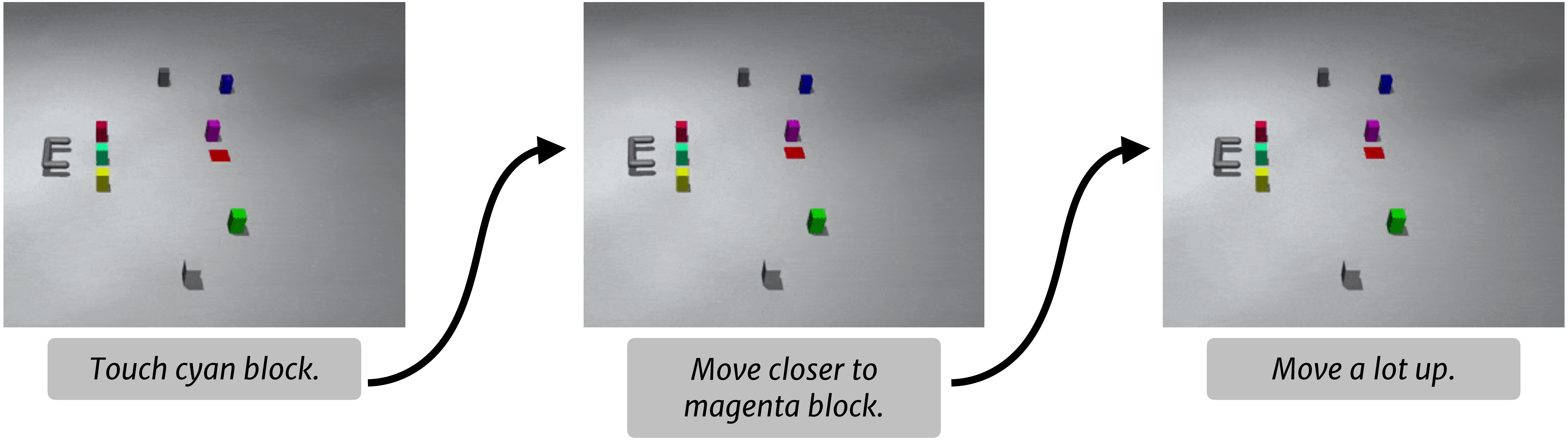


Move closer to magenta block.



Move a lot up.

