BEHAVIOR-1K: A Benchmark for Embodied AI with 1,000 Everyday Activities and Realistic Simulation

Chengshu Li*, Ruohan Zhang*, Josiah Wong*, Cem Gokmen*, Sanjana Srivastava*, Roberto Martín-Martín*, Chen Wang*, Gabrael Levine*, Michael Lingelbach, Jiankai Sun, Mona Anvari, Minjune Hwang, Manasi Sharma, Arman Aydin, Dhruva Bansal, Samuel Hunter, Kyu-Young Kim, Alan Lou, Caleb R Matthews, Ivan Villa-Renteria, Jerry Huayang Tang, Claire Tang, Fei Xia, Silvio Savarese, Hyowon Gweon, Karen Liu, Jiajun Wu, Li Fei-Fei

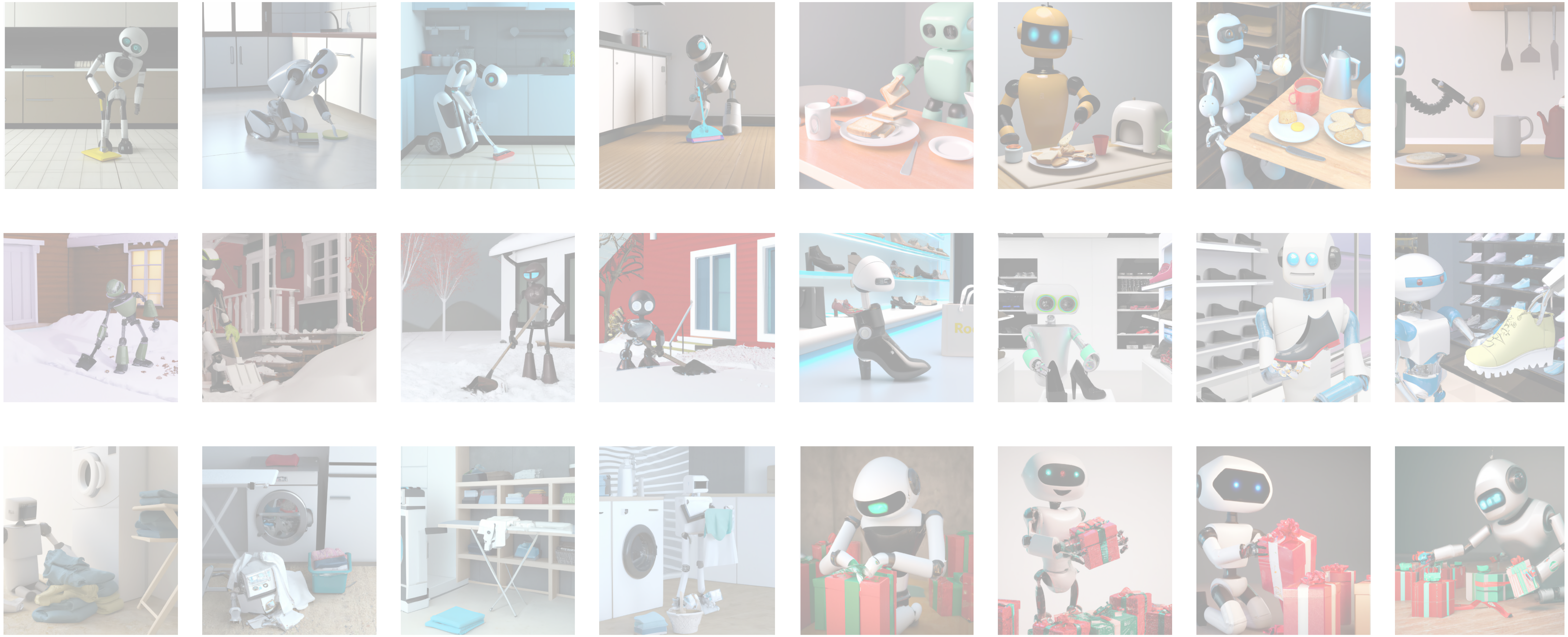
Stanford University Human-Centered Artificial Intelligence

Stanford University

STANFORD VISION & LEARNING

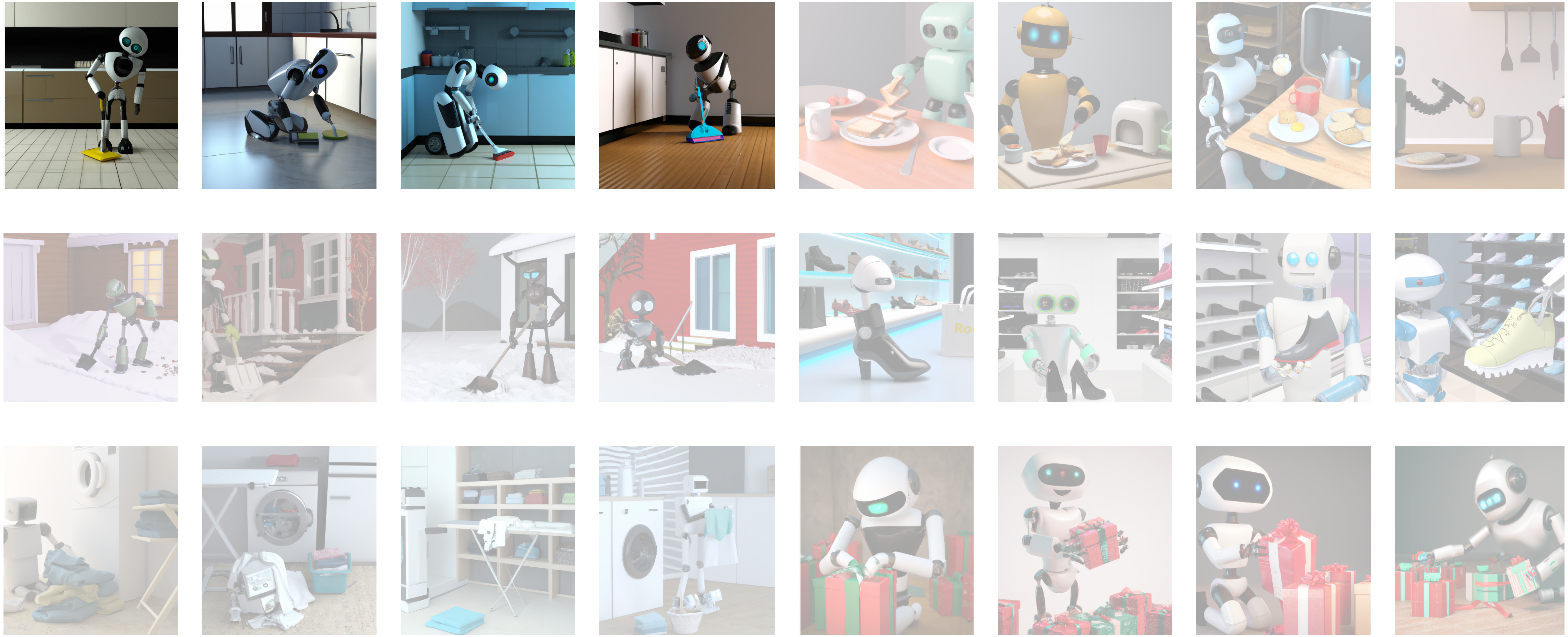
NVIDIA OMNIVERSE

What kind of activities?
“Would you like a robot to help you with…”



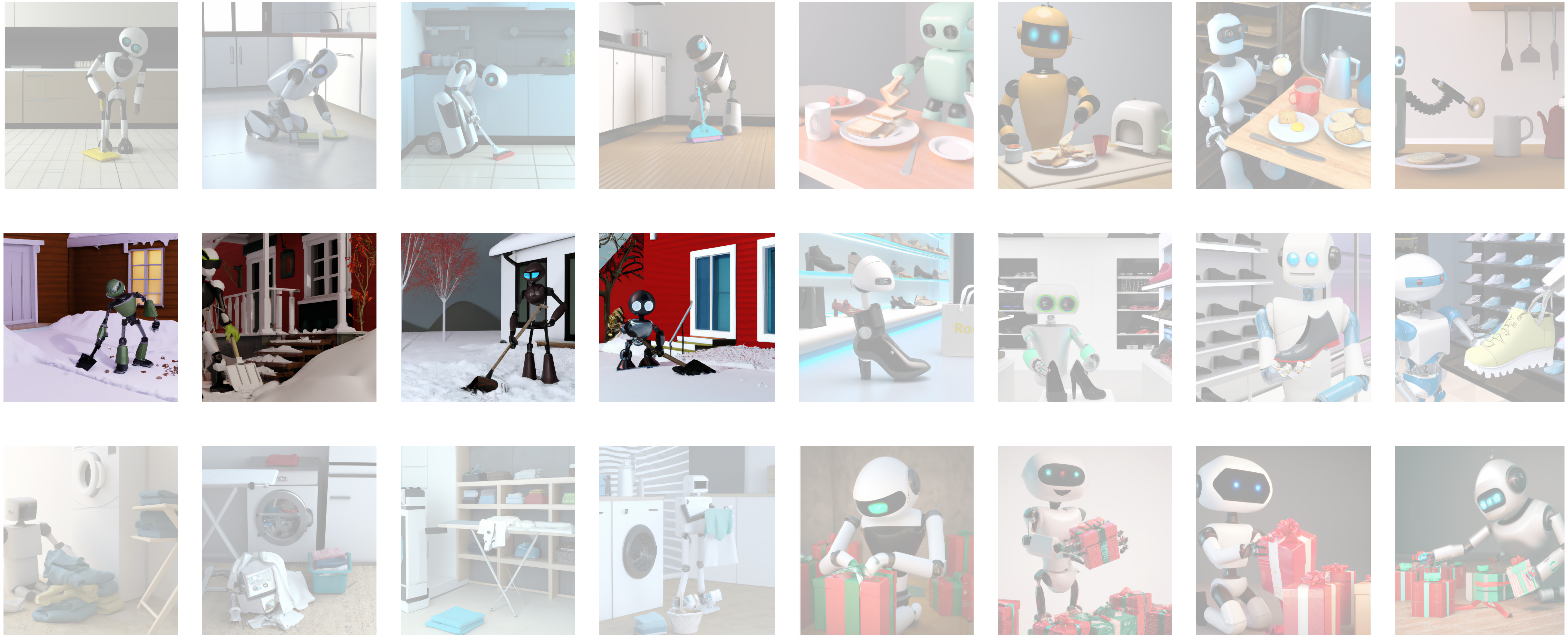
Images Generated by DALL-E 2

…cleaning the kitchen floor?



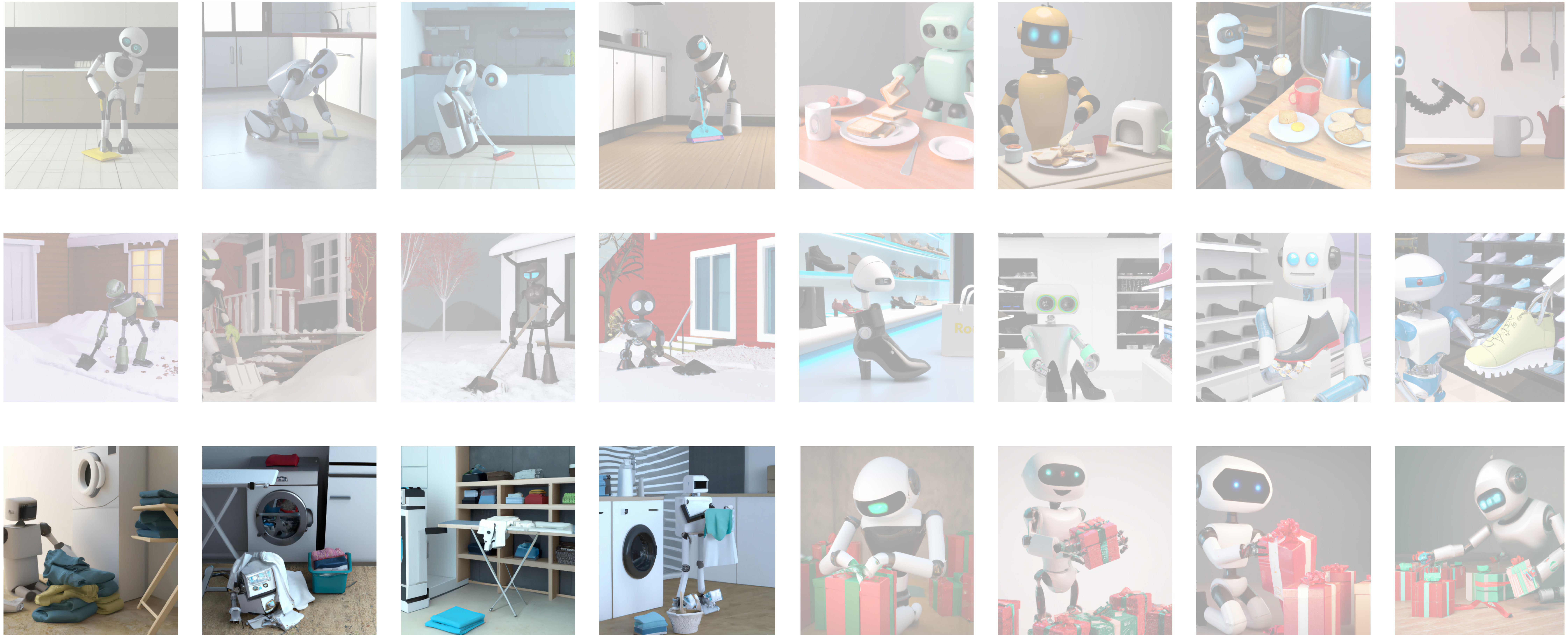
Images Generated by DALL-E 2

…shoveling the snow?



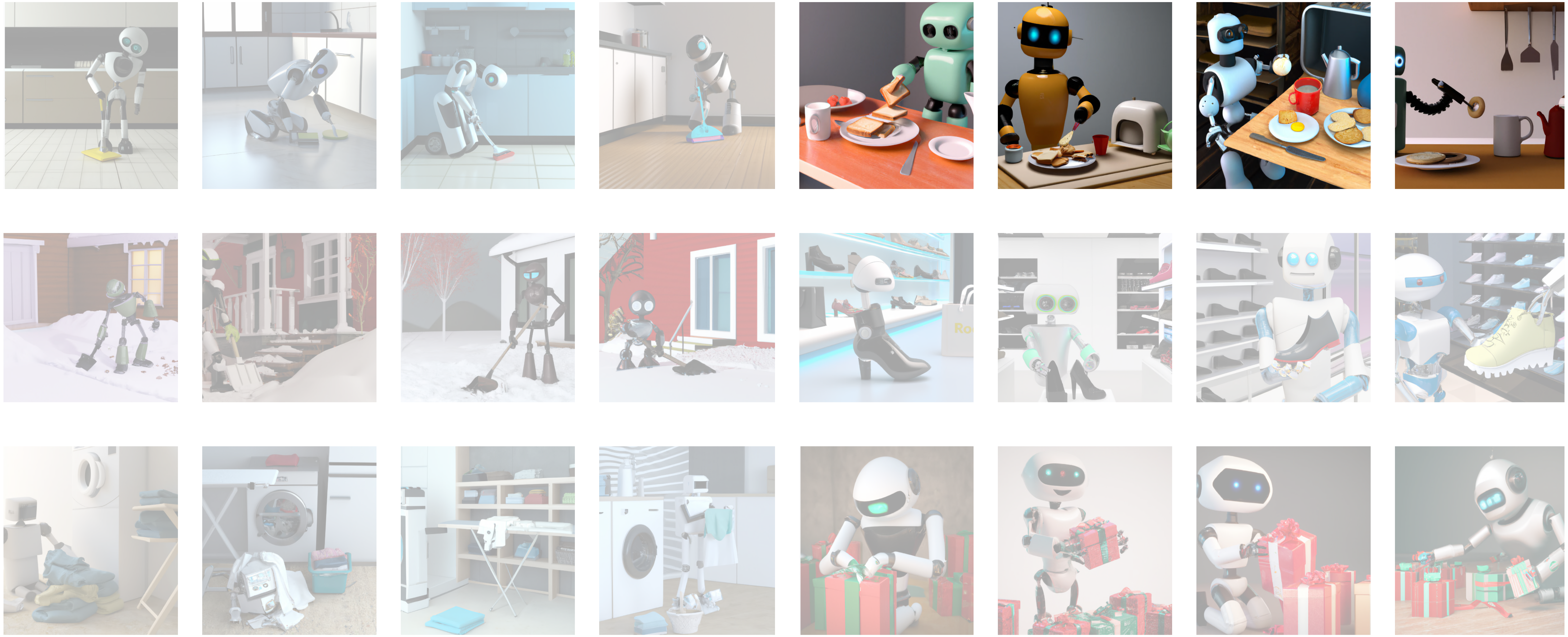
Images Generated by DALL-E 2

…folding laundry?



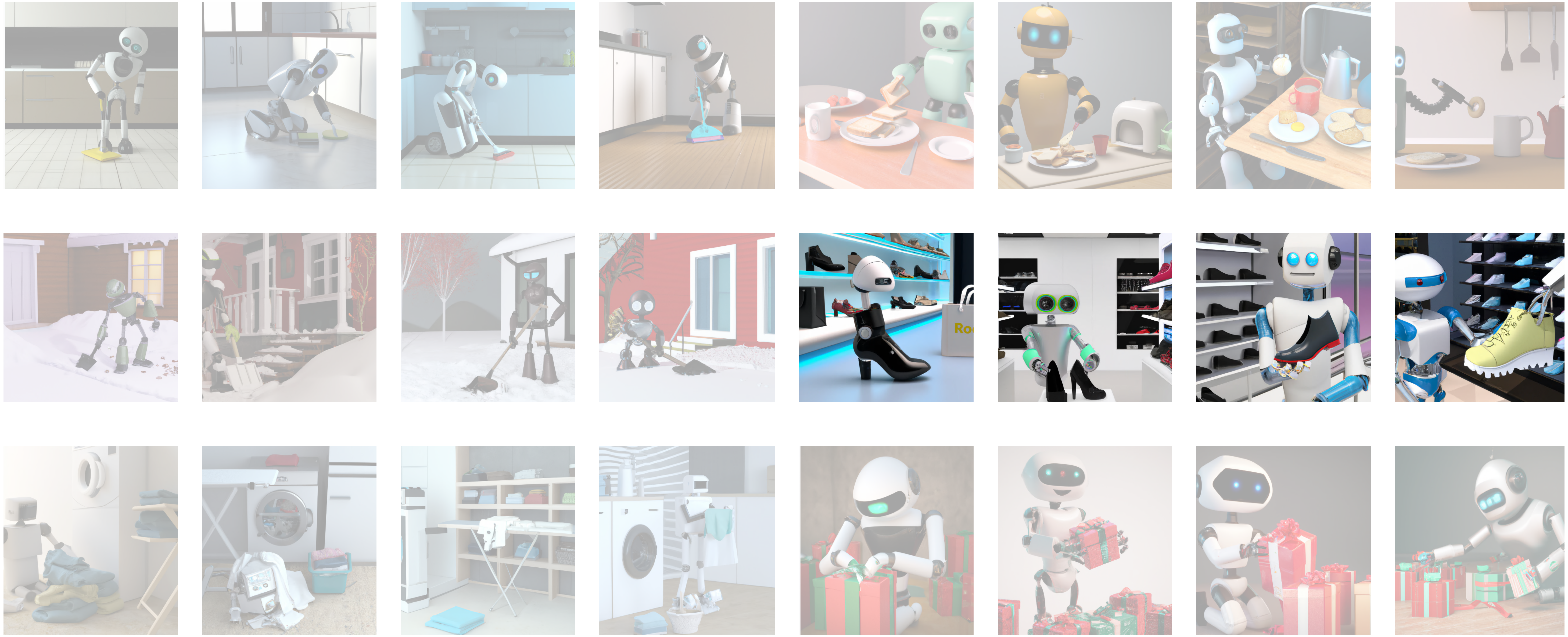
Images Generated by DALL-E 2

…cooking breakfast?



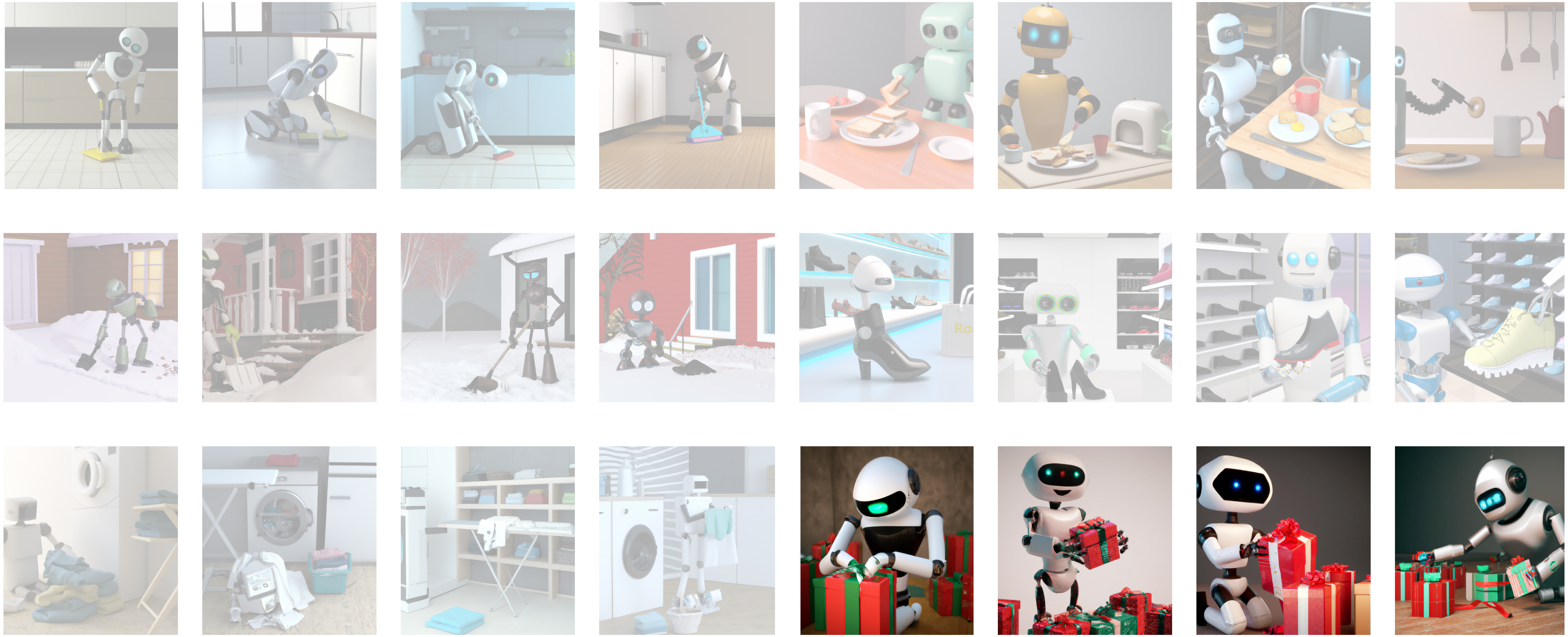
Images Generated by DALL-E 2

…buying shoes?



