Lazy Belief-Space Task and Motion Planning for Robots Acting in Partially Observable Environments

Caelan Reed Garrett

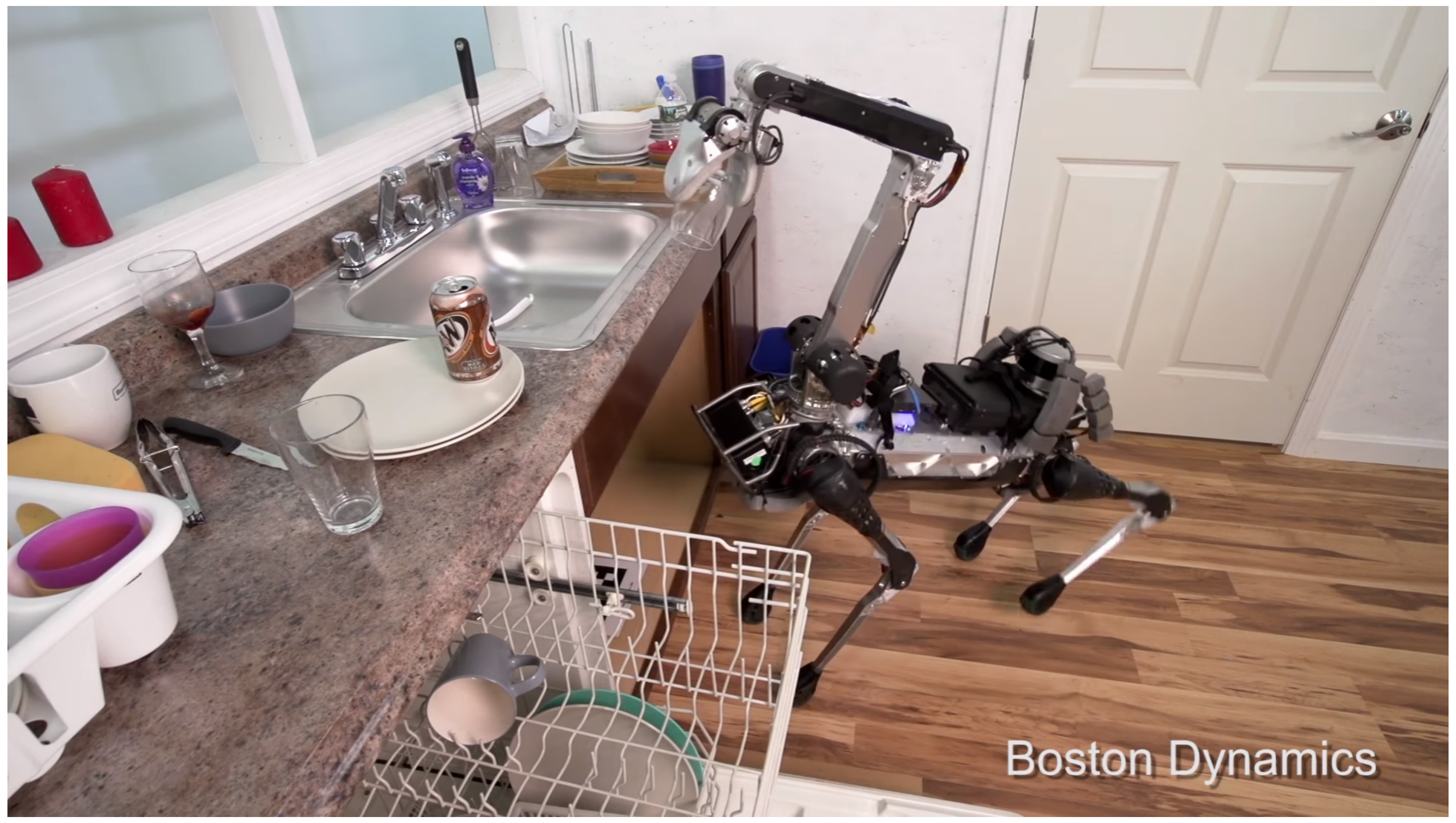
Advisors: Tomás Lozano-Pérez and Leslie Pack Kaelbling

01/24/2020 @ UW

web.mit.edu/caelan

Planning for Autonomous Robots

- Robot must select both **high-level** actions & **low-level** controls
- **Application areas:** semi-structured and human environments



Household

Warehouse fulfilment

Food service

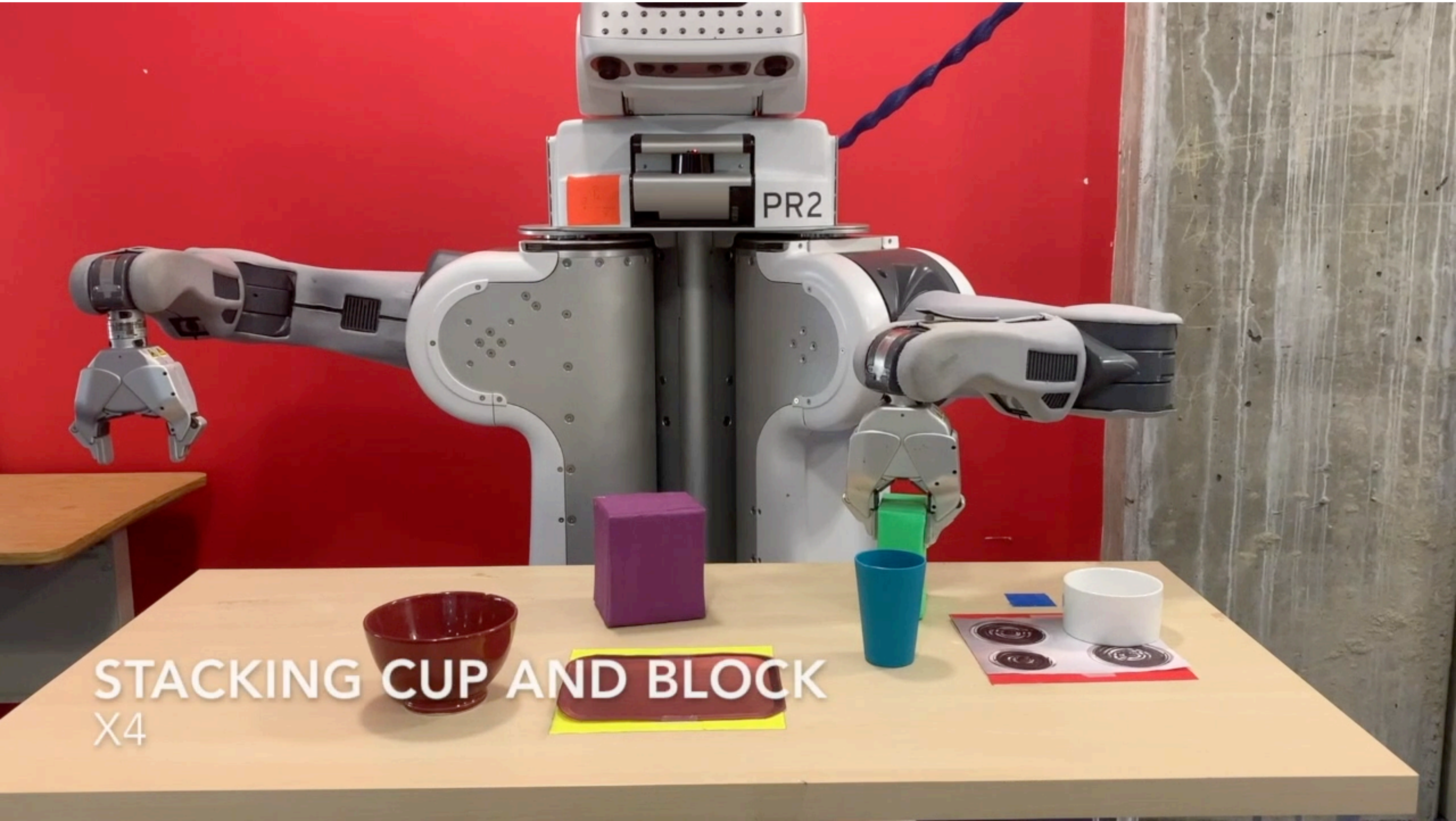
Construction

Task and Motion Planning (TAMP)

- Plan in a **high-dimensional** and **hybrid** space
- **Variables**
  - **Continuous**: robot configuration, object poses, door joint positions, …
  - **Discrete**: is-on, is-in-hand, is-holding-water, is-cooked, …
- **Actions:** move, pick, place, push, pull, pour, detect, cook, …

Manipulation: “Cooking”

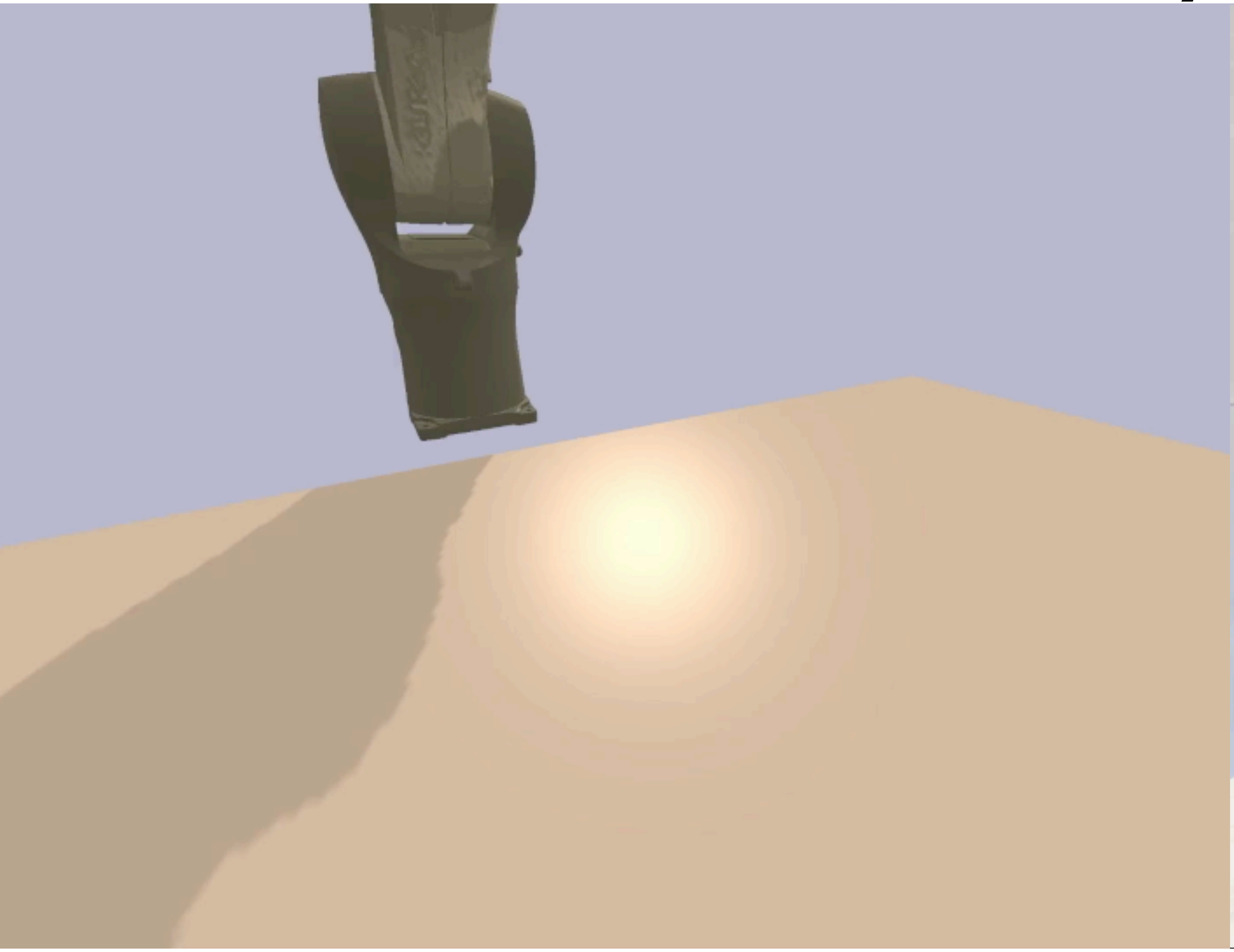
Manipulation: Stacking



Manipulation: Preparing “Coffee”

Automated Fabrication: Voronoi

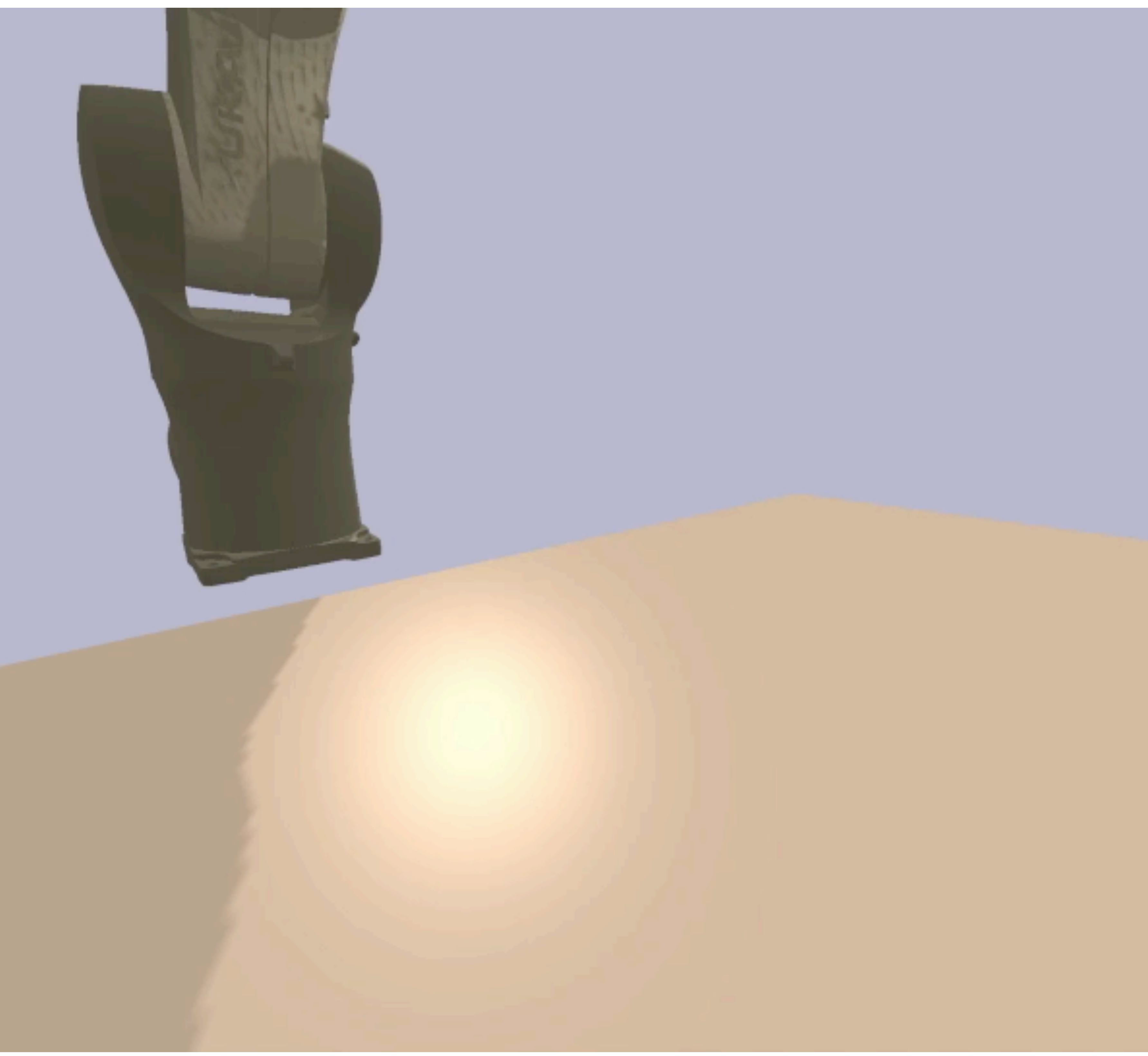
- Plan sequence of **306** 3D printing extrusions
- Collision, kinematic, **stability** and **stiffness** constraints

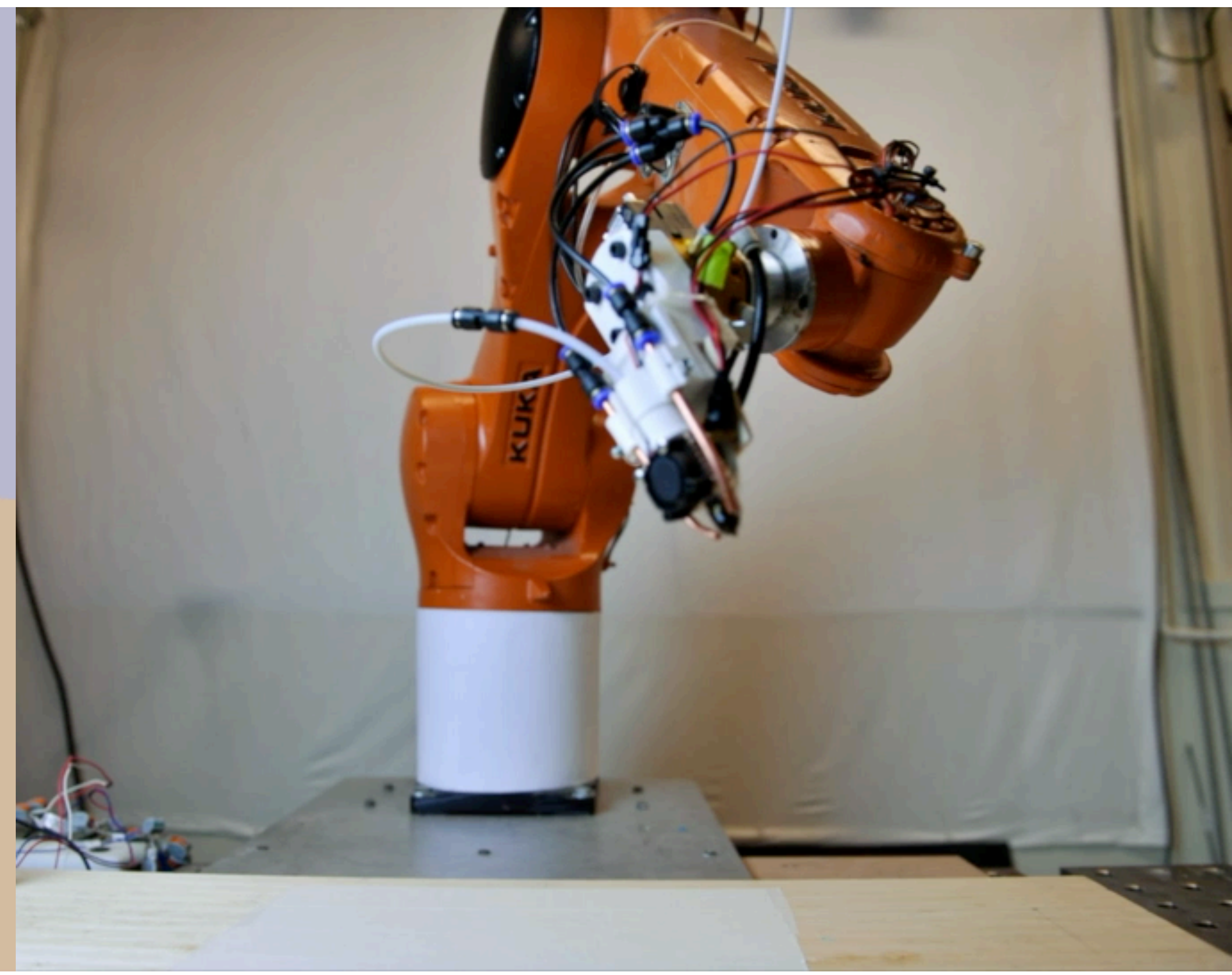


[Huang, Garrett, & Mueller 2018]

Automated Fabrication: Klein Bottle

- Plan sequence of **246** 3D printing extrusions
- Collision, kinematic, **stability** and **stiffness** constraints





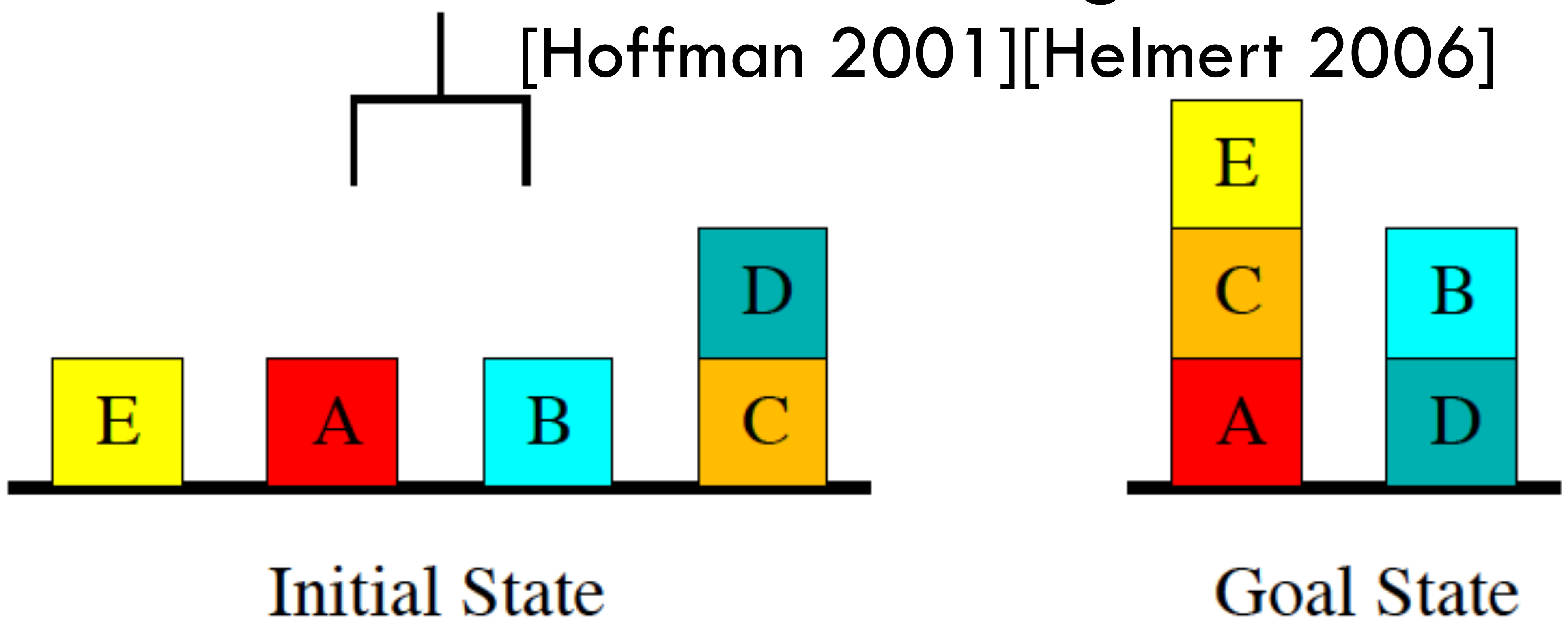
[Garrett, Huang, Lozano-Pérez, & Mueller TBA]