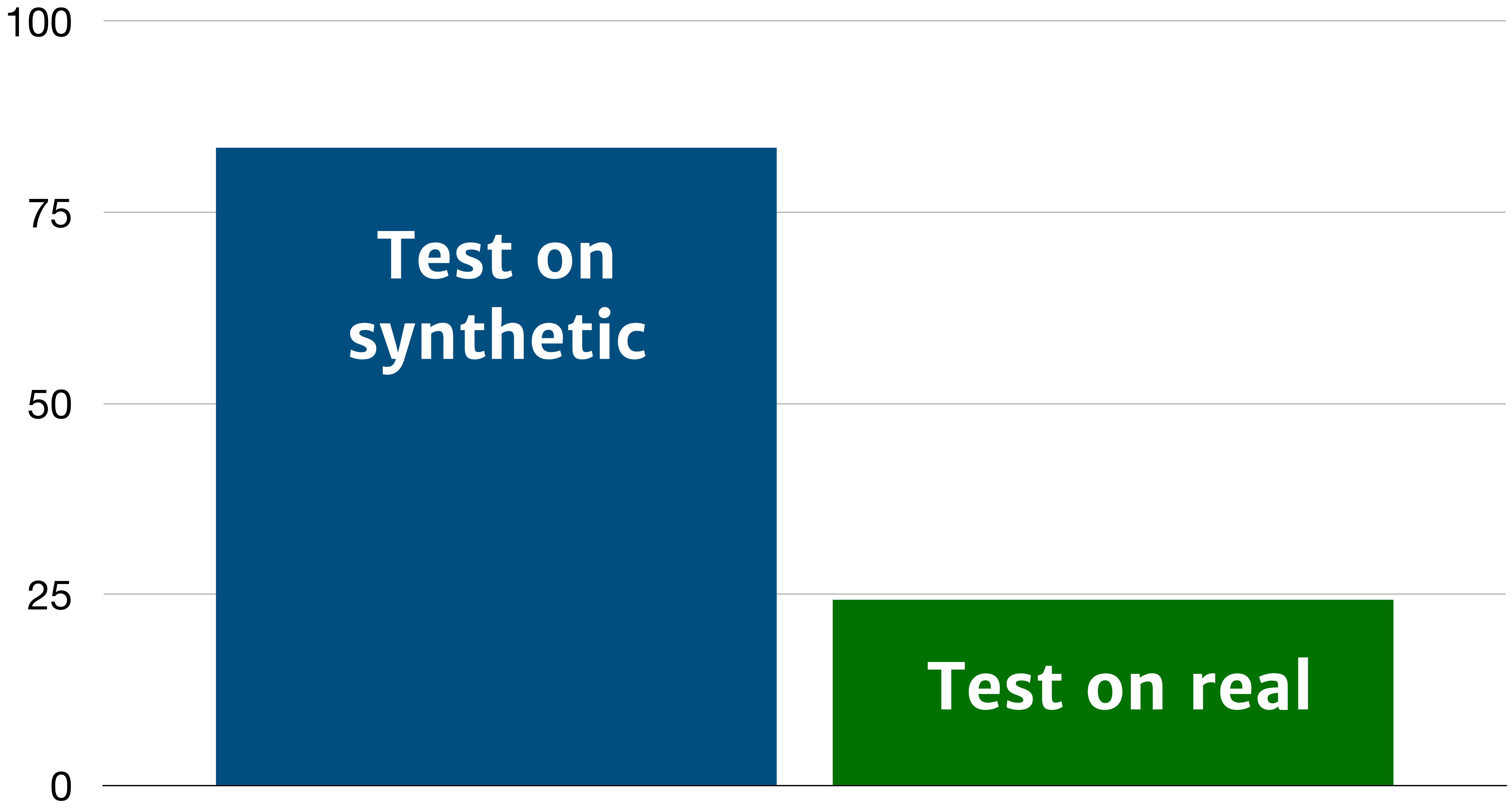
Fake data



“Instructions” are synthesized from a grammar because of sample inefficiency!