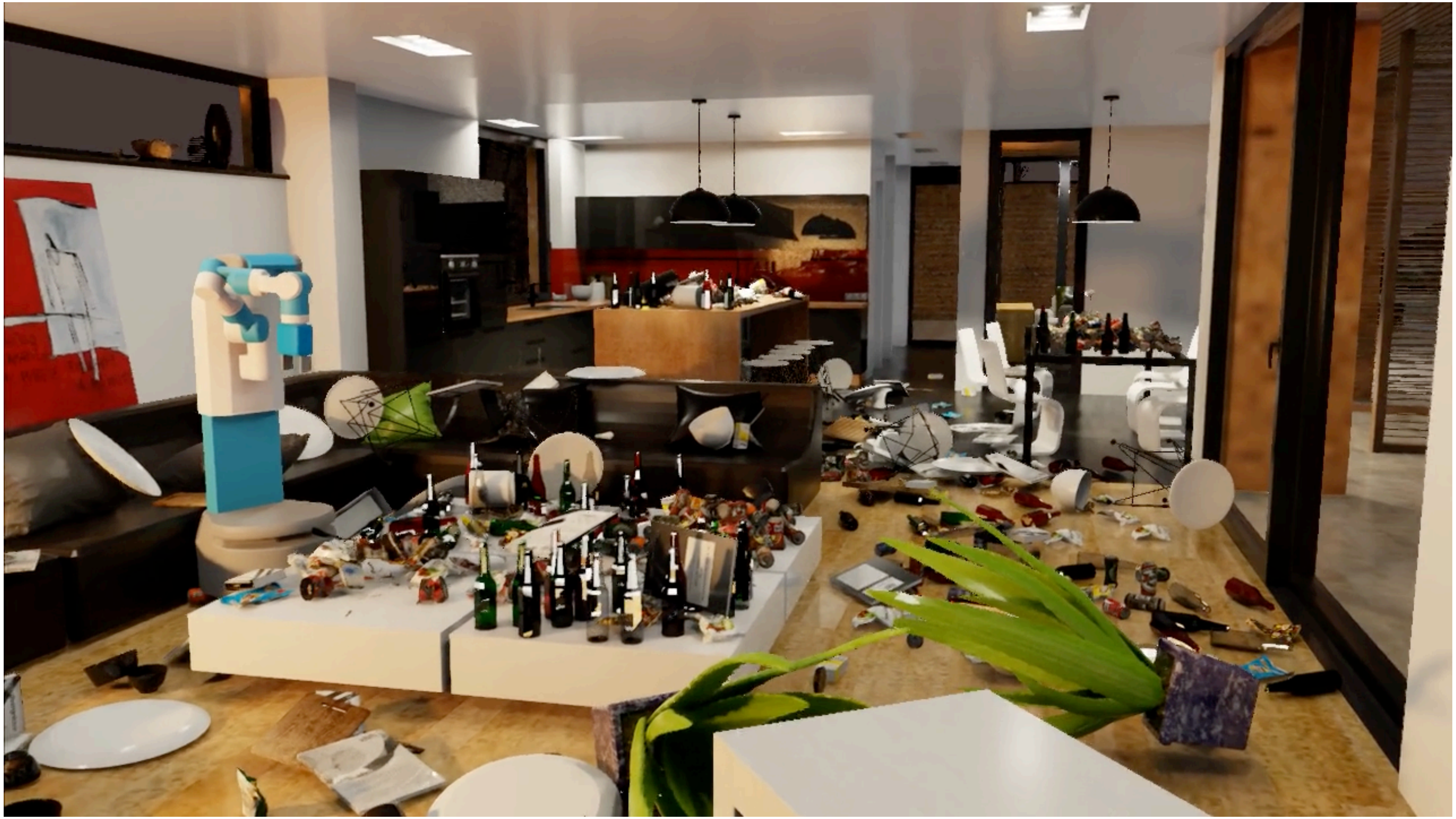
Images Generated by DALL-E 2

…opening Christmas gifts?



Images Generated by DALL-E 2

…cleaning after a wild party?



Robot learning benchmark today
Tasks hand-crafted or selected by researchers

What do people actually want robots to do?

Sources of data	"How much would you benefit if a robot did this for you?"	Participants

U.S. BUREAU OF LABOR STATISTICS

eurostat

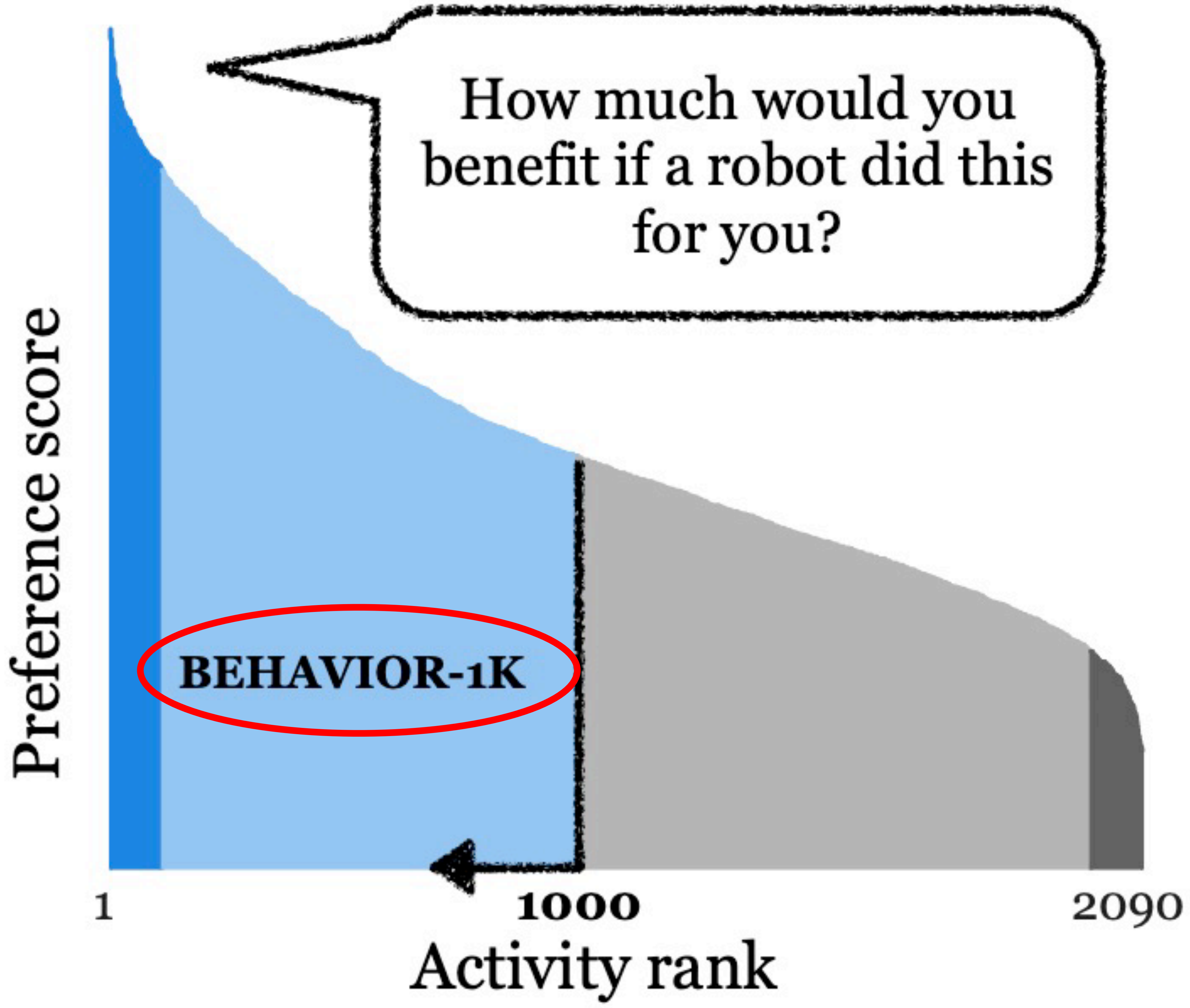
CENTRE FOR TIME USE RESEARCH

wikiHow to do anything

amazon mechanical turk

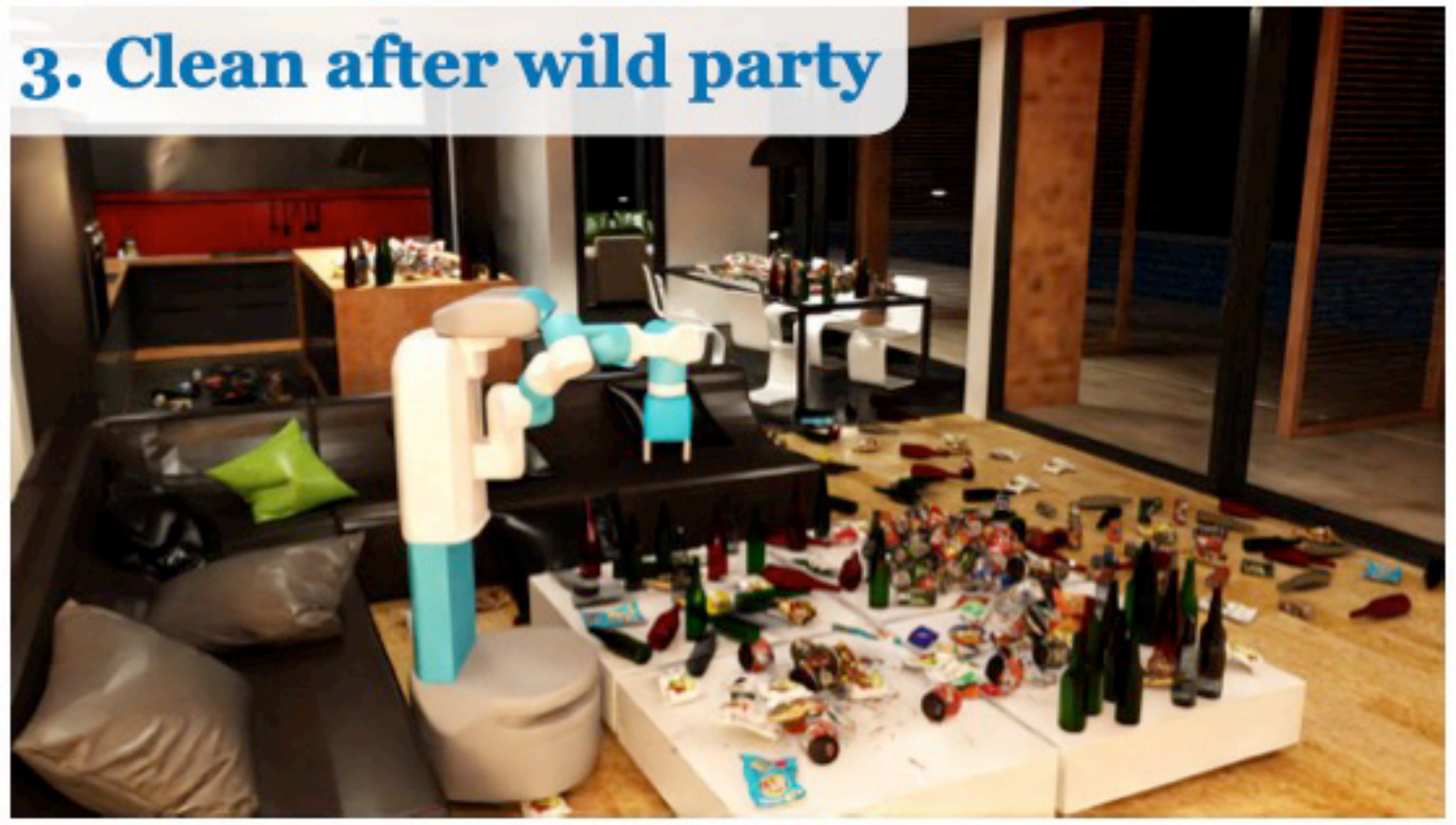
2000+ everyday activities	10-pt Likert scale	1,461 participants Age 20-78; 44%F 56%M Income $0-524k

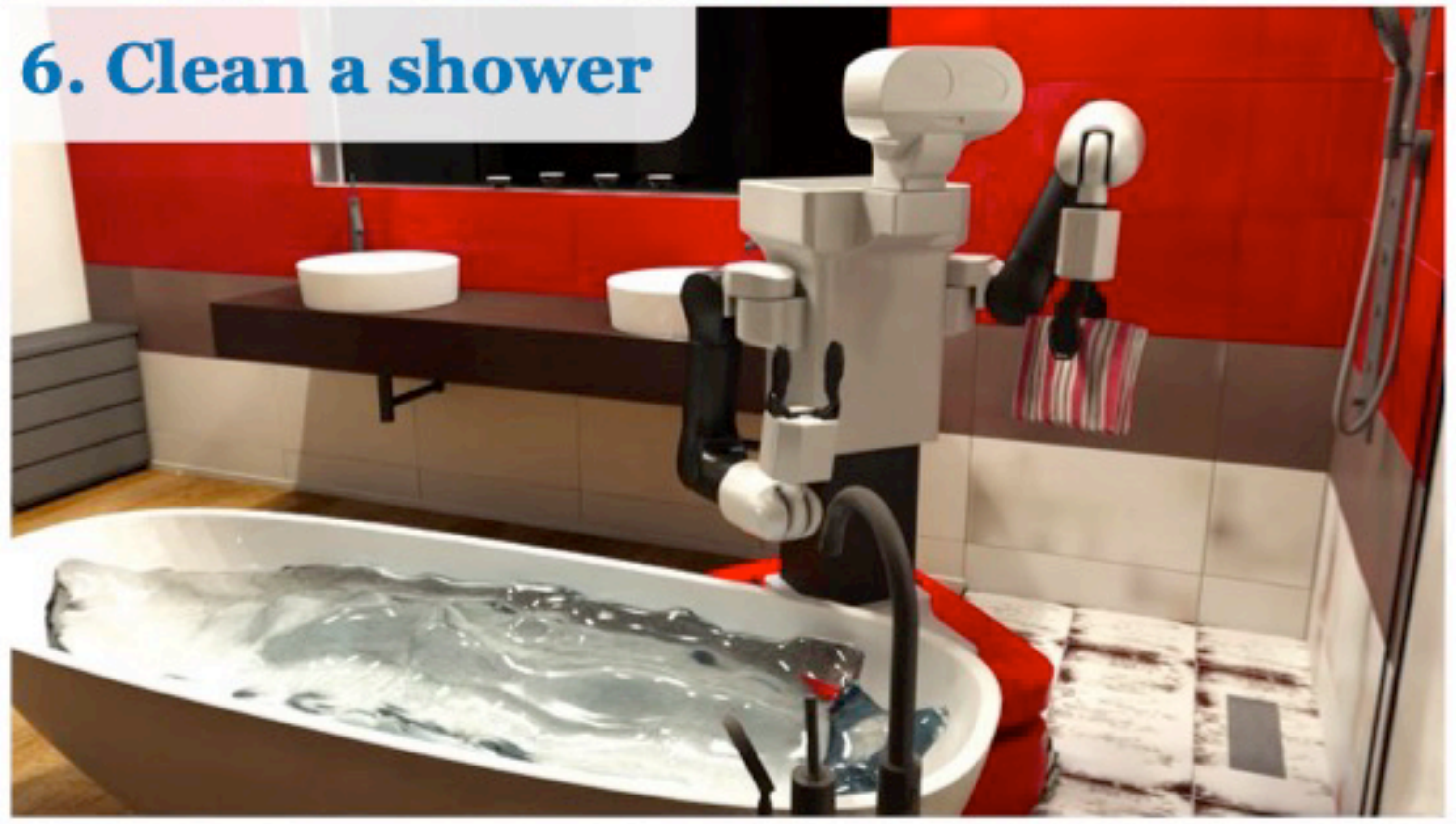
…people responded!



1. Wash floor
2. Clean bathrooms
3. **Clean after a wild party**
4. Clean floors
5. Mop floors
6. **Clean a shower**
7. Clean toilet
8. Clean bathtub

2083. Open X-mas presents
2084. Play snooker
2085. Balayage
2086. Throw darts
2087. Mix baby cereal
2088. Buy a ring
2089. Play squash
2090. Opening presents





What’s needed for a robot to learn to solve a task?

3. Clean after a wild party

“What”…

does it mean to “clean after a wild party”?

“Where”…

should this cleaning occur (environments / scenes)?

“How”…

can a robot learn to clean?

“What”…defines a **task**?

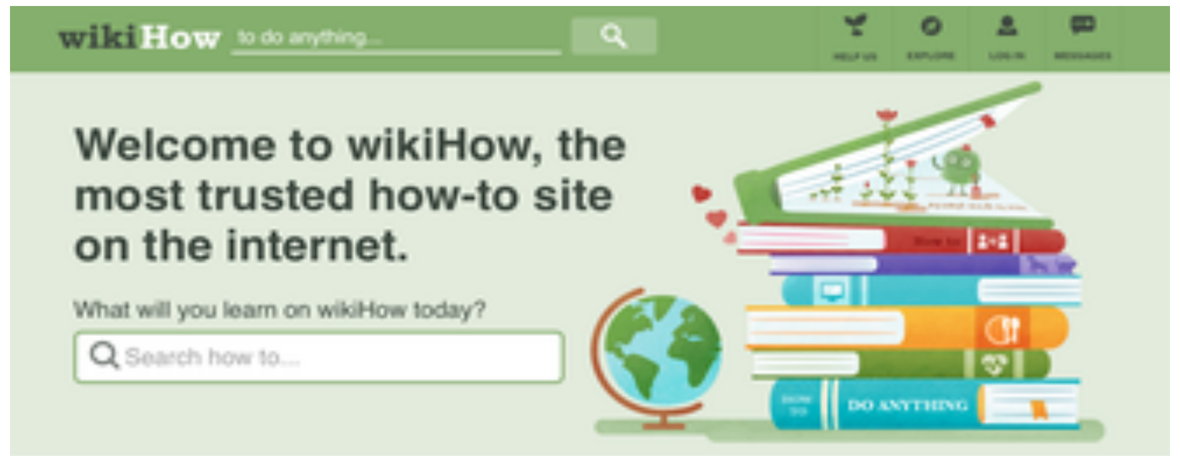
Prior work info
hand specified, ad-hoc, etc.

“What”…defines a **task**?

Our method:
Programmatic, standardized (and crowdsourced) using BDDL!

BEHAVIOR-1K activity definitions

Combining human and machine effort to define 1,000 activities

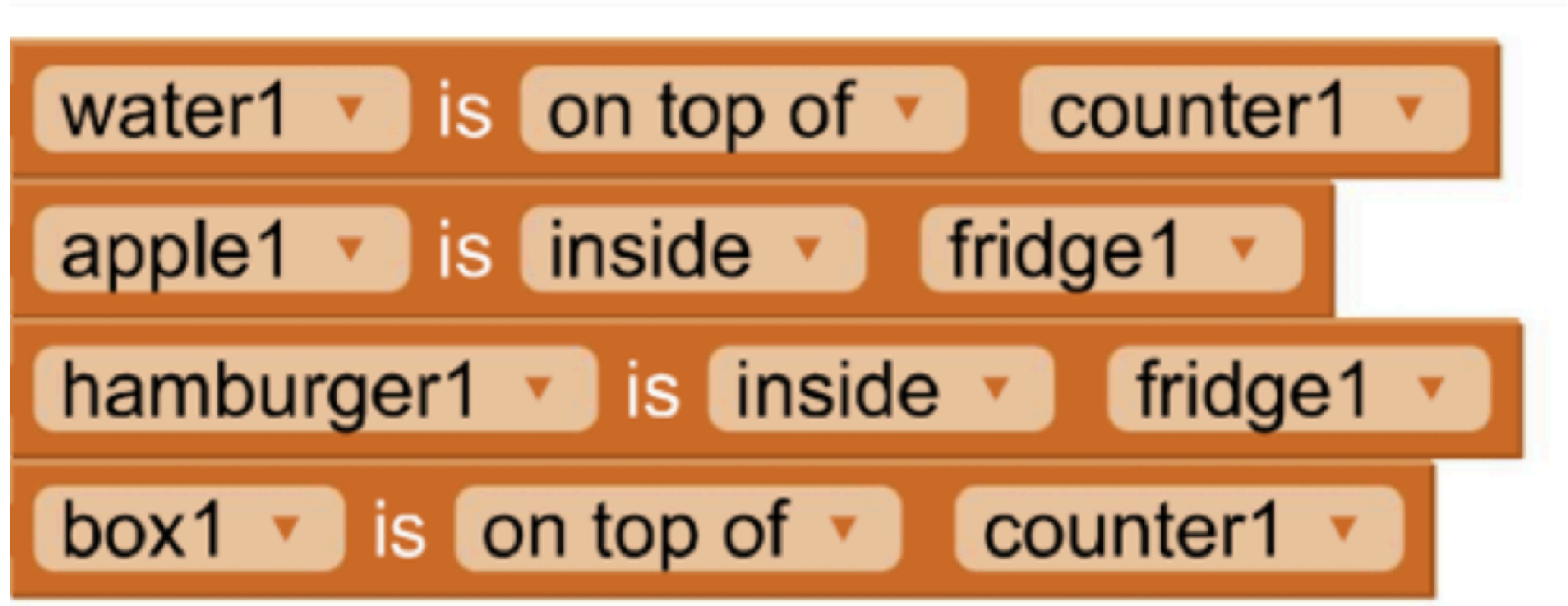


What objects are needed for each activity?

Properties: cookable, sliceable, Freezable, burnable, deformable, …

Cooking temperature: 58°C

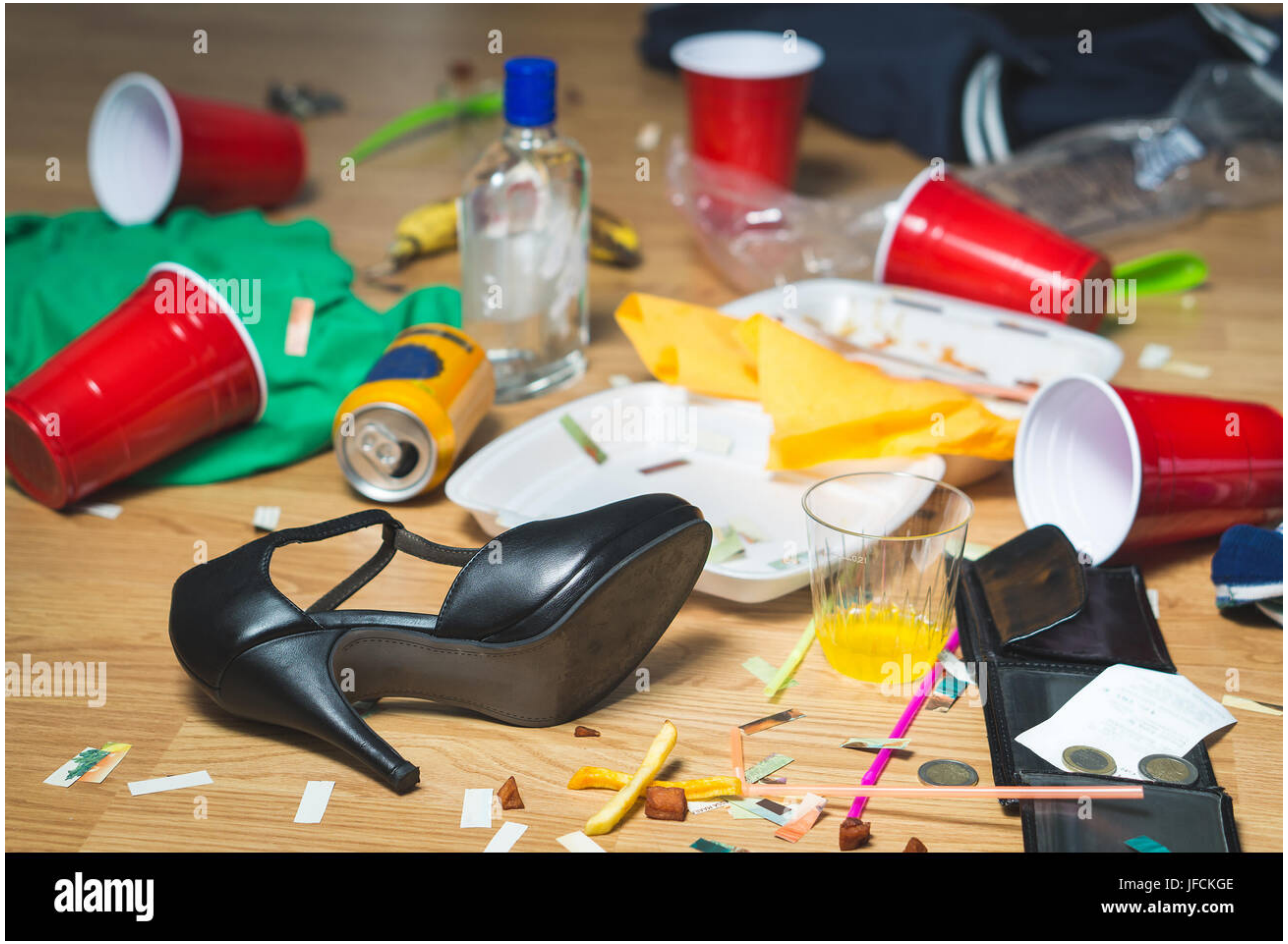
What object properties are necessary to define the activities?