Background

Classical (Task) Planning

- Plan in a large **discrete** space with **many variables**
- **Planning languages:** STRIPS/PDDL [Fikes 1971] [Aeronautiques 1998]
  - **Facts:** boolean state variables
  - **Parameterized** actions
    - **Preconditions** test validity
    - **Effects** change the state
- **Heuristic search** algorithms [Hoffman 2001][Helmert 2006]

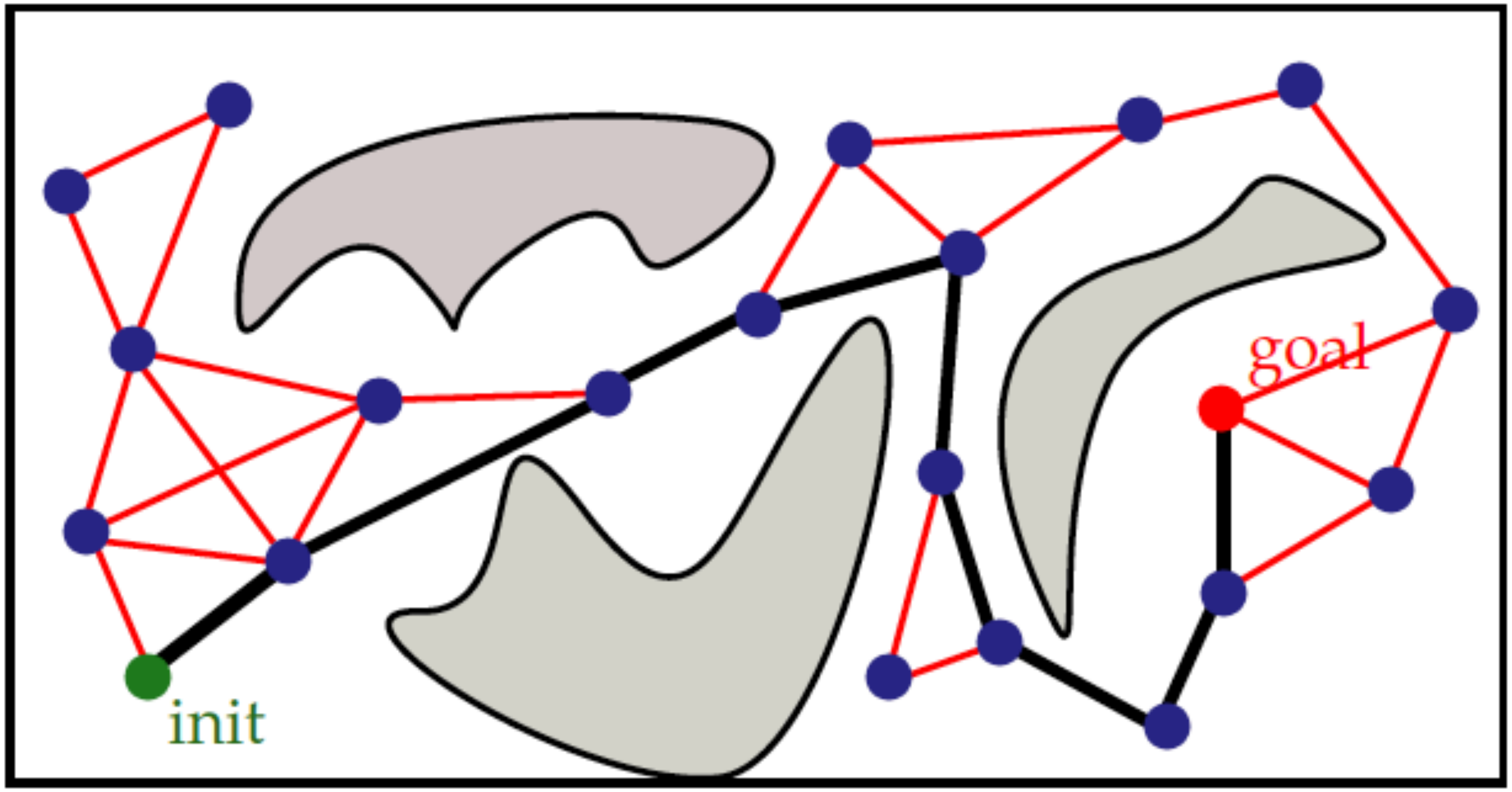
```
(:action stack
  :parameters (?b1 ?b2)
  :precondition (and
    (Holding ?b1)
    (Clear ?b2))
  :effect (and
    (HandEmpty)
    (On ?b1 ?b2)
    (not (Holding ?b1))
    (not (Clear ?b2))))
```



Sampling-Based Motion Planning

- Plan in a **continuous** configuration space
  1. **Sample** robot configurations (often randomly)
  2. **Connect** nearby configurations if collision-free path
  3. **Search** for a path within resulting graph

- Probabilistic Roadmap (**PRM**) [Kavraki 1994]
- **Lazy PRM** [Bohlin 2000]
- RRT [Kuffner 2000]
- RRT* [Karaman 2011]
- …



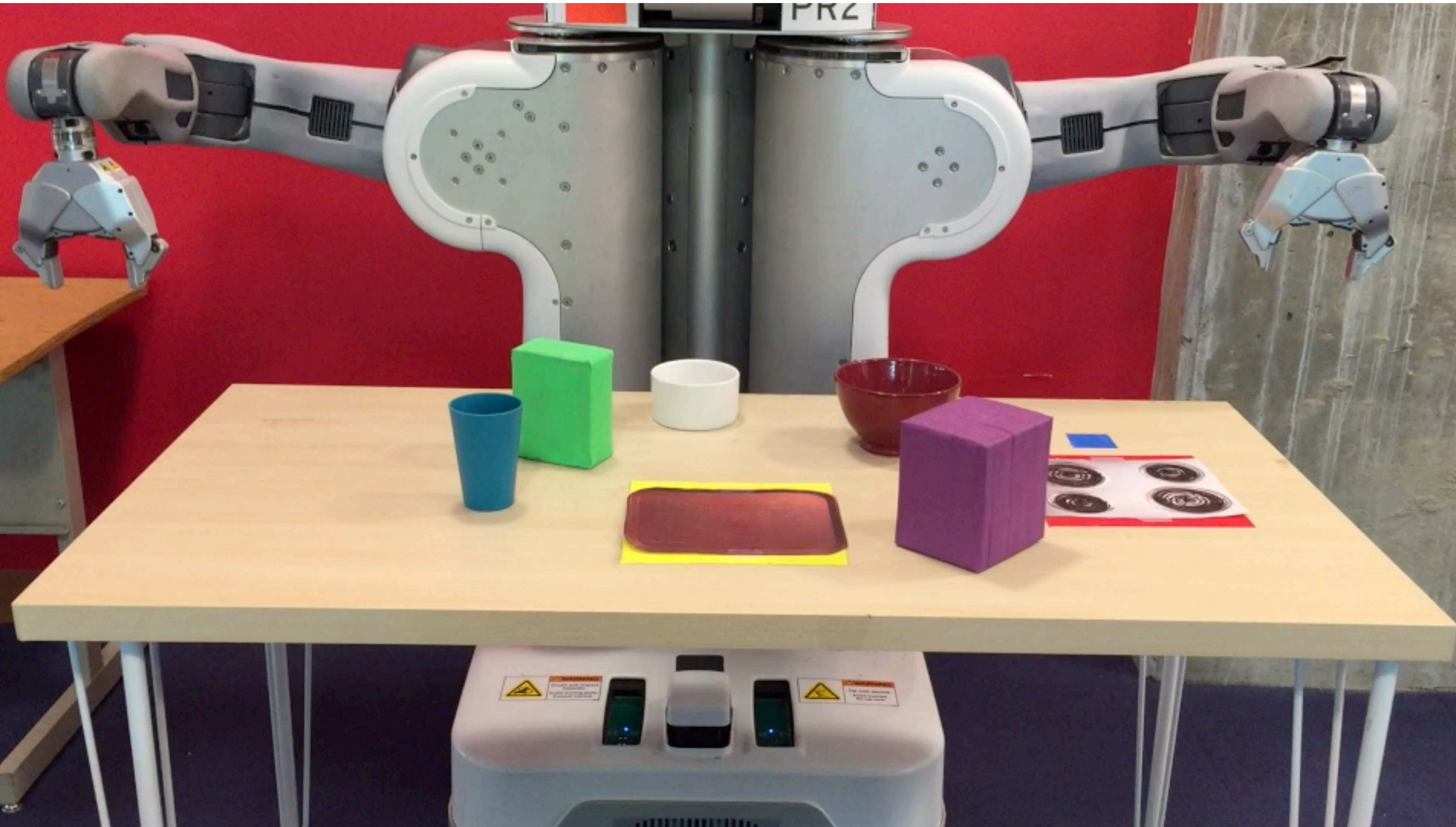
[Fig from Erion Plaku]

Geometric Constraints Affect Plan

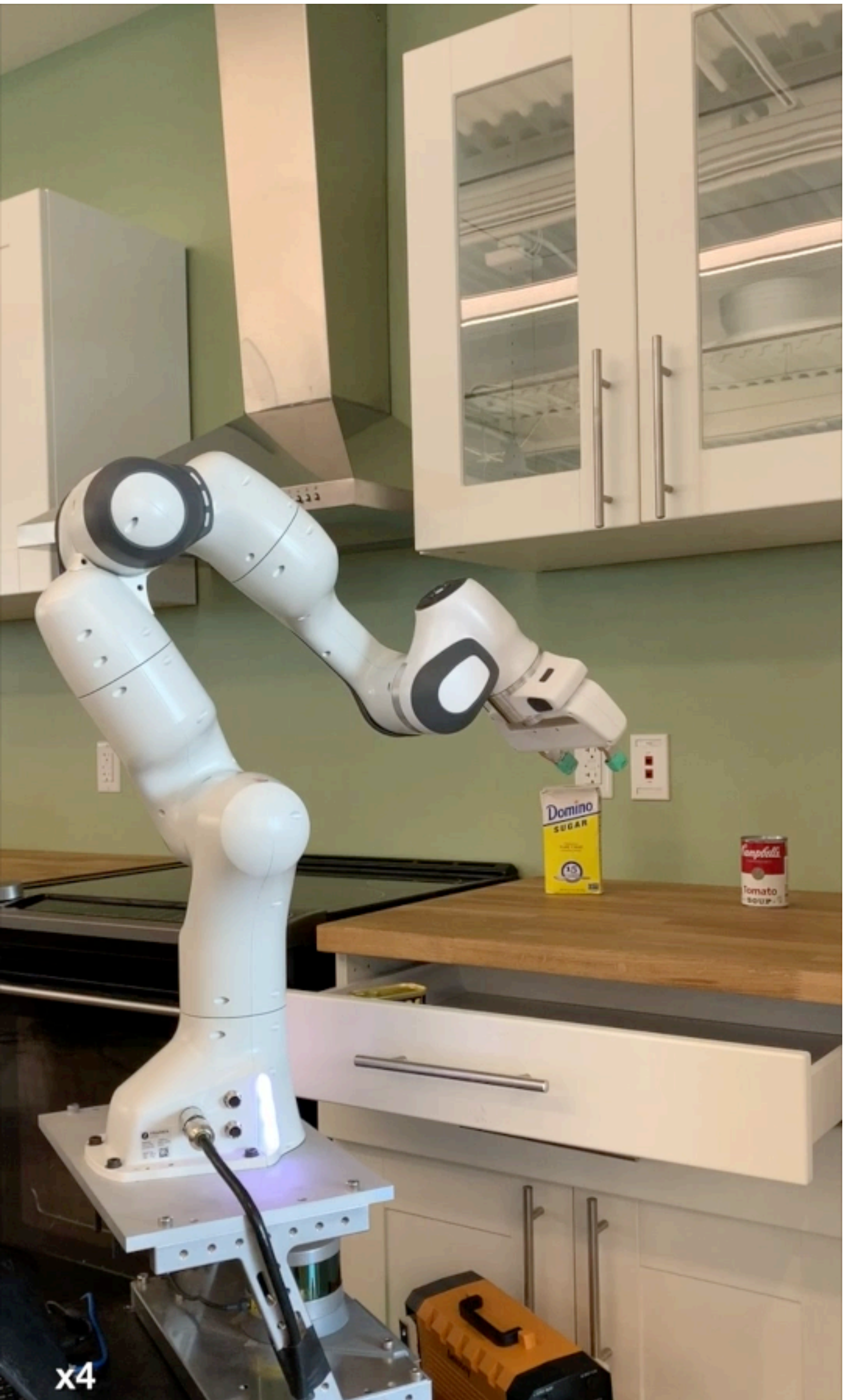
- **Inherits challenges** of both motion & classical planning
  - **High-dimensional, continuous** state-spaces
  - State-space **exponential** in number of variables
  - **Long horizons**

- Continuous constraints **limit high-level strategies**
  - Kinematics, reachability, joint limits, collisions grasp, visibility, stability, stiffness, torque limits, …

Pouring Among Obstacles



Spam in Left Cabinet & Doors Closed



- Physical constraints can be subtle!
- Robot forced to **regrasp** the object
  - Change from a **top** grasp to a **side** grasp
- **Non-monotonic** problem
  - Plan must **undo goals** to solve
  - **Open** then later **close** the cabinet door

Prior Work

- **Multi-Modal Motion Planning -** *Alami, Siméon et al., Hauser and Latombe, Plaku and Hager, Barry et al., Toussaint, Vega-Brown and Roy*
  - <u>Inefficient in high-dimensional state-spaces</u>
- **Semantic Attachments -** *Dornhege et al., Erdem et al., Lagriffoul et al., Dantam et al., Ferrer-Mestres et al.*
  - <u>Assumes an a priori state-space discretization</u>
- **Task & Motion Interface -** *Gravot et al., Cambon et al., Kaelbling and Lozano-Pérez, Srivastava et al., Garrett et al.*
  - <u>Inflexible to new domains</u>
- **No general-purpose, flexible framework** for modeling a variety of TAMP domains

Our Approach: PDDLStream

- Extends Planning Domain Description Language (**PDDL**)
  - Modular & **domain-independent**
- Enables the specification of **sampling procedures**
  - Can encode domains with **infinitely-many** actions

- Admits **generic** algorithms that operate using the samplers as **blackbox inputs**
  - The user only needs to specify the samplers
- **Probabilistically complete** when given samplers that densely sample the appropriate constraints

PDDLStream Language

[Garrett, Lozano-Pérez, Kaelbling 2020]

PDDL: Factored Action Language

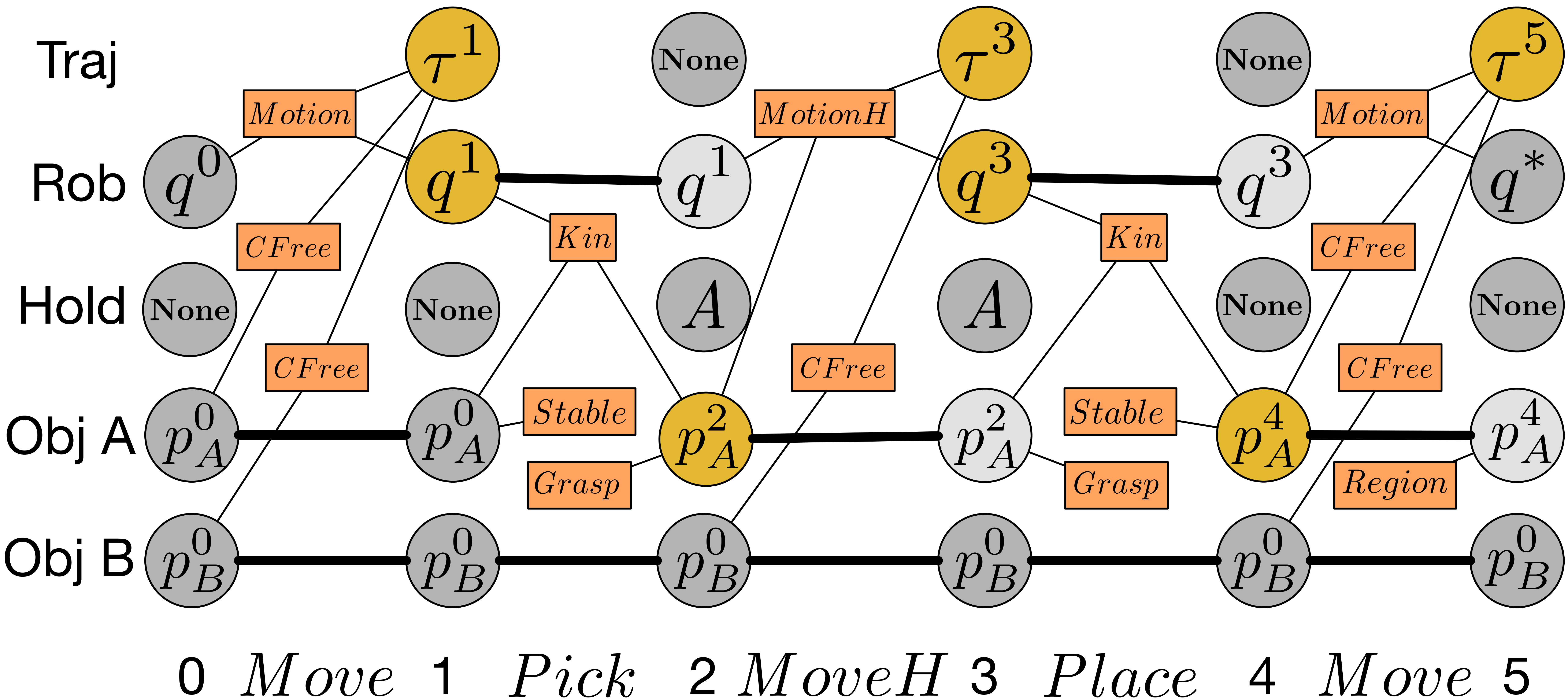
- Efficient discrete **search algorithms** that exploit logically **factored** state & action structure

- Actions encodes the **difference** between two states using preconditions & effects
  - Most variables are **unchanged**
  - Transitions can be described using **few parameters**

```
(:action stack
  :parameters (?b1 ?b2)
  :precondition (and (Holding ?b1 (Clear ?b2))
  :effect (and (HandEmpty) (On ?b1 ?b2)
                (not (Holding ?b1)) (not (Clear ?b2))))
```

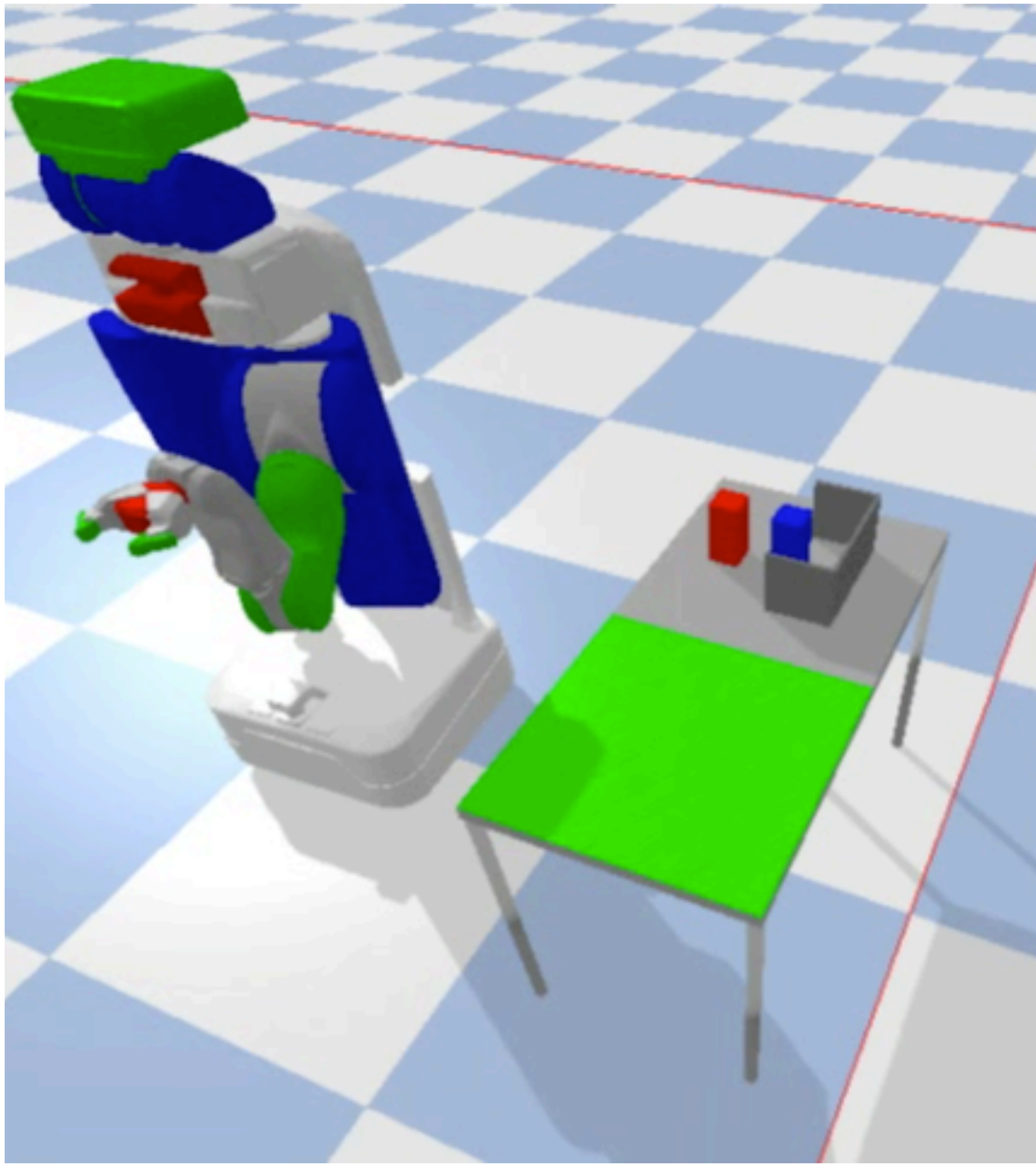
Hybrid Constraint Network

- **Sparse interactions** between variables
- Only 7 out of 29 parameters are **free parameters**



