Design Choices in Modern Embodied AI Environments

Building and Working in Environments for Embodied AI (part III)

CVPR 2022 Tutorial

UC San Diego

清華大學
Tsinghua University

SFU SIMON FRASER UNIVERSITY

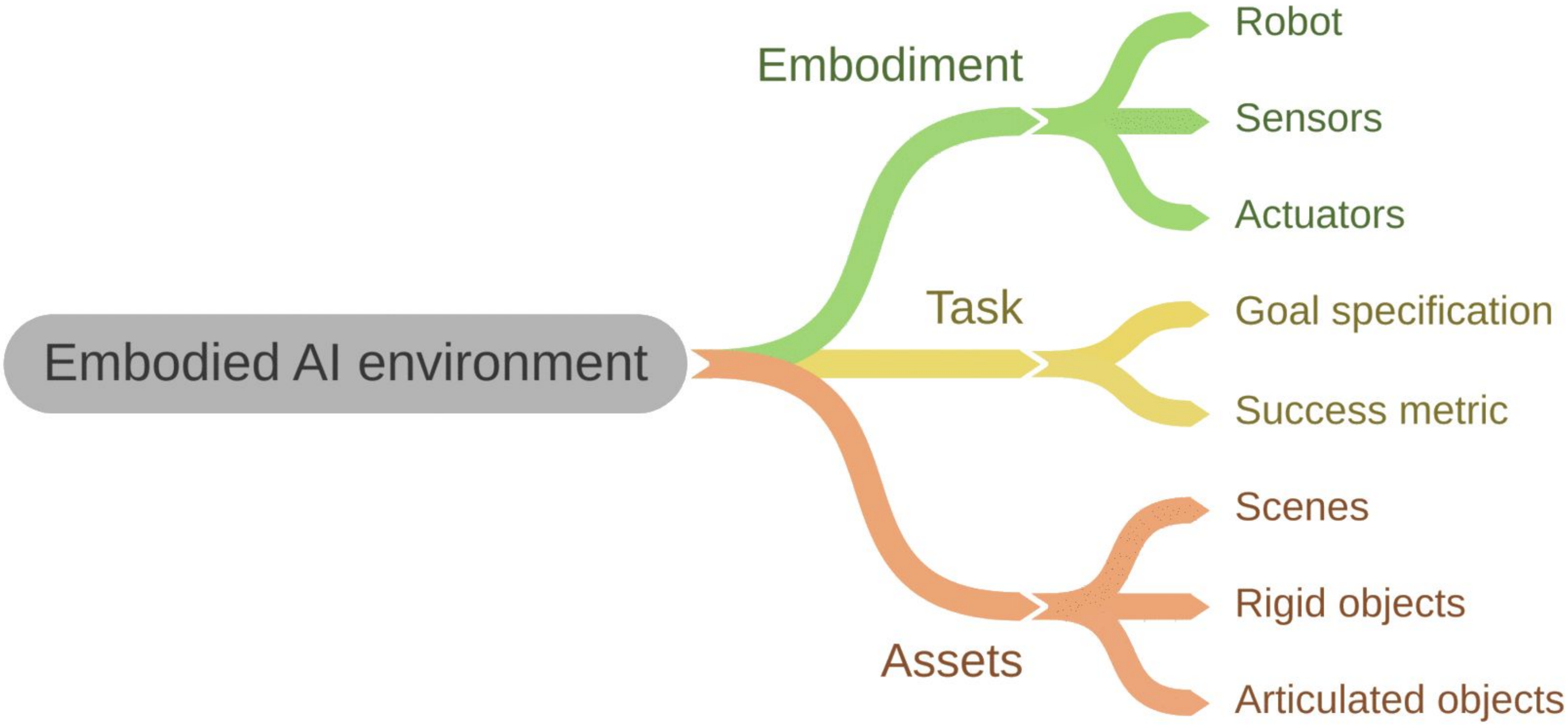
Overview

- We are talking about:
  - How to **design embodiment**?
  - How to **define a task**?
  - What kind of **assets** are needed?
- We will show several case studies of existing Embodied AI environments

Overview

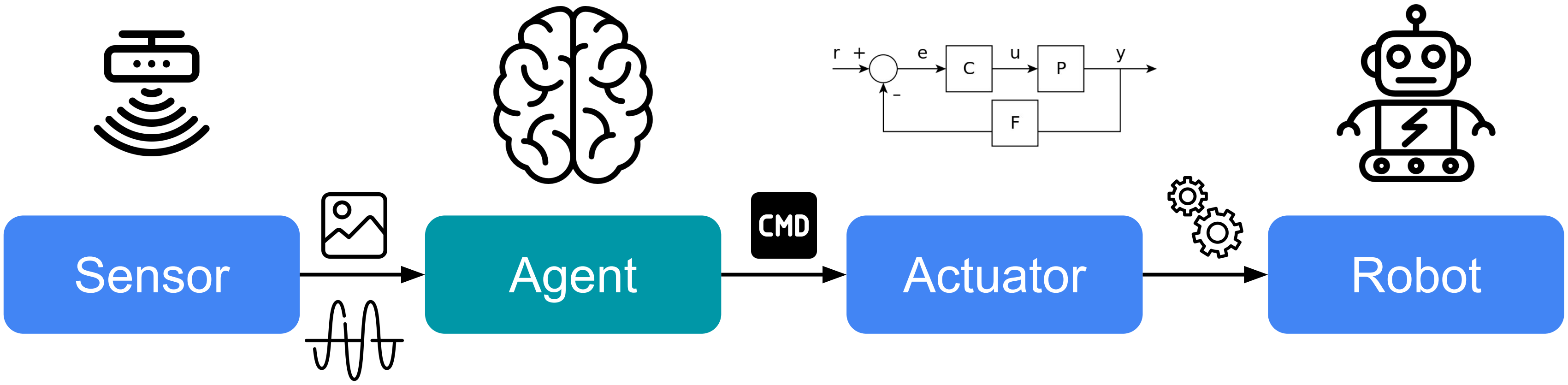
- This section is intended for who want to:
    - Find a suitable environment to start their projects
    - Have a deeper look at environments they are using
    - Customize or create a new environment
- All design choices are discussed in the simulation context