What is a flexible and natural way for humans to define activities?

"**Where**"…should the task be done?

Real Life?
(wild parties are
expensive):)

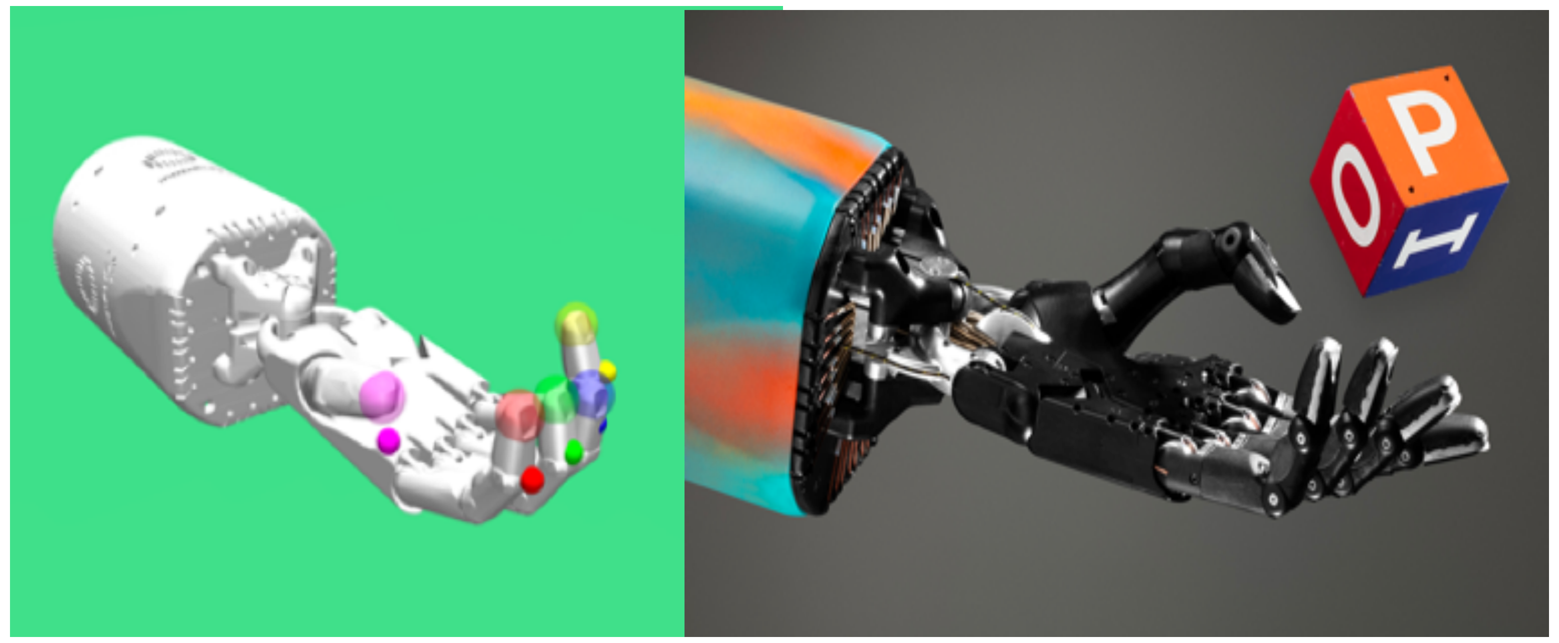


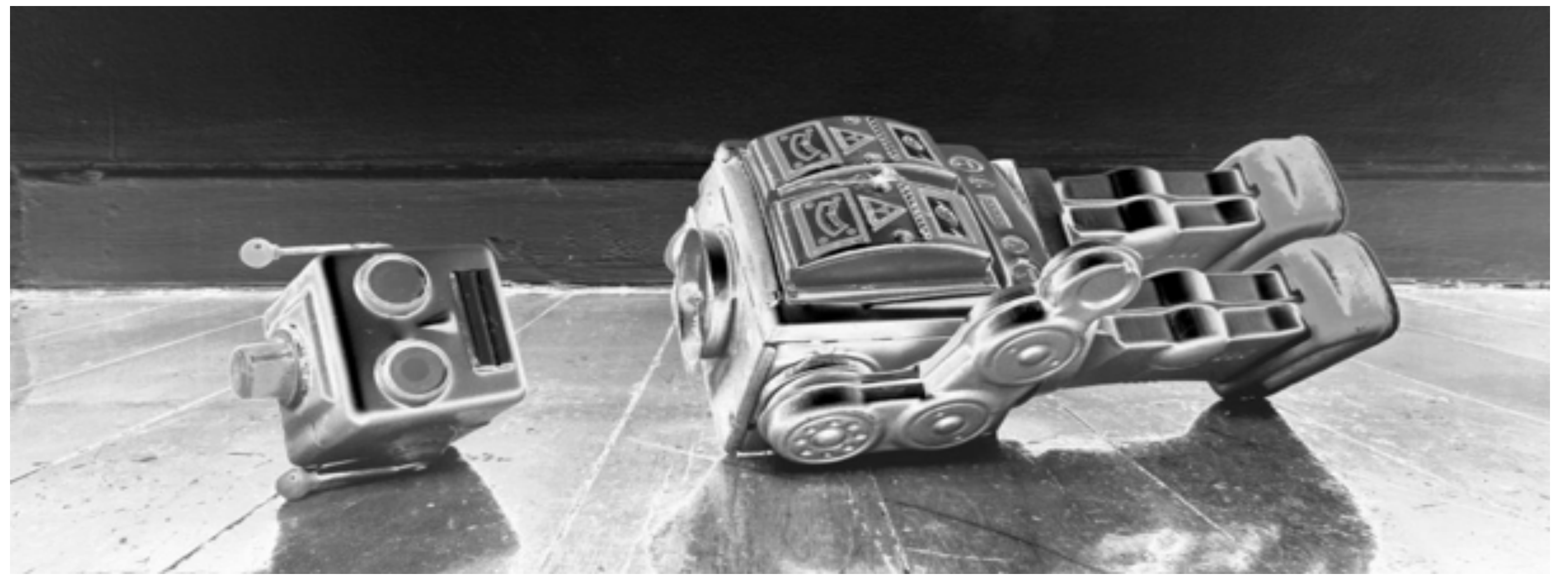
What about Simulation?

1 Fast training, transferrable with sim2real

2 Safe for robots and environment

3 Accessible and reproducible benchmarking

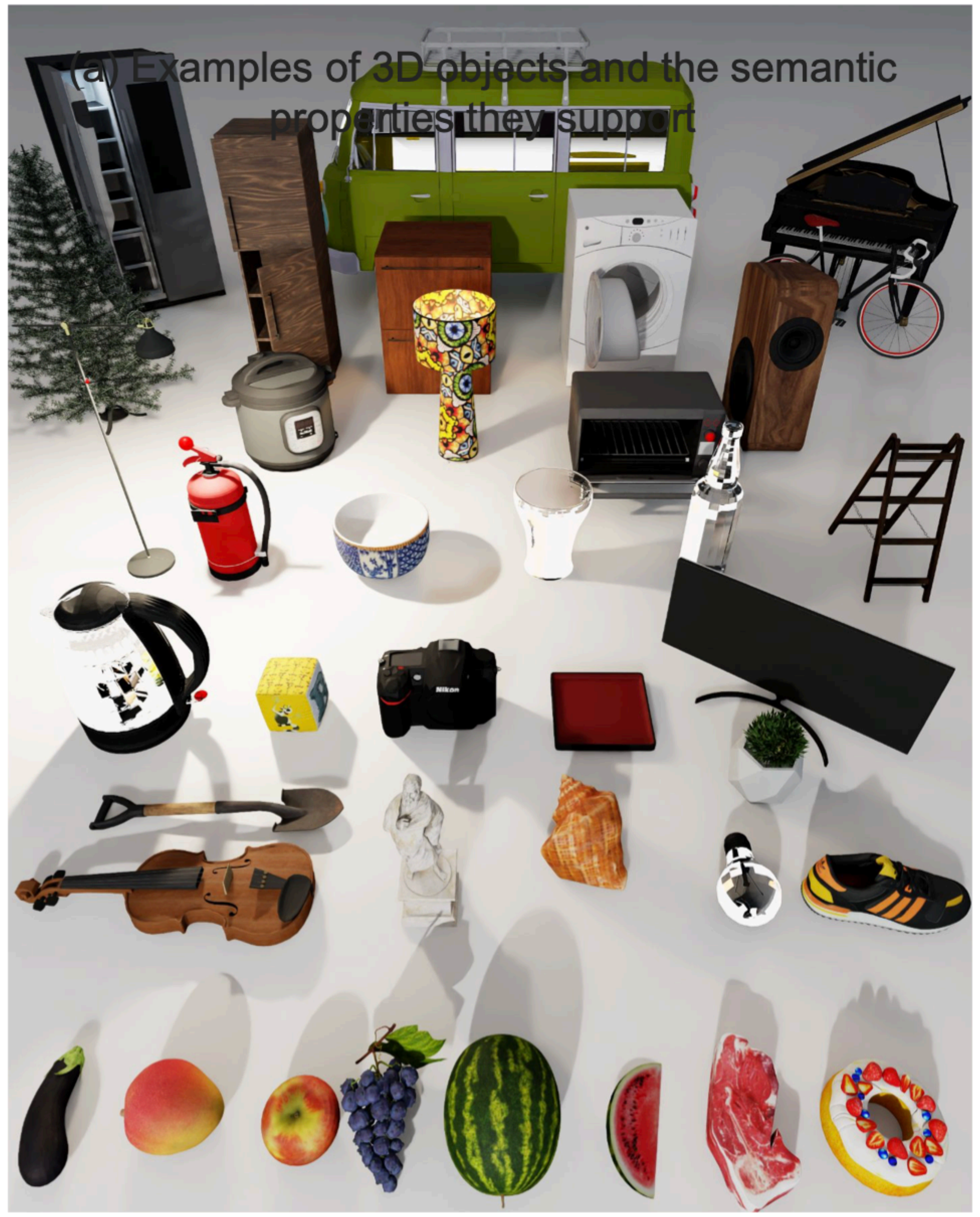


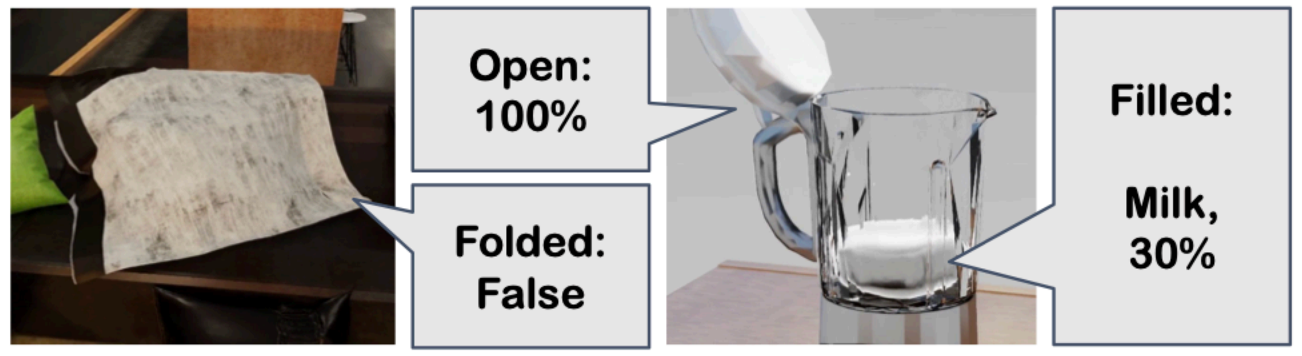


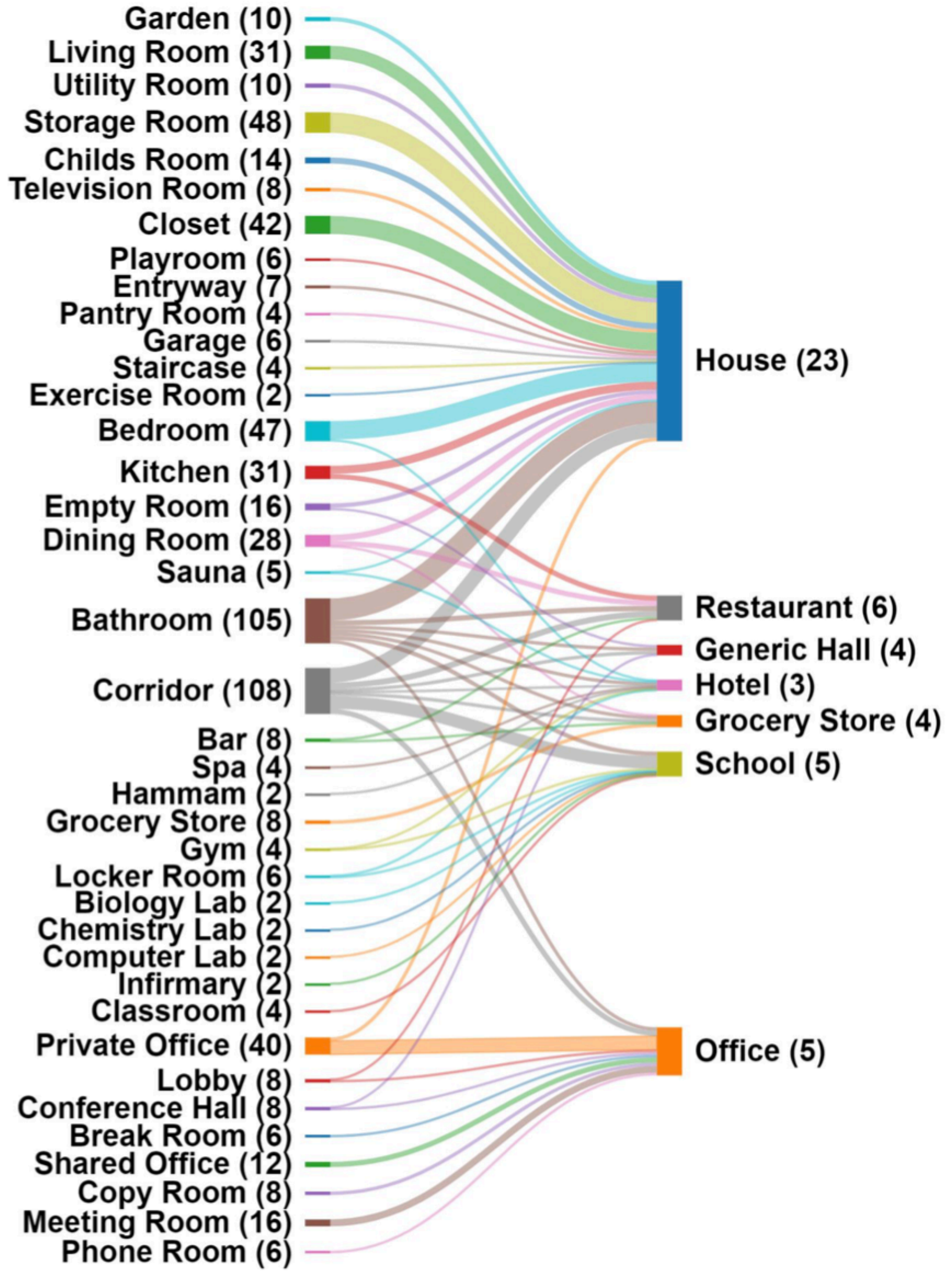
Dexterity, OpenAI'19; ImageNet, 2009 [Deng et al.]

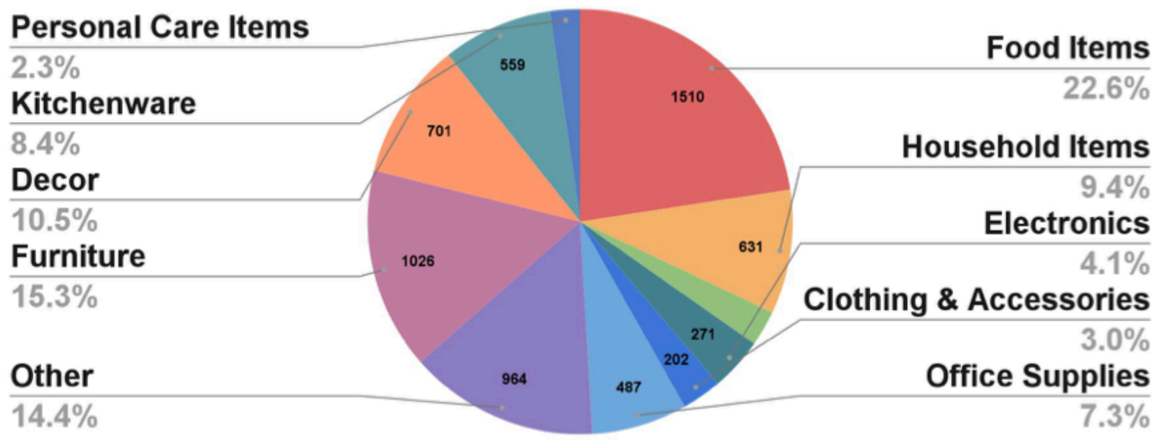
BEHAVIOR-1K Asset Suite

Large-scale, diverse, curated dataset of realistic scenes and objects

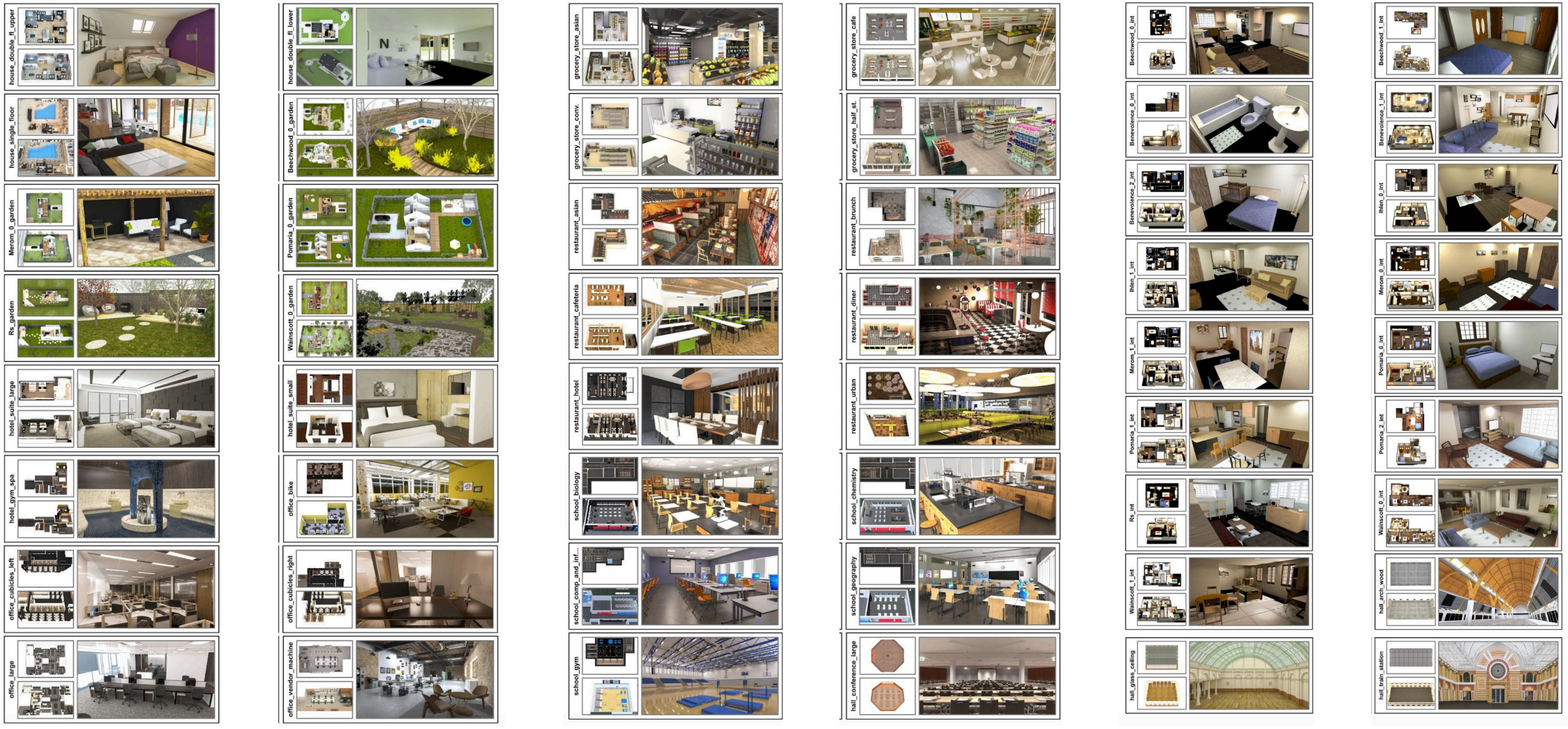




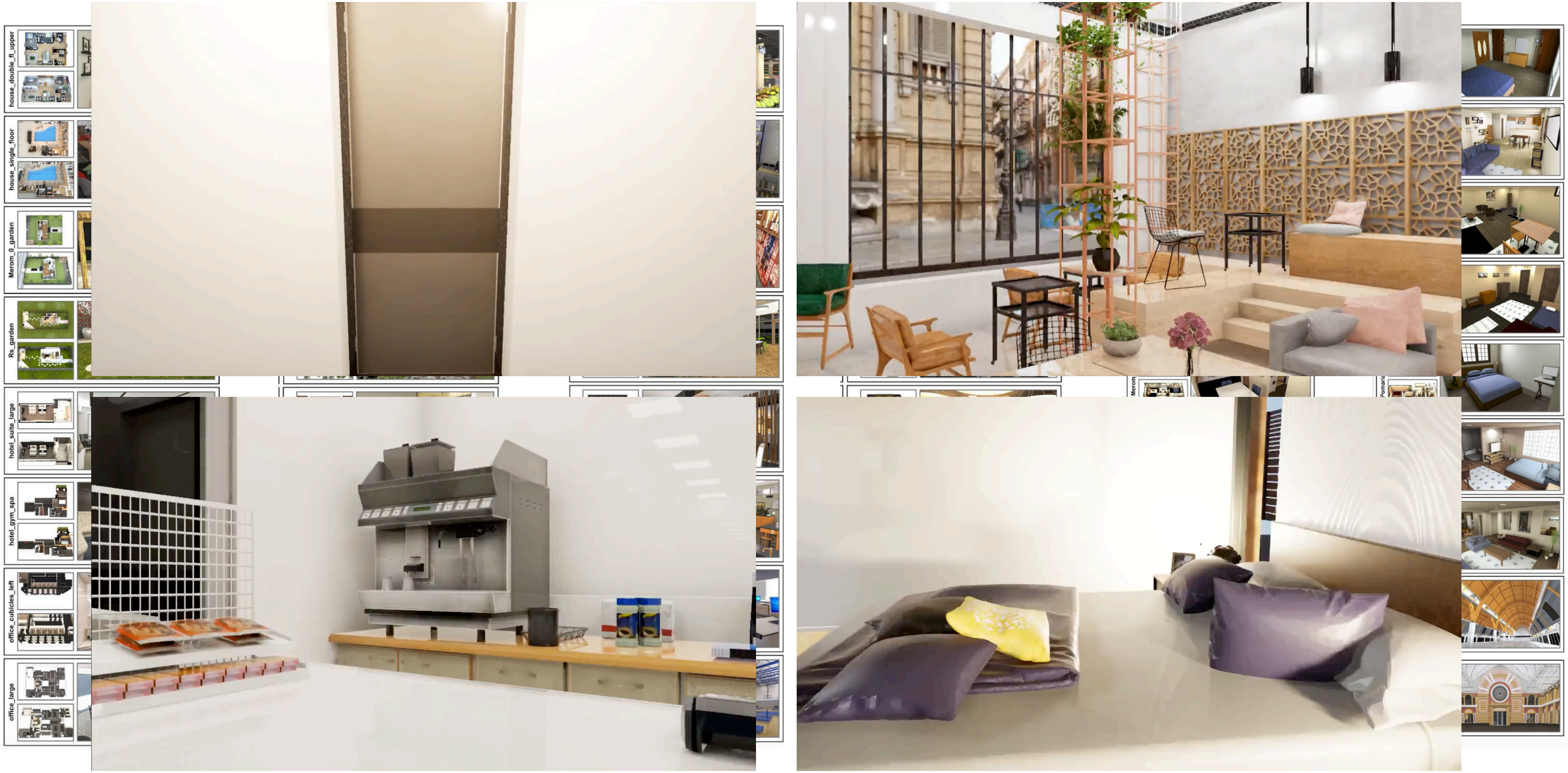




A focus on ecological and diverse scenes and objects:
50 fully interactive environments



A focus on ecological and diverse scenes and objects:
50 fully interactive environments

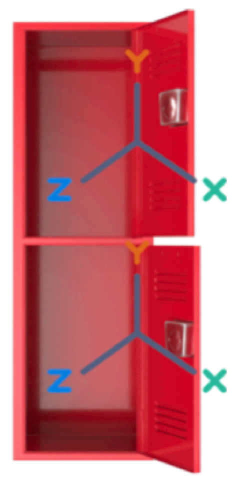


A focus on ecological and diverse scenes and objects: **1200+ categories, 5000+ 3D models, 30+ properties**

Semantic

Properties: cookable, sliceable, freezable, burnable, deformable
…
Cooking temperature: 58°C
…

Physical



Articulation annotation (joint type, origin, axis, limit)
Mass, friction, CoM, …
Canonical size and orientation

TODO

better asset visualization (e.g. image from BVS paper)

Linking Task Specifications and Task Objects

TODO: Example BDDL

TODO: Example Assets

TODO: Section on Asset Curation Process
i.e.: How it’s done, highlights, etc.