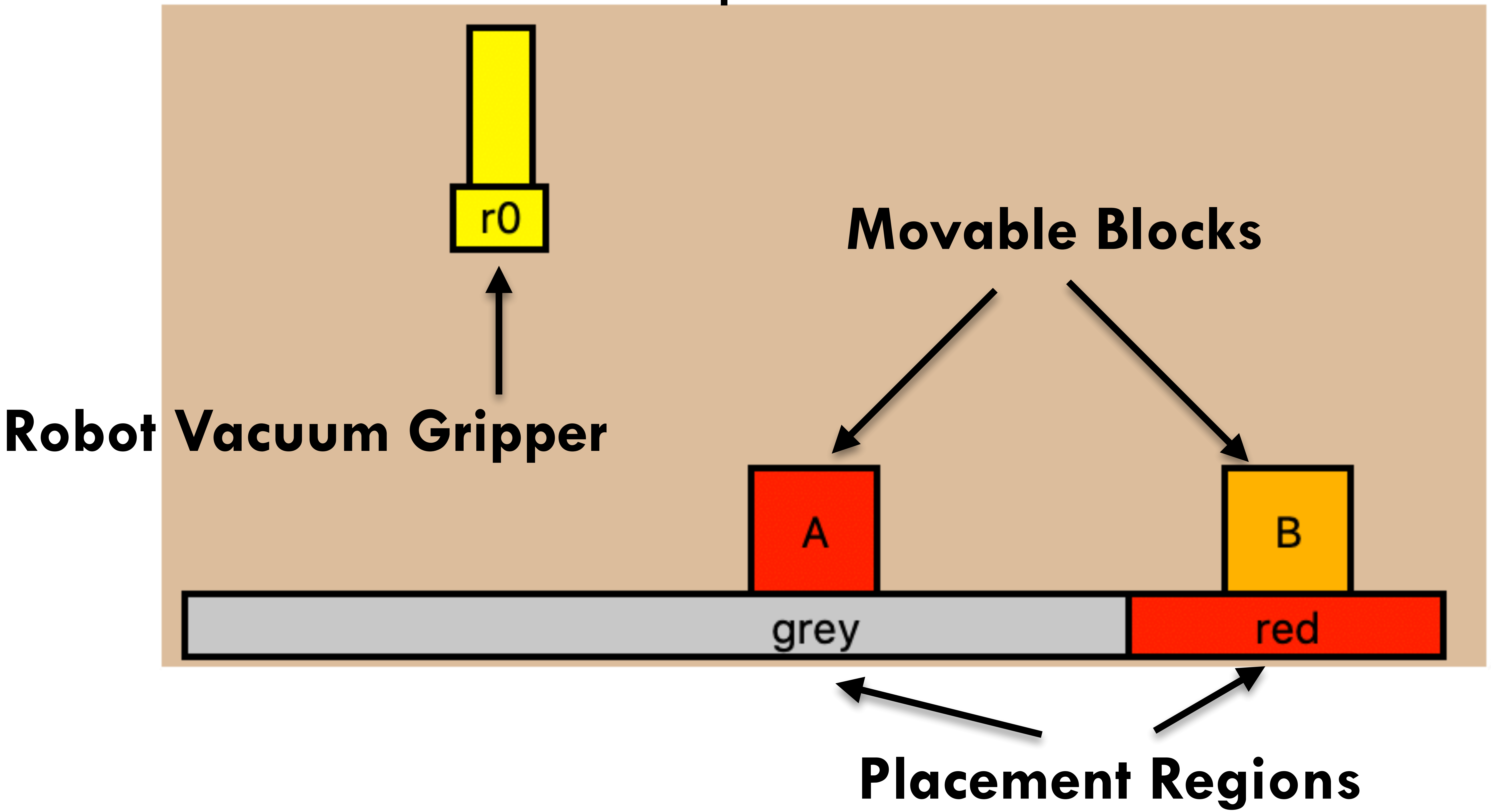
Motivating Pick & Place Example

- Single red object **prevents** a goal blue object from being grasped
- Focus on a compact **2D version**
- Formulation almost the same for 3D



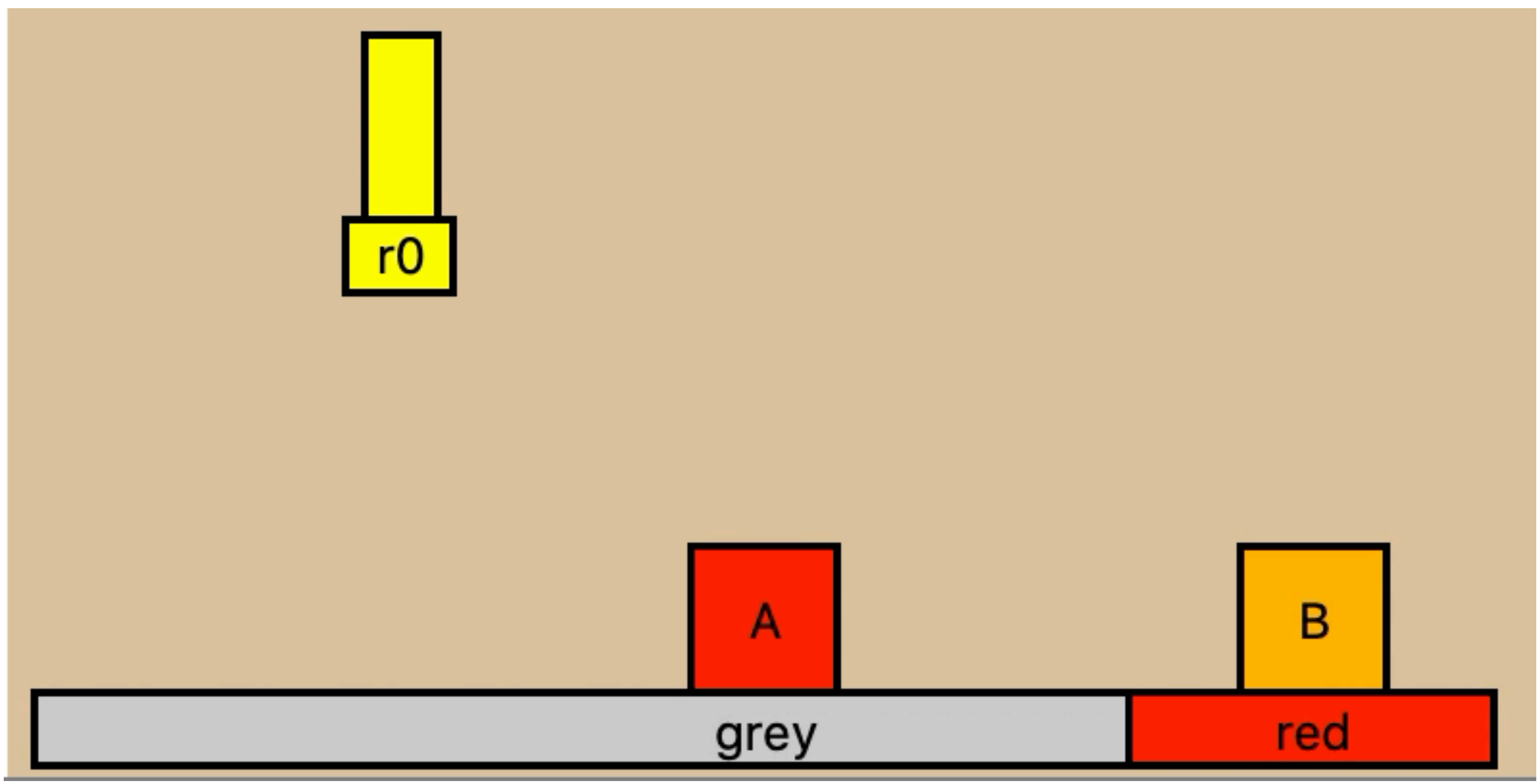
2D Pick-and-Place Example

- **Goal**: block **A** within the **red** region
- Robot and block poses are continuous [*x*, *y*] pairs
- Block **B** obstructs the placement of **A**



2D Pick-and-Place Solution

- One (of infinitely many) possible solutions
  - `move, pick` **`B`**`, move, place` **`B`**`,`
    `move, pick` **`A`**`, move, place` **`A`**



2D Pick-and-Place Initial & Goal

- Some constants are **numpy arrays**
- **Static initial facts** - value is **constant** over time
  - (Block, A), (Block, B), (Region, red), (Region, grey), (Conf, **[-7.5 5.]**), (Pose, A, **[0. 0.]**), (Pose, B, **[7.5 0.]**), (Grasp, A, **[0. -2.5]**), (Grasp, B, **[0. -2.5]**)
- **Fluent initial facts** - value **changes** over time
  - (AtConf, **[-7.5 5.]**), (HandEmpty), (AtPose, A, **[0. 0.]**), (AtPose, B, **[7.5 0.]**)
- **Goal formula:** `(exists (?p) (and (Contained A ?p red) (AtPose A ?p)))`

2D Pick-and-Place Actions

- Typical PDDL action description except that arguments are **high-dimensional** & **continuous**!
- To use the actions, must **prove** the following **static facts:**

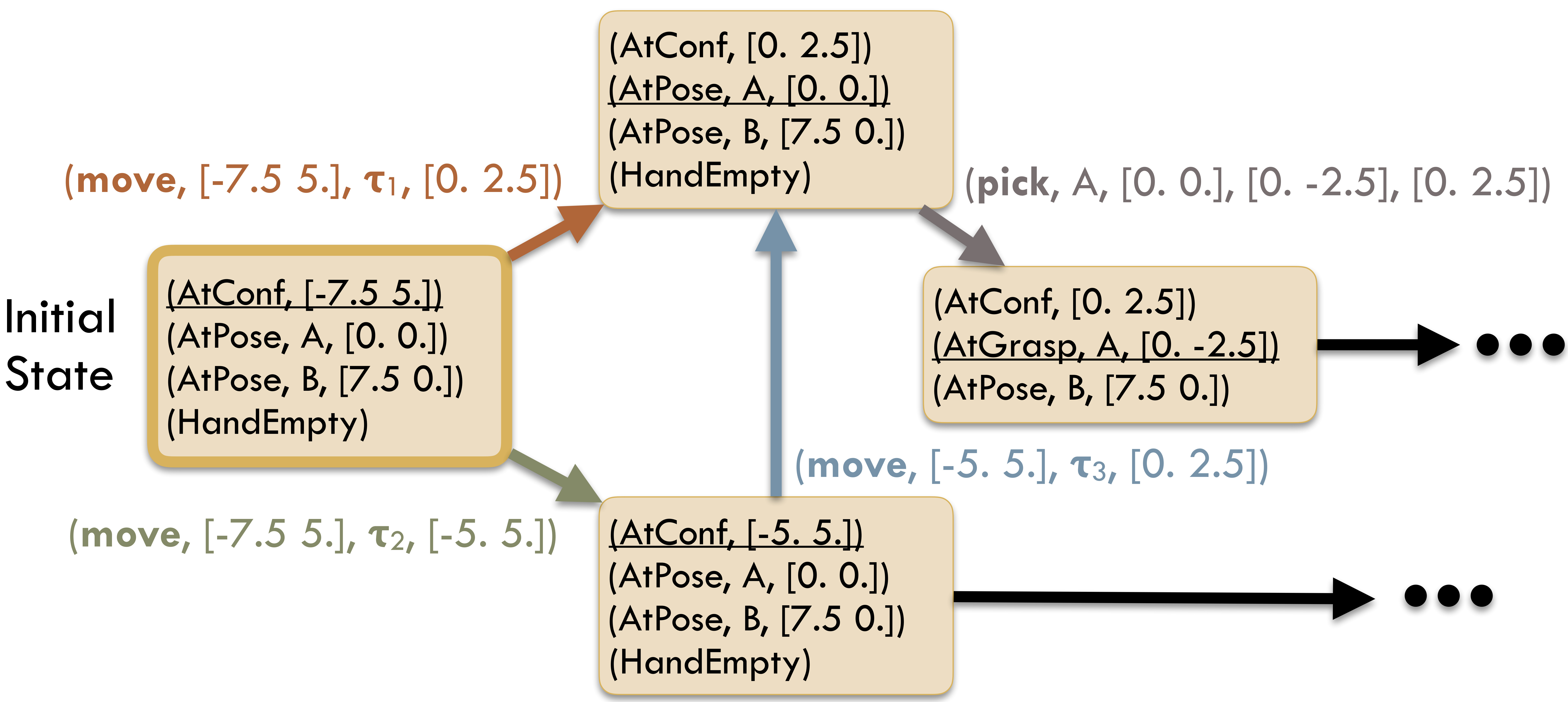
`(Motion ?q1 ?t ?q2), (Kin ?b ?p ?g ?q)`

```
(:action move
  :parameters (?q1 ?t ?q2)
  :precondition (and (Motion ?q1 ?t ?q2) (AtConf ?q1))
  :effect (and (AtConf ?q2) (not (AtConf ?q1))))

(:action pick
  :parameters (?b ?p ?g ?q)
  :precondition (and (Kin ?b ?p ?g ?q)
                     (AtConf ?q) (AtPose ?b ?p) (HandEmpty))
  :effect (and (AtGrasp ?b ?g)
               (not (AtPose ?b ?p)) (not (HandEmpty))))
```

BFS in Discretized State-Space

- Suppose we were **given** the following additional static facts:
  - (Motion, [-7.5 5.], τ_1, [0. 2.5]), (Motion, [-7.5 5.], τ_2, [-5. 5.]), (Motion, [-5. 5.], τ_3, [0. 2.5]), (Kin, A, [0. 0.], [0. -2.5], [0. 2.5]), …



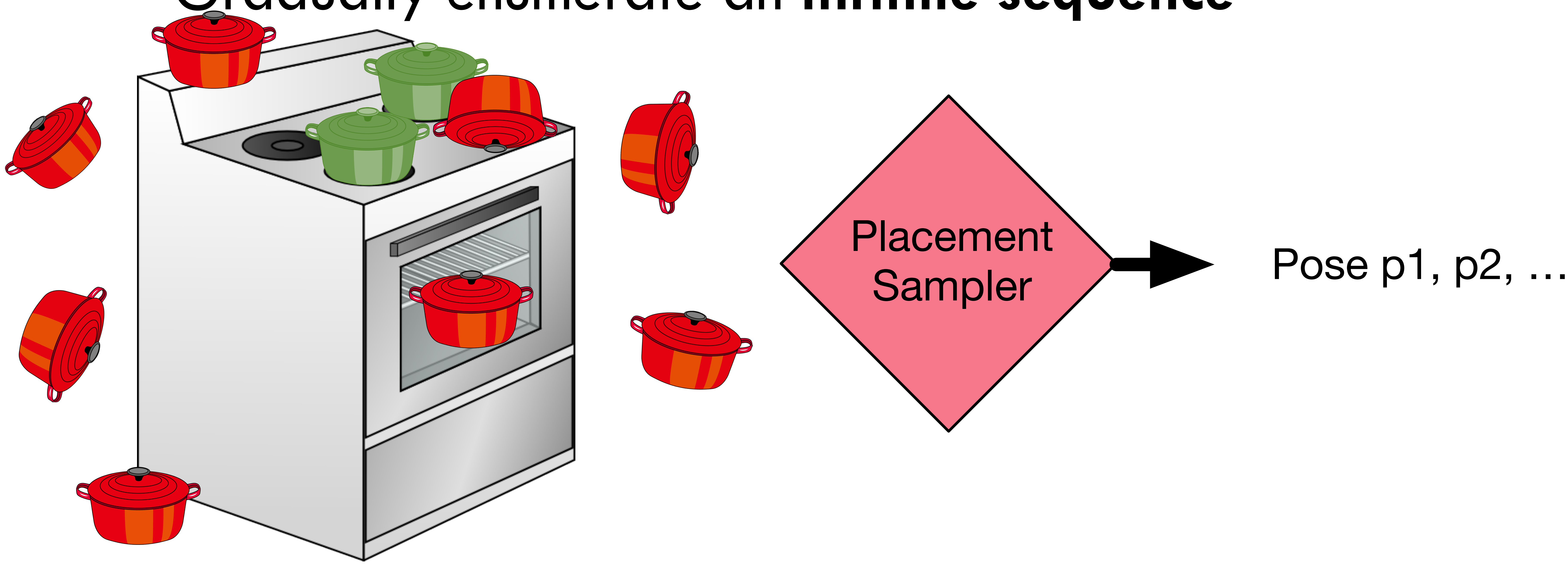
No a Priori Discretization

- **Values given at start:**
  - 1 initial configuration: (Conf, **[-7.5 5.]**)
  - 2 initial poses: (Pose, A, **[0. 0.]**), (Pose, B, **[7.5 0.]**)
  - 2 grasps: (Grasp, A, **[0. -2.5]**), (Grasp, B, **[0. -2.5]**)

- **Planner needs to find:**
  - 1 pose within a region: `(Contain A ?p red)`
  - 1 collision-free pose: `(CFree A ?p ? B ?p2)`
  - 4 grasping configurations: `(Kin ?b ?p ?g ?q)`
  - 4 robot trajectories: `(Motion ?q1 ?t ?q2)`

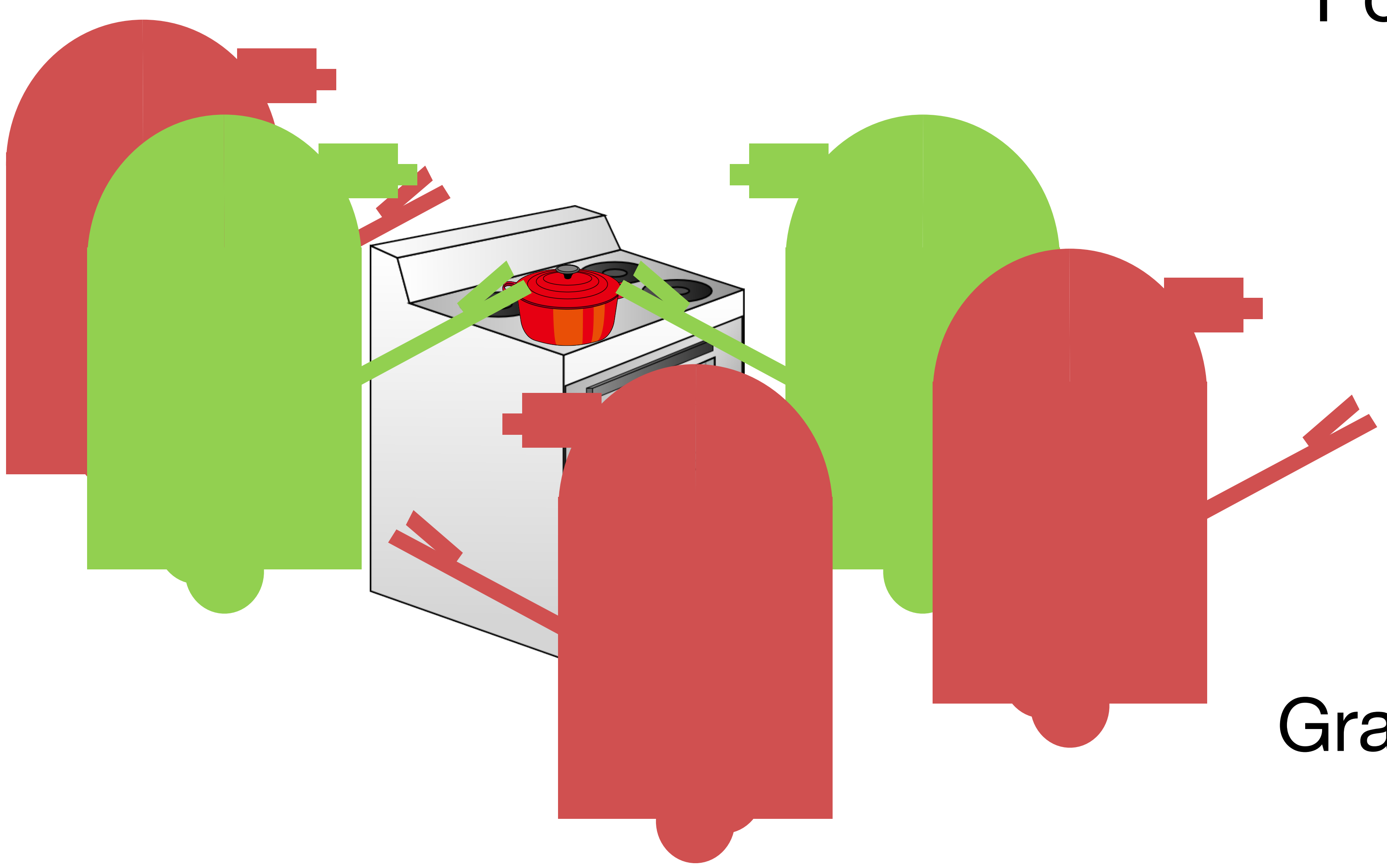
What Samplers Do We Need?