Most Common Design Choices



Design Choice: Embodiment

- Robot hardware
- Sensors
- Actuators



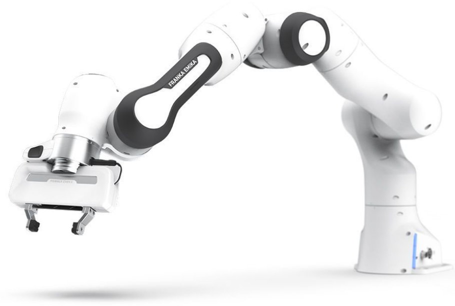
Figures from Wikipedia and flaticon

Embodiment: Robot Hardware

- Physical body with various sensorimotor capacities
- Shape the cognition and action

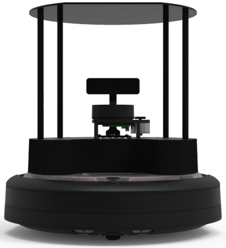
Legged robot

Unitree A1 Robot



Manipulator

Franka Emika Panda



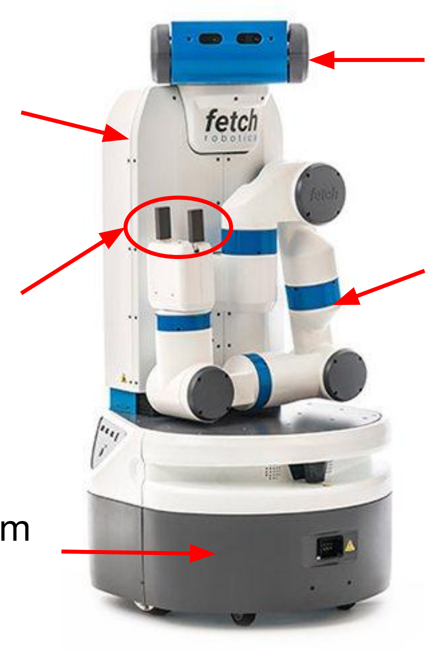
Wheeled robot

TurtleBot

Mobile manipulator

Fetch

Common Robot Components



Head

Torso

Gripper
(end-effector)

Arm

Mobile platform
(base)

Embodiment: Sensor

- Common types of sensors
  - Visual perception
  - Robot proprioception
  - Haptic/tactile perception
  - GPS and compass (IMU)

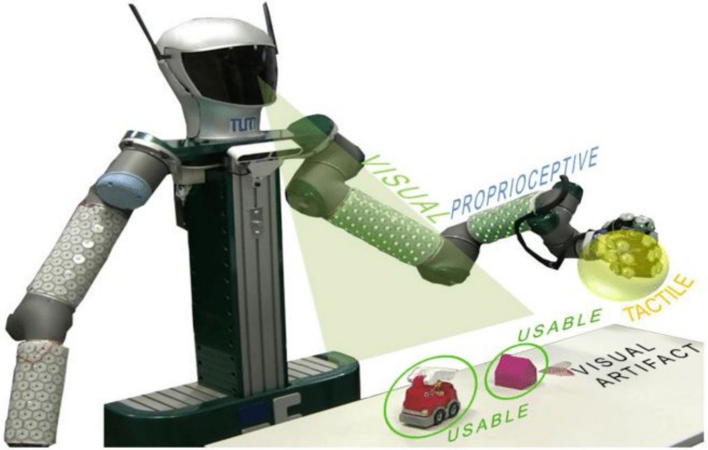
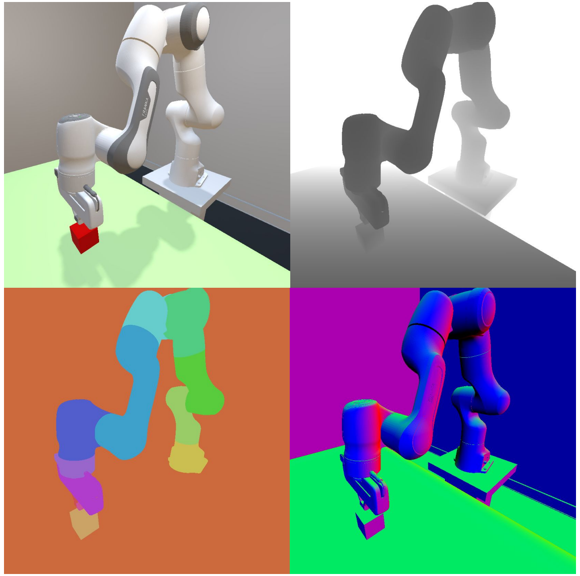


Figure from Yielding Self-Perception in Robots Through Sensorimotor Contingencies

Sensor: Visual Perception

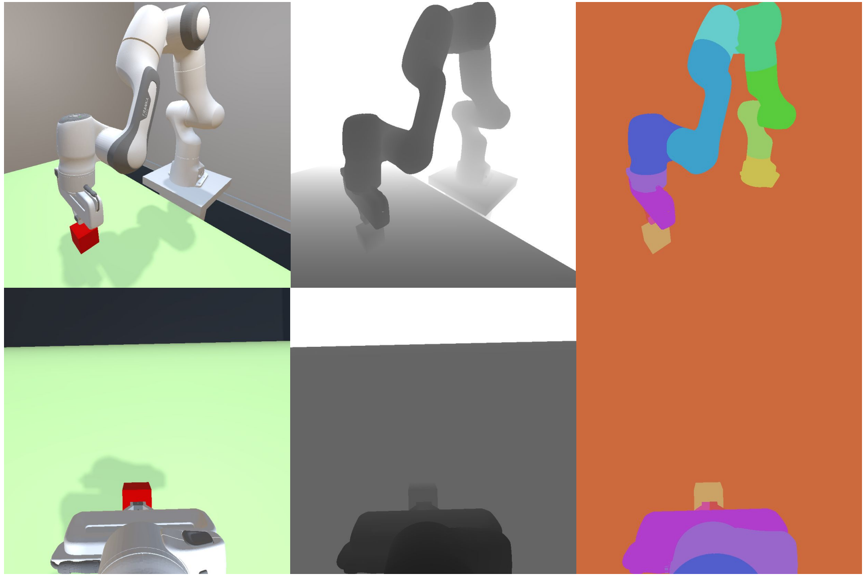
- Common signal types
  - RGB
  - Depth
  - Semantic/instance masks
- Other signal types
  - Normal
  - Optical flow
  - …