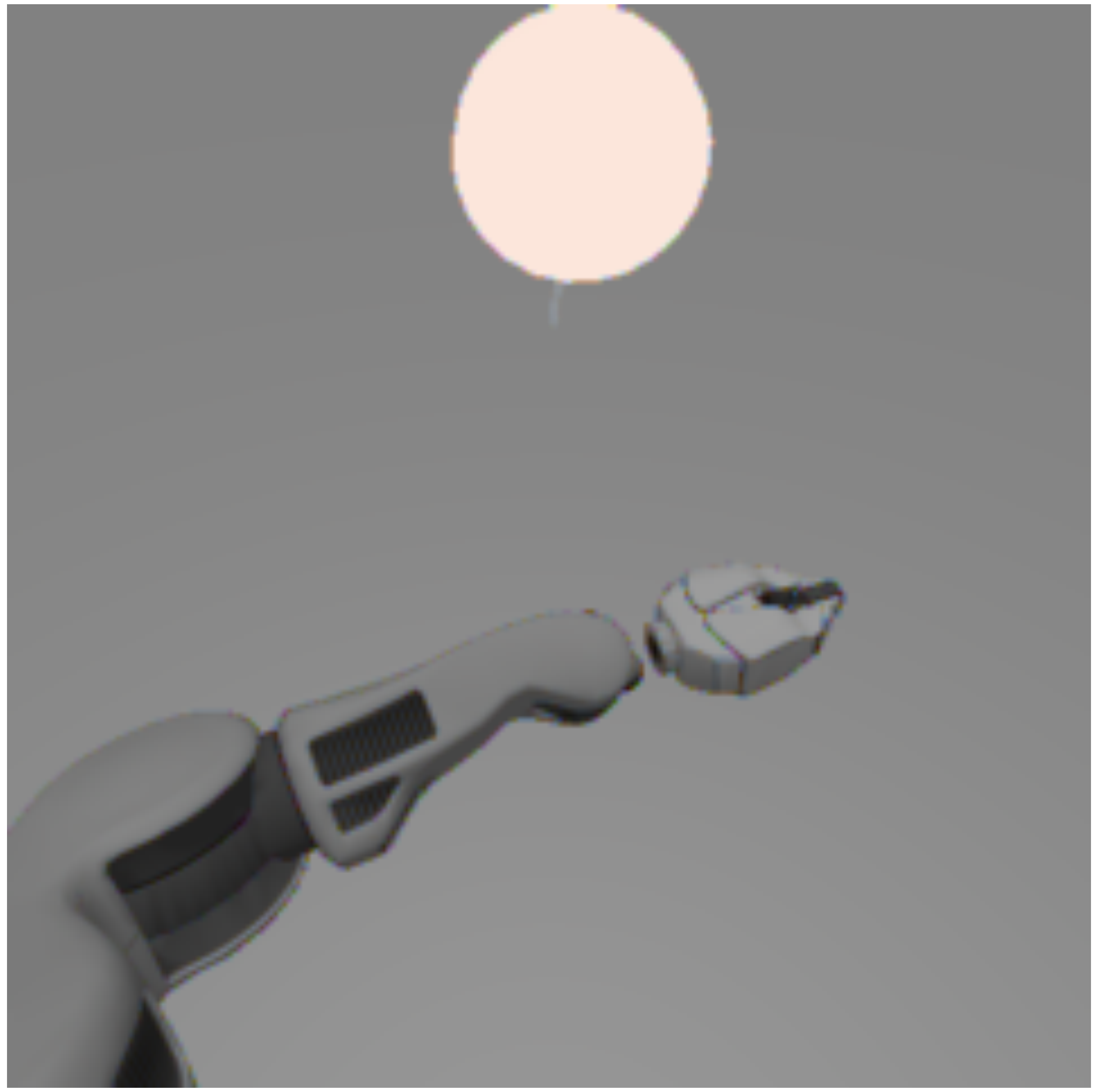
Fake data

Train on synthetic language and



[Blukis et al., CoRL ’18]

"Sim-to-real transfer"



[e.g. Tzeng et al. 2016, Adapting Deep Visuomotor Representations with Weak Pairwise Constraints]

"Sim-to-real transfer"

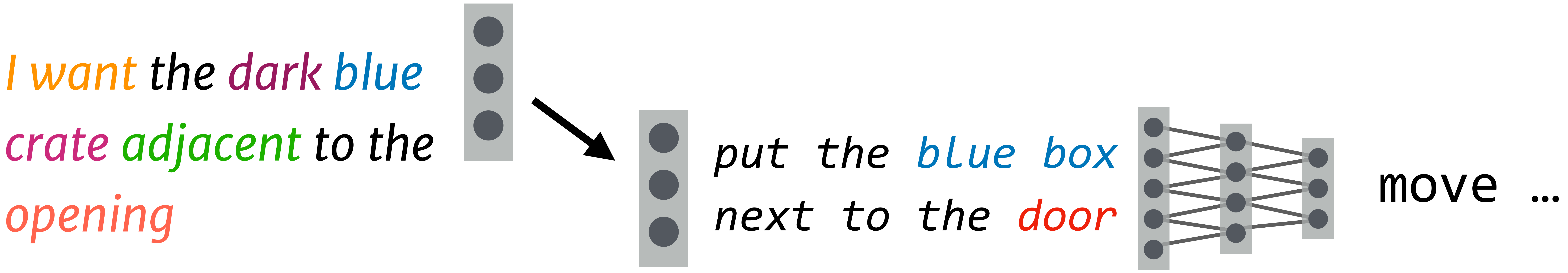
neighborhood with largest number of restaurants with thai cuisine

neighborhood with the most Thai restaurants

[e.g. Tzeng et al. 2016, Adapting Deep Visuomotor Representations with Weak Pairwise Constraints]

"Sim-to-real transfer"