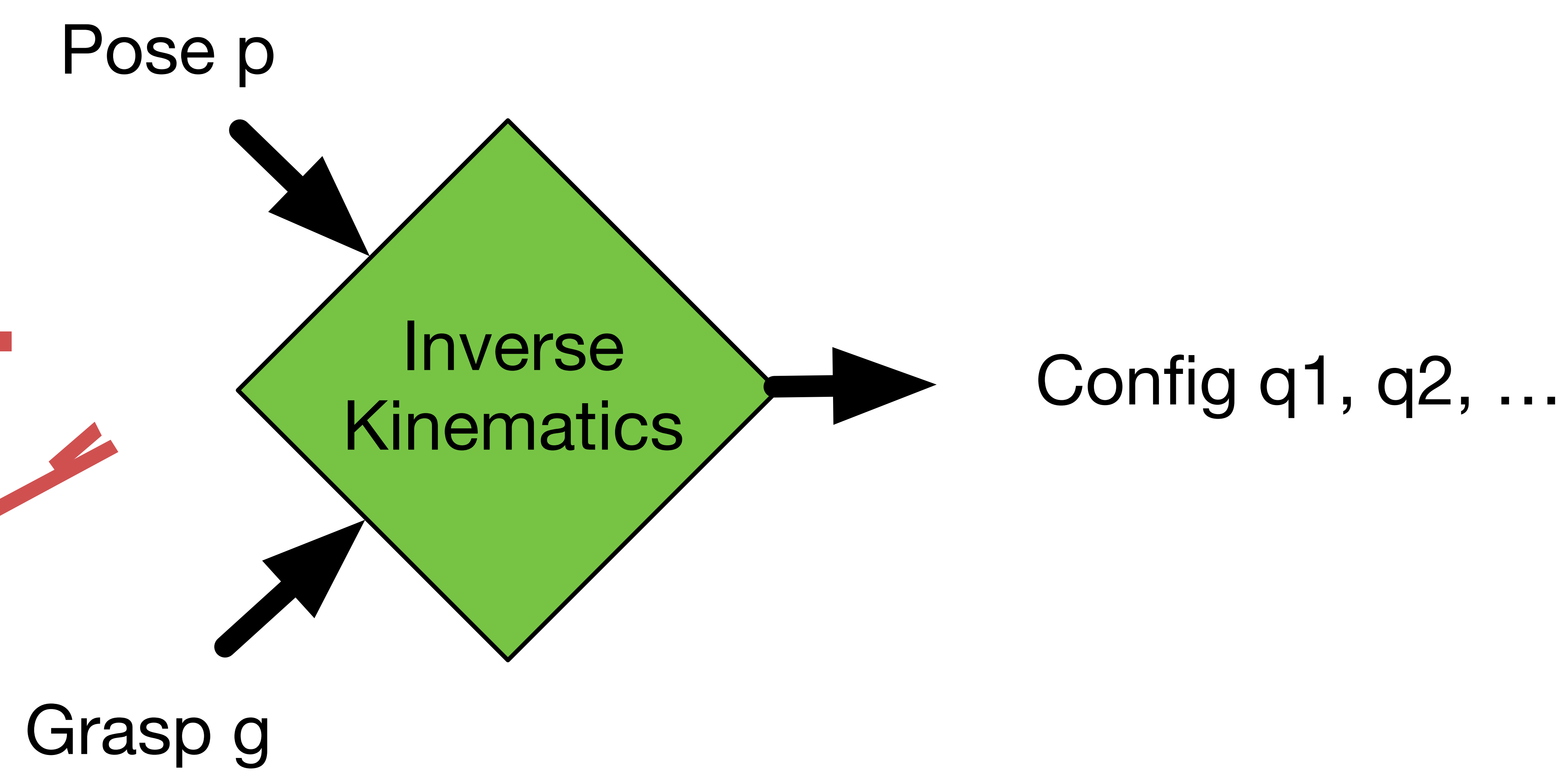
- **Low-dimensional** placement stability constraint (`Contain`)
  - *i.e.* 1D manifold embedded in 2D pose space
- Directly **sample values that satisfy the constraint**
- May need **arbitrarily many** samples
  - Gradually enumerate an **infinite sequence**



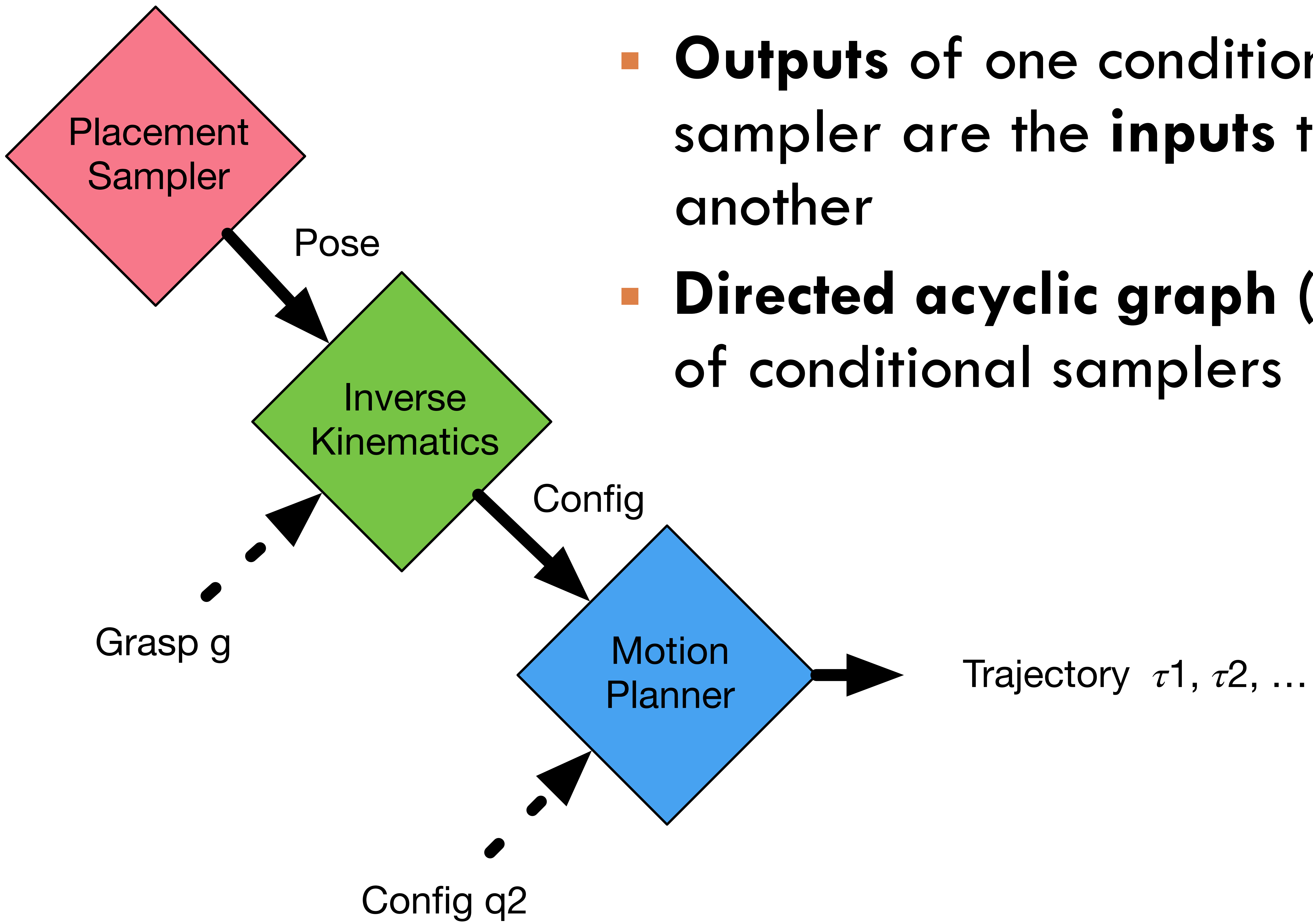
Intersection of Constraints

- **Kinematic constraint** (`Kin`) involves poses, grasps, and configurations
- **Conditional samplers** - samplers with inputs





Composing Conditional Samplers



- **Outputs** of one conditional sampler are the **inputs** to another
- **Directed acyclic graph (DAG)** of conditional samplers

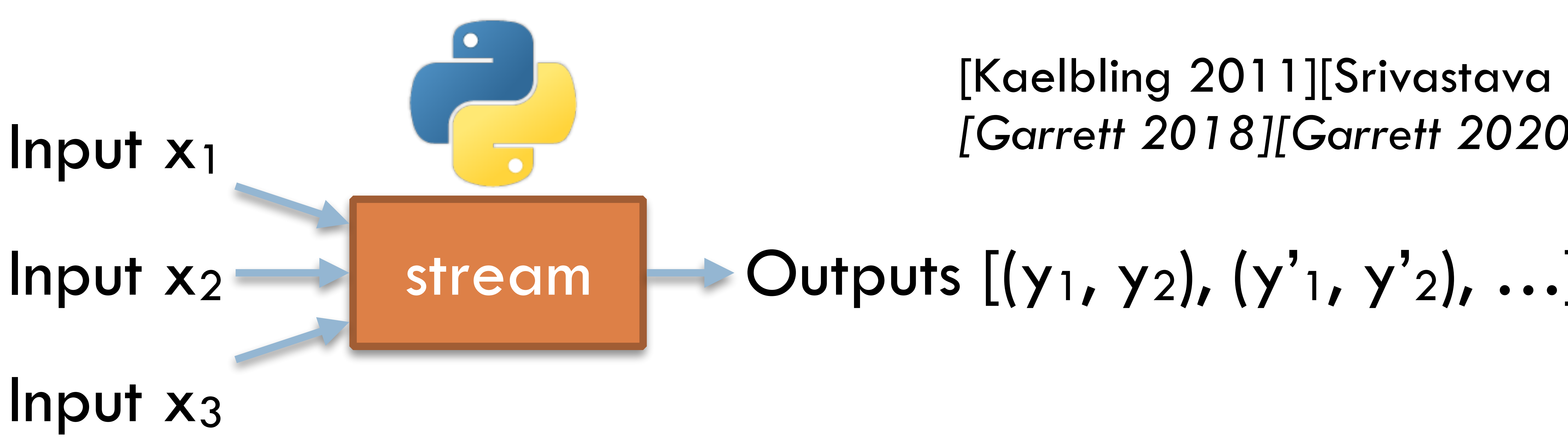
Stream: a function to a generator

- **Advantages**
  - Programmatic implementation
  - Compositional
  - Supports infinite sequences

```python
def stream(x1, x2, x3):
    i = 0
    while True:
        y1 = i*(x1 + x2)
        y2 = i*(x2 + x3)
        yield (y1, y2)
        i += 1
```

- **Stream** - function from an **input object tuple** (x_1, x_2, x_3) to a (potentially infinite) sequence of **output object tuples** $[(y_1, y_2), (y'_1, y'_2), \ldots]$

[Kaelbling 2011][Srivastava 2014]
[Garrett 2018][Garrett 2020a]



Stream Certified Facts

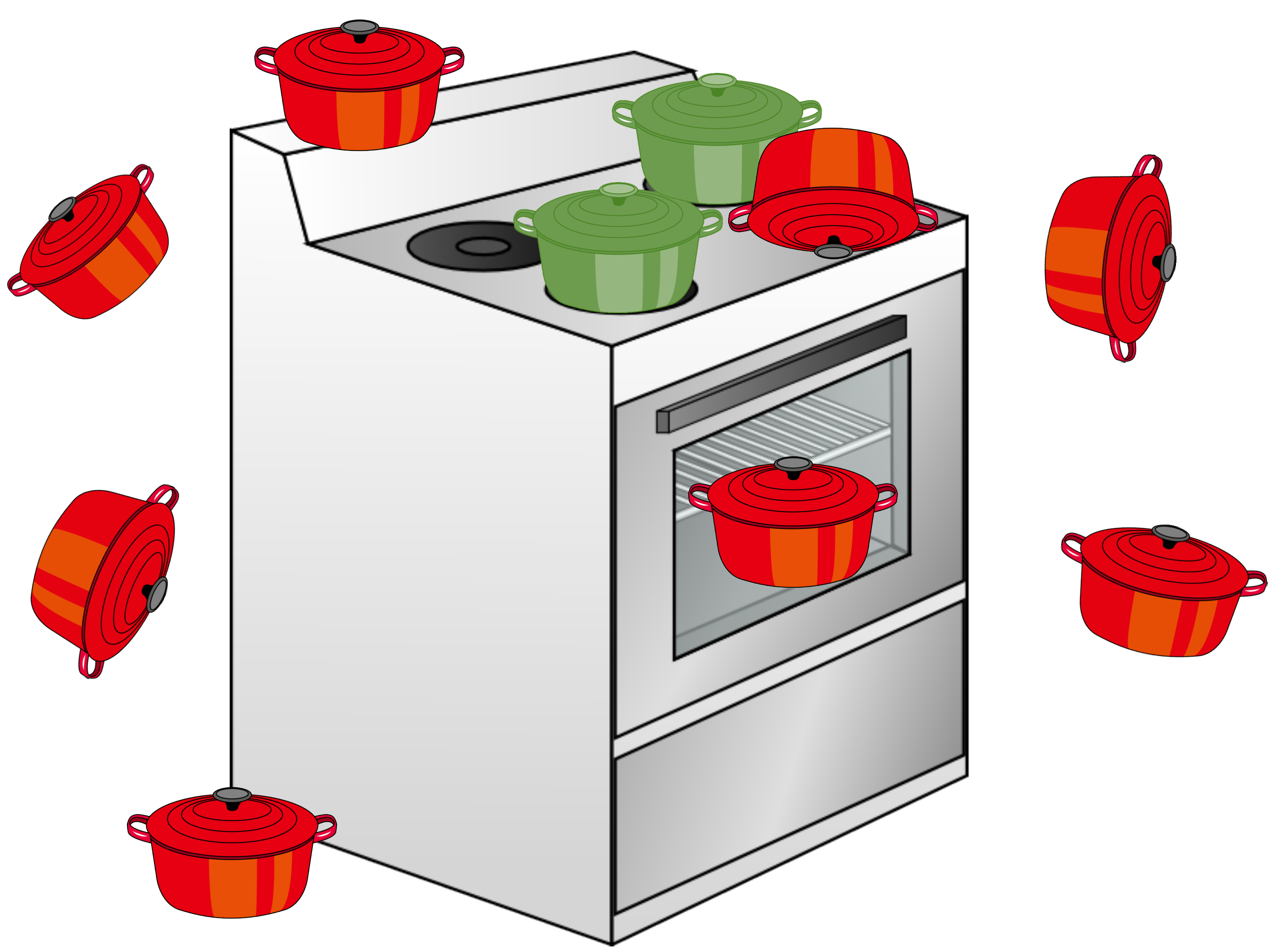
- Objects alone aren't helpful: **what do they represent?**
  - Communicate semantics using **predicates**!

- Augment stream specification with:
  - **Domain facts** - static facts declaring legal **inputs**
    - e.*g*. only configurations can be motion inputs

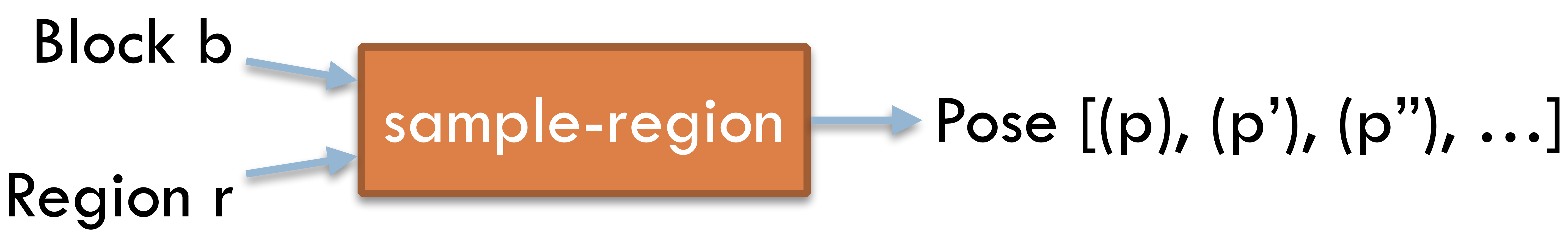
  - **Certified facts** - static facts that all **outputs** satisfy with their corresponding **inputs**
    - e.*g*. poses sampled from a region are within it

Sampling Contained Poses


```
(:stream sample-region
  :inputs (?b ?r)
  :domain (and (Block ?b) (Region ?r))
  :outputs (?p)
  :certified (and (Pose ?b ?p) (Contain ?b ?p ?r)))
```



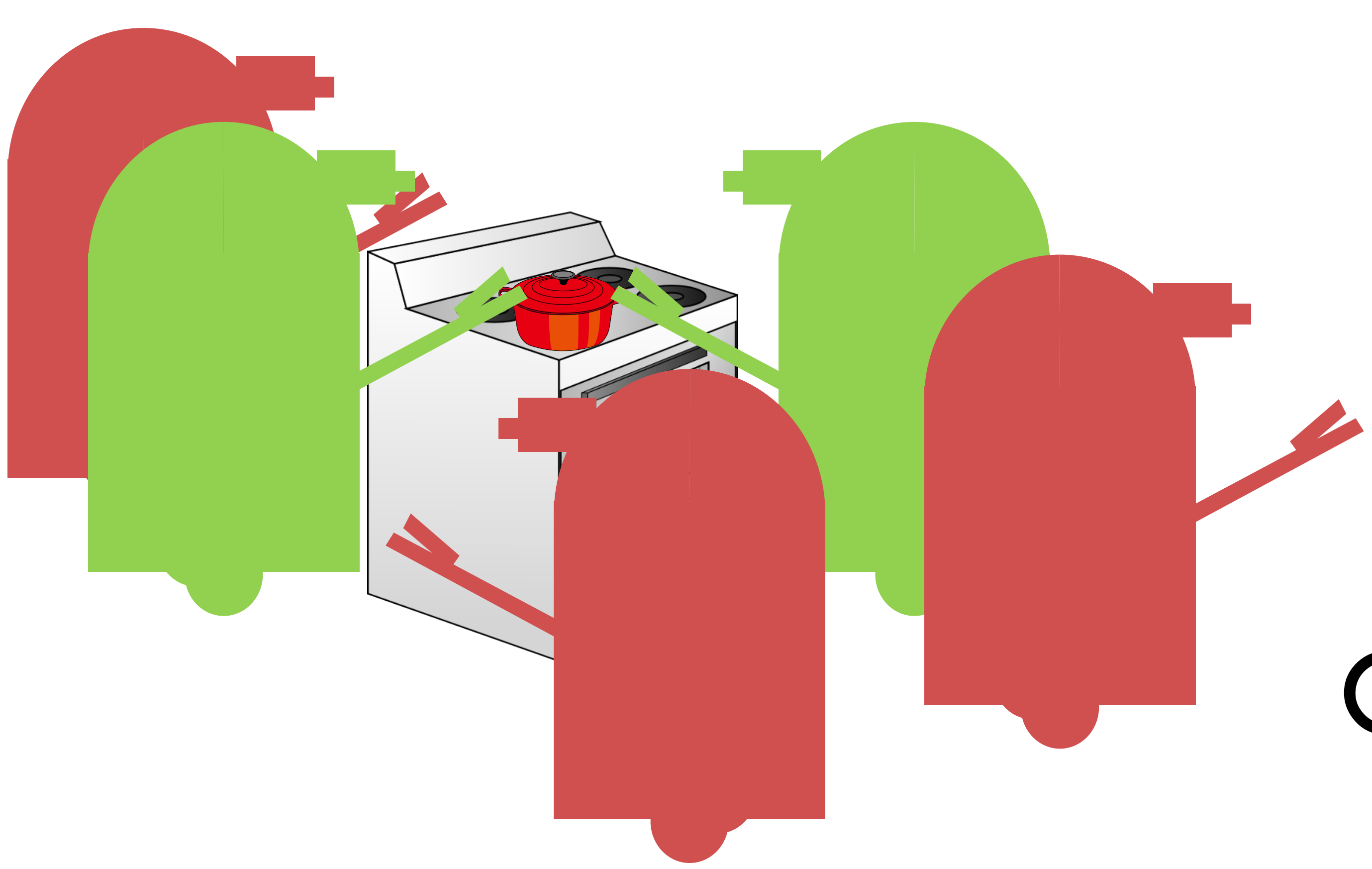
```python
def sample_region(b, r):
  x_min, x_max = REGIONS[r]
  w = BLOCKS[b].width
  while True:
      x = random.uniform(x_min + w/2,
                         x_max - w/2)
      p = np.array([x, 0.])
      yield (p,)
```