ManiSkill 2022: rgb, depth, semantic, normal

Sensor: Visual Perception

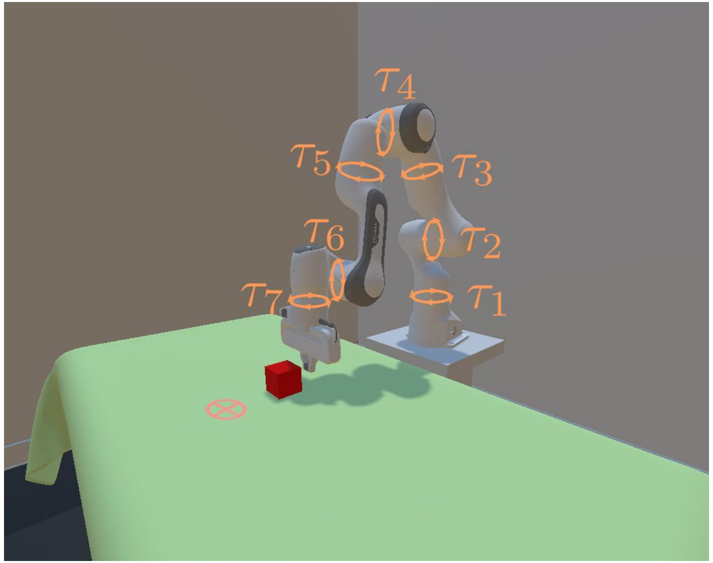
- Common placement
  - ○ Head
  - ○ Arm
  - ○ Third-view



Top: third-view; bottom: arm (wrist)

Sensor: Robot Proprioception

- Joint state
  - Position
  - Velocity
  - Torque
- Link state
  - End-effector pose



Sensors: Haptic/Tactile Perception

- Whether the gripper is grasping something
- Force-torque sensor: measuring external force and torque

Robotiq force-torque sensor

Sensors: GPS and Compass

- Odometry: GPS (relative position) and Compass (relative orientation)
- Used to localize the agent relative to its initial pose

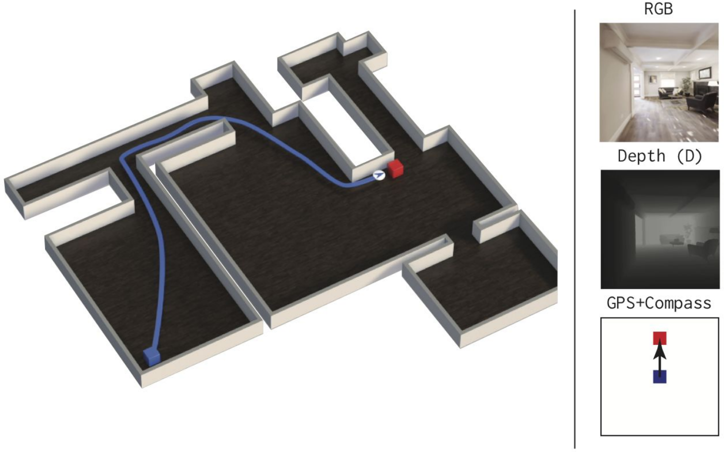
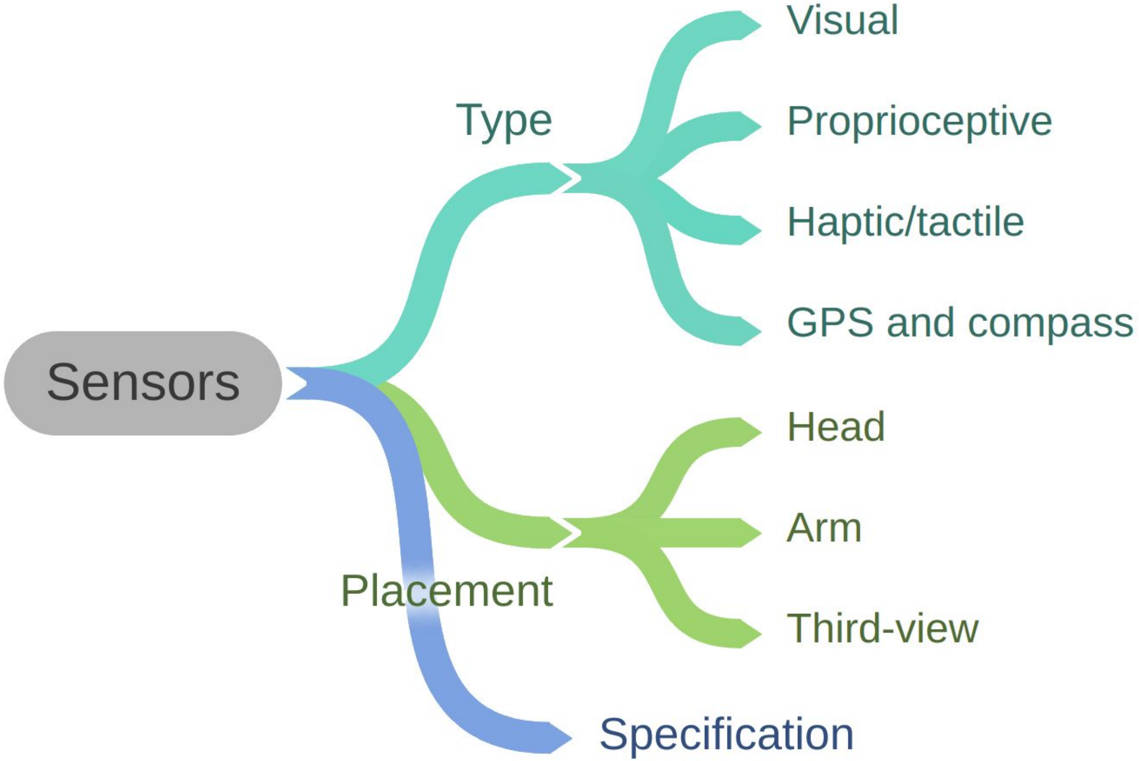