TODO: Example Asset
Vis

TODO: Example Asset
Vis

B1K Dashboard
i.e.: This is where BDDL Task Defs, Object Visualization, Properties, intersect

TODO

Dashboard & knowledge base
(you can use it independently of OG)

Finally, the most important question…

“How”…

can a robot learn to
solve a task?

Finally, the most important question…

We don't have an answer):

But! We want to propose a tool that can ***accelerate*** research progress towards solving the B1K benchmark…


Exit


Exit

OmniGibson

OmniGibson

OmniGibson

why omnigibson?

why not... Habitat

- fast
- scalable

⚠ **non-interactive**

OmniGibson

why not... AI2THOR

- configurable scenes
- diverse object states

⚠ **simplified robot action**

Habitat

OmniGibson

why not... TDW ThreeDWorld

- cloth & fluid
- sound

⚠ **only change of pose**

AI2THOR

Habitat

OmniGibson

Realism score [1-5] ↑

OmniGibson

3.20

Pushing realism in perception and physics

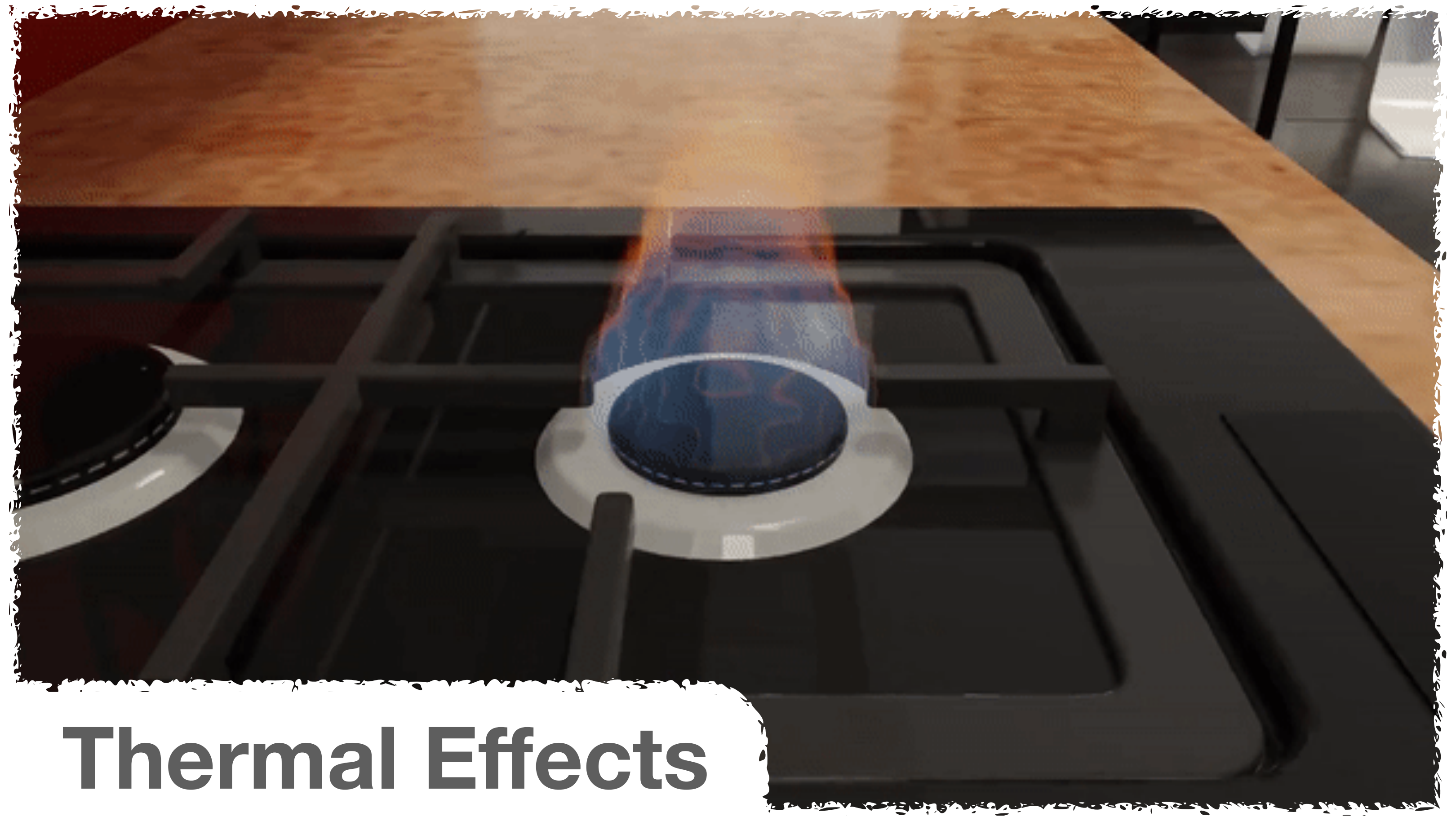
Habitat 1.74

TDW ThreeDWorld 1.65

AI2THOR 1.73

iGibson 1.69

OmniGibson | Pushing realism in perception and physics


Thermal Effects


Transition Machines


Lights & Reflections


Fluids


Deformables


Transparency

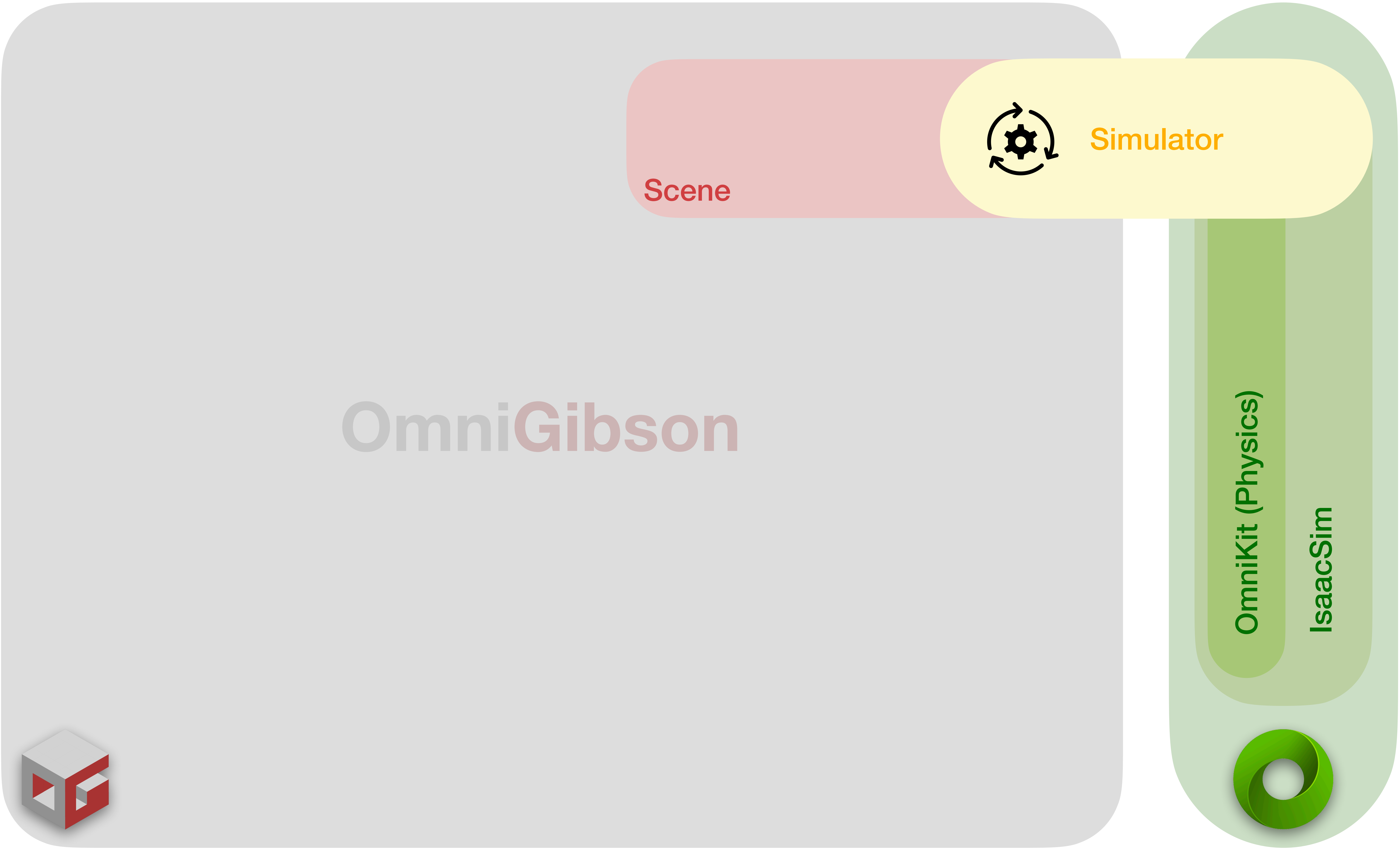
OmniGibson

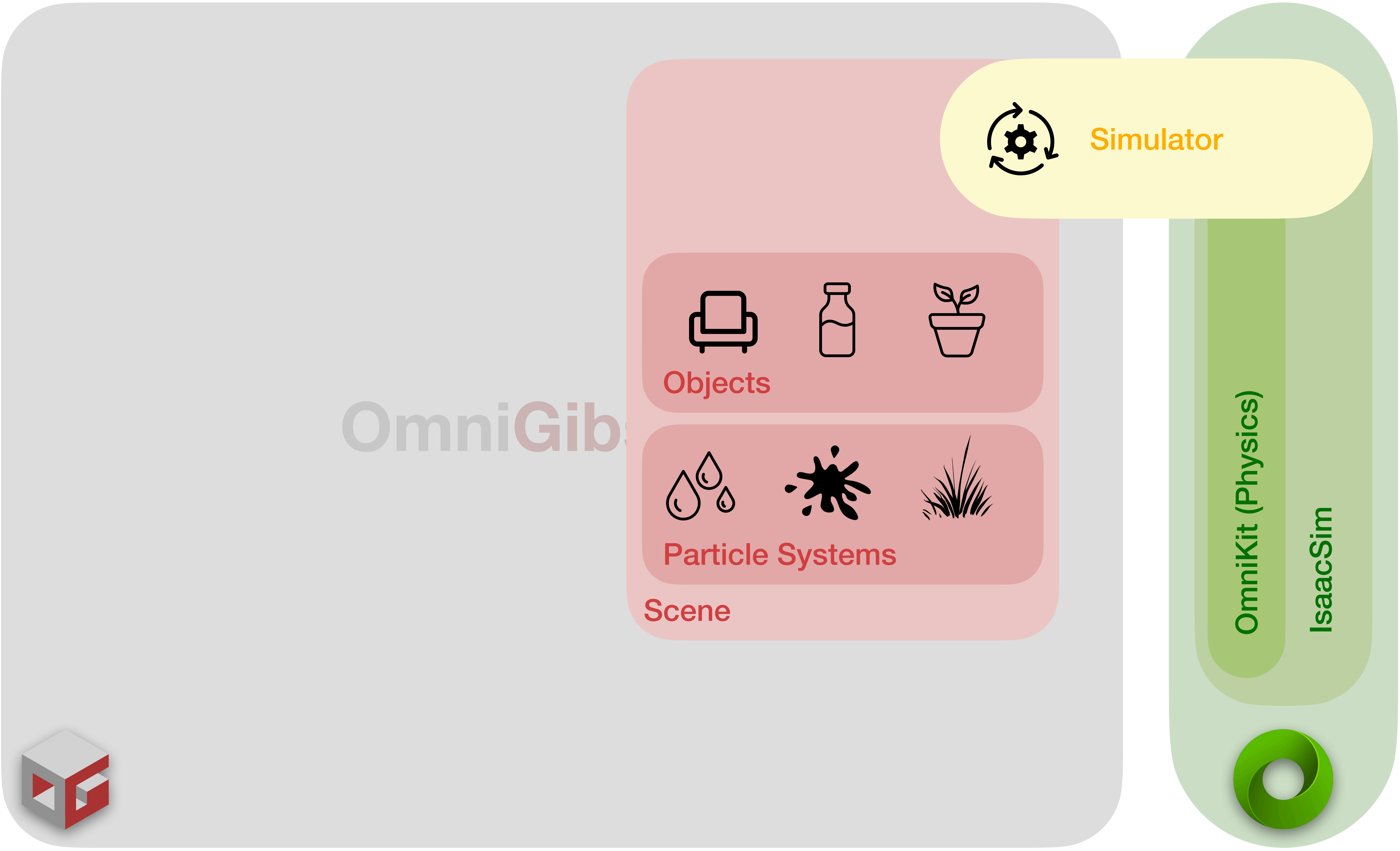
OmniGibson

architecture overview

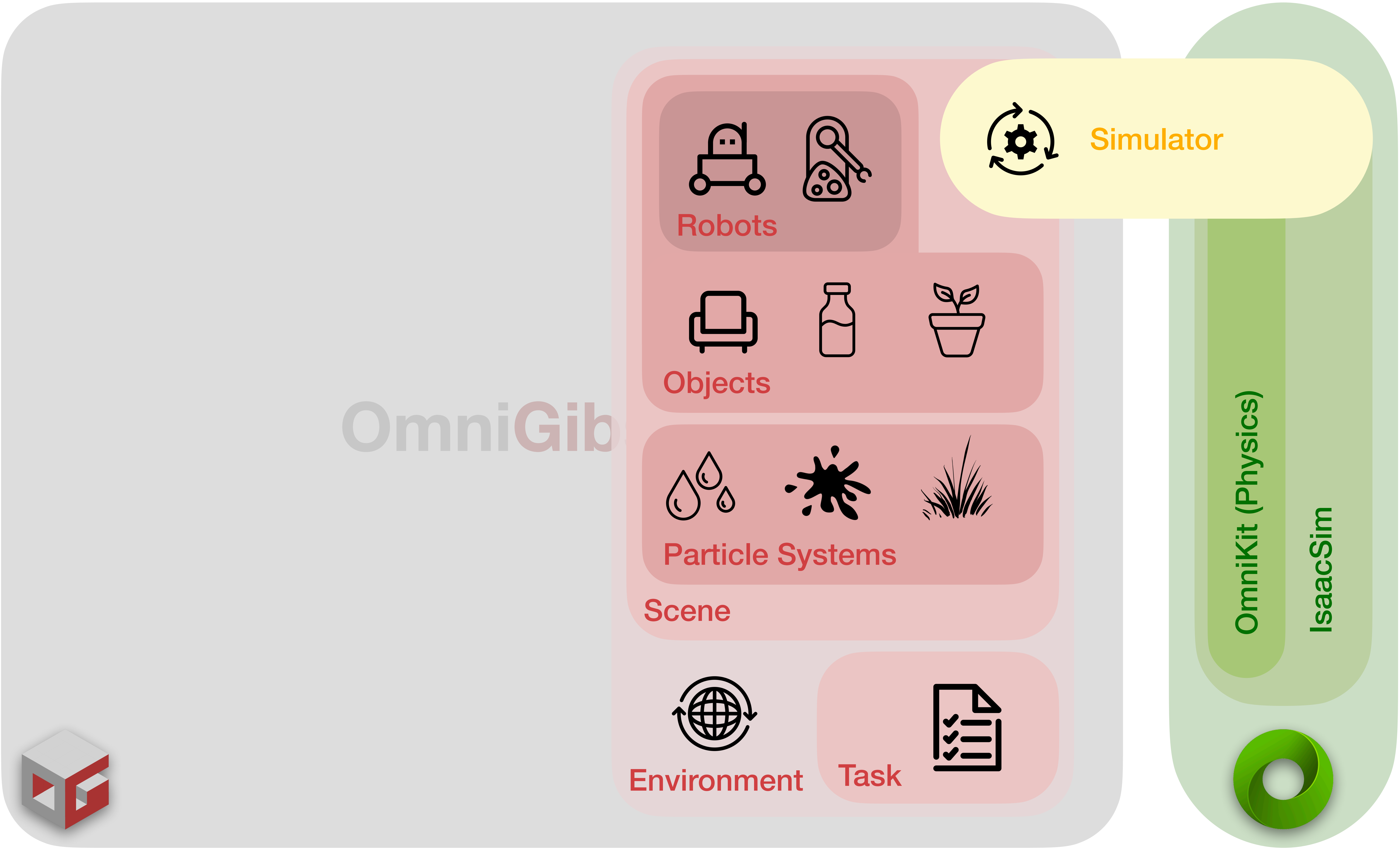
OmniGibson

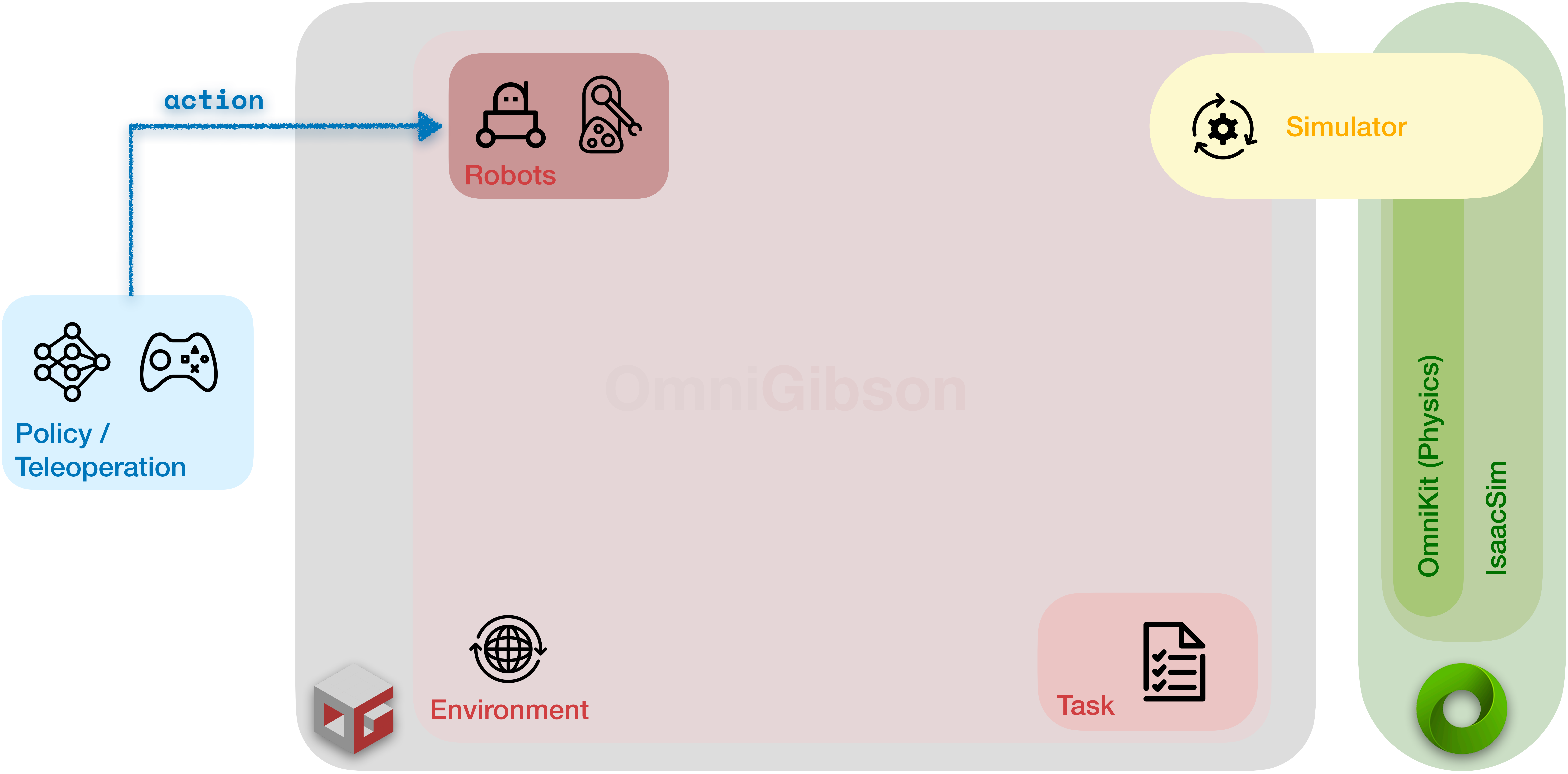

OmniGibson
Simulator
OmniKit (Physics)
IsaacSim


Simulator
Scene
OmniGibson
OmniKit (Physics)
IsaacSim

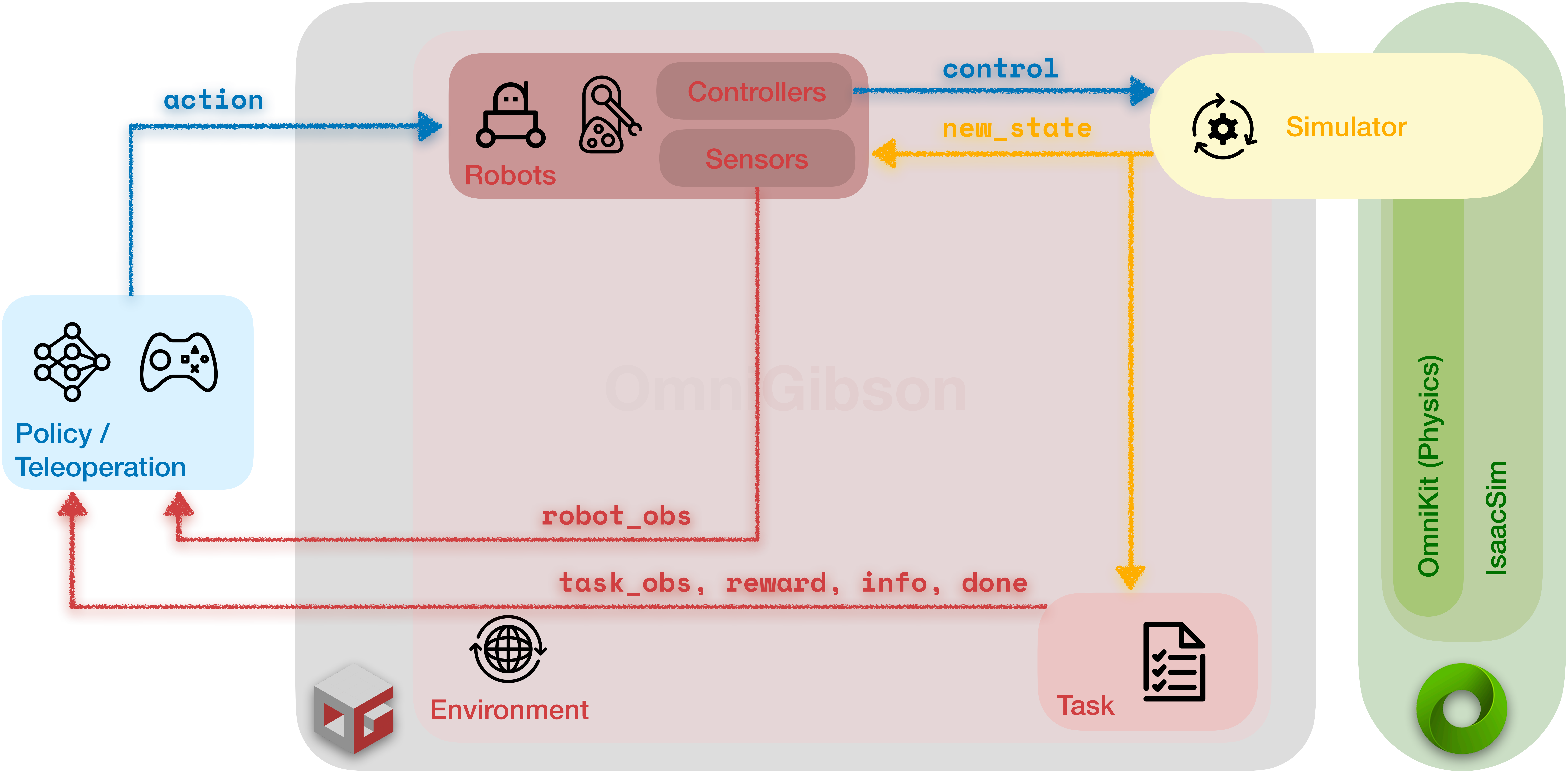


OmniGibs
Simulator
Objects
Particle Systems
Scene
OmniKit (Physics)
IsaacSim


OmniGib
Robots
Objects
Particle Systems
Scene
Environment
Task
Simulator
OmniKit (Physics)
IsaacSim



action
Robots
Simulator
OmniGibson
Policy /
Teleoperation
OmniKit (Physics)
IsaacSim
Environment
Task



action
control
new_state
Controllers
Sensors
Robots
Simulator
Policy /
Teleoperation
OmniGibson
robot_obs
task_obs, reward, info, done
Environment
Task
OmniKit (Physics)
IsaacSim