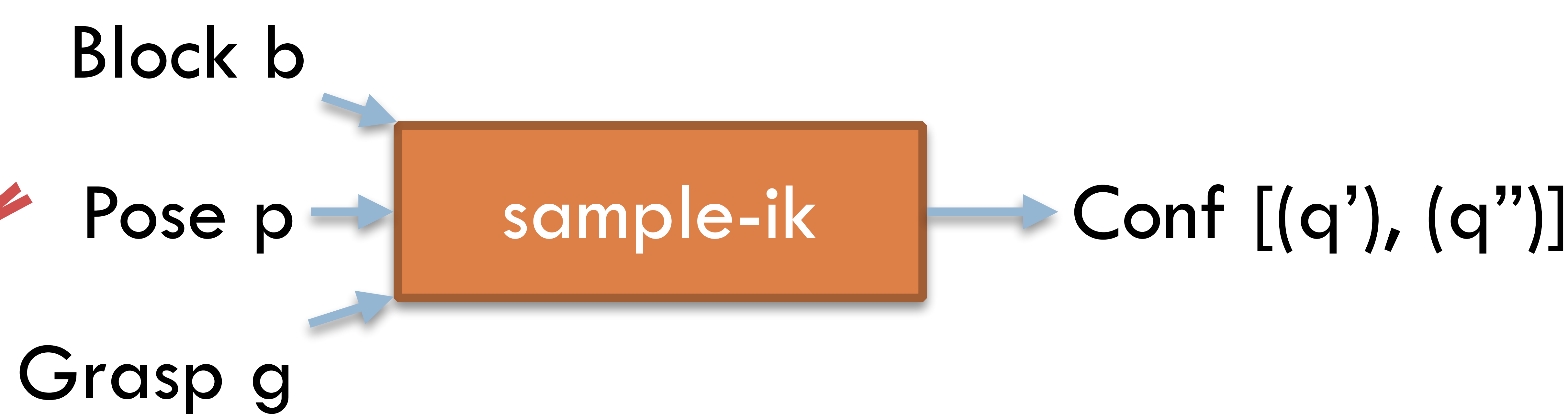


Sampling IK Solutions

- **Inverse kinematics** (IK) to produce robot grasping configuration
- Trivial in 2D, non-trial in general (e.g. 7 DOF arm)

```
(:stream sample-ik
  :inputs (?b ?p ?g)
  :domain (and (Pose ?b ?p) (Grasp ?b ?g))
  :outputs (?q)
  :certified (and (Conf ?q) (Kin ?b ?p ?g ?q)))
```





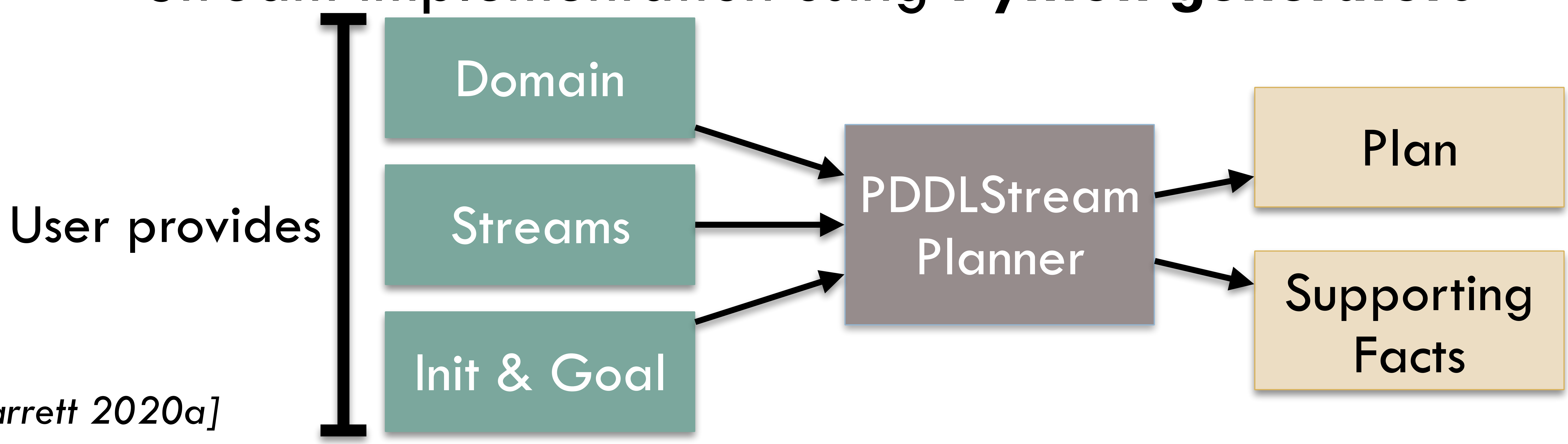
Calling a Motion Planner

- “Sample” **multi-waypoint trajectories**
- Use off-the-shelf motion planner (e.g. RRT)

```
(:stream sample-motion
    :inputs (?q1 ?q2)
    :domain (and (Conf ?q1) (Conf ?q2))
    :outputs (?t)
    :certified (and (Traj ?t) (Motion ?q1 ?t ?q2)))
```

PDDLStream = PDDL + Streams

- **Domain dynamics** (*domain.pddl*): *declares* actions
- **Stream properties** (*stream.pddl*)
  - Declares stream inputs, outputs, and certified facts
- **Problem** and **stream implementation** (*problem.py*)
  - Initial state, **Python constants**, & goal formula
  - Stream implementation using **Python generators**



[Garrett 2020a]

PDDLStream Algorithms

[Garrett, Lozano-Pérez, Kaelbling 2020]

Two PDDLStream Algorithms

- **PDDLStream planners decide** which streams to use
- Algorithms alternate between **searching & sampling:**
  1. **Search** a finite PDDL problem for plan
  2. **Modify** the PDDL problem (depending on the plan)

- Search implemented using **off-the-shelf algorithms**
  - **Off-the-shelf AI planner** - FastDownward
    - Exploits **factoring** in its search heuristics (*e.g.* h_{FF})

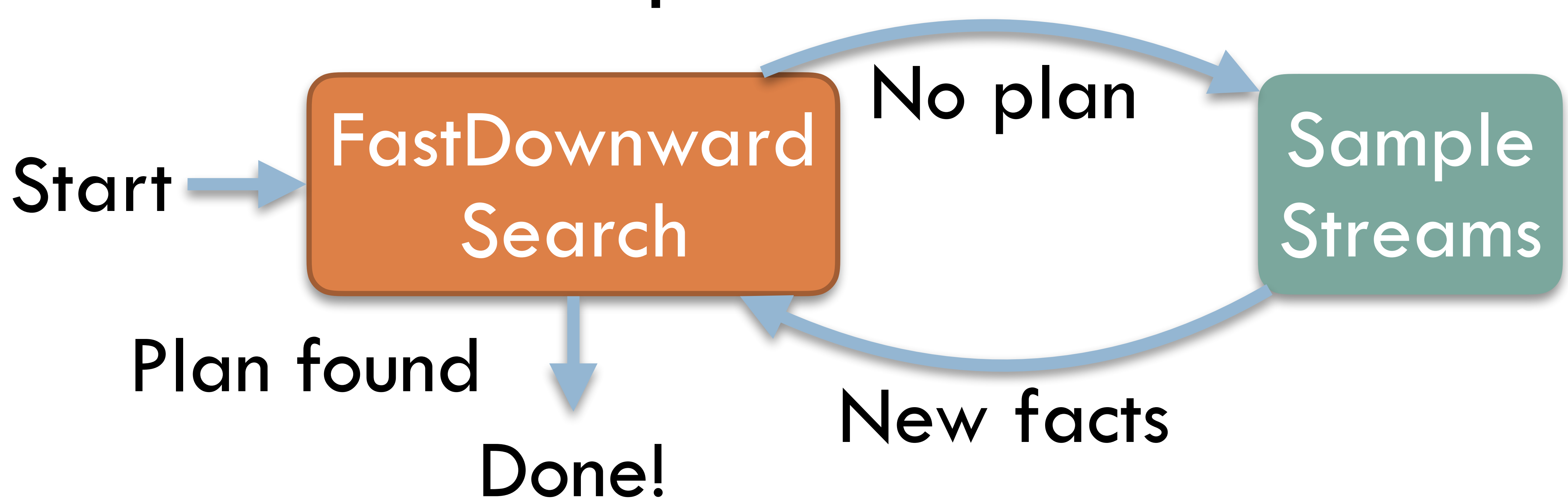
- **Probabilistically complete** given *sufficient* samplers

[Garrett 2018][Garrett 2020a]

Incremental Algorithm

- Incrementally construct all possible initial facts
- Periodically check if a solution exists
- Repeat:

1. **Compose** and **evaluate** a finite number of streams to unveil more facts in the initial state
2. **Search** the current PDDL problem for plan
3. **Terminate** when a plan is found

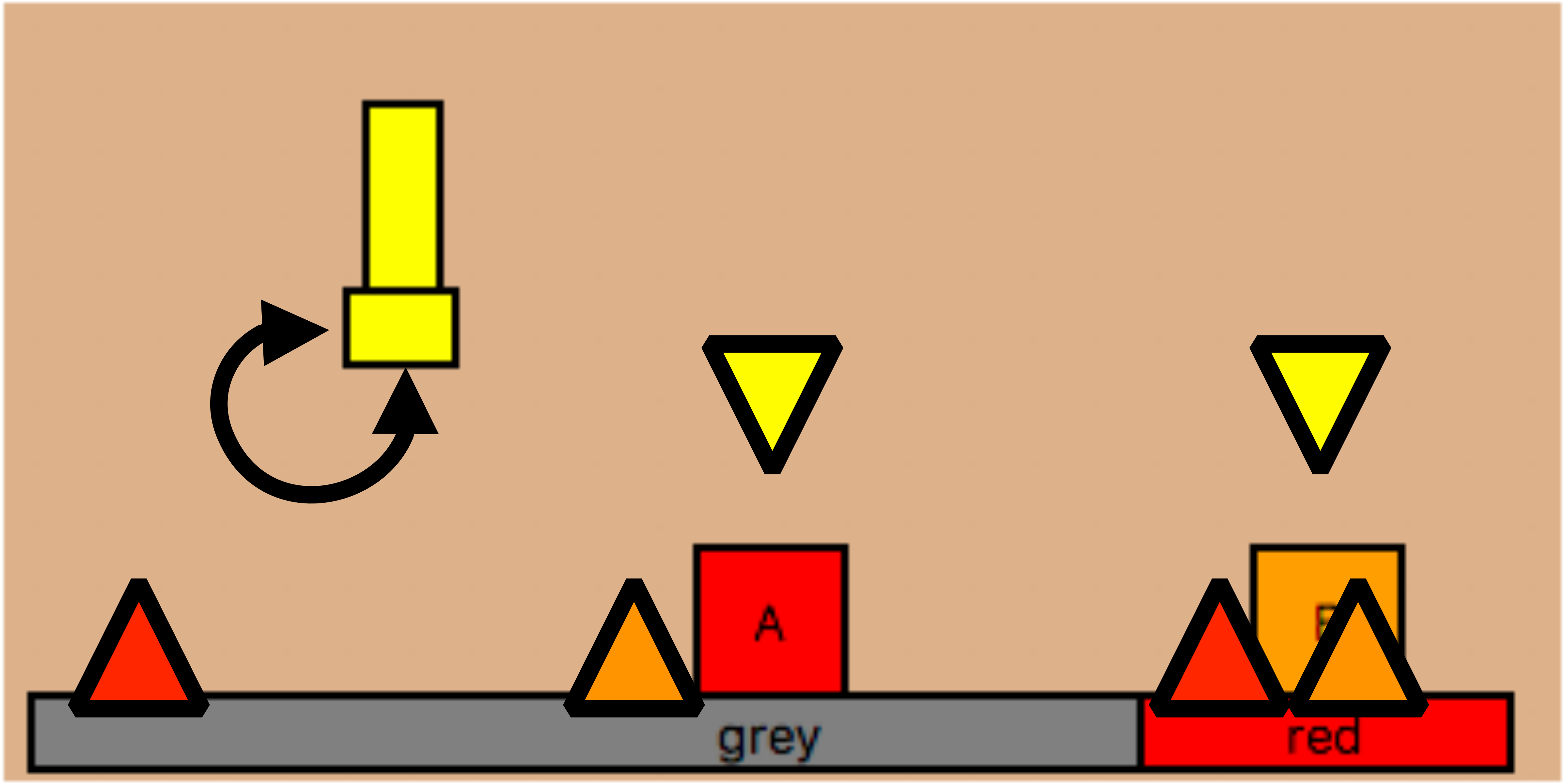


[Garrett 2018]
[Garrett 2020a]

Incremental: Sampling Iteration 1

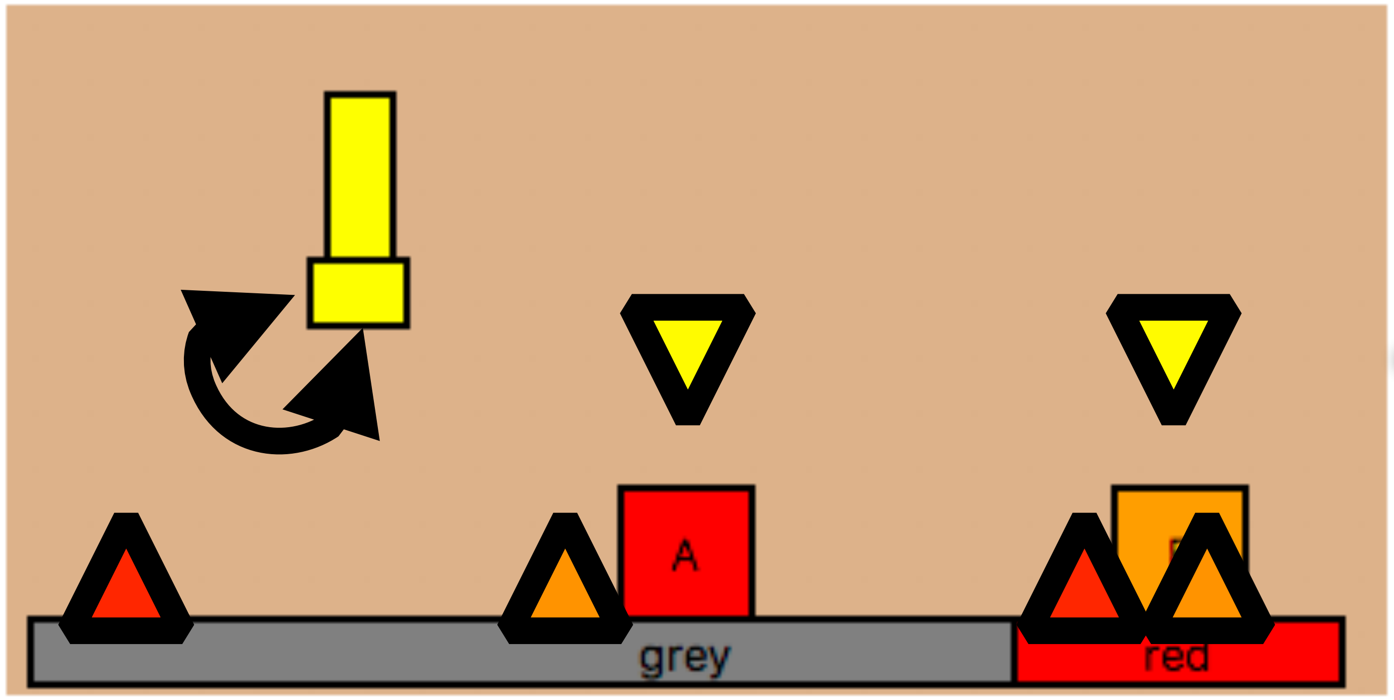
Iteration 1 - 14 stream evaluations

- **Sampled:**
  - 2 new robot configurations:
  - 4 new block poses:
  - 2 new trajectories:



Incremental: Search Iteration 1

- Pass current discretization to FastDownward
- If **infeasible**, the current set of samples is insufficient

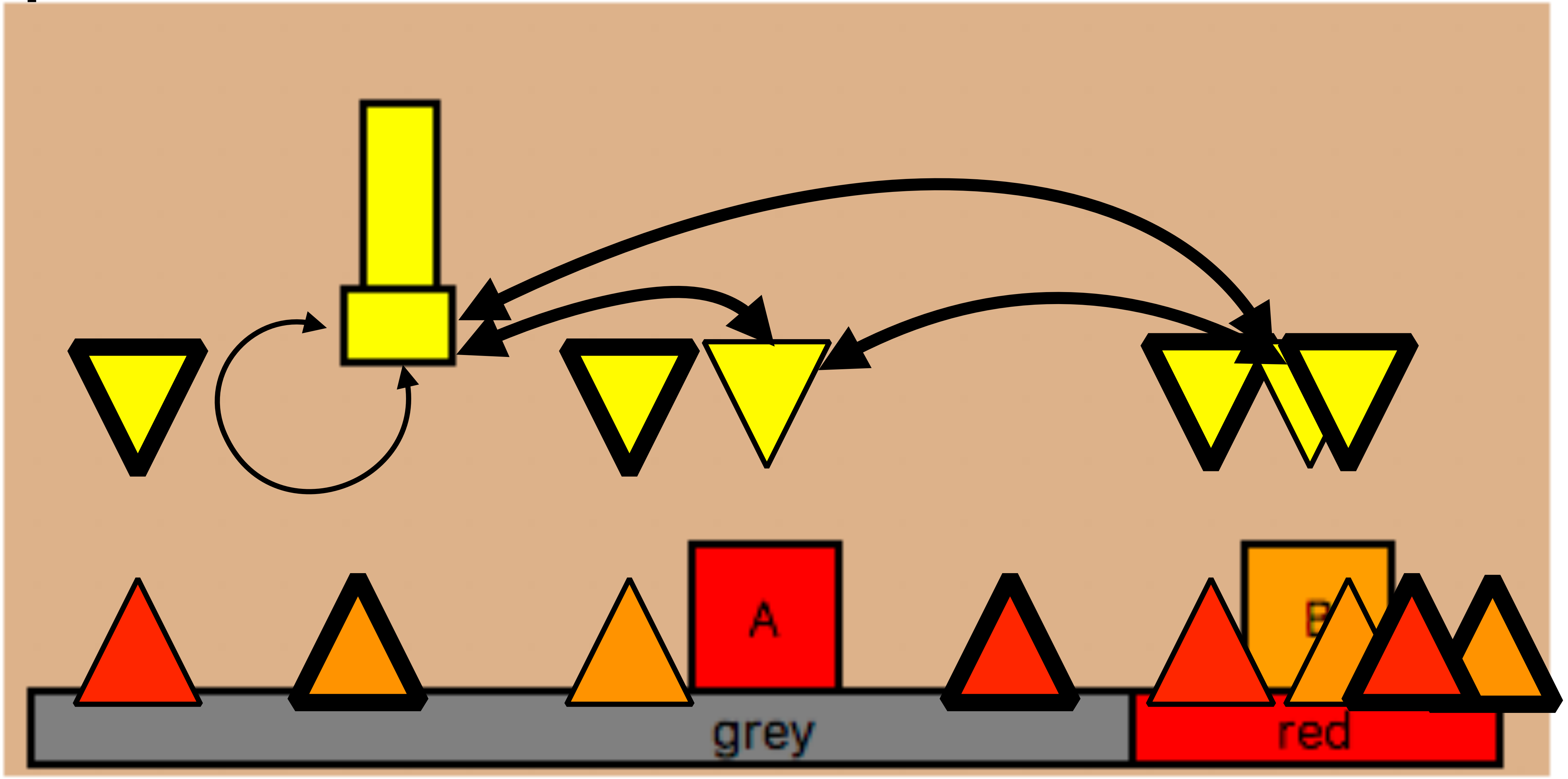


FastDownward Search → Infeasible!