Figure from DD-PPO

Design Choices for Sensors



Sensors

Type

Visual

Proprioceptive

Haptic/tactile

GPS and compass

Placement

Head

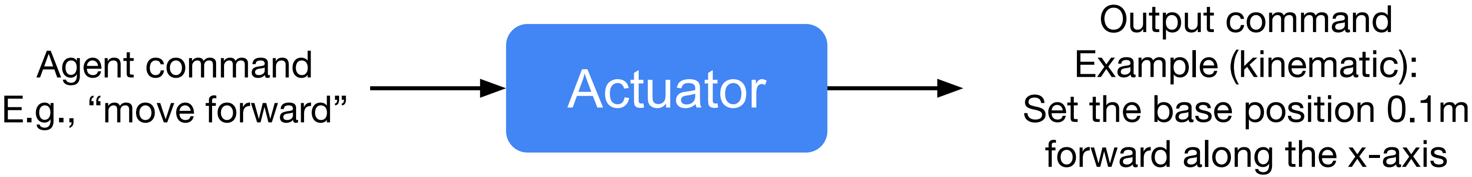
Arm

Third-view

Specification

Embodiment: Actuator

- *“An actuator is a component of a machine that is responsible for moving and controlling a mechanism or system”* from Wikipedia



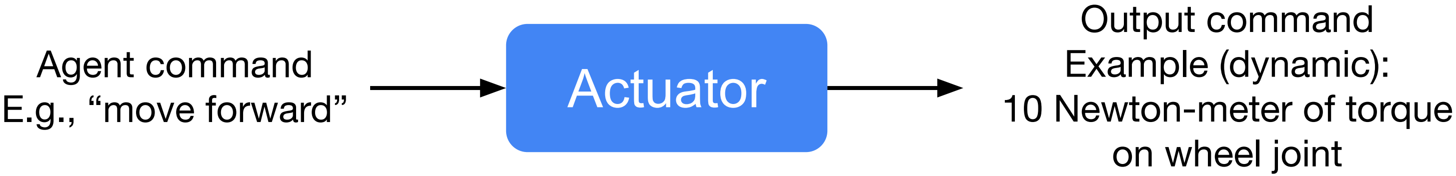
Agent command
E.g., “move forward”

Actuator

Output command
Example (kinematic):
Set the base position 0.1m
forward along the x-axis

Embodiment: Actuator

- *“An actuator is a component of a machine that is responsible for moving and controlling a mechanism or system”* from Wikipedia



Agent command
E.g., “move forward”

Actuator

Output command
Example (dynamic):
10 Newton-meter of torque
on wheel joint

Actuator: Example

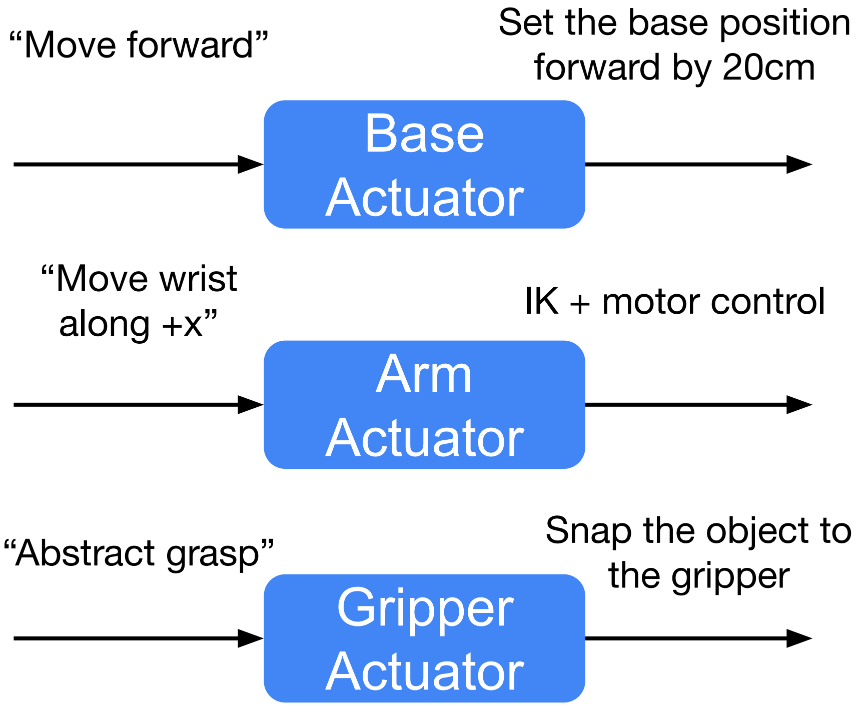
ManipulaTHOR

A Framework for Visual Object Manipulation

Ai2

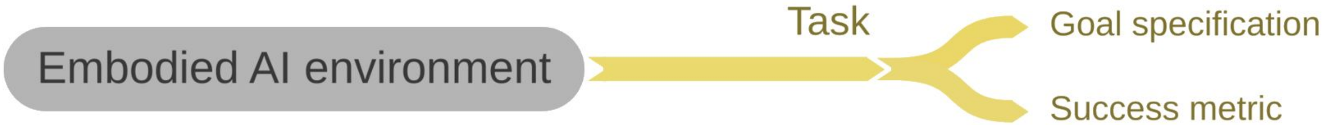
Allen Institute for AI

ArmPointNav



Design Choice: Task Specification

- Goal specification
- Success metric



Task: Goal Specification

- Geometric position
- Object category
- Semantic/instance masks
- Language instruction
- Goal image
- Successful demonstration
- …