OmniGibson

OmniGibson

highlighted features


1
Large-Scale
Scene Generation
2
Modular
Robot Control
OmniGibson
4
Customizable
Physics Transitions
3
Kinematic & Semantic
Object States

OmniGibson

Easy as...

```
import omnigibson as og

env = og.Environment(cfg)
obs, rew, done, info = env.step([])
```

Zero-length action when
no robot is loaded

1 **Large-Scale Scene Generation**

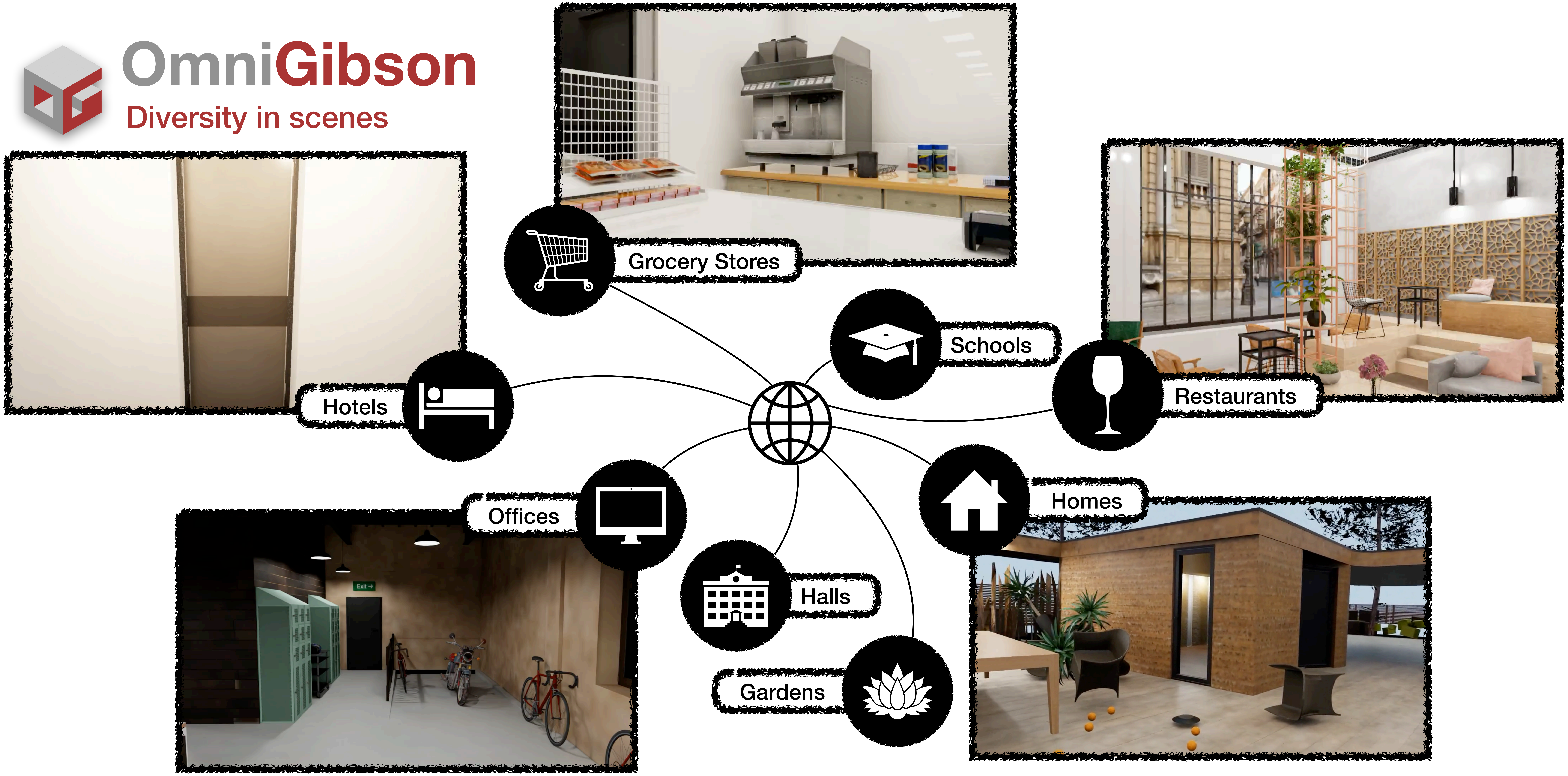
2 Modular Robot Control

3 Kinematic & Semantic Object States

4 Customizable Physics Transitions

OmniGibson

Diversity in scenes



1. **Large-Scale Scene Generation**
2. Modular Robot Control
3. Kinematic & Semantic Object States
4. Customizable Physics Transitions

OmniGibson

Diversity in scenes

50 large-scale scenes

3 5 4 5 4 5 18 6

```
import omnigibson as og

env = og.Environment({
  "scene": {
    "type": "InteractiveTraversableScene",
    "scene_model": "grocery_store_conv",
  },
})
```

1 Large-Scale Scene Generation

2 Modular Robot Control

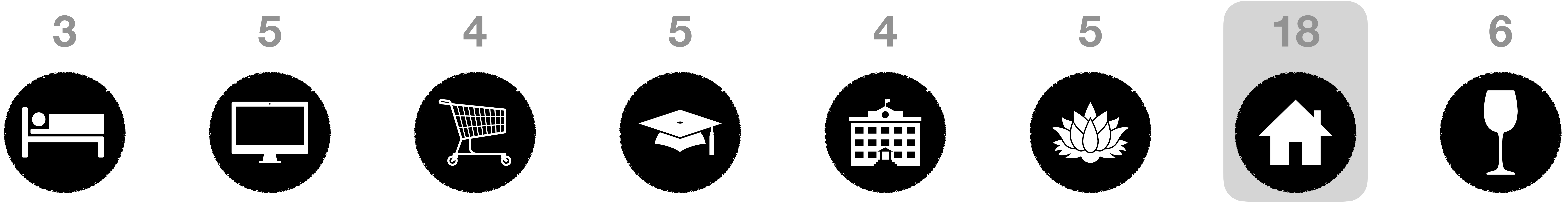
3 Kinematic & Semantic Object States

4 Customizable Physics Transitions

OmniGibson

Diversity in scenes

50 large-scale scenes




```
import omnigibson as og

env = og.Environment({
   "scene": {
     "type": "InteractiveTraversableScene",
     "scene_model": "house_single_floor",
   },
})
```

1 **Large-Scale Scene Generation**

2 Modular Robot Control

3 Kinematic & Semantic Object States

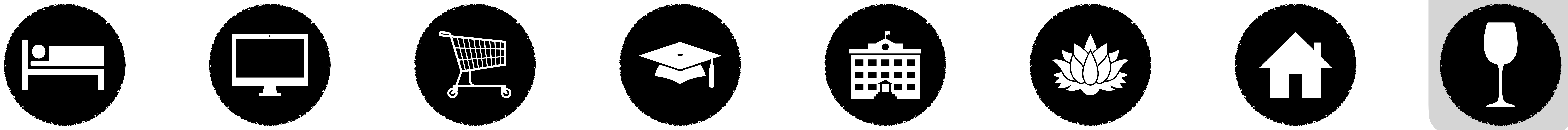
4 Customizable Physics Transitions

OmniGibson

Diversity in scenes

50 large-scale scenes

3 5 4 5 4 5 18 6



```
import omnigibson as og

env = og.Environment({
  "scene": {
    "type": "InteractiveTraversableScene",
    "scene_model": "restaurant_brunch",
  },
})
```

1 Large-Scale Scene Generation

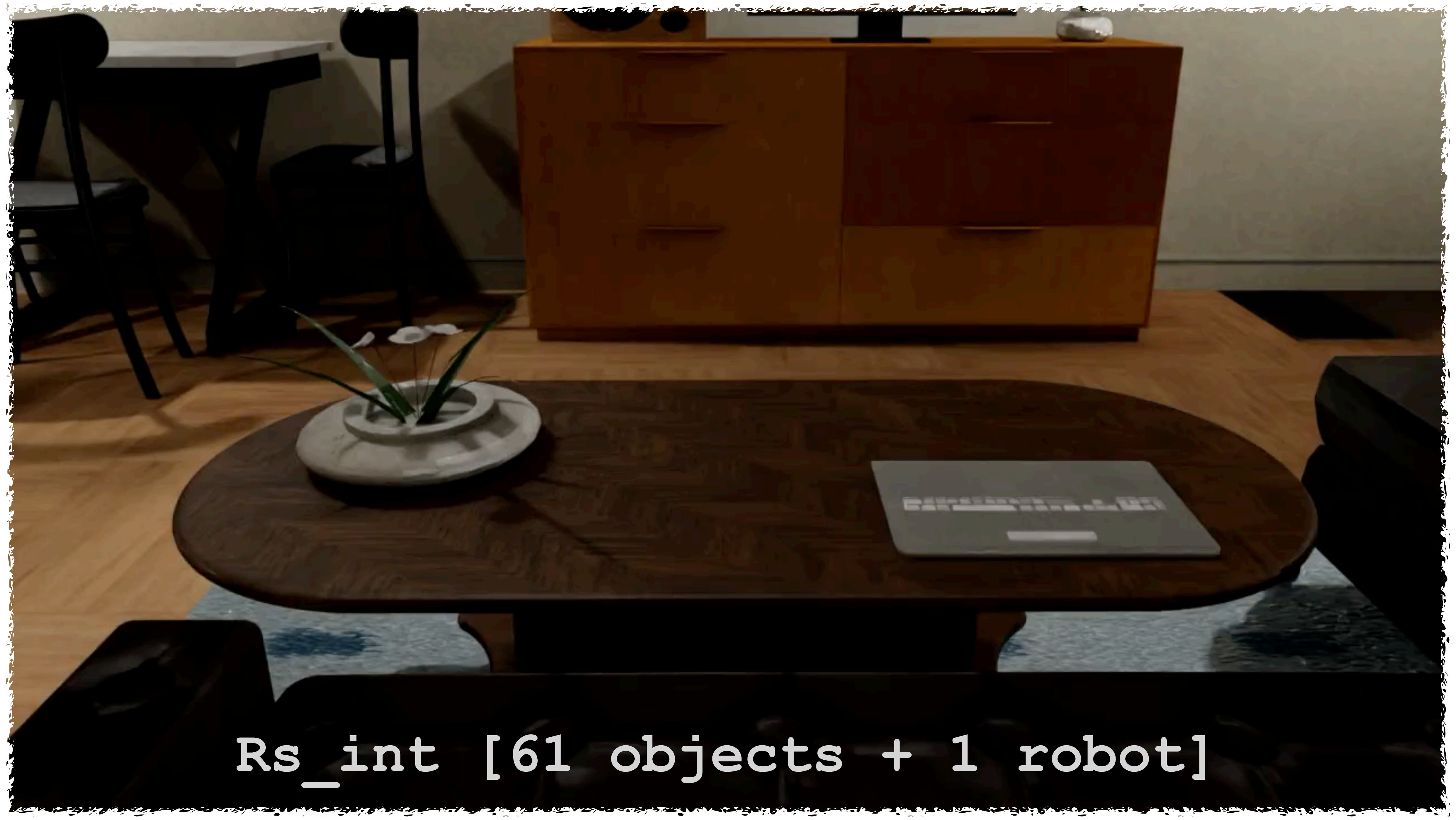
2 Modular Robot Control

3 Kinematic & Semantic Object States

4 Customizable Physics Transitions

OmniGibson

Performance



Rs_int		restaurant_hotel
73 FPS	With object states	64 FPS
78 FPS	No object states	68 FPS
212 FPS	No object states / No rendering	416 FPS

1 **Large-Scale Scene Generation**

2 Modular Robot Control

3 Kinematic & Semantic Object States

4 Customizable Physics Transitions

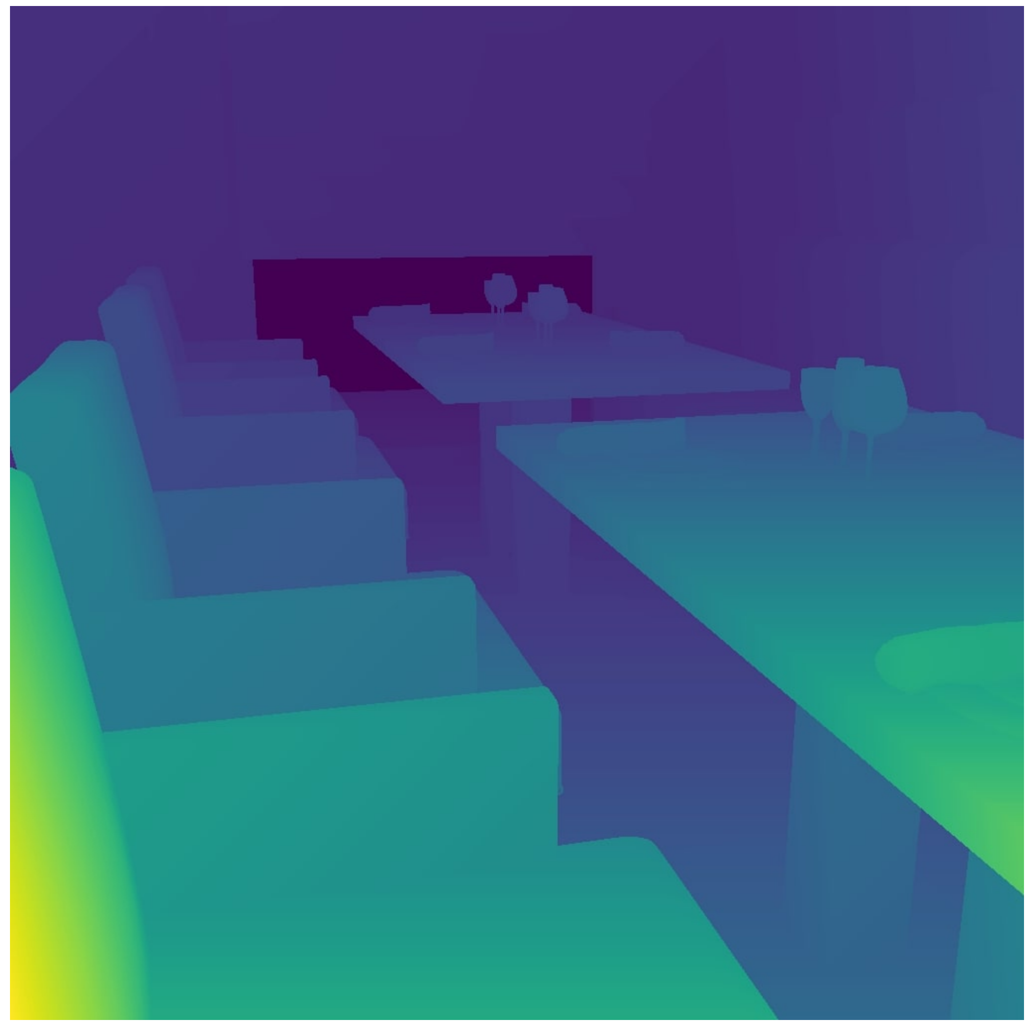
OmniGibson

Robot Observations

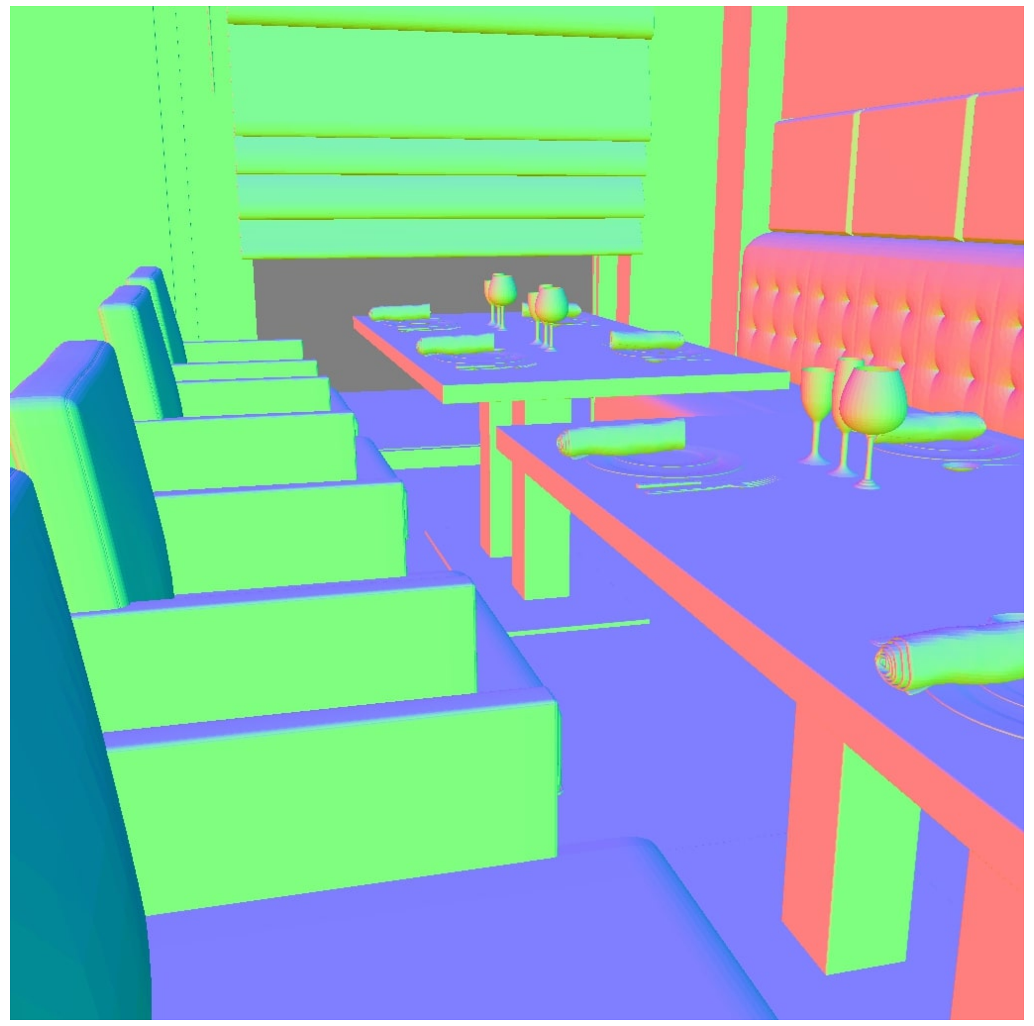
```
# Modify obs modalities dynamically!
robot = env.robots[0]
robot.add_obs_modality("normal")
robot.add_obs_modality("seg_instance")
robot.remove_obs_modality("rgb")
obs = robot.get_obs()
```

obs["robot0:Camera_sensor_eyes_rgb"]

obs["robot0:Camera_sensor_eyes_depth"]



obs["robot0:Camera_sensor_eyes_normal"]



obs["robot0:Camera_sensor_eyes_seg_instance"]

1 Large-Scale Scene Generation

2 Modular Robot Control

3 Kinematic & Semantic Object States

4 Customizable Physics Transitions

TODO

Cem and Sujay's scene graph building

OmniGibson

Robot Controllers

q $\dot{q}$ τ

Joint Control

Differential Drive

Inverse Kinematics

Multi-Finger Gripper

1 Large-Scale Scene Generation

2 Modular Robot Control

3 Kinematic & Semantic Object States

4 Customizable Physics Transitions

OmniGibson

Robot Teleoperation

`omnigibson/examples/robots/robot_control_example.py`

Robot Camera View!