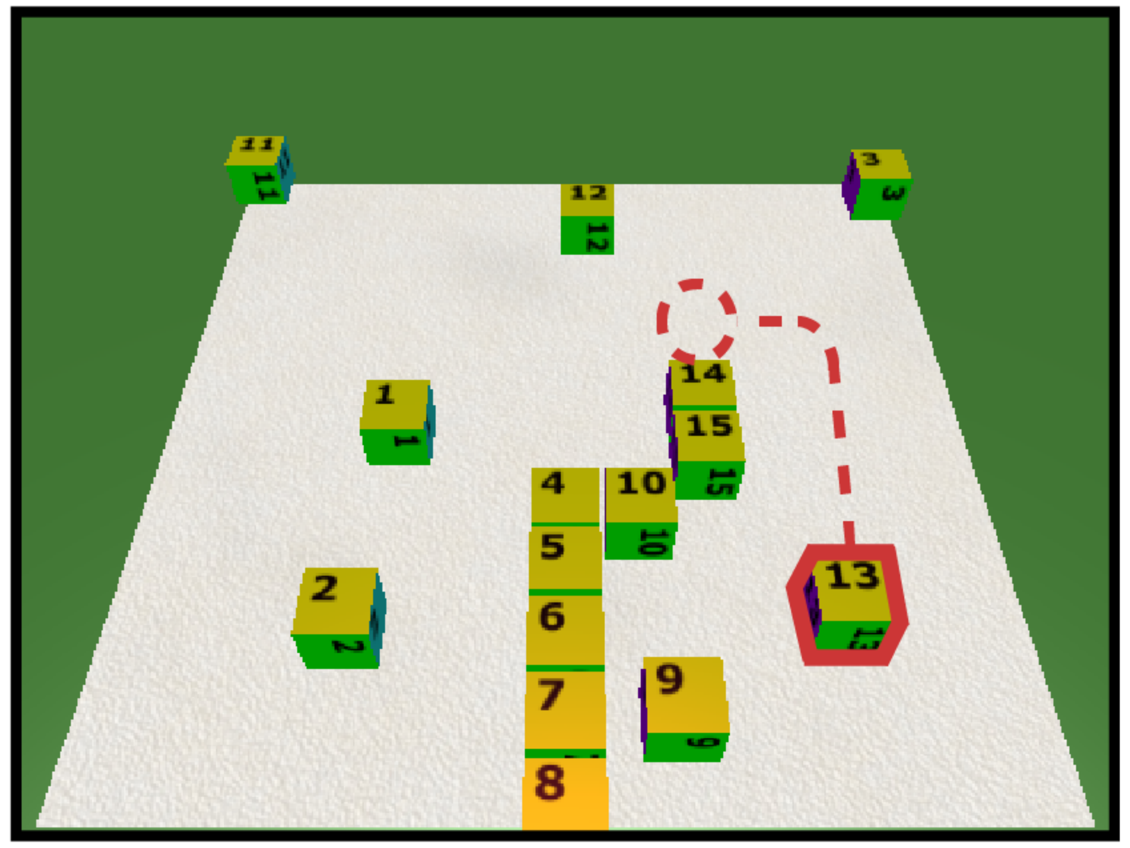
1. project input queries onto the set of synthetic sentences

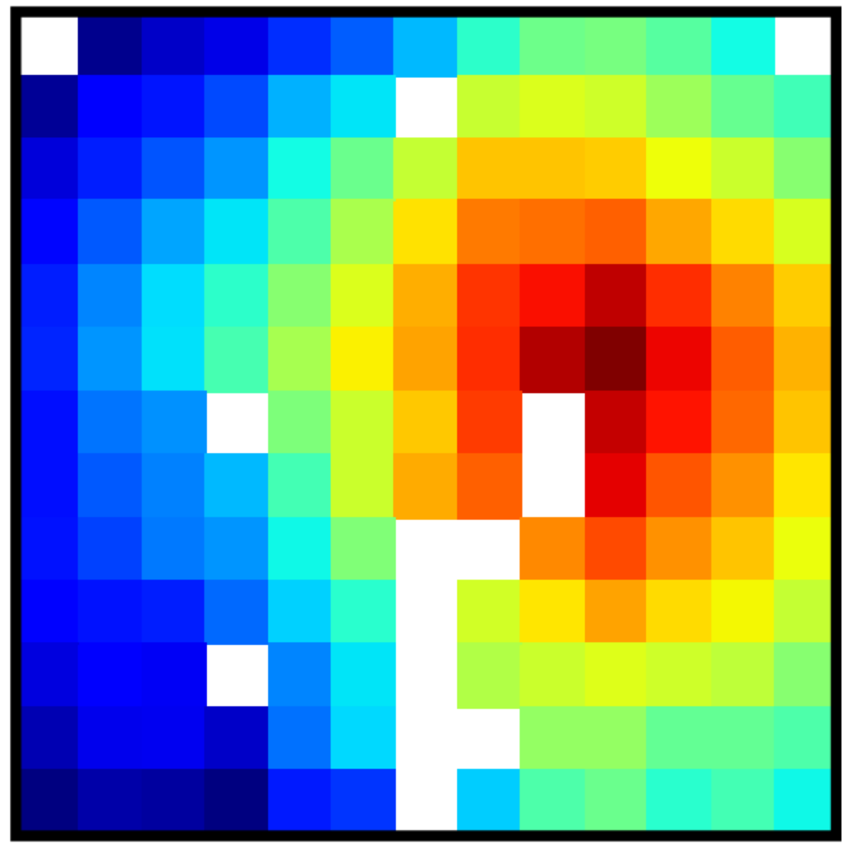


2. interpret with a model trained only on synthetic data

[Marzoev et al '20]