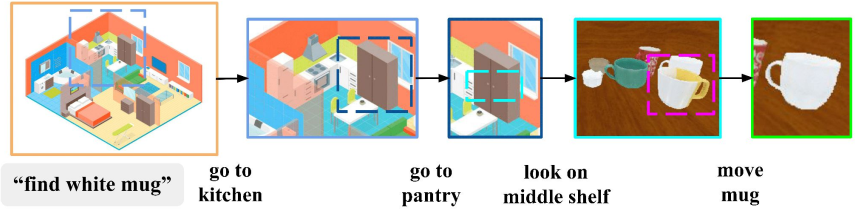
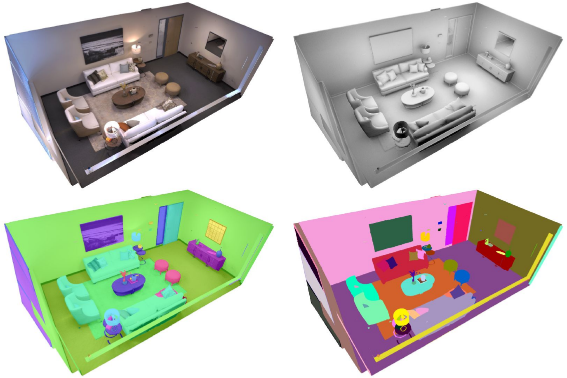


Figure from Multi-Layer Semantic and Geometric Modeling with Neural Message Passing in 3D Scene Graphs for Hierarchical Mechanical Search

Task: Success Metric

- Common success metrics
  - Navigation: The agent stops close enough to the goal
  - Manipulation: The pose of the target object is close enough to the goal
- Additional success requirements
  - The arm returns to a resting position
  - The robot is static
  - The states of target objects satisfy success predicates
  - …

Design Choice: Assets



Non-interactive scenes

[Replica Dataset]

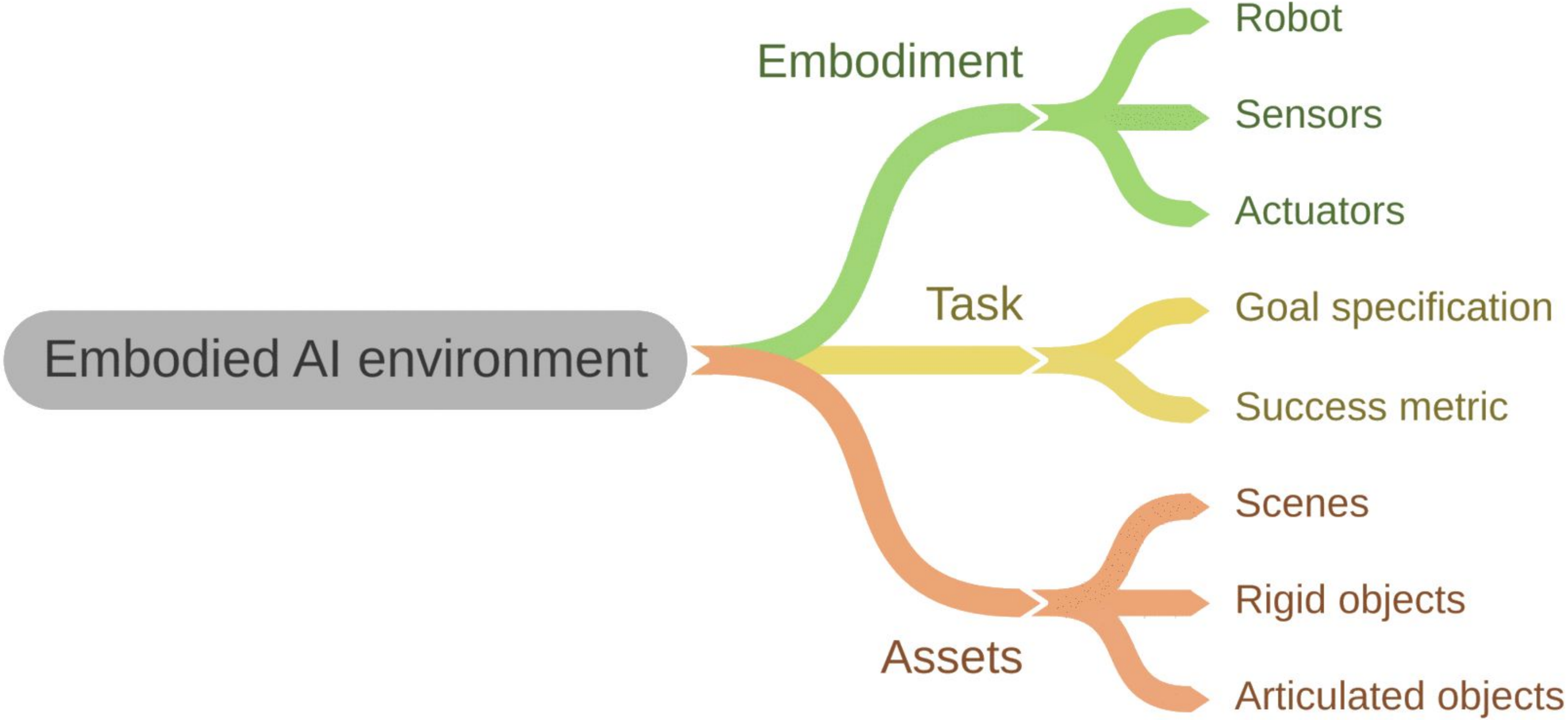
Rigid objects

[YCB Object Set]