1 Large-Scale Scene Generation

2 Modular Robot Control

3 Kinematic & Semantic Object States

4 Customizable Physics Transitions

OmniGibson

Object State Interface

Dictionary mapping state
name to state instance

```
obj.states[<STATE_NAME>]
```

1 Large-Scale Scene Generation

2 Modular Robot Control

3 Kinematic & Semantic Object States

4 Customizable Physics Transitions

OmniGibson

Object State Interface

```
obj.states[<STATE_NAME>]
OnTop
Under
ToggledOn
Frozen
Sliced
Soaked
Filled
...
```

1 Large-Scale Scene Generation

2 Modular Robot Control

3 Kinematic & Semantic Object States

4 Customizable Physics Transitions

OmniGibson
Object State Interface

```
obj.states[<STATE_NAME>].get_value(*args) -> bool
           OnTop
           Under
           ToggledOn
           Frozen
           Sliced
           Soaked
           Filled
           ...
```

State-specific arguments, e.g. other object for checking `Under`

1 Large-Scale Scene Generation

2 Modular Robot Control

3 **Kinematic & Semantic Object States**

4 Customizable Physics Transitions

OmniGibson

Object State Interface

```
.get_value(*args) -> bool
```

```
obj.states[<STATE_NAME>].set_value(*args, bool)
```

```
OnTop
Under
ToggledOn
Frozen
Sliced
Soaked
Filled
...
```

1 Large-Scale Scene Generation

2 Modular Robot Control

3 Kinematic & Semantic Object States

4 Customizable Physics Transitions

OmniGibson

Kinematic States

States that describe **physical relationships**

Examples: `OnTop`, `Under`, `Inside`

```
microwave.states[OnTop].set_value(table, True)
```

OmniGibson

Kinematic States

States that describe **physical relationships**

Examples: `OnTop`, `Under`, `Inside`

```
microwave.states[OnTop].set_value(table, True)
```

```
microwave.states[Under].set_value(table, True)
```

1 Large-Scale Scene Generation

2 Modular Robot Control

3 Kinematic & Semantic Object States

4 Customizable Physics Transitions

OmniGibson

Kinematic States

States that describe **physical relationships**

Examples: `OnTop`, `Under`, `Inside`

```
microwave.states[OnTop].set_value(table, True)
```

```
microwave.states[Under].set_value(table, True)
```

```
snack_box.states[Inside].set_value(shelf, True)
```

1 Large-Scale Scene Generation

2 Modular Robot Control

3 Kinematic & Semantic Object States

4 Customizable Physics Transitions

OmniGibson

Semantic States - Temperature

States that describe **temperature dynamics**

Examples: `Cooked`, `Frozen`, `Burnt`, `HeatSource`, `Heated`

```
cabinet.states[Cooked]
    .set_value(True)
```

1 Large-Scale Scene Generation

2 Modular Robot Control

3 Kinematic & Semantic Object States

4 Customizable Physics Transitions

OmniGibson

Semantic States - Temperature

States that describe **temperature dynamics**

Examples: `Cooked`, `Frozen`, `Burnt`, `HeatSource`, `Heated`

```
cabinet.states[Cooked]
  .set_value(True)
```

```
cabinet.states[Frozen]
  .set_value(True)
```

1 Large-Scale Scene Generation

2 Modular Robot Control

3 Kinematic & Semantic Object States

4 Customizable Physics Transitions

OmniGibson

Semantic States - Temperature

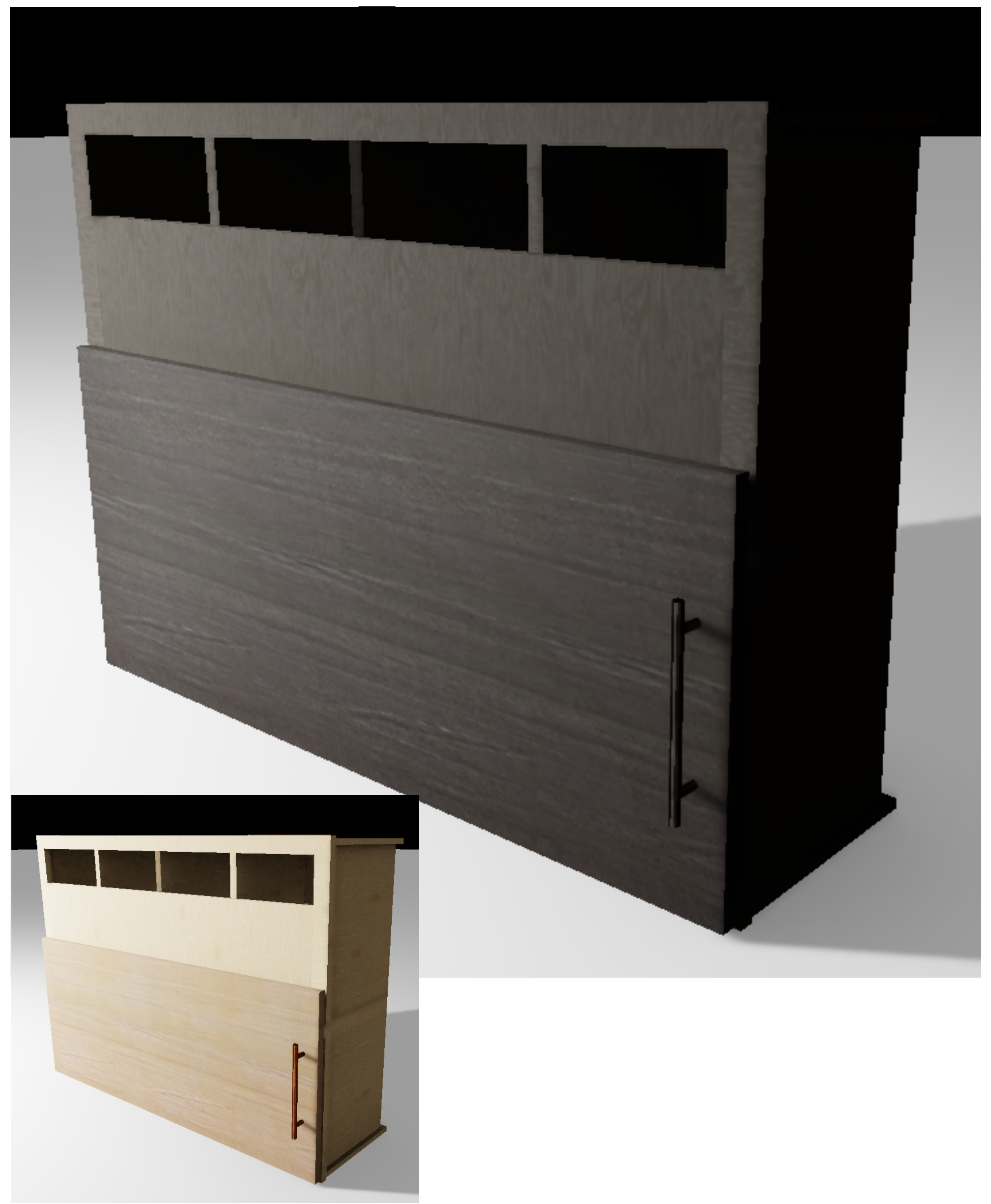
States that describe **temperature dynamics**

Examples: `Cooked`, `Frozen`, `Burnt`, `HeatSource`, `Heated`

`cabinet.states[Cooked].set_value(True)`

`cabinet.states[Frozen].set_value(True)`

`cabinet.states[Burnt].set_value(True)`



1 Large-Scale Scene Generation

2 Modular Robot Control

3 Kinematic & Semantic Object States

4 Customizable Physics Transitions

OmniGibson

Semantic States - Temperature

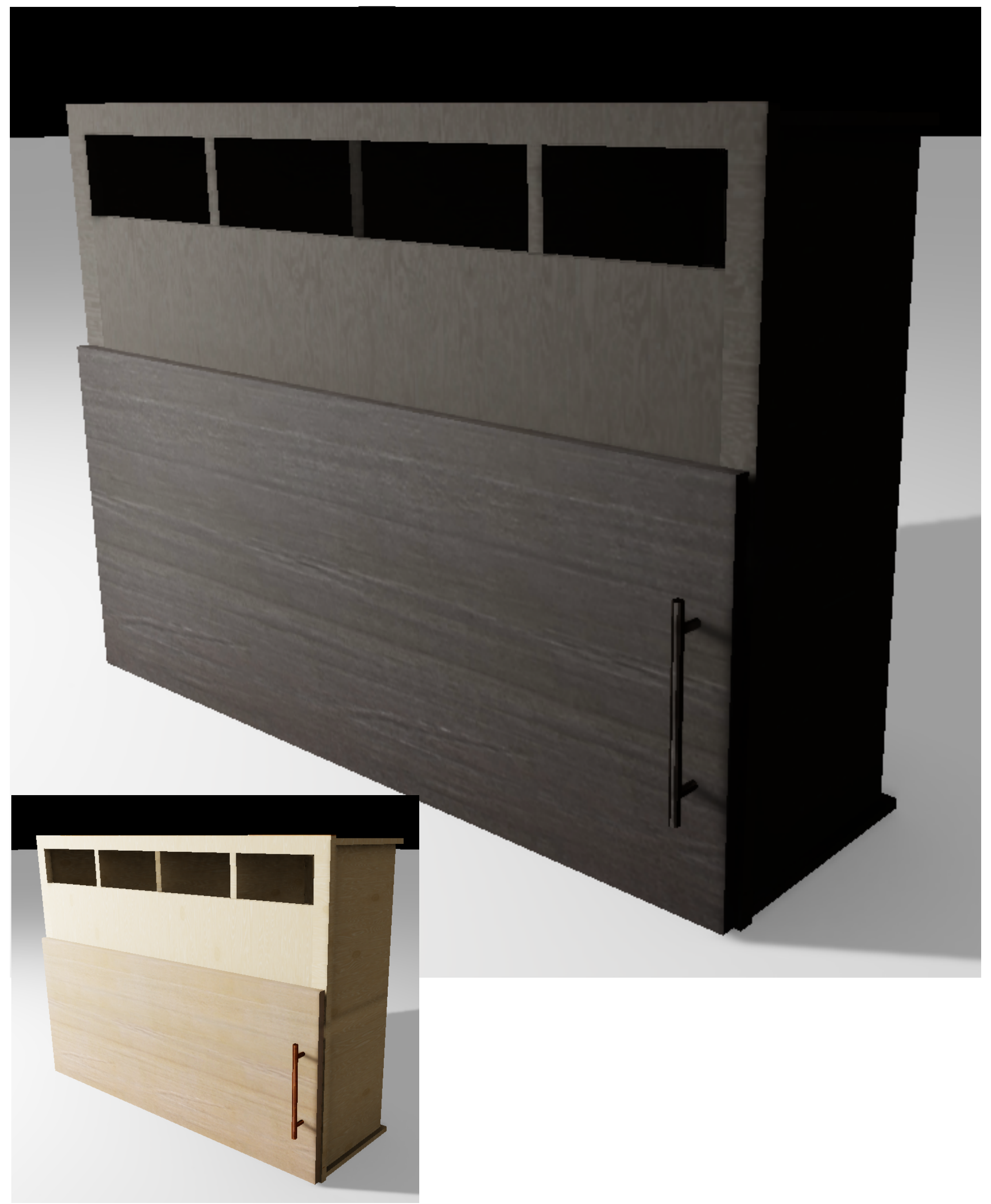
States that describe **temperature dynamics**

Examples: `Cooked`, `Frozen`, `Burnt`, `HeatSource`, `Heated`

`cabinet.states[Cooked].set_value(True)`

`cabinet.states[Frozen].set_value(True)`

`cabinet.states[Burnt].set_value(True)`



`stove.states[ToggledOn].set_value(True)`

1 Large-Scale Scene Generation

2 Modular Robot Control

3 Kinematic & Semantic Object States

4 Customizable Physics Transitions

OmniGibson

Semantic States - Temperature

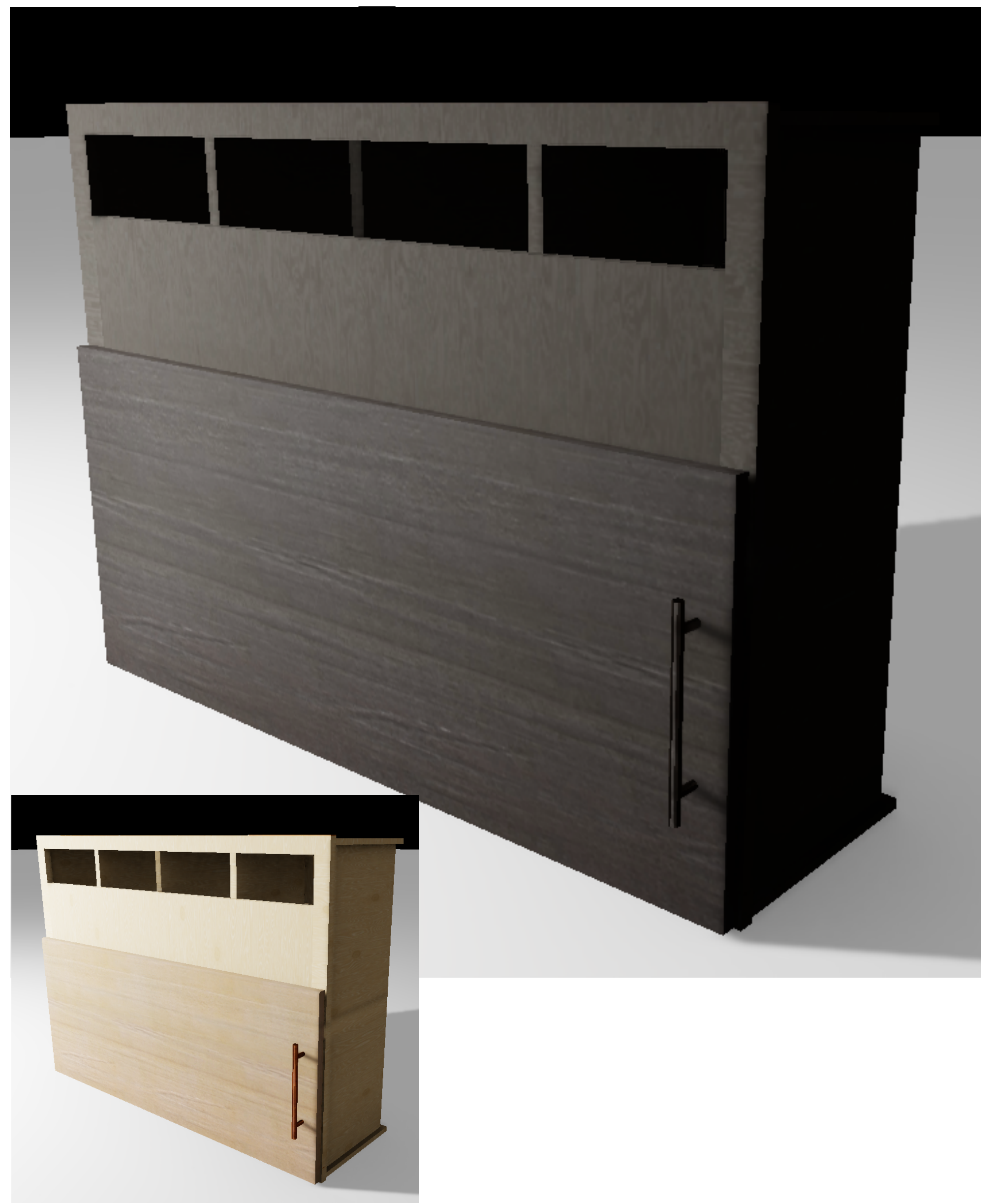
States that describe **temperature dynamics**

Examples: `Cooked`, `Frozen`, `Burnt`, `HeatSource`, `Heated`

`cabinet.states[Cooked].set_value(True)`

`cabinet.states[Frozen].set_value(True)`

`cabinet.states[Burnt].set_value(True)`



`stove.states[ToggledOn].set_value(True)`

`pie.states[Heated].set_value(True)`

1 Large-Scale Scene Generation

2 Modular Robot Control

3 Kinematic & Semantic Object States

4 Customizable Physics Transitions

OmniGibson

Semantic States - Covered

States that generate **visual** or **physical particles**

Examples: `Covered with Stain`, `Covered with Water`

`table.states[Covered].set_value(StainSystem, False)`

`table.states[Covered].set_value(StainSystem, True)`

1 Large-Scale Scene Generation

2 Modular Robot Control

3 Kinematic & Semantic Object States

4 Customizable Physics Transitions

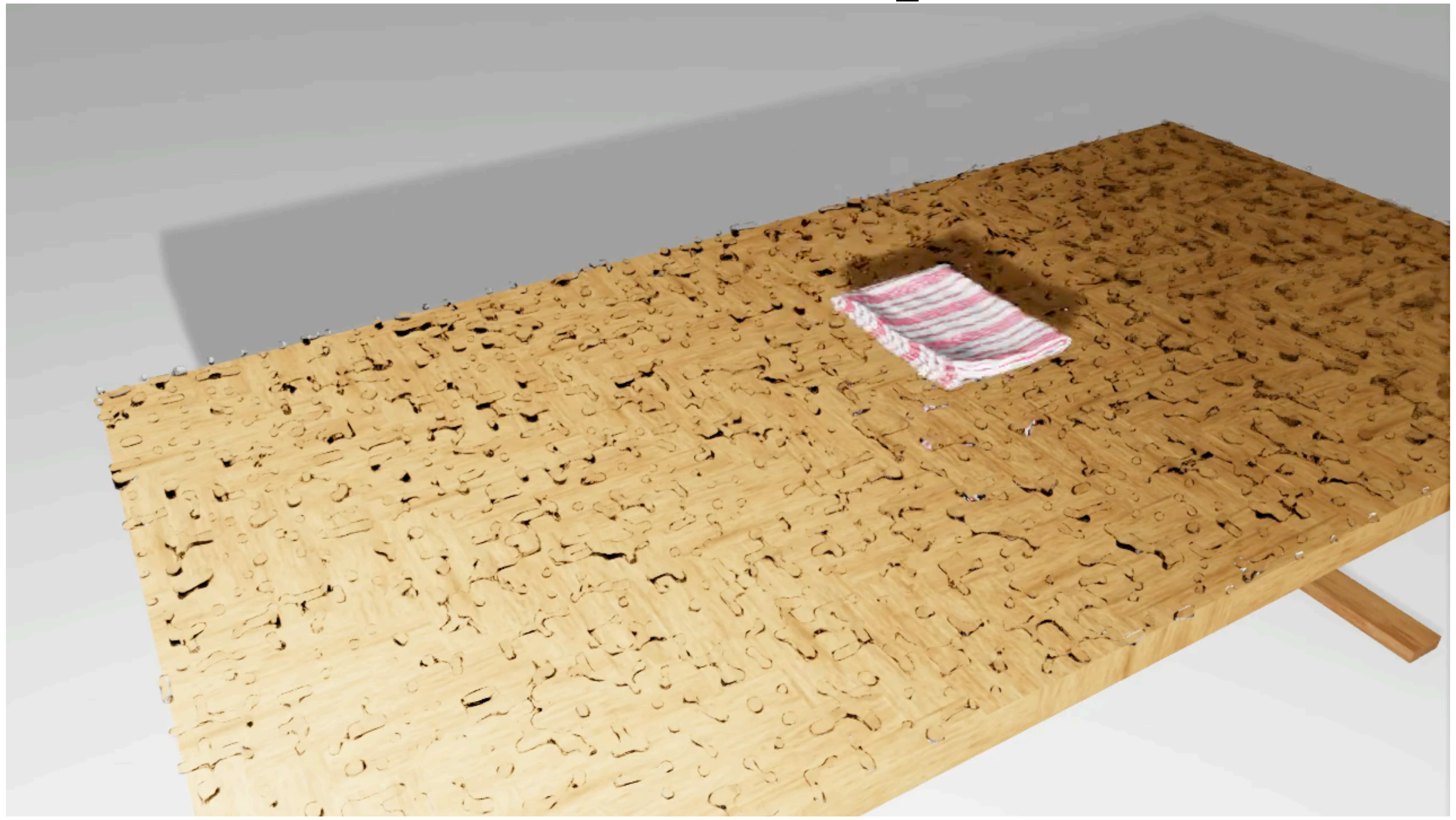
OmniGibson

Semantic States - Particle Modifiers

States that modify the current set of **visual** or **physical particles**
Examples: **`ParticleApplier`**, **`ParticleRemover`**

`cloth.states[ToggledOn].set_value(True)`

`cloth.states[ToggledOn].set_value(True)`



1 Large-Scale Scene Generation

2 Modular Robot Control

3 Kinematic & Semantic Object States

4 Customizable Physics Transitions

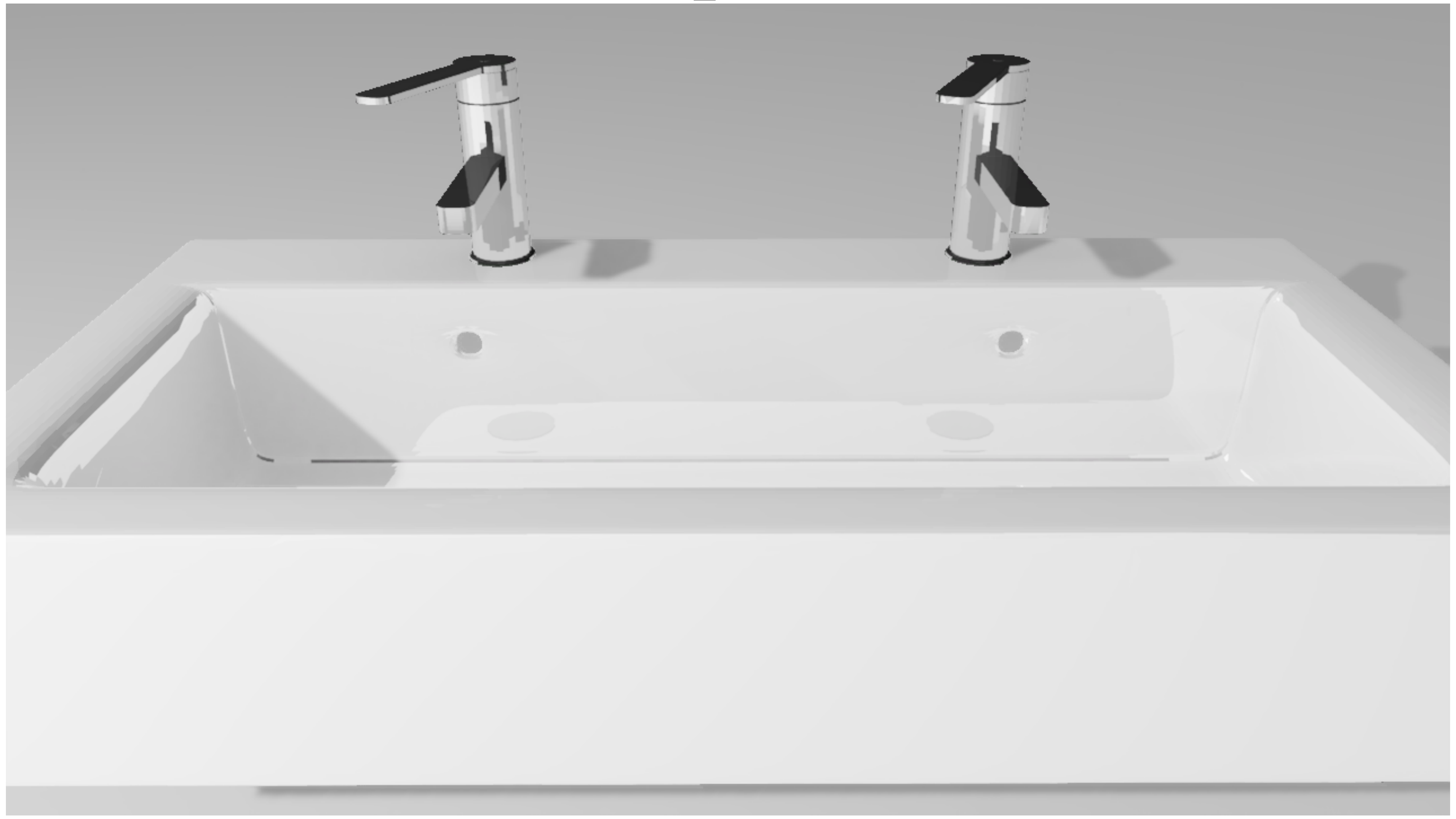
OmniGibson

Semantic States - Fluids

States that facilitate **fluid interaction**

`Filled`: Whether an object **contains** a threshold amount of a specific **`fluid`**

```
sink.states[Filled].set_value(WaterSystem, False)
```



```
sink.states[Filled].set_value(WaterSystem, True)
```