Incremental: Sampling Iteration 2

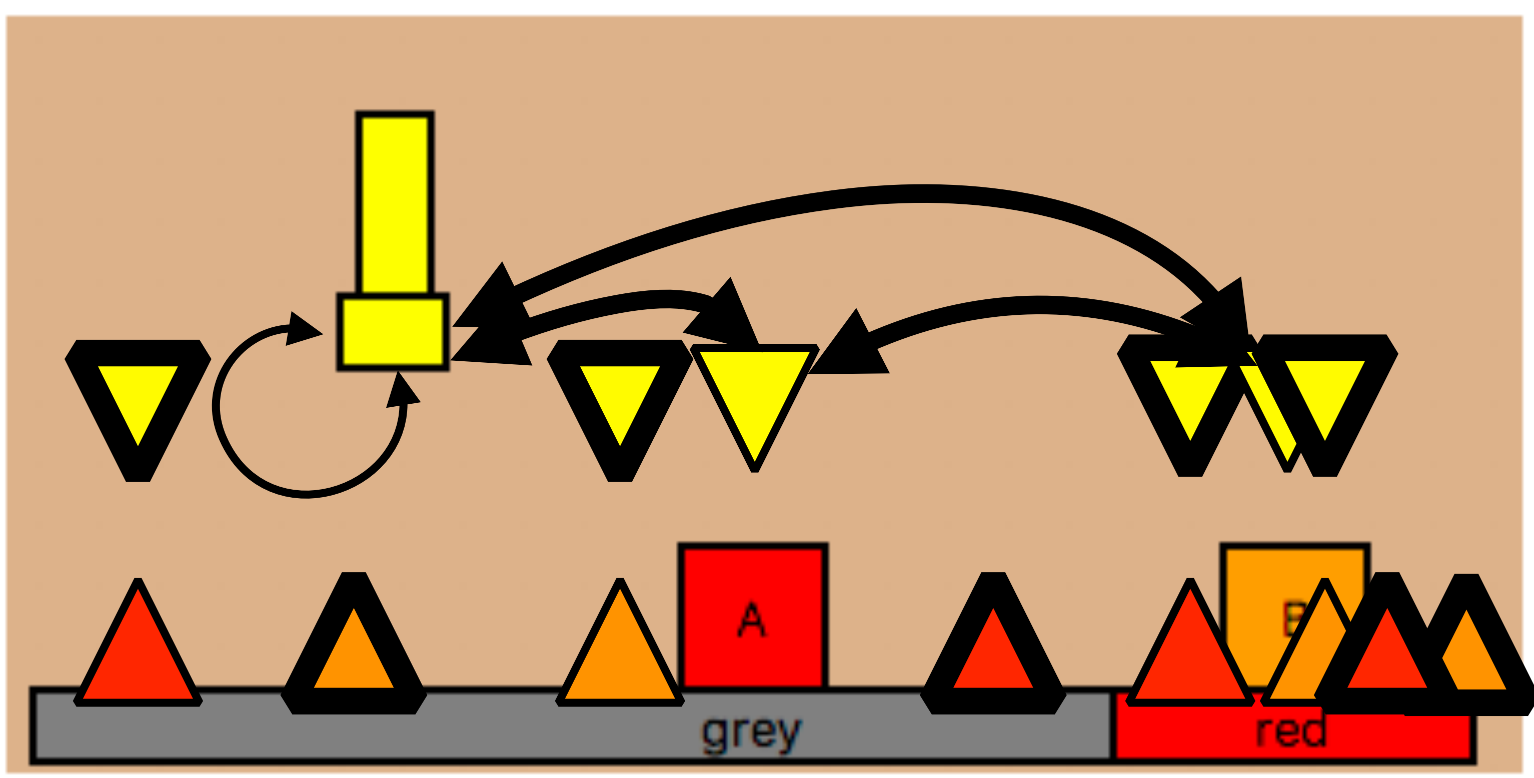
Iteration 2 - 54 stream evaluations

- **Sampled:**
  - 4 new robot configurations:
  - 4 new block poses:
  - 10 new trajectories:



Incremental: Search Iteration 2

- Pass current discretization to FastDownward
- If **infeasible**, the current set of samples is insufficient



Incremental Example: Iterations 3-4

Iteration 3 - 118 stream evaluations

Iteration 4 - 182 stream evaluations

Solution:

1) **move** [-7.5 5.] [[-7.5 5.], [-7.5 5.], [7.5 5.], [7.5 2.5]] [7.5 2.5]
2) **pick B** [7.5 0.] [0. -2.5] [7.5 2.5]
3) **move** [7.5 2.5] [[7.5 2.5], [7.5 5.], [10.97 5.], [10.97 2.5]] [10.97 2.5]
4) **place B** [10.97 0.] [0. -2.5] [10.97 2.5]
5) **move** [10.97 2.5] [[10.97 2.5], [10.97 5.], [0. 5.], [0. 2.5]] [0. 2.5]
6) **pick A** [0. 0.] [0. -2.5] [0. 2.5]
7) **move** [0. 2.5] [[0. 2.5], [0. 5.], [7.65 5.], [7.65 2.5]] [7.65 2.5]
8) **place A** [7.65 0.] [0. -2.5] [7.65 2.5]

- **Drawback** - many unnecessary samples produced
  - **Computationally expensive** to generate
  - Induces **large discrete-planning problems**

Optimistic Stream Outputs

- Many TAMP streams are exceptionally **expensive**
  - Inverse kinematics, motion planning, collision checking
- **Only** query streams that are **identified** as useful
  - Plan with **optimistic hypothetical** outputs [Srivastava 2014]
- Inductively create **unique first-class** placeholder object for each stream instance output (has **#** as its prefix)

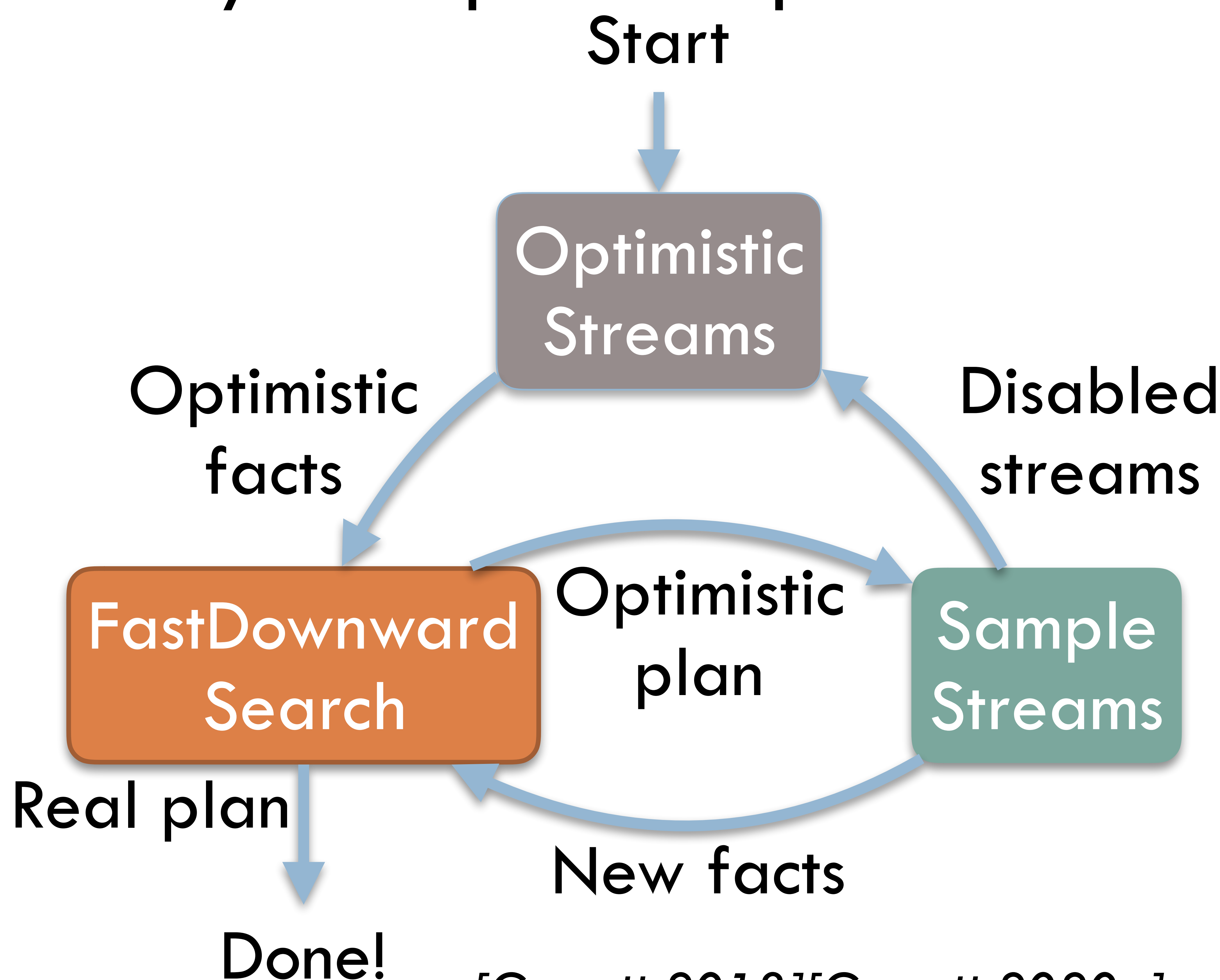
Optimistic evaluations:

1. **s-region:**(block-A, red-region)->(**<u>#p0</u>**)
2. **s-ik:**(block-A, [0. 0.], [0. -2.5])->(**<u>#q0</u>**),
3. **s-ik:**(block-A, **<u>#p0</u>**, [0. -2.5]) ->(**<u>#q2</u>**)

[Garrett 2018]
[Garrett 2020a]

Focused Algorithm

- **Lazily** plan using optimistic outputs **before** real outputs
- **Recover** set of streams used by the optimistic plan
- Repeat:
  1. Construct active **optimistic** objects
  2. **Search** with **real & optimistic** objects
  3. If **only real objects** used, **return plan**
  4. **Sample** used streams
  5. **Disable** used streams

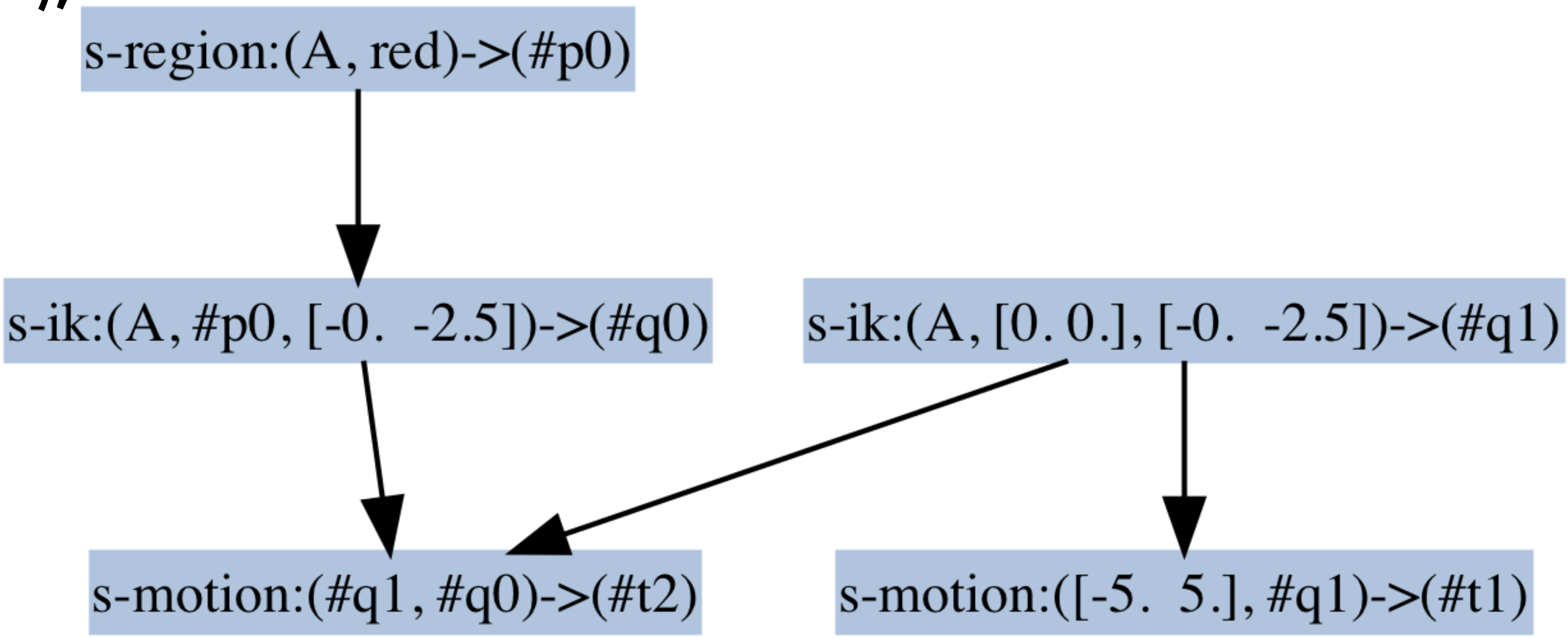


[Garrett 2018][Garrett 2020a]

Focused Example 1

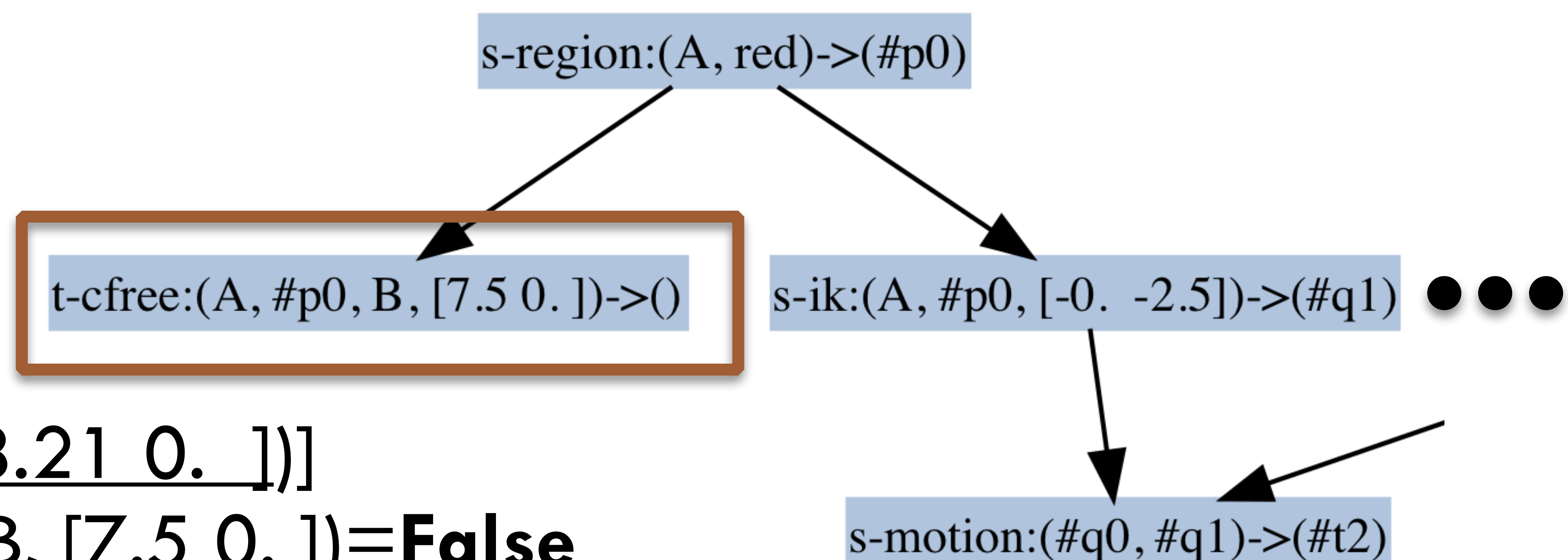
Optimistic Plan:
move([-5. 5.], #t0, #q0), **pick**(A, [0. 0.], [-0. -2.5], #q0),
move(#q0, #t2, #q1), **place**(A, #p0, [-0. -2.5], #q1)

Constraints:
(kin, A, #q0, #p0, [-0. -2.5]),
(kin, A, #q1, [0. 0.], [-0. -2.5]),
(motion, [-5. 5.], #t1, #q1),
(motion, #q1, #t2, #q0),
(contain, A, #p0, red),



Focused Example 2: Iteration 1

Optimistic Plan:
move([-5. 5.], #t0, #q0), **pick**(A, [0. 0.], [-0. -2.5], #q0),
move(#q0, #t2, #q1), **place**(A, #p0, [-0. -2.5], #q1)
Constraints:
(cfree, A, #p0, B, [7.5 0.]), (contain, A, #p0, red),
(kin, A, #q0, [0. 0.], [-0. -2.5]), (kin, A, #q1, #p0, [-0. -2.5]),
(motion, #q0, #t2, #q1), (motion, [-5. 5.], #t0, #q0)



Stream evaluations:
1.**s-region:**(A, red)->[([8.21 0.])]
2.**t-cfree:**(A, [8.21 0.], B, [7.5 0.])=**False**
These stream instances are **removed** from subsequent searches

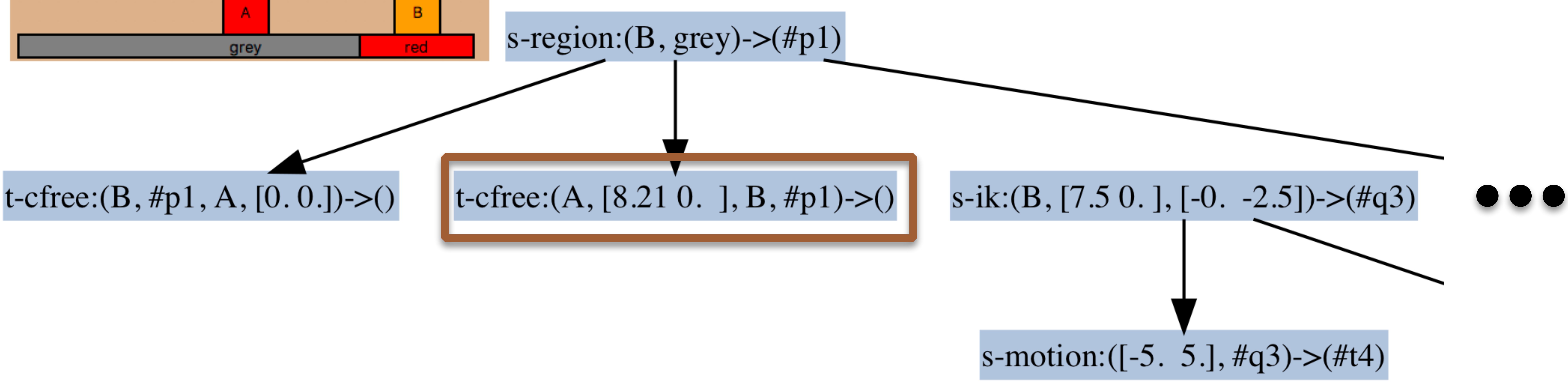
Focused Example: Iteration 2

Optimistic Plan:

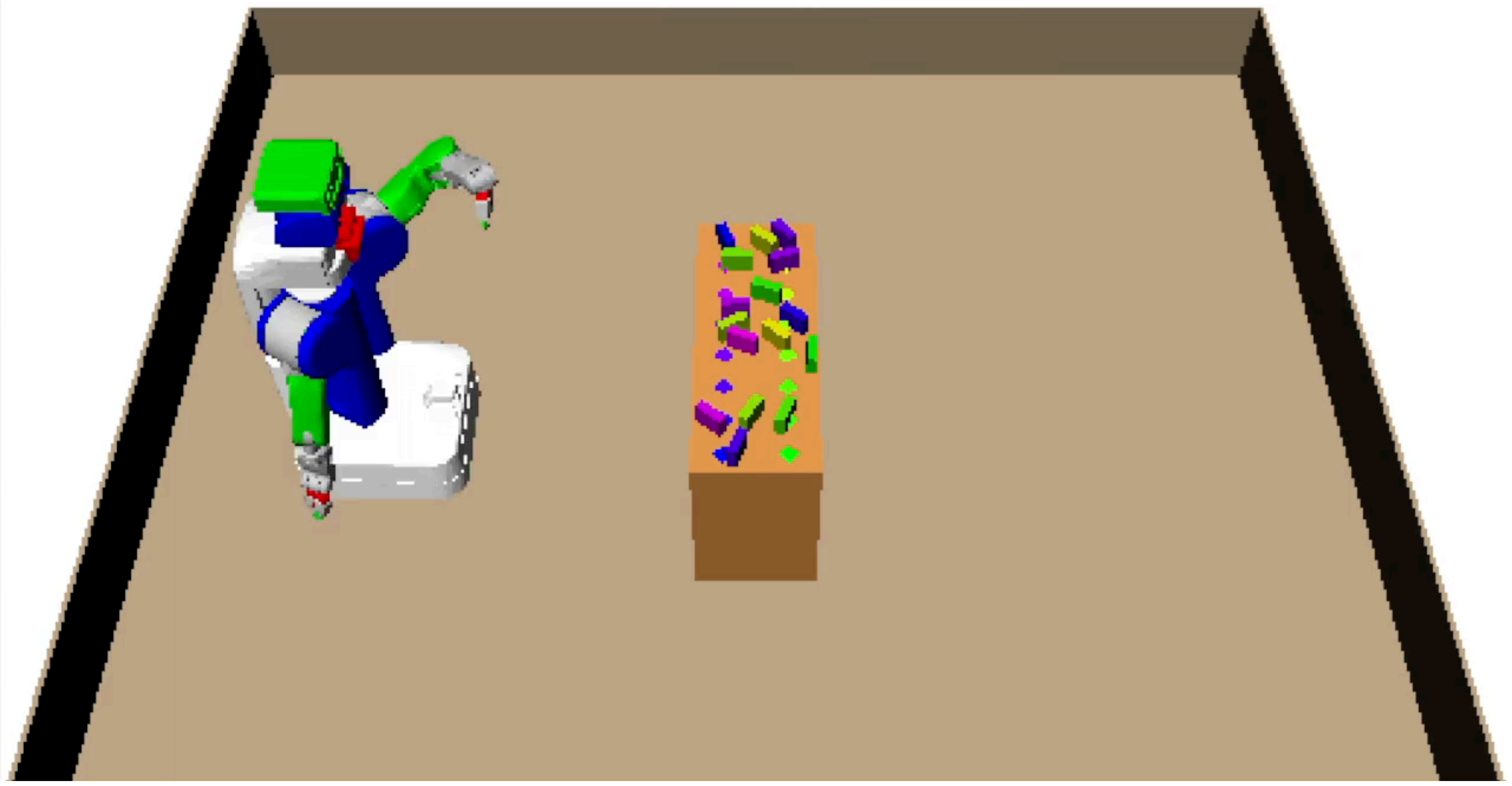
move([-5. 5.], #t4, #q2), **pick**(B, [7.5 0.], [-0. -2.5], #q2), **move**(#q2, #t9, #q3), **place**(B, #p1, [-0. -2.5], #q3), **move**(#q3, #t6, #q0), **pick**(A, [0. 0.], [-0. -2.5], #q0), **move**(#q0, #t8, #q4), **place**(A, [8.21 0.], [-0. -2.5], #q4)