Articulated objects

[PartNet-Mobility Dataset]

Recap: Design Choices



- Embodied AI environment
  - Embodiment
    - Robot
    - Sensors
    - Actuators
  - Task
    - Goal specification
    - Success metric
  - Assets
    - Scenes
    - Rigid objects
    - Articulated objects

Physical vs. Non-physical Simulation

- Certain design choices do not need physical simulation
  - Sensor: (perfect) semantic/instance mask
  - Actuator: high-level input command and kinematic output command, e.g., like “pick an apple” implemented by setting the position of apple without simulation
  - Assets: non-interactive scenes

Case Studies:
AI Habitat, AI2THOR, ManiSkill

The cases studied here are not exhaustive. There are also other embodied AI environments, like BEHAVIOR (iGibson), MetaWorld, Robosuite, RLBench, ThreeDWorld, etc.

Case Study I: AI Habitat

- Habitat 1.0: Navigation
  - PointNav
  - ObjectNav
  - Instruction following
  - Embodied QA

The figures and videos in this section are from Habitat unless specified

Case Study I: AI Habitat

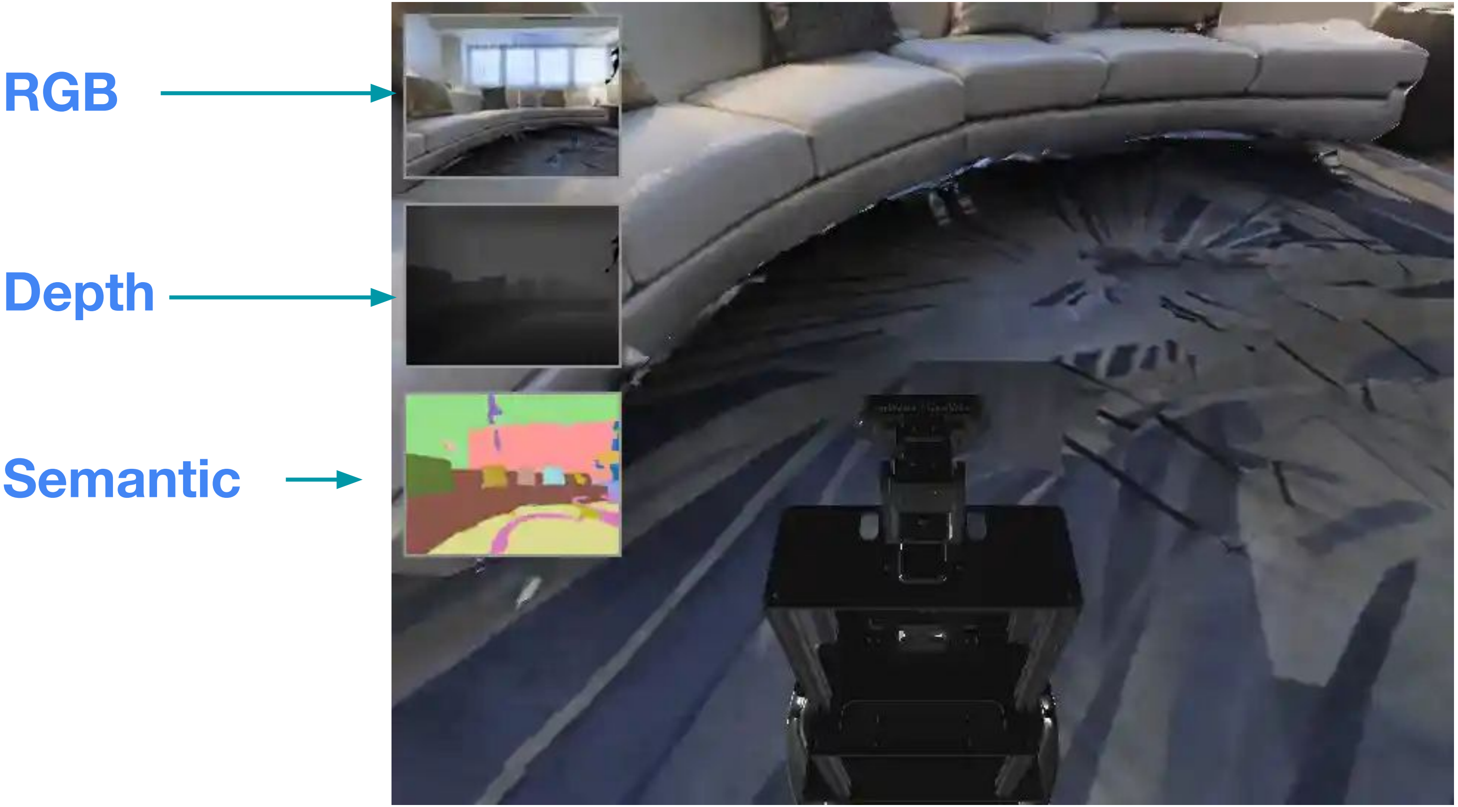
- Habitat 2.0: Mobile manipulation
  - Home Assistant Benchmark
  - Under active development
- We focus on Habitat 1.0 in this talk

Video from https://sites.google.com/view/hab-m3

Embodiment

- Robot: a cylindrical primitive shape with a diameter of 0.2m and a height of 1.5m
- Visual sensors
    - RGB-D camera at a height of 1.5m and oriented to face ‘forward’
    - Semantic instance masks (optional, for training)
- Proprioceptive sensors
    - Perfect GPS and compass (to compute relative goal position)

Embodiment: Visual Sensors



The robot is only for visualization

Embodiment: Actuator

- Input: a discrete action in *move forward* (0.25m), *turn left* (10 deg), *turn right* (10 deg) and stop
- Output: set the pose of the navigation agent without physical simulation
- Navigation constraints and collision response are based on **NavMesh** (explained in the next slide)

NavMesh

- A navigation mesh (NavMesh) is a collection of two-dimensional convex polygons (i.e., a polygon mesh) that define which areas of an environment are traversable by an agent with a particular embodiment.