1 Large-Scale Scene Generation

2 Modular Robot Control

3 Kinematic & Semantic Object States

4 Customizable Physics Transitions

OmniGibson

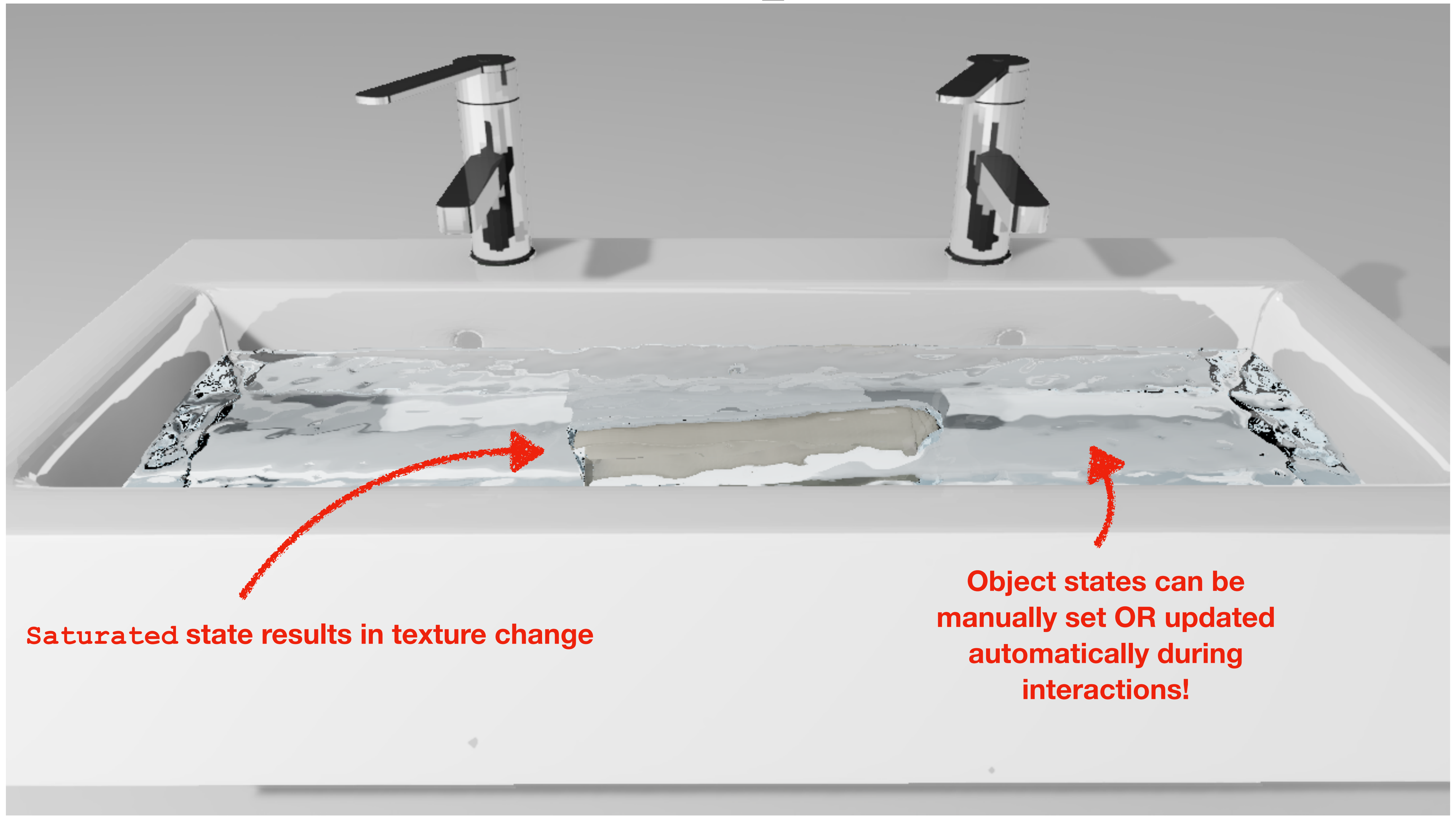
Semantic States - Fluids

States that facilitate **fluid interaction**

`Filled`: Whether an object **contains** a threshold amount of a specific **`fluid`**
`Saturated`: Whether an object has **absorbed** a threshold amount of a specific **`fluid`**

```
napkin.states[Saturated].set_value(WaterSystem, False)
```

```
napkin.states[Saturated].set_value(WaterSystem, True)
```



1 Large-Scale Scene Generation

2 Modular Robot Control

3 Kinematic & Semantic Object States

4 Customizable Physics Transitions

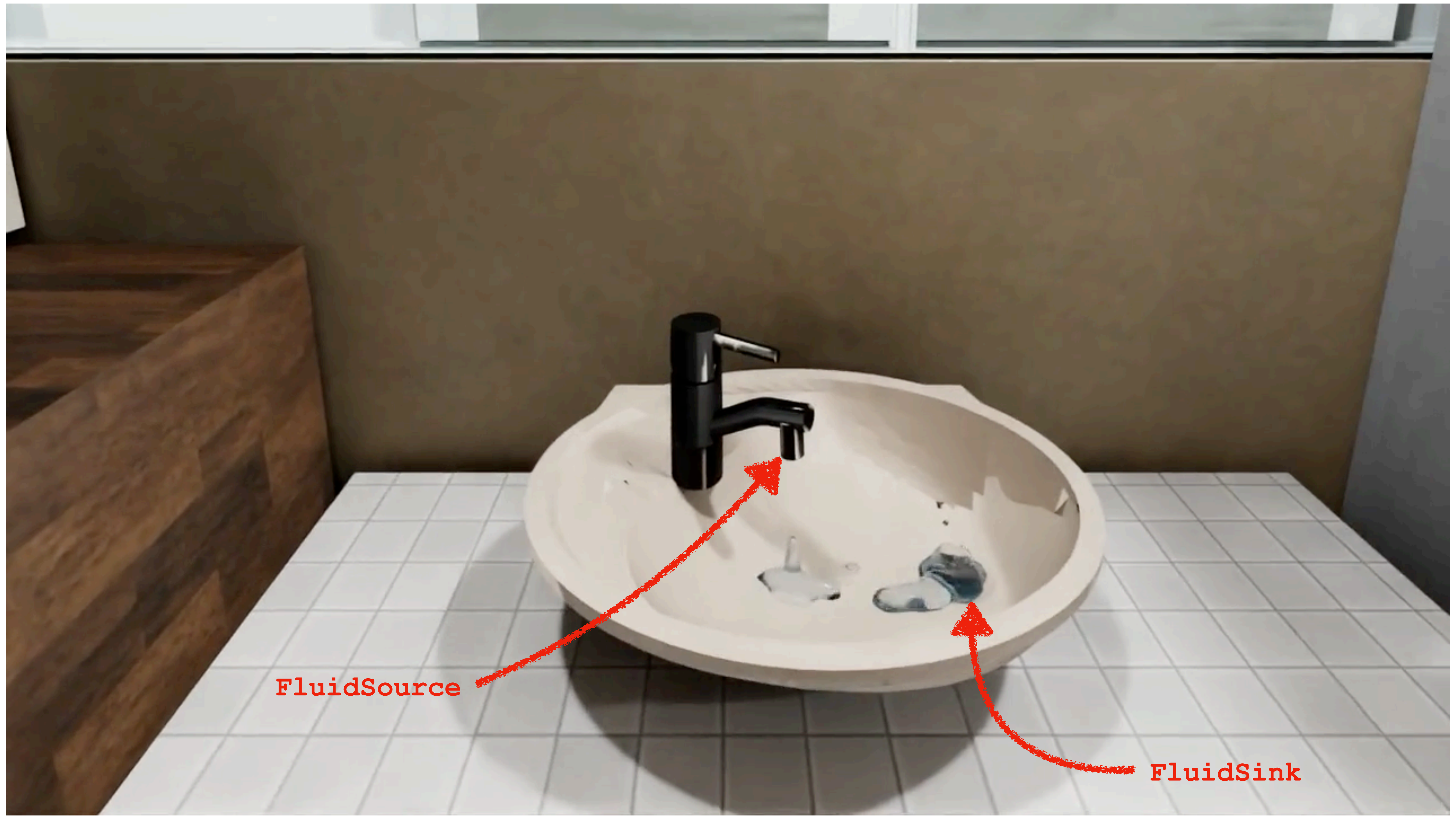
OmniGibson

Semantic States - Fluids

States that facilitate **fluid interaction**

`Filled`: Whether an object **contains** a threshold amount of a specific **`fluid`**
`Saturated`: Whether an object has **absorbed** a threshold amount of a specific **`fluid`**
`FluidSource / FluidSink`: Whether an object can **generate / absorb** arbitrary amounts of a specific **`fluid`**

`sink.states[ToggledOn].set_value(True)`



1 Large-Scale Scene Generation

2 Modular Robot Control

3 Kinematic & Semantic Object States

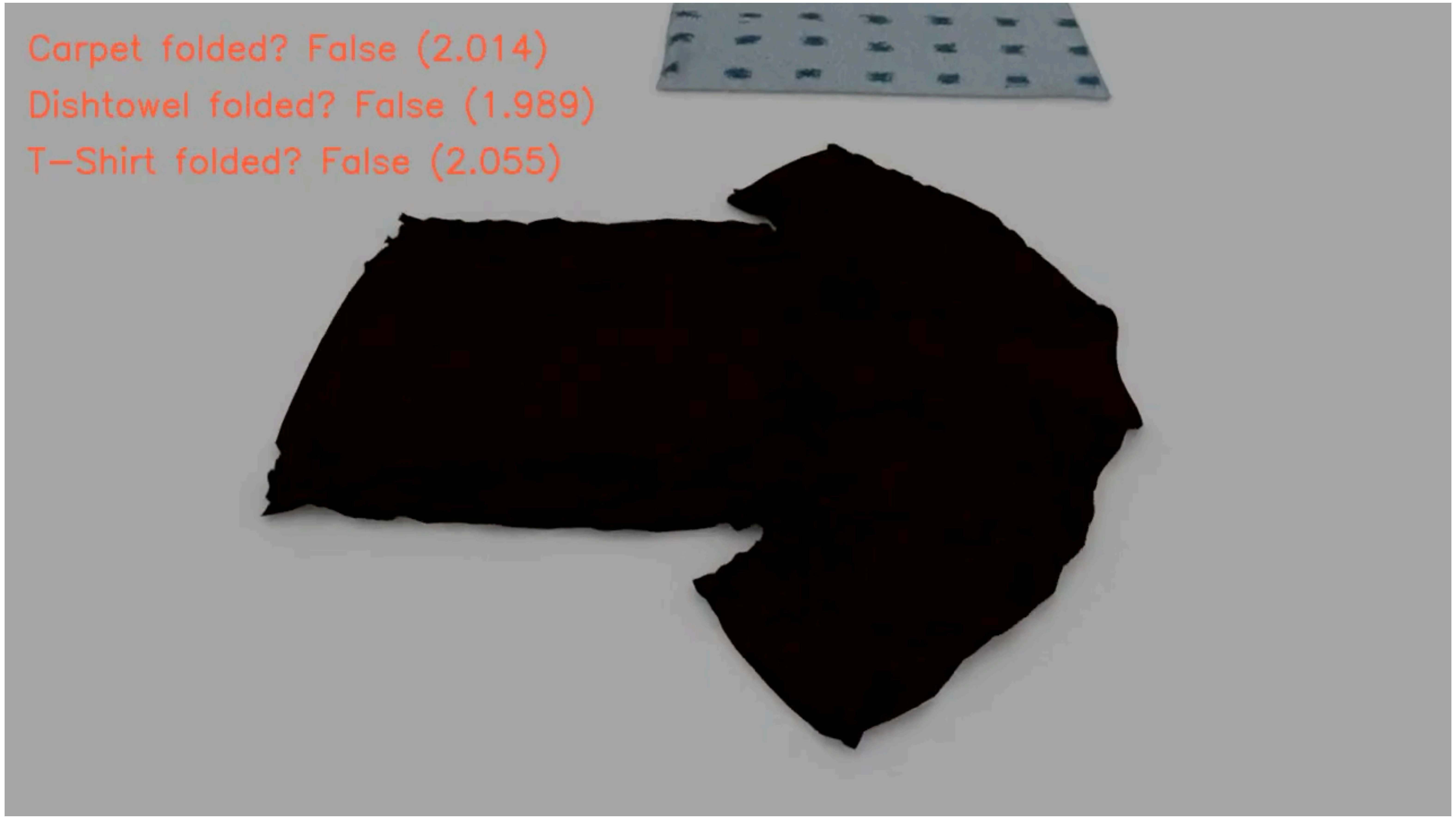
4 Customizable Physics Transitions

OmniGibson

Semantic States - Cloth

States that facilitate **cloth interaction**

`Folded` / `Unfolded`: Whether an object's **overlap** meets specific criteria



OmniGibson

Physics Transition Interface

```
class TransitionRule:

def condition(self, *args) -> bool

def transition(self, *args)
```

Determines whether a transition should occur

What should happen when a transition is triggered

Allows us to capture arbitrarily complex **physical phenomena**!

1 Large-Scale Scene Generation

2 Modular Robot Control

3 Kinematic & Semantic Object States

4 **Customizable Physics Transitions**

OmniGibson

Physics Transition Examples

```
class SlicingRule:

def condition(self, *args) -> bool

    # Return True if a slicer object is touching
    a sliceable object with sufficient force


def transition(self, *args)

    # Remove the sliceable object and
    import its sliced component objects
```

1 Large-Scale Scene Generation

2 Modular Robot Control

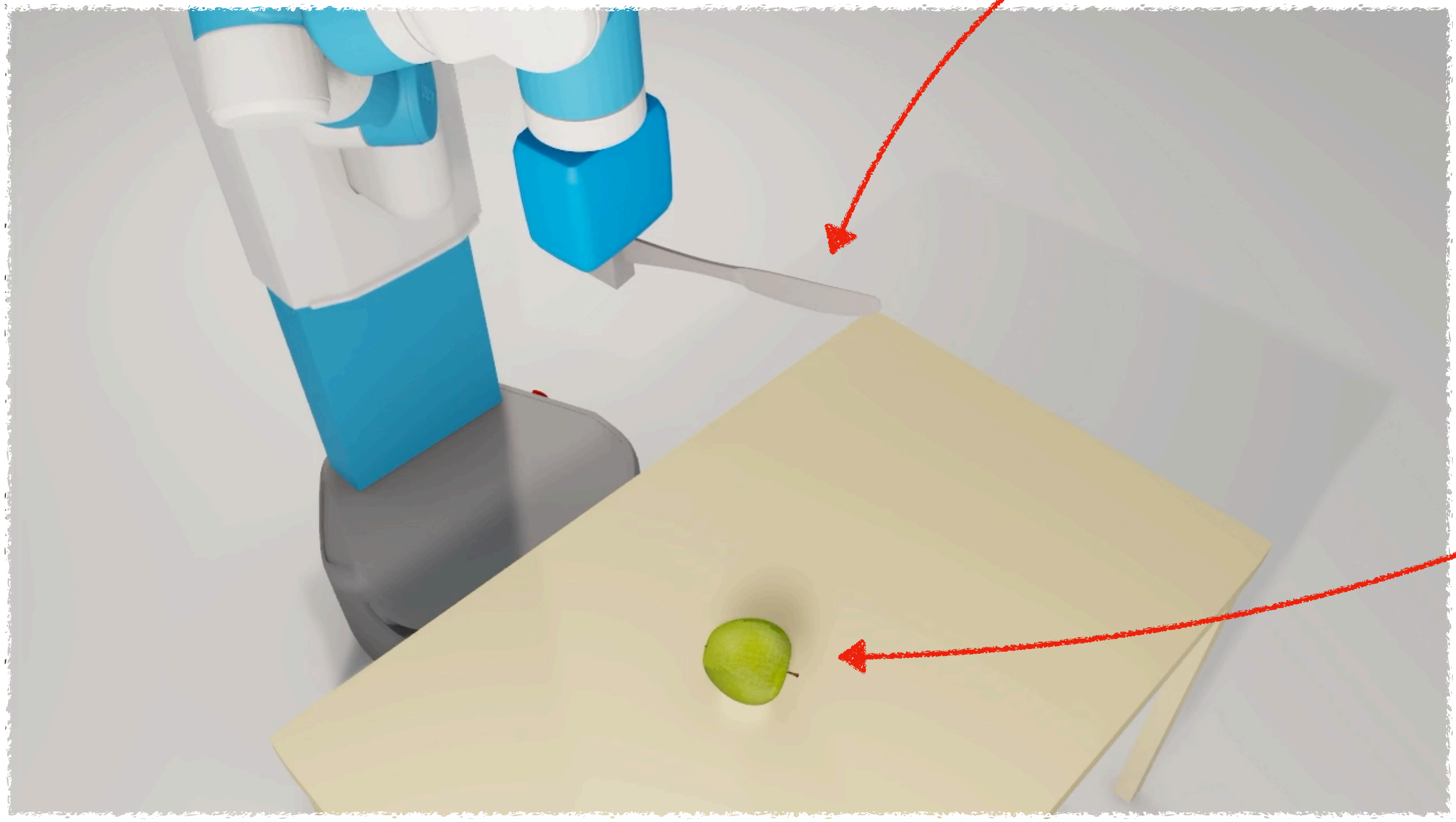
3 Kinematic & Semantic Object States

4 **Customizable Physics Transitions**

OmniGibson

Physics Transition Examples

```
class SlicingRule:
```



1 Large-Scale Scene Generation

2 Modular Robot Control

3 Kinematic & Semantic Object States

4 Customizable Physics Transitions

OmniGibson

Physics Transition Examples

```
class StrawberrySmoothie:
```

1 Large-Scale Scene Generation

2 Modular Robot Control

3 Kinematic & Semantic Object States

4 Customizable Physics Transitions

Tools for Accelerating Robot Learning

- Domain Randomization (Visual + Physical)
- Teleoperation Interfaces (Devices + VR)
- Skill Primitives
-

Domain Randomization: Perceptual

e.g.: Lighting, Color, etc.

TODO

Customizable, modular scene configuration + synthetic data generation

Domain Randomization: Physical
e.g.: Instance Randomization

TODO

Scene-level / task sampling

Teleoperation Interface: Spacemouse

TODO

Spacemouse Teleloperation

Teleoperation Interface: VR

TODO

VR Teleoperation

Skill Primitives

TODO

Primitives (Ayano & Minjune & Sujay)
(maybe) Integration with CuRobo

Have Any Tasks Been Solved?

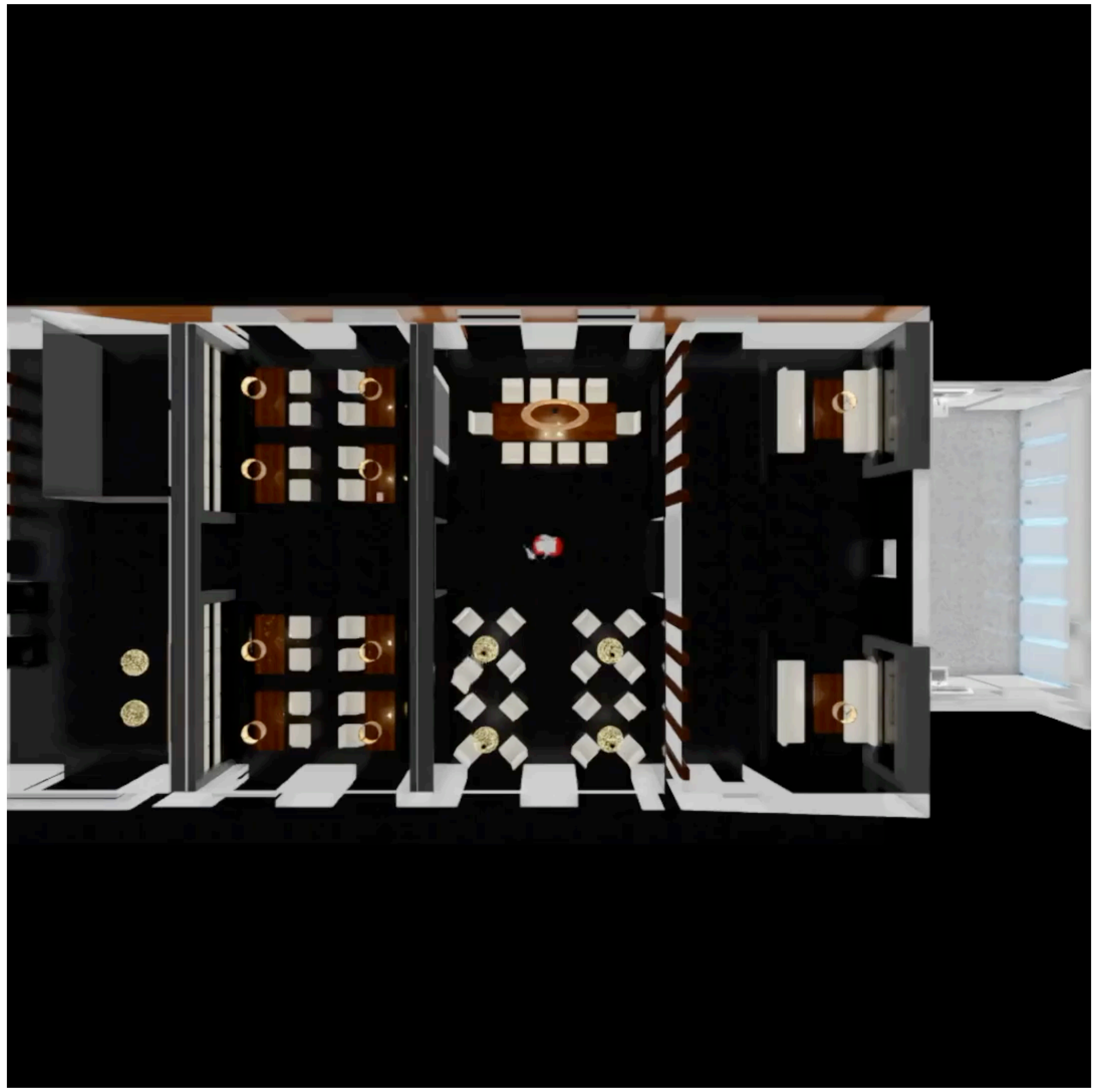
Current B1K Efforts for solving the benchmark:

- **Initial experiment study (CoRL Paper)**
- **Sanjana’s LLM Generated Plans**
- **Large-Scale RL + Skill Primitives (Cem + Sujay)**
- **Sim2Real “Digital Cousins” (Josiah)**

How do today's algorithms do on BEHAVIOR?