t-cfree:(A, [8.21 0.], B, [7.5 0.]) previously **failed**

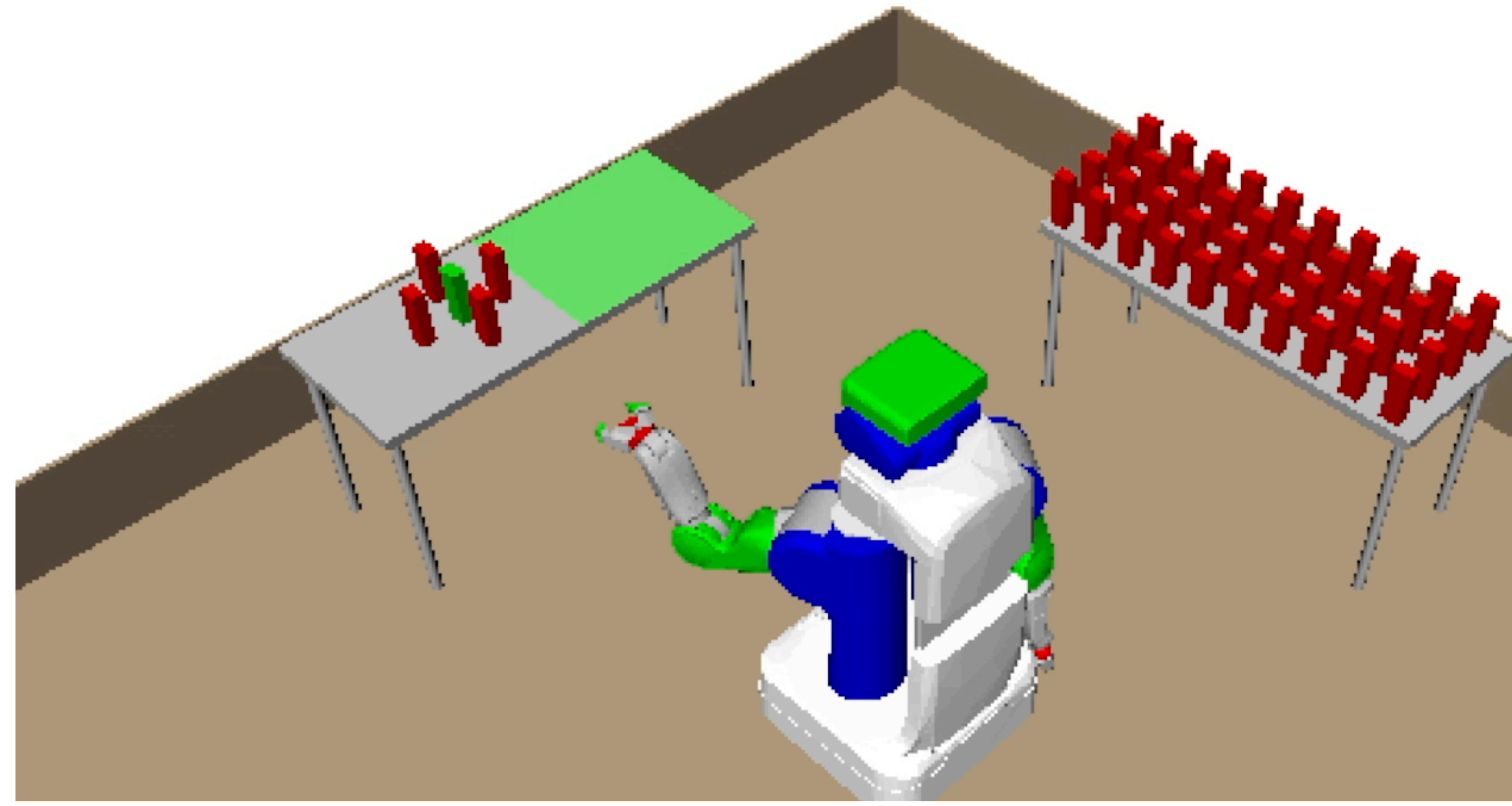
t-cfree:(A, [8.21 0.], B, #p1) might **succeed**



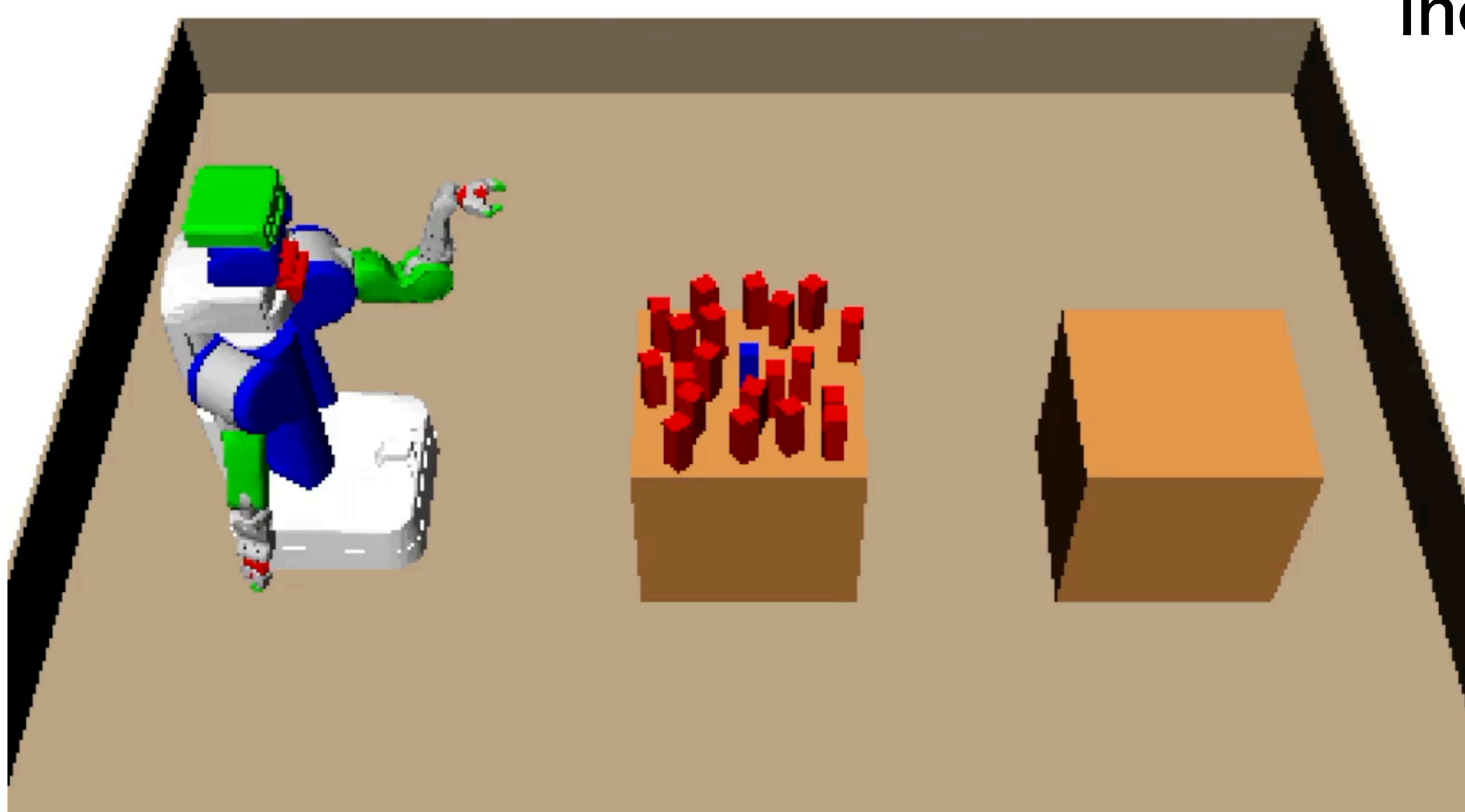
Focused Outperforms Incremental



Incremental ~20s
Focused ~10s



Incremental 120+
Focused ~25s



Incremental 120+
Focused ~20s

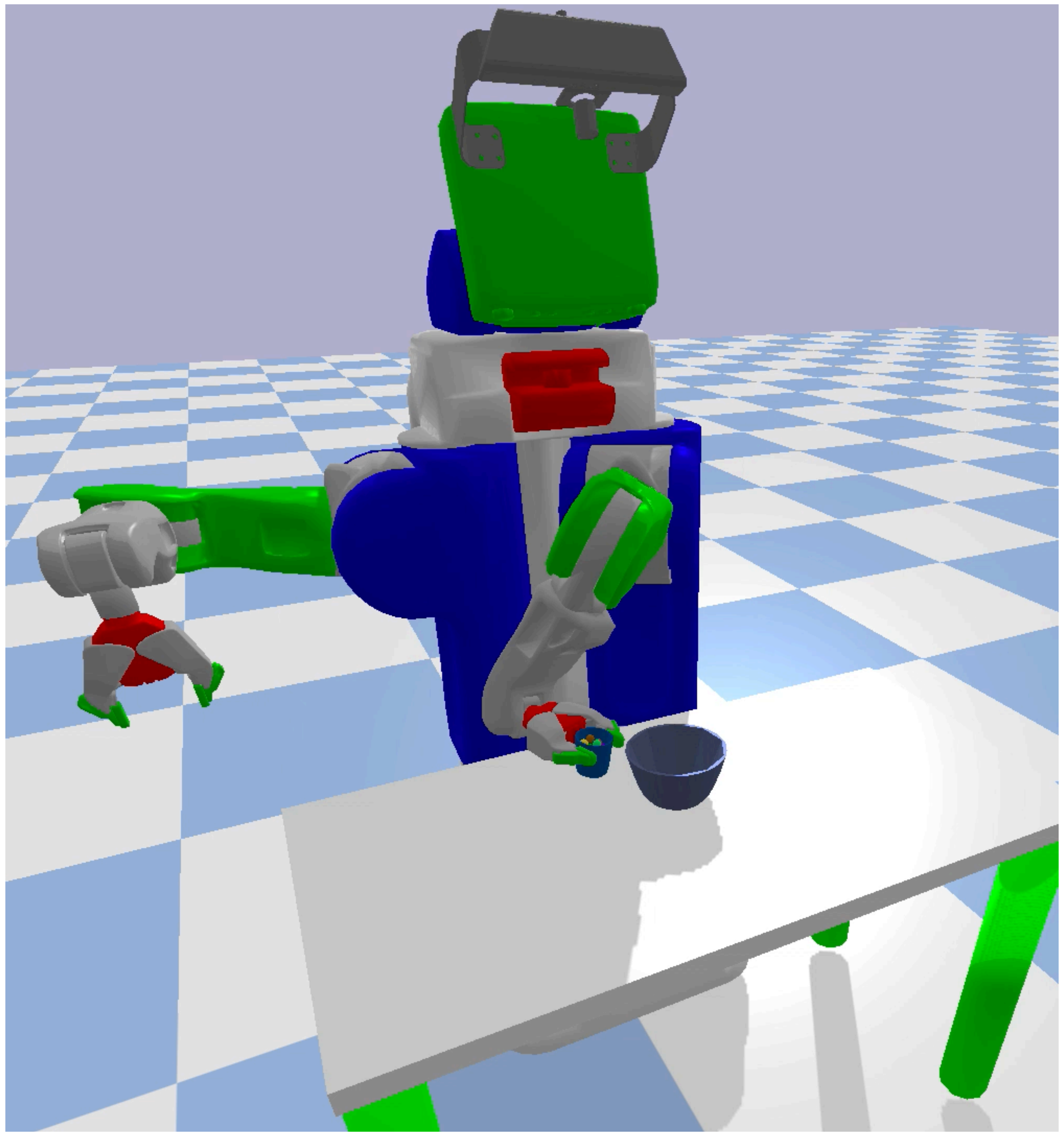
[Garrett 2018]

Learning Samplers

Learning Samplers for Dynamic Skills

- Learn **many** good pours
  - % particle mass in bowl
- **Gaussian Processes** Regression
  - **Uncertainty** quantification

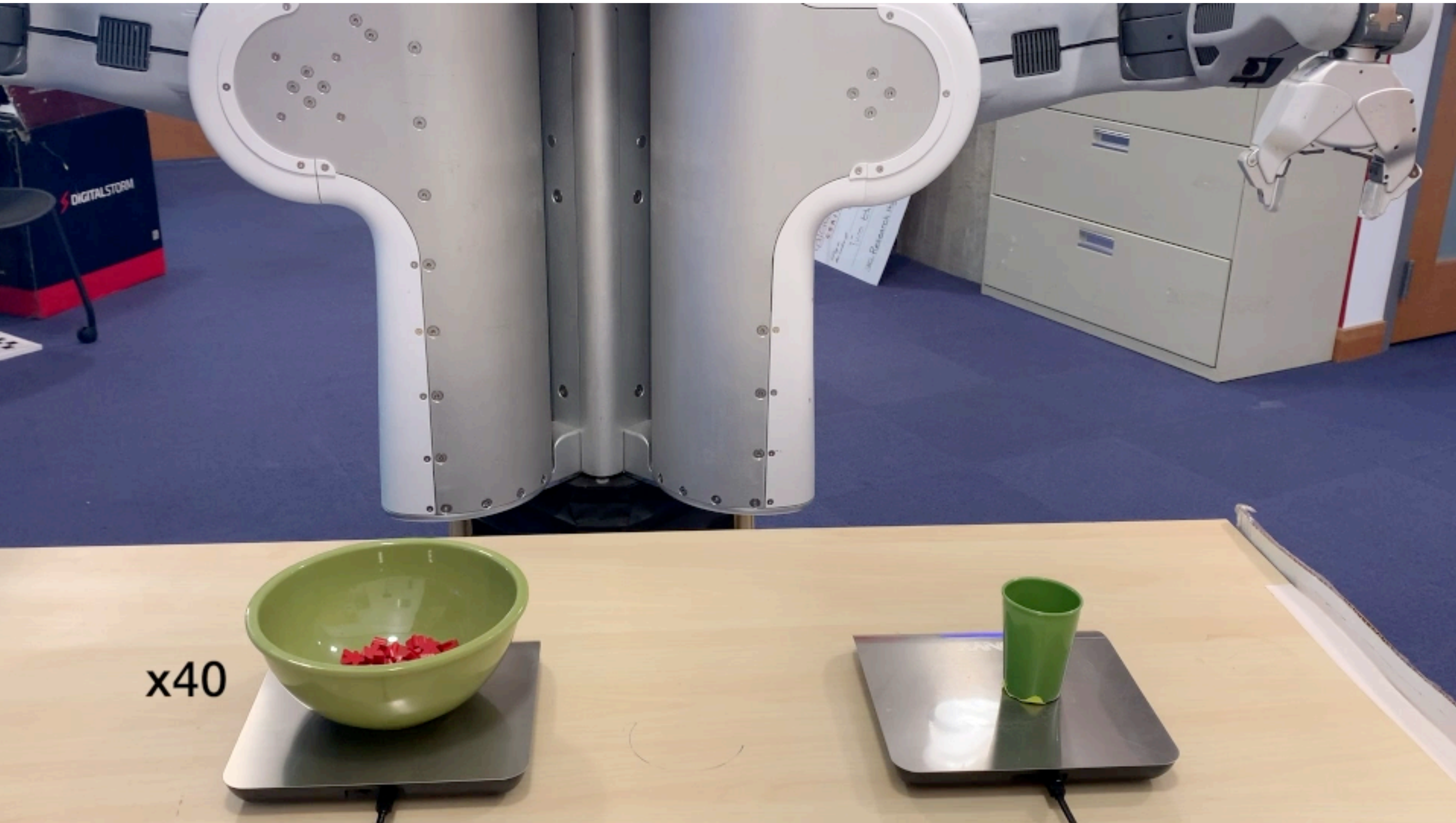
$$\textbf{score}(\underbrace{\overbrace{\mathit{radius, height},}^{\text{given}}\ \overbrace{\color{red}{\mathit{position, axis, pitch}}}^{\color{red}{\text{sample}}}}_{\theta}) > 0$$



```
(:action pour
 :parameters (?arm ?bowl ?pose ?cup ?grasp ?conf ?traj)
 :precondition (and (GoodPour ?arm ?bowl ?pose ?cup ?grasp …))
 :effect (and (HasWater ?bowl) (not (HasWater ?cup))))
```

[Wang, Garrett, Kaelbling, and Lozano-Pérez 2018]

PDDLStream Planning to Collect Data



Training: Active Exploration

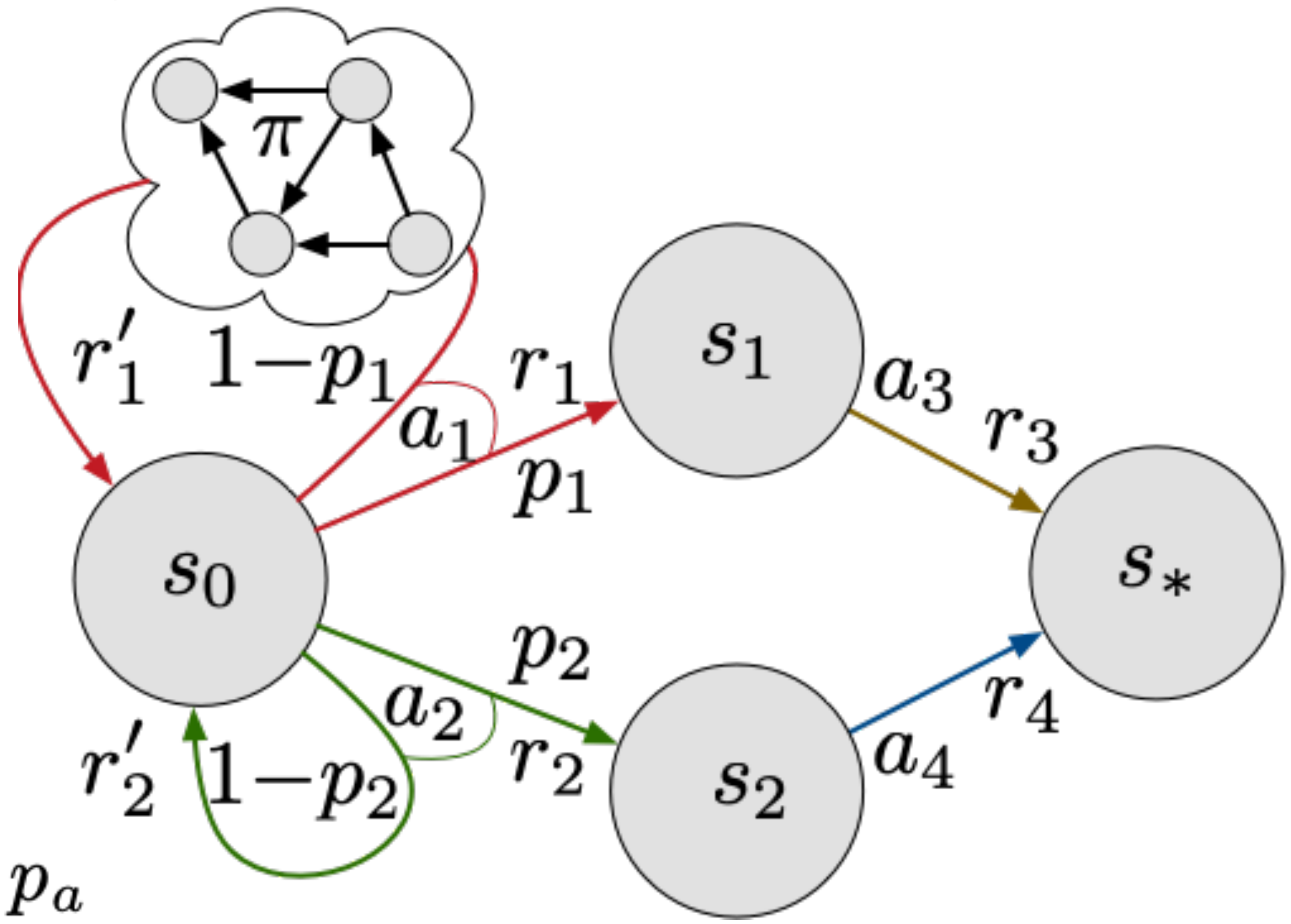
Testing: Predicted Best Sample

TAMP Under Uncertainty

[Garrett, Paxton, Lozano-Pérez, Kaelbling, & Fox 2020]

Addressing Stochasticity

- Compute policy **online** via **replanning**
- Approximate MDP as a **cost-sensitive deterministic** problem (**determinization**) [Yoon 2007]
  - Allows us to leverage PDDLStream
- Class of simple MDPs where **deterministic planning is optimal**
- Combines **reward & probability** of transition



$$\hat{c}_a = -r_a - \sum_{t=1}^{\infty} r'_a(1-p_a)^t = -r_a - r'_a \frac{1-p_a}{p_a}$$

[Keyder 2008]

Addressing Partial Observability

- **Occlusions** due to limited field of view, doors, drawers, movable objects, robot, …
- Update a **belief** (probability distribution) over states
- Plan in the **space of beliefs** (belief space planning)

[Kaelbling 2013]

- **Factor** into a pose belief per movable object
- Use **particle-based** beliefs to capture multi-modal distributions

Movable Object Occlusion

Observation Actions

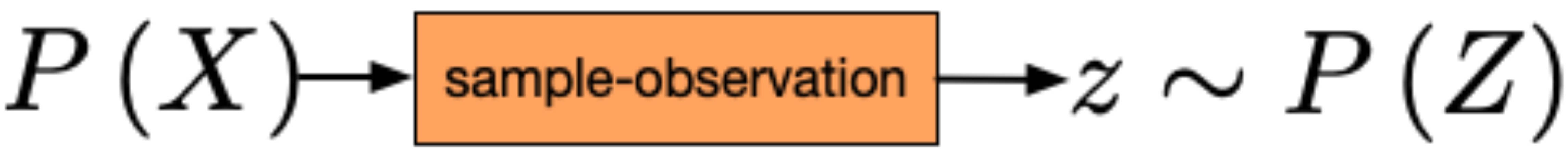
- State variables are **distributions over values**
- Intentionally take **observation actions**

```
(:action detect
 :param (?o ?pb1 ?obs ?pb2)
 :pre (and (BeliefUpdate ?o ?pb1 ?obs ?pb2)
  (AtPoseB ?o ?pb1) (BVisible ?o ?pb1 ?obs))
 :eff (and (AtPoseB ?o ?pb2) (not (AtPoseB ?o ?pb1))
  (incr (total-cost) (ObsCost ?o ?pb1 ?obs))))
```