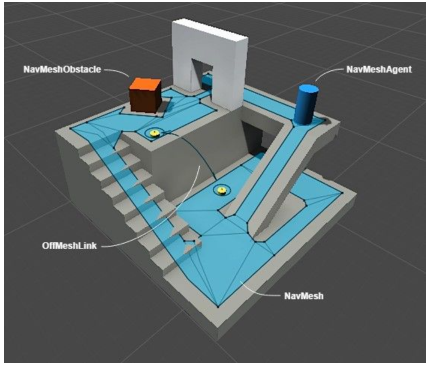
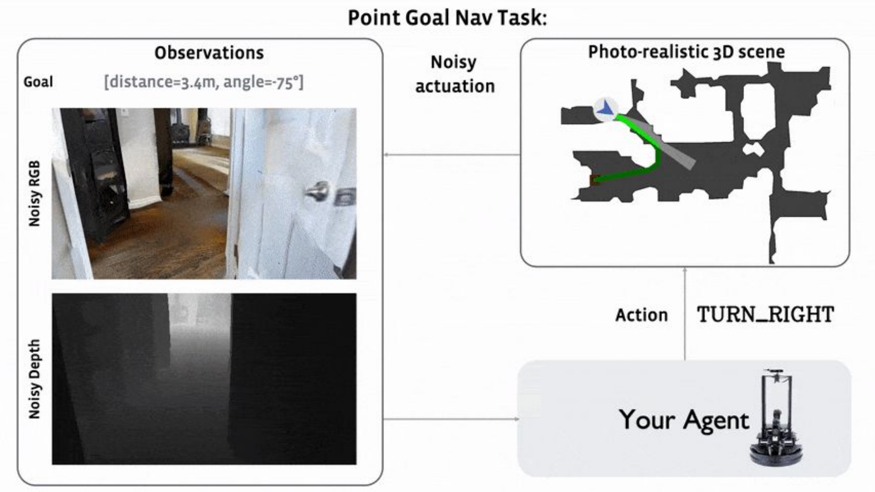


Figure from Dave Johnston's blog

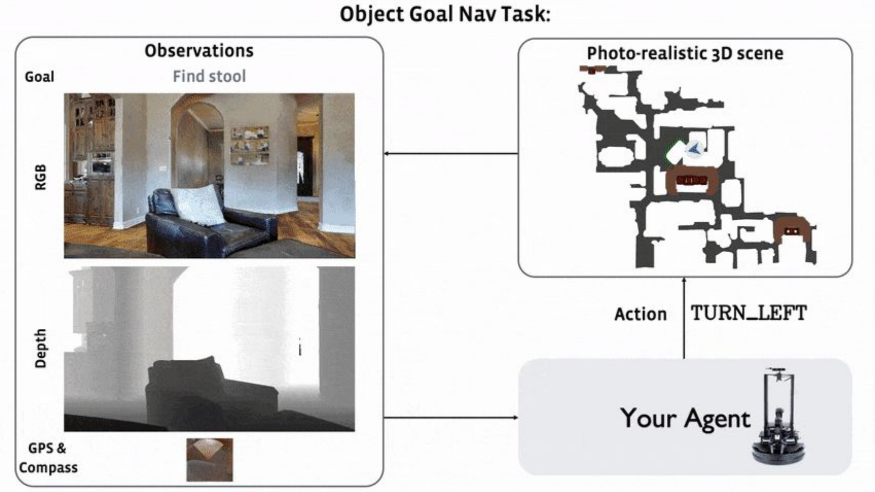
Task Specification

- PointNav
    - Geometric goal: a 2D position relative to the agent's start location
    - *An episode is successful if on calling the STOP action, the agent is within 0.36m (2x agent-radius) of the goal position.*



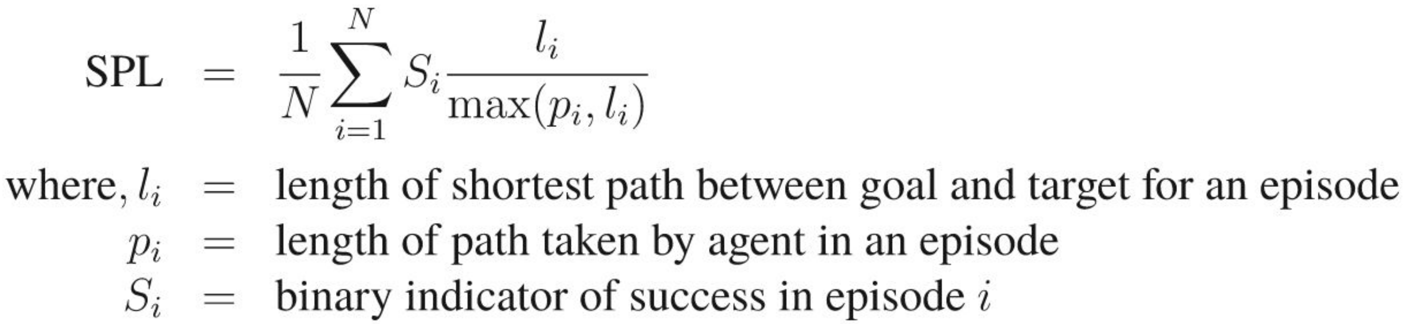
Task Specification

- ObjectNav
  - Semantic goal: a category
  - *An episode is successful if on calling the STOP action, the agent is within 1.0m from any instance of the target object category AND the object can be viewed by an oracle*