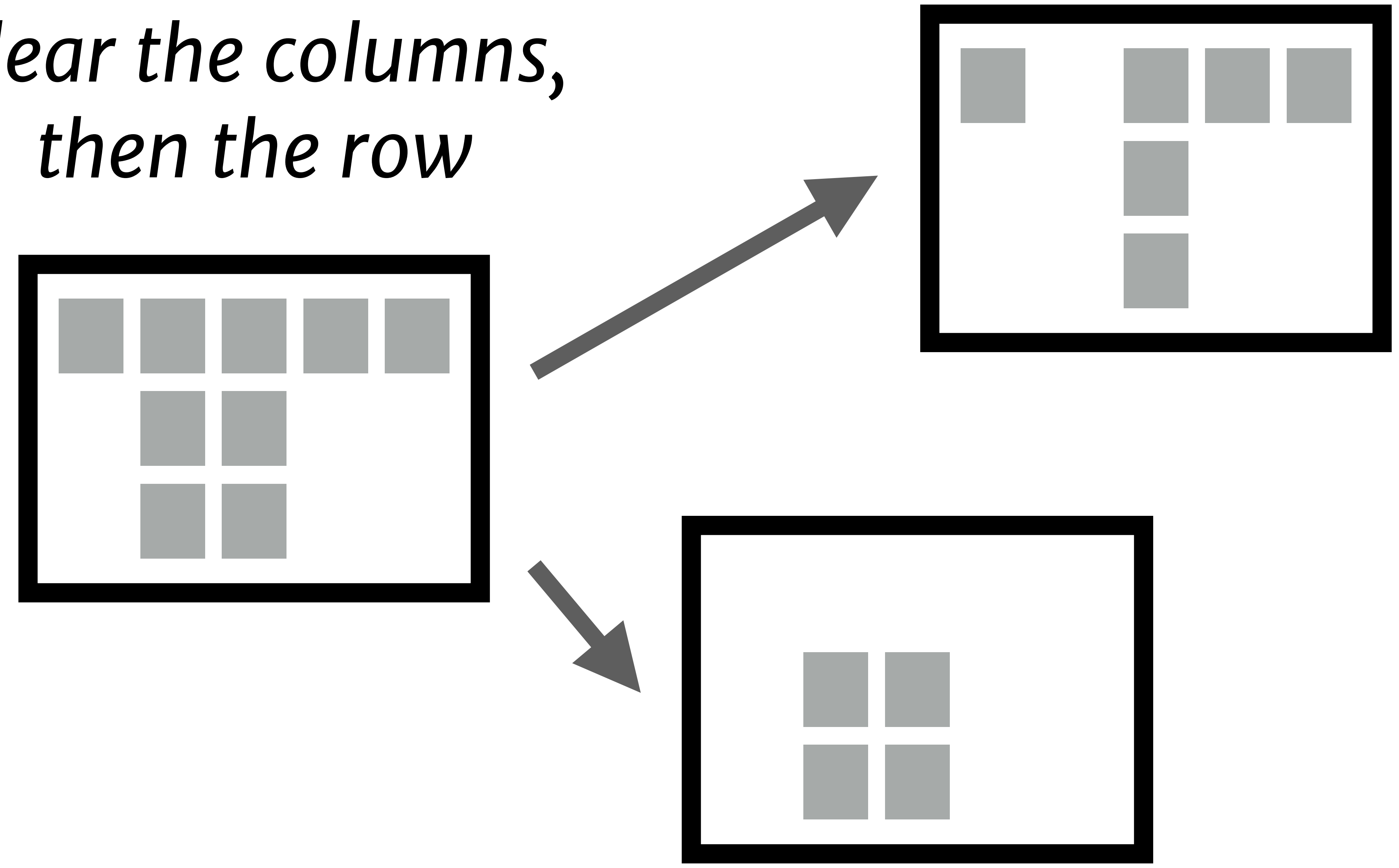
Neural planning



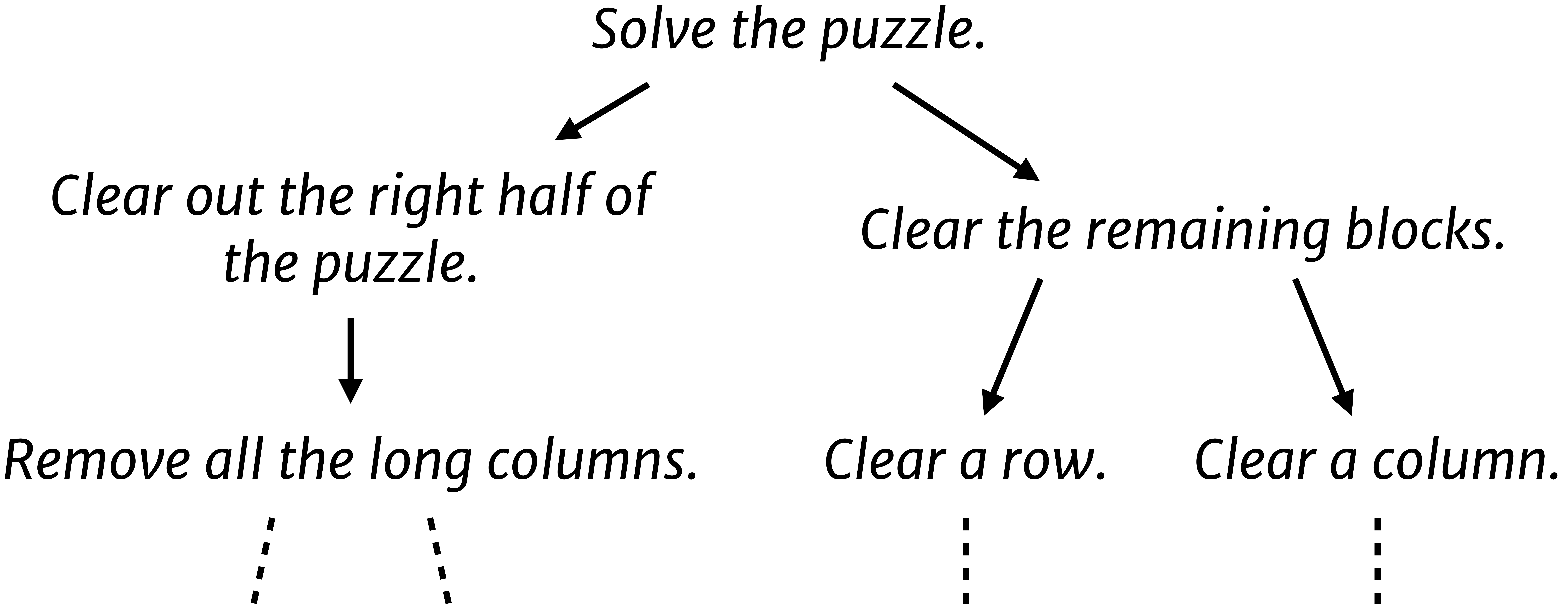


Take block 13 and place it directly above block 14 so they are almost touching.

Clear the columns, then the row



Natural language subgoals



Conclusions

Instruction following ⇔ policy learning

But need to think carefully about state tracking, planning, compositionality

Instruction following ⇒ other tasks

Language generation, machine teaching, structured exploration

Challenges

Better data efficiency, smarter inference

References

Branavan et al. *Reinforcement learning for mapping instructions to actions.* ACL 2009.

Mei et al. *Listen, attend, and walk: neural mapping of navigational instructions to action sequences.* AAAI 2016.

Tellex et al. *Understanding natural language commands for robotic navigation and mobile manipulation.* NCAI 2011.

Andreas & Klein. *Alignment-based compositional semantics for instruction following.* EMNLP 2016.

Andreas et al. *Modular multitask reinforcement learning with policy sketches.* ICML 2017.

Fried et al. *Unified pragmatic models for generating and following instructions.* NAACL 2018.

Co-Reyes et al. *Guiding policies with language via meta-learning.* ICLR 2019.

Andreas et al. *Learning with latent language.* NAACL 2018.

Marzoev et al. *Unnatural language processing: bridging the gap between synthetic and real language data.* Preprint.