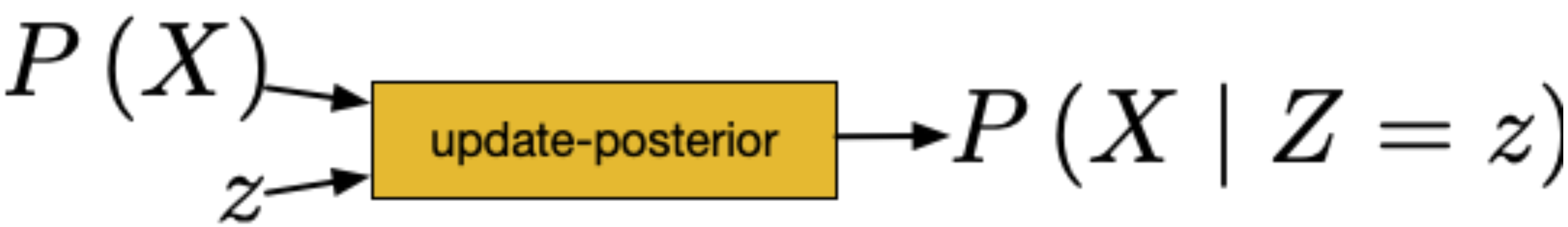
- Streams used sample and compute **belief dynamics**
- Visible with **high probability** precondition
- Action costs incorporate **observation likelihoods**

Observation Streams

- Streams operate on **distributions** rather than **points**
- **Sample** possible observations



- **Deterministic belief update** given observation



- Compute the **likelihood of an observation**

- Incorporate into `detect` cost

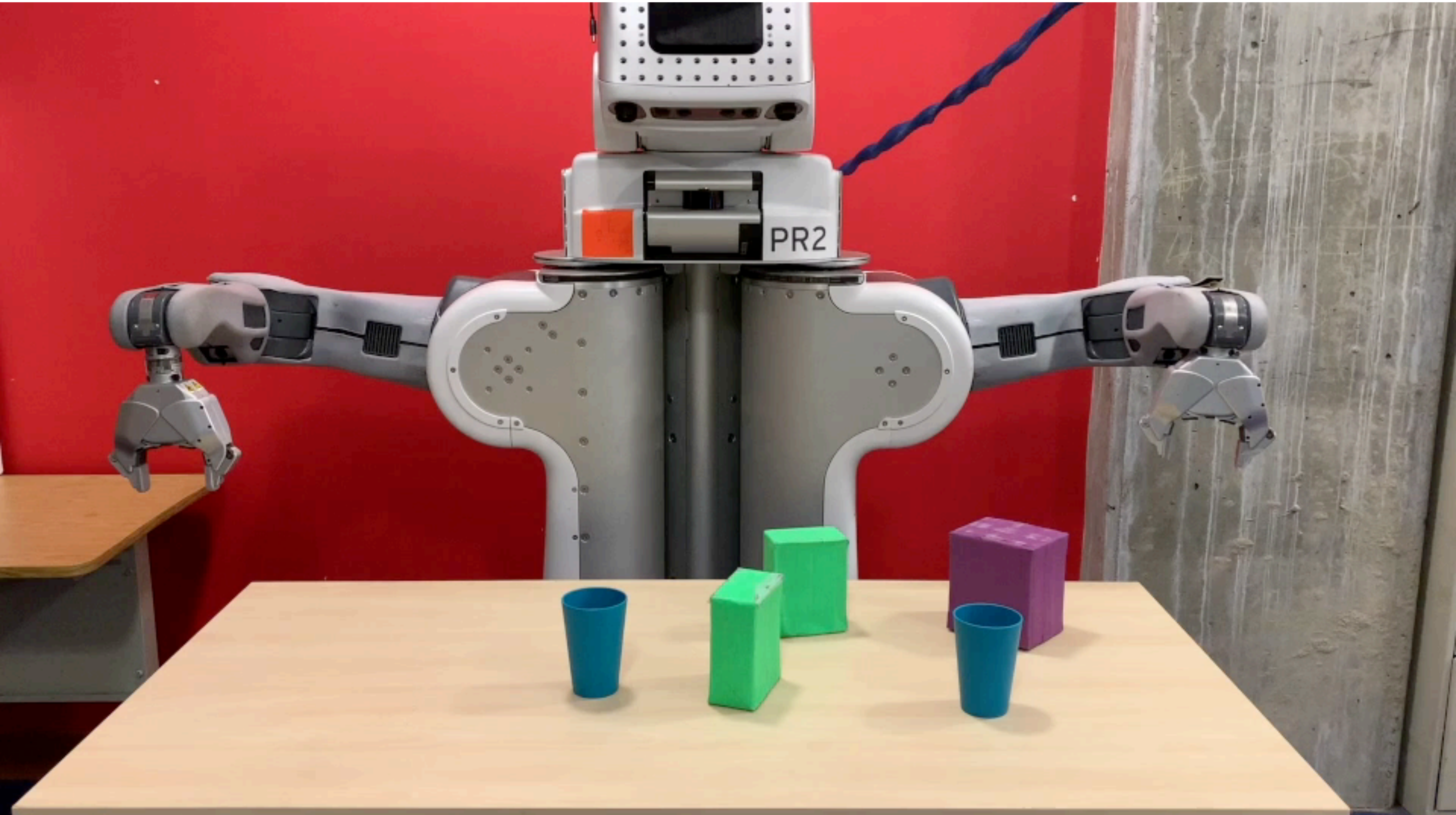
Prior: Spam in One of the Drawers

Prior: Spam in One of the Drawers

Takeaways

- **PDDLStream:** planning language that supports **sampling procedures** as blackbox streams
- **Lazy/optimistic** planning intelligently queries only a small number of samplers (focused algorithm)
- React online to stochasticity using **cost-sensitive, deterministic** replanning
- Define **streams over distributions** to perform belief-space planning

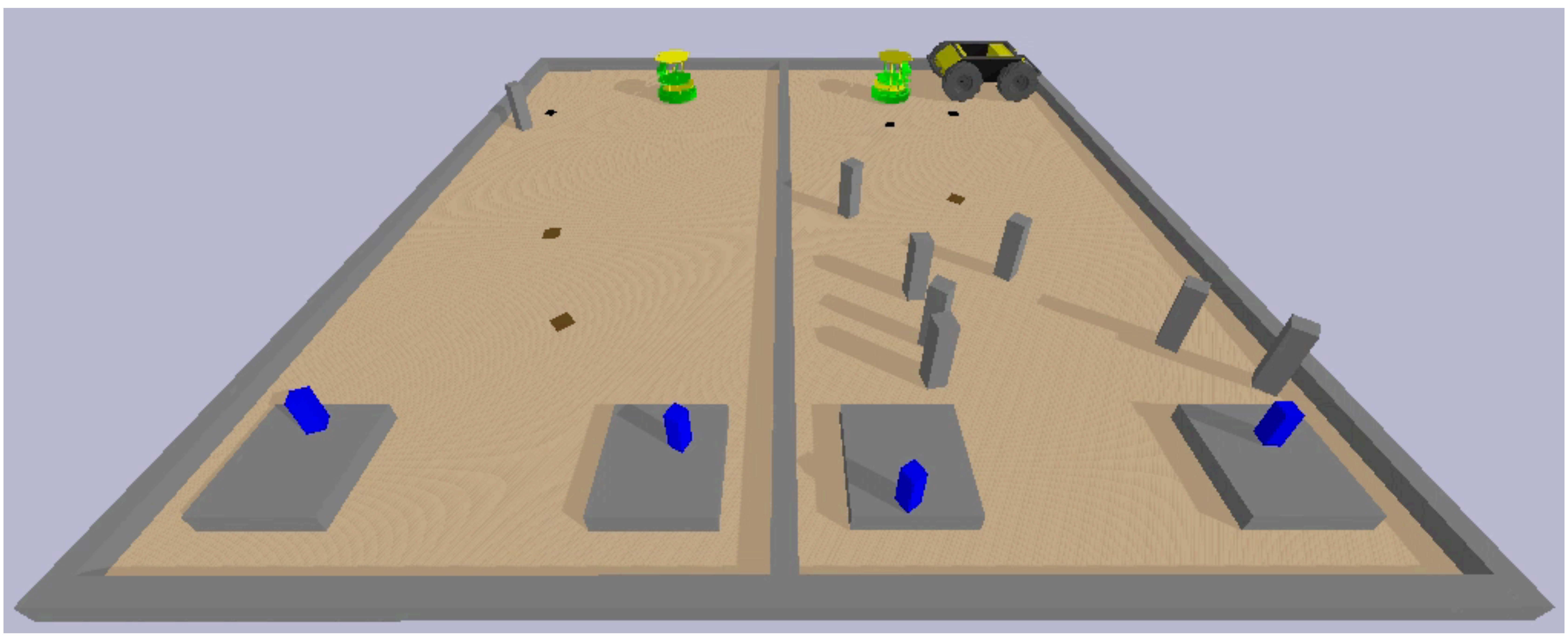
Questions? (and Outtakes!)



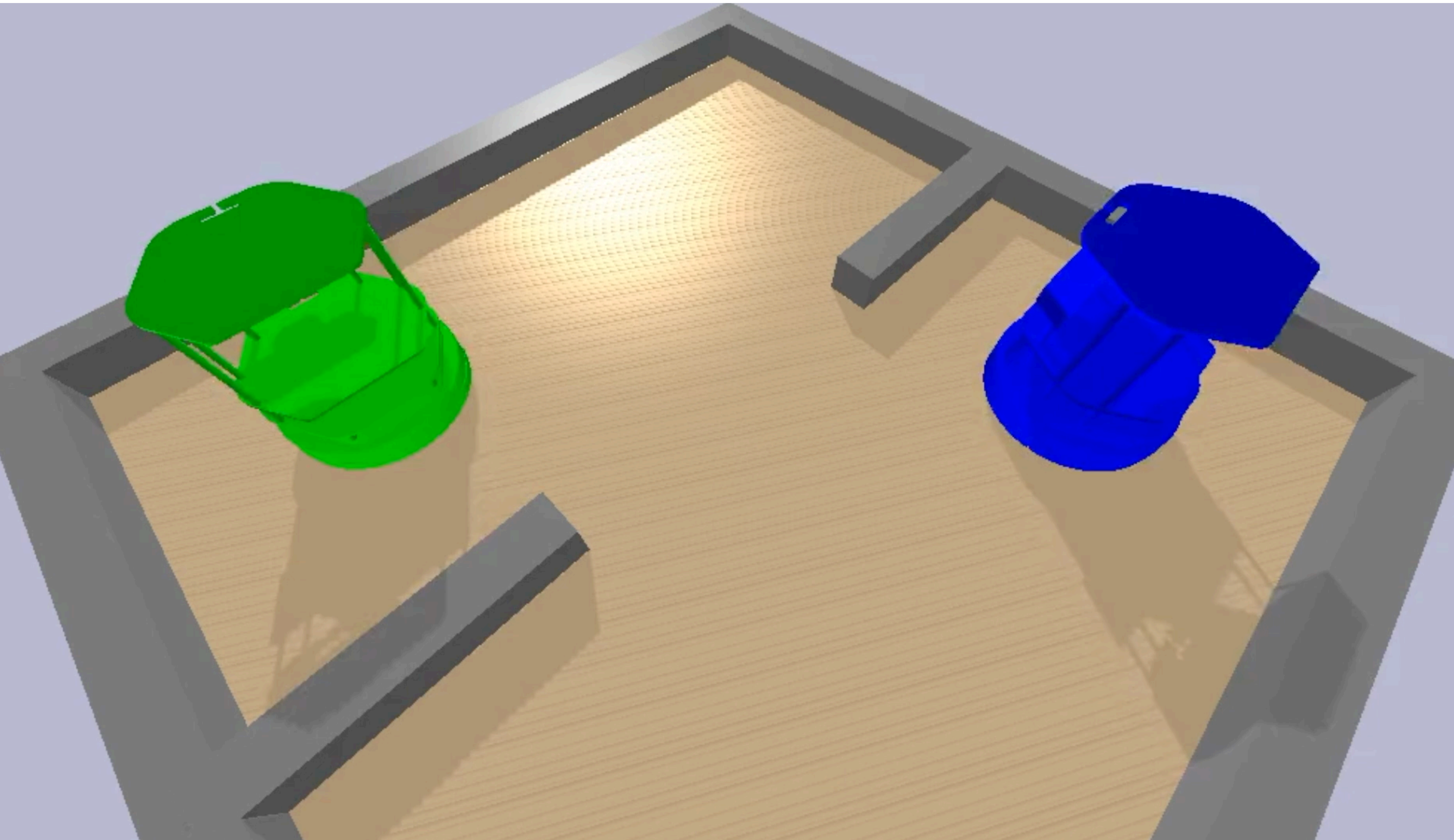
Multi-Robot TAMP

Centralized Turtlebot Imaging

- Rovers domain with **visibility** and **reachability**
- How can we plan for **simultaneous** execution?
  - Use a **temporal planner** as search subroutine (e.g. **Temporal FastDownward**) [Eyerich 2009]



Swap Initial Configurations



References

Classical Planning

- **[Fikes 1971]** Fikes, R.E. and Nilsson, N.J., 1971. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, *2*(3-4), pp.189-208.
- **[Aeronautiques 1998]** Aeronautiques, C., Howe, A., Knoblock, C., McDermott, I.D., Ram, A., Veloso, M., Weld, D., SRI, D.W., Barrett, A., Christianson, D. and Friedman, M., 1998. PDDL| The Planning Domain Definition Language.
- **[Hoffman 2001]** Hoffmann, J. and Nebel, B., 2001. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, *14*, pp.253-302.
- **[Helmert 2006]** Helmert, M., 2006. The fast downward planning system. *Journal of Artificial Intelligence Research*, *26*, pp.191-246.
- **[Eyerich 2009]** Eyerich, P., Mattmüller, R. and Röger, G. (2009) "Using the Context-enhanced Additive Heuristic for Temporal and Numeric Planning," in *Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS)*. AAAI Press, pp. 130–137.

Motion Planning

- **[Kavraki 1994]** Kavraki, L., Svestka, P. and Overmars, M.H., 1994. Probabilistic roadmaps for path planning in high-dimensional configuration spaces (Vol. 1994).
- **[Bohlin 2000]** Bohlin, R. and Kavraki, L.E., 2000, April. Path planning using lazy PRM. In Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065) (Vol. 1, pp. 521-528). IEEE.
- **[Kuffner 2000]** Kuffner Jr, J.J. and LaValle, S.M., 2000, April. RRT-connect: An efficient approach to single-query path planning. In *ICRA* (Vol. 2).
- **[Kuffner 2001]** LaValle, S.M. and Kuffner Jr, J.J., 2001. Randomized kinodynamic planning. *The International Journal of Robotics Research*, *20*(5), pp.378-400.
- **[Karaman 2011]** Karaman, S. and Frazzoli, E., 2011) Sampling-based Algorithms for Optimal Motion Planning. *International Journal of Robotics Research (IJRR)*, Sage Publications, 30(7), pp. 846–894.

Prediscretized Planning

- **[Dornhege 2009]** Dornhege, C., Eyerich, P., Keller, T., Trüg, S., Brenner, M. and Nebel, B., 2009, October. Semantic attachments for domain-independent planning systems. In *Nineteenth International Conference on Automated Planning and Scheduling*.
- **[Erdem 2011]** Erdem, E., Haspalamutgil, K., Palaz, C., Patoglu, V. and Uras, T., 2011, May. Combining high-level causal reasoning with low-level geometric reasoning and motion planning for robotic manipulation. In *2011 IEEE International Conference on Robotics and Automation* (pp. 4575-4581). IEEE.
- **[Lagriffoul 2014]** Lagriffoul, F., Dimitrov, D., Bidot, J., Saffiotti, A. and Karlsson, L., 2014. Efficiently combining task and motion planning using geometric constraints. *The International Journal of Robotics Research*, *33*(14), pp.1726-1747.
- **[Ferrer-Mestres 2017]** Ferrer-Mestres, J., Frances, G. and Geffner, H., 2017. Combined task and motion planning as classical AI planning. *arXiv preprint arXiv:1706.06927*.
- **[Dantam 2018]** Dantam, N.T., Kingston, Z.K., Chaudhuri, S. and Kavraki, L.E., 2018. An incremental constraint-based framework for task and motion planning. *The International Journal of Robotics Research*, *37*(10), pp.1134-1151.
- **[Lo 2018]** Lo, S.Y., Zhang, S. and Stone, P., 2018, July. PETLON: Planning Efficiently for Task-Level-Optimal Navigation. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems* (pp. 220-228). International Foundation for Autonomous Agents and Multiagent Systems.
- **[Huang 2018]** Huang, Y., Garrett, C.R. and Mueller, C.T., 2018. Automated sequence and motion planning for robotic spatial extrusion of 3D trusses. *Construction Robotics*, *2*(1-4), pp.15-39.

Multi-Modal Motion Planning

- **[Alami 1994]** Alami, R., Laumond, J.P. and Siméon, T., 1994. Two manipulation planning algorithms. In *WAFR Proceedings of the workshop on Algorithmic foundations of robotics* (pp. 109-125). AK Peters, Ltd. Natick, MA, USA.
- **[Siméon 2004]** Siméon, T., Laumond, J.P., Cortés, J. and Sahbani, A., 2004. Manipulation planning with probabilistic roadmaps. *The International Journal of Robotics Research*, *23*(7-8), pp.729-746.
- **[Hauser 2011]** Hauser, K. and Ng-Thow-Hing, V., 2011. Randomized multi-modal motion planning for a humanoid robot manipulation task. *The International Journal of Robotics Research*, *30*(6), pp.678-698.
- **[Plaku 2010]** Plaku, E. and Hager, G. (2010) "Sampling-based Motion Planning with Symbolic, Geometric, and Differential Constraints," in *IEEE International Conference on Robotics and Automation (ICRA)*. Available at: http://ieeexplore.ieee.org/document/5509563/.
- **[Barry 2013]** Barry, J., Kaelbling, L.P. and Lozano-Pérez, T., 2013, May. A hierarchical approach to manipulation with diverse actions. In *2013 IEEE International Conference on Robotics and Automation* (pp. 1799-1806). IEEE.
- **[Toussaint 2015]** Toussaint, M., 2015, June. Logic-geometric programming: An optimization-based approach to combined task and motion planning. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.
- **[Vega-Brown 2016]** Vega-Brown, W. and Roy, N., 2016, December. Asymptotically optimal planning under piecewise-analytic constraints. In *Workshop on the Algorithmic Foundations of Robotics*.
- **[Toussaint 2018]** Toussaint, M., Allen, K., Smith, K.A. and Tenenbaum, J.B., 2018. Differentiable Physics and Stable Modes for Tool-Use and Manipulation Planning. In *Robotics: Science and Systems*.

Task and Motion Planning

- **[Gravot 2005]** Gravot, F., Cambon, S. and Alami, R., 2005. aSyMov: a planner that deals with intricate symbolic and geometric problems. In *Robotics Research. The Eleventh International Symposium* (pp. 100-110). Springer, Berlin, Heidelberg.
- **[Kaelbling 2011]** Kaelbling, L. P. and Lozano-Pérez, T. Hierarchical task and motion planning in the now. *2011 IEEE International Conference on Robotics and Automation*, Shanghai, 2011, pp. 1470-1477.
- **[Srivastava 2014]** Srivastava, S., Fang, E., Riano, L., Chitnis, R., Russell, S. and Abbeel, P., 2014, May. Combined task and motion planning through an extensible planner-independent interface layer. In *2014 IEEE international conference on robotics and automation (ICRA)* (pp. 639-646). IEEE.
- **[Garrett 2017]** Garrett, C.R., Lozano-Perez, T. and Kaelbling, L.P., 2017. FFRob: Leveraging symbolic planning for efficient task and motion planning. *The International Journal of Robotics Research*, *37*(1), pp.104-136.
- **[Garrett 2018]** Garrett, C.R., Lozano-Pérez, T. and Kaelbling, L.P., 2018. Sampling-based methods for factored task and motion planning. *The International Journal of Robotics Research*, *37*(13-14), pp.1796-1825.
- **[Garrett 2020a]** Garrett, C. R., Lozano-Pérez, T. and Kaelbling, L. P. (2020) "PDDLStream: Integrating Symbolic Planners and Blackbox Samplers," in *International Conference on Automated Planning and Scheduling (ICAPS)*. Available at: https://arxiv.org/abs/1802.08705.

Probabilistic & Partially-Observable

- **[Kaelbling 1998]** Kaelbling, L.P., Littman, M.L. and Cassandra, A.R., 1998. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, *101*(1-2), pp.99-134.
- **[Yoon 2007]** Yoon, S.W., Fern, A. and Givan, R., 2007, September. FF-Replan: A Baseline for Probabilistic Planning. In *ICAPS* (Vol. *7*, pp. 352-359).
- **[Keyder 2008]** Keyder, E. and Geffner, H., 2008. The HMDPP planner for planning with probabilities. *Sixth International Planning Competition at ICAPS*, *8*.
- **[Kaelbling 2013]** Kaelbling, L.P. and Lozano-Pérez, T., 2013. Integrated task and motion planning in belief space. *The International Journal of Robotics Research*, *32*(9-10), pp.1194-1227.
- **[Garrett 2020b]** Garrett, C. R. *et al.* (2020) “Online Replanning in Belief Space for Partially Observable Task and Motion Problems,” in *International Conference on Robotics and Automation (ICRA)*. Available at: https://arxiv.org/abs/1911.04577.