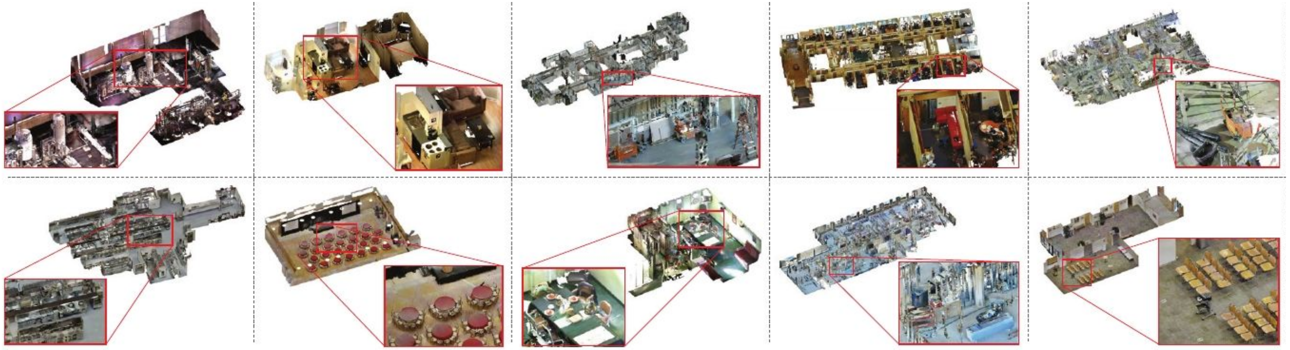
Task Specification: Metrics

- Success weighted by Path Length (SPL)
  - Measure the efficiency to reach the goal

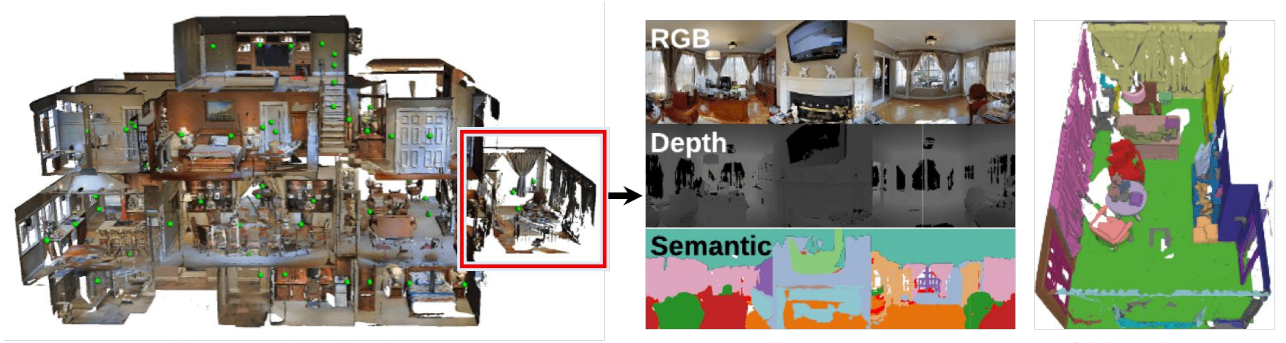


$$
\begin{aligned}
\text{SPL} &= \frac{1}{N}\sum_{i=1}^{N} S_i \frac{l_i}{\max(p_i, l_i)} \\
\text{where, } l_i &= \text{length of shortest path between goal and target for an episode} \\
p_i &= \text{length of path taken by agent in an episode} \\
S_i &= \text{binary indicator of success in episode } i
\end{aligned}
$$

Assets: Scenes



Gibson



Textured 3D Mesh

Panoramas

Object Instances

Matterport3D

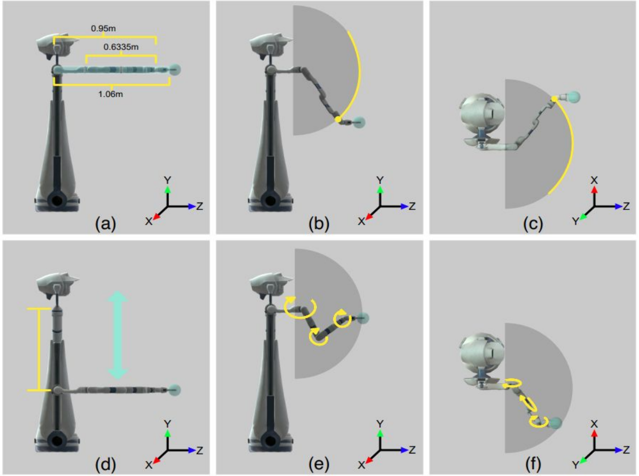
Case Study II: AI2THOR

- iTHOR:
  - Navigation and high-level object interaction
  - Instruction following
  - Embodied QA
- ManipulaTHOR:
  - Mobile manipulation (pick-and-place)

The figures and videos in this section are from AI2THOR unless specified

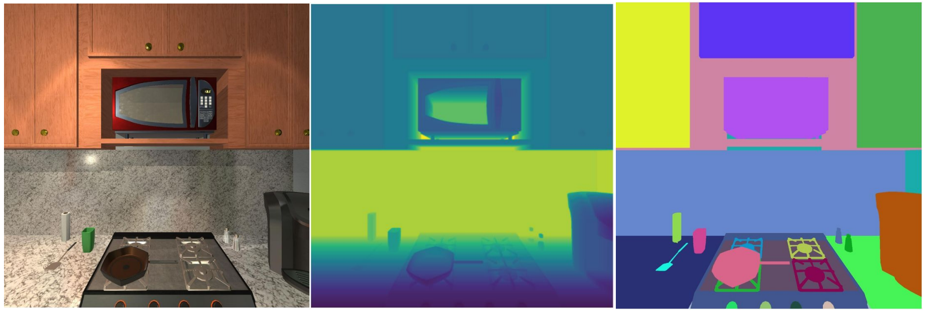
Embodiment: Robot

- Arm based on Kinova Gen3
- The arm height is adjustable



Embodiment: Visual Perception

- Types of sensor signals
  - RGB-D
  - Instance masks and 3D bounding boxes (optional for training)
- Camera placement
  - Mounted on the head, adjustable (height and angle)