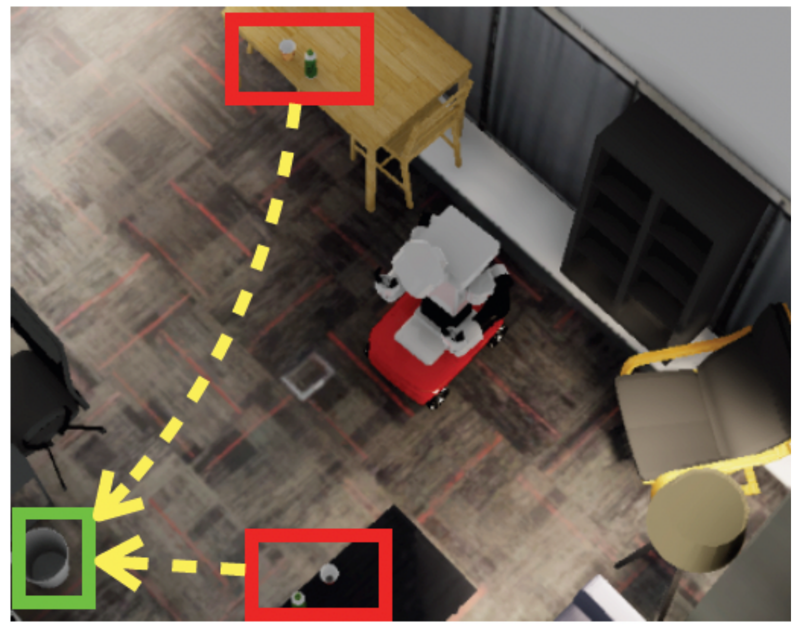
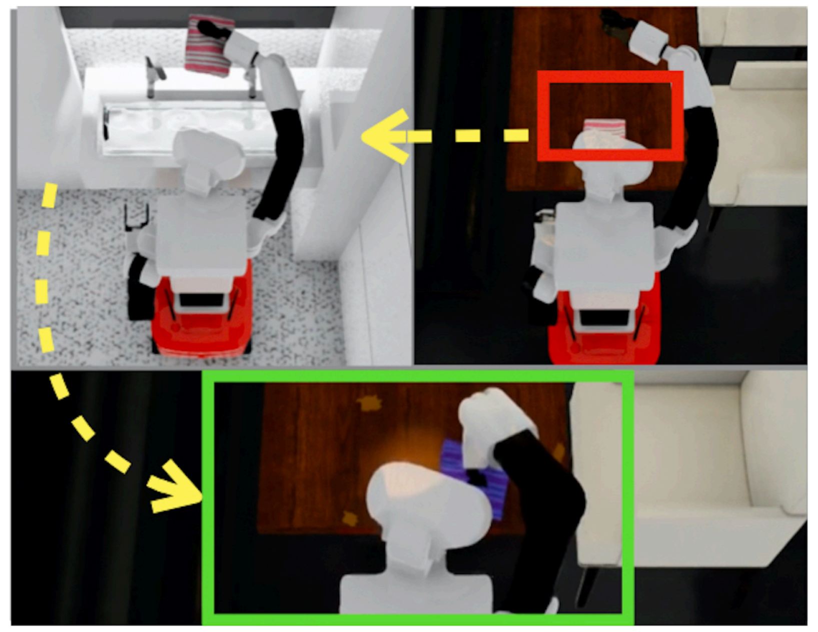
Store decoration

Collect trash



Clean table

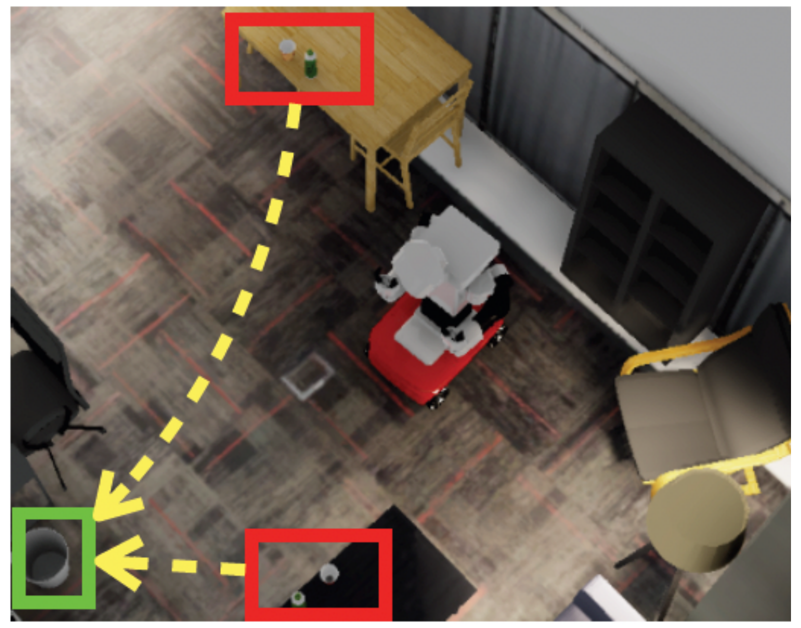
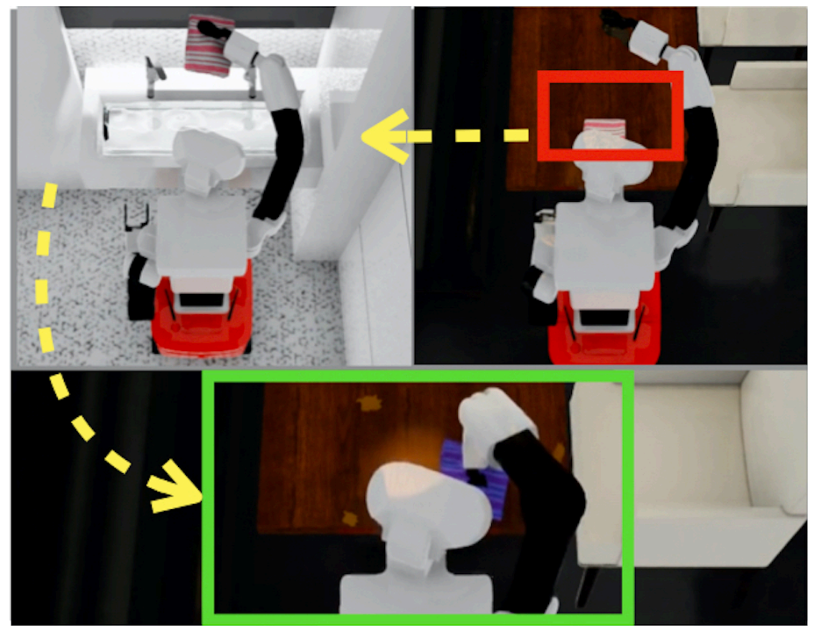
How do today's algorithms do on BEHAVIOR?
Assume different levels of privileged information

		Store decoration	Collect trash	Clean table
				
Low-level action space (Soft Actor-Critic[1])		0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
High-level action primitives (Proximal Policy Opt.[2])				
action prim. + memory (Proximal Policy Opt.[2])				

Takeaway #1: the model with low-level action space completely fails to learn because the activities are long-horizon, have sparse reward, and require many different skills.

[1] Haarnoja et al. "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor." ICML. PMLR, 2018; [2] Schulman, et al. "Proximal policy optimization algorithms." arXiv 2017

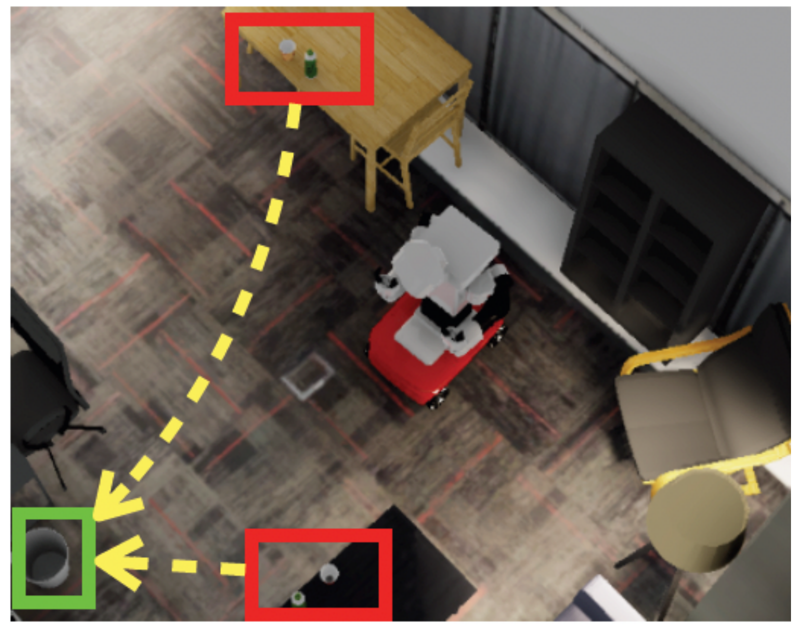
How do today's algorithms do on BEHAVIOR?
Assume different levels of privileged information

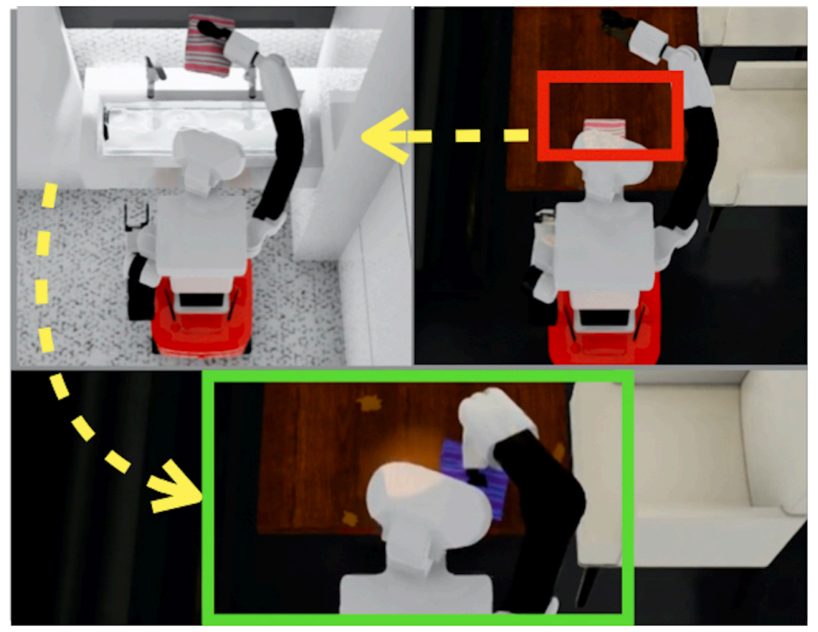
		Store decoration	Collect trash	Clean table
				
Low-level action space (Soft Actor-Critic[1])		0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
High-level action primitives (Proximal Policy Opt.[2])		0.48 ± 0.06	0.42 ± 0.02	0.77 ± 0.08
action prim. + memory (Proximal Policy Opt.[2])				

Takeaway #2: the model with high-level action primitives (e.g. Nav, Pick, Place) achieves better success by leveraging privileged information and temporally extended action space.

[1] Haarnoja et al. "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor." ICML. PMLR, 2018; [2] Schulman, et al. "Proximal policy optimization algorithms." arXiv 2017

How do today's algorithms do on BEHAVIOR?
Assume different levels of privileged information





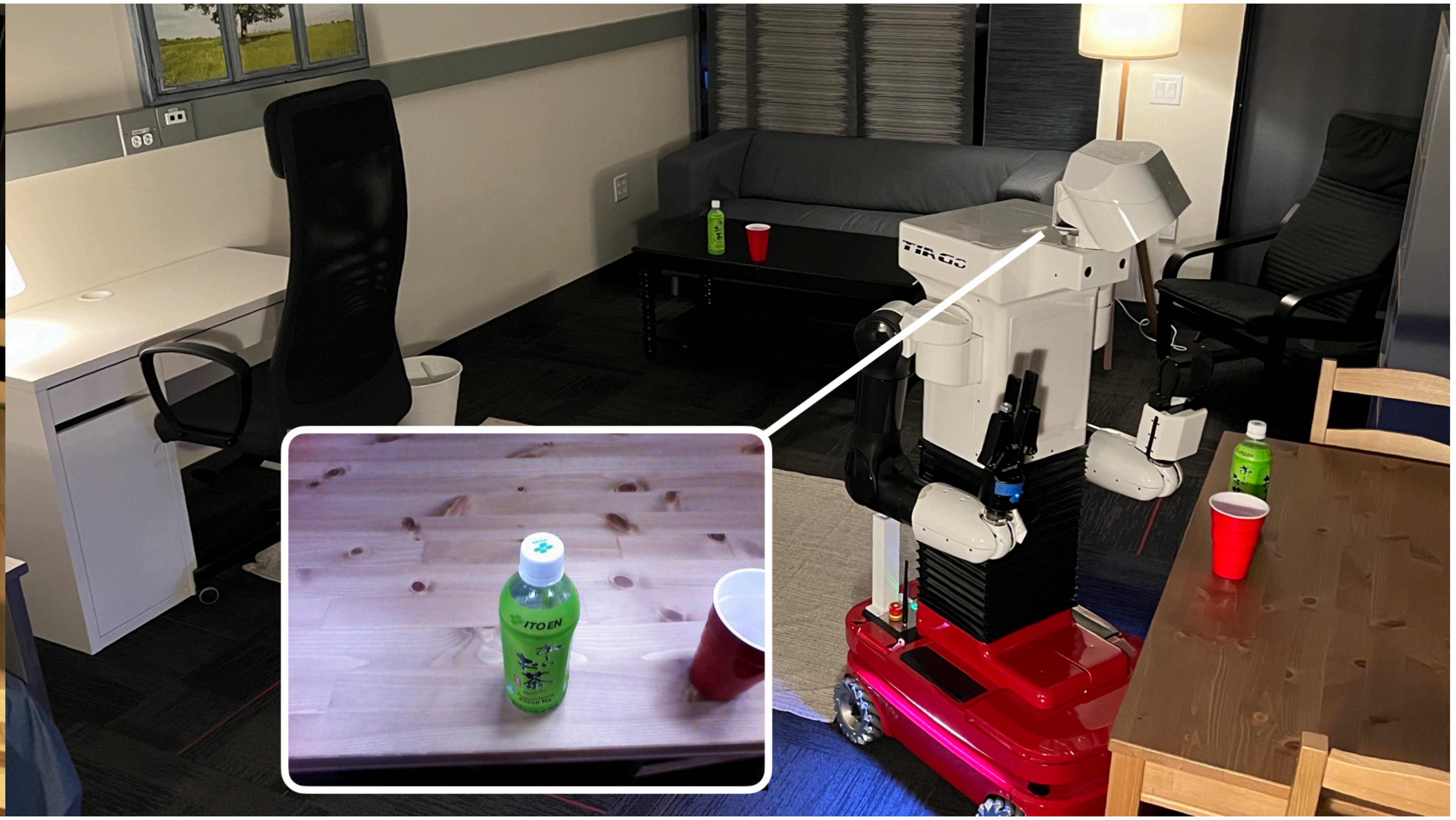
		Store decoration	Collect trash	Clean table
Low-level action space (Soft Actor-Critic[1])		0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
High-level action primitives (Proximal Policy Opt.[2])		0.48 ± 0.06	0.42 ± 0.02	0.77 ± 0.08
action prim. + memory (Proximal Policy Opt.[2])		0.55 ± 0.05	0.63 ± 0.03	0.99 ± 0.02

Takeaway #3: memory brings in extra improvements for success rate and task efficiency across the board, especially for long-horizon activities.

[1] Haarnoja et al. "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor." ICML. PMLR, 2018; [2] Schulman, et al. "Proximal policy optimization algorithms." arXiv 2017

The ultimate goal of BEHAVIOR is real-world robotics:
Assessing sim-real gap, and testing sim2real transfer

Simulation in BEHAVIOR's OmniGibson environment



Real world – "Marvin" the Tiago robot

The ultimate goal of BEHAVIOR is real-world robotics:
Realistic simulation allows for better zero-shot sim2real transfer

LLM-Generated Plans

TODO

- Sanjana's LLM generated plans

Large Scale RL + Skill Primitives

TODO

- Cem / Sujay RL Project

Sim2Real “Digital Cousins”

TODO

Sim2real (Josiah's project)

OmniGibson

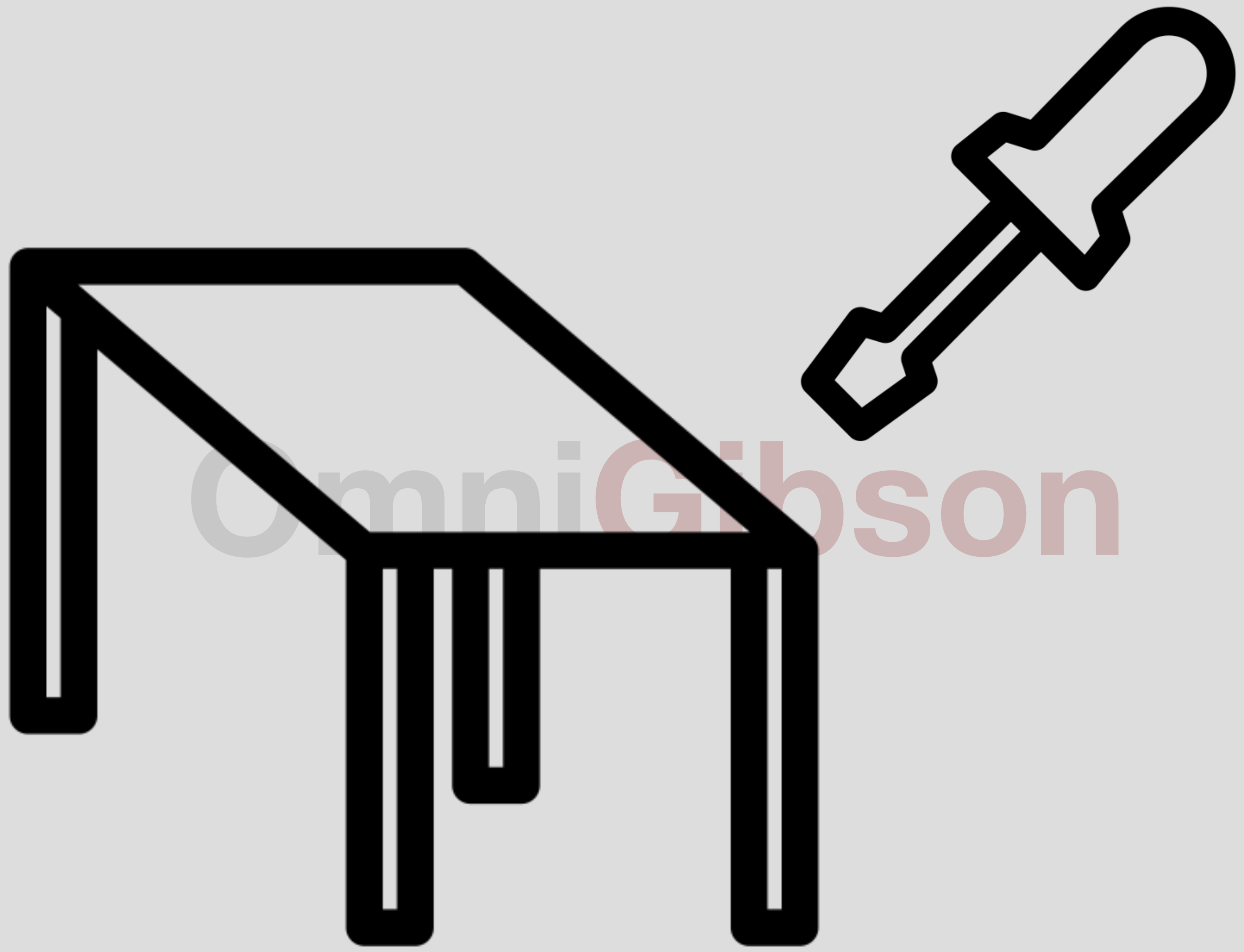
OmniGibson

research challenges

how to accurately capture **complex** states

How to define **hanging** an object?

how to accurately capture **complex** states

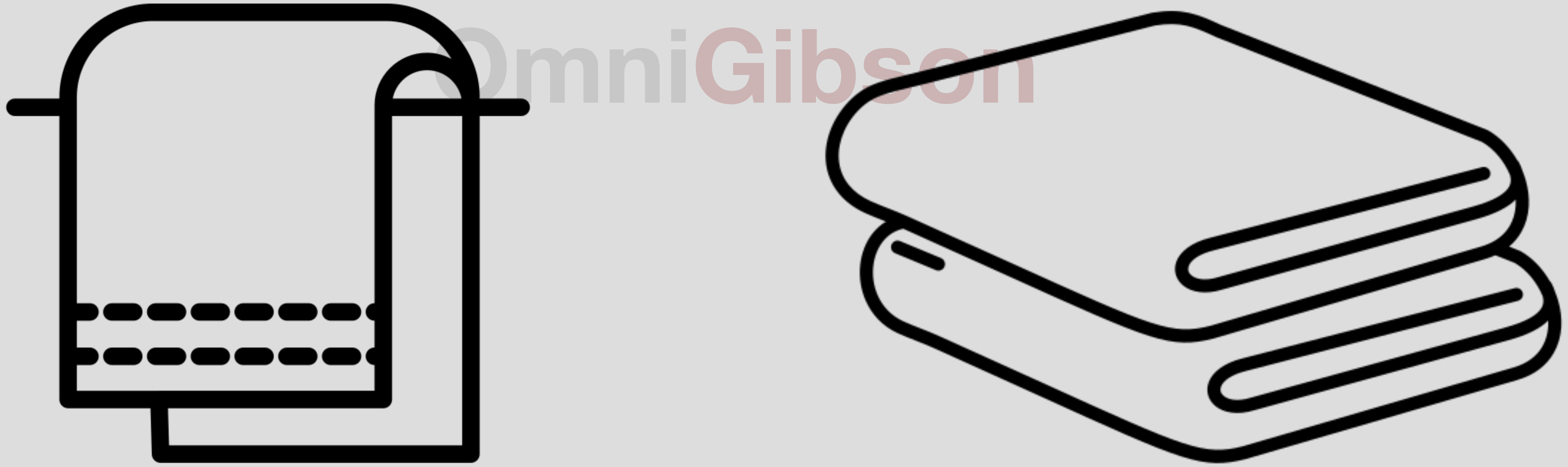


How to define **assembling** subcomponents?

how to accurately capture **complex** states

how to programmatically manipulate **cloth**

→ can we programmatically modify individual cloth particles?



how to accurately capture **complex** states

how to programmatically manipulate **cloth**

→ can we programmatically modify individual cloth particles?

how to model **multi-fluid** interaction

→ will multiple particle systems be able to collide in a future release?

OmniGibson

Try it out!

github.com/StanfordVL/OmniGibson

OmniGibson is powered by NVIDIA OMNIVERSE


Dieter Fox


Animesh Garg


Ajay Mandlekar


Renato Gasoto


Viktor Makoviychuk


Ignacio Llamas


Alain Denzler


Damien Fagnou


Rev Lebaredian


Beau Perschall


Simon Schirm


Gavan Woolery


Hammad Mazhar


Liila Torabi

Pierre Terdiman


Philipp Reist


Milad Rakhsha


Ales Borovicka

OmniGibson is part of BEHAVIOR

STANFORD VISION & LEARNING

TOYOTA RESEARCH INSTITUTE

HAI Stanford University Human-Centered Artificial Intelligence

NVIDIA OMNIVERSE

BEHAVIOR


Chengshu Li


Ruohan Zhang


Josiah Wong


Cem Gokmen


Sanjana Srivastava


Roberto Martín-Martín


Chen Wang

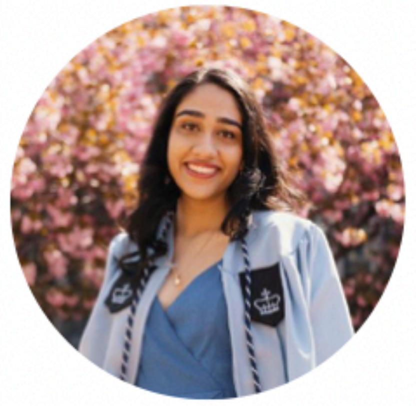

Gabrael Levine


Michael Lingelbach


Jiankai Sun


Mona Avari


Minjune Hwang


Manasi Sharma


Arman Aydin


Dhruva Bansal

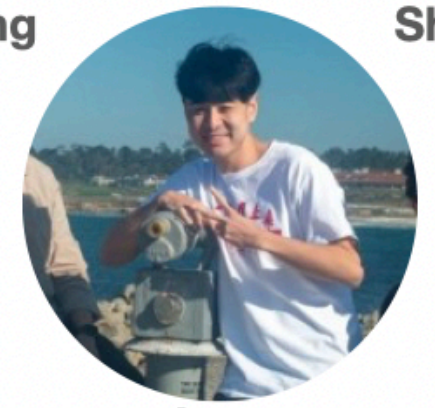

Samuel Hunter


Kyu-Young Kim


Alan Lou


Caleb Matthews


Ivan Villa-Renteria


Jerry Tang


Claire Tang


Fei Xia


Wensi Ai


Yunzhu Li


Silvio Savarese


Hyowon Gweon


Karen Liu


Jiajun Wu


Fei-Fei Li

STANFORD VISION & LEARNING

TOYOTA RESEARCH INSTITUTE

HAI Stanford University Human-Centered Artificial Intelligence

NVIDIA OMNIVERSE

Office of Naval Research Science & Technology

NSF

BEHAVIOR

BEHAVIOR.STANFORD.EDU

Try out our beta release today!

> **To augment people**
> (activity selection from human survey)

> **Large-scale & diverse**
> (activities, scenes, objects, states)

> **Realistic & ecological**
> (distribution, rendering, fluids, deformations)

TODO

Communication channel
- Discord
- Colab
- Github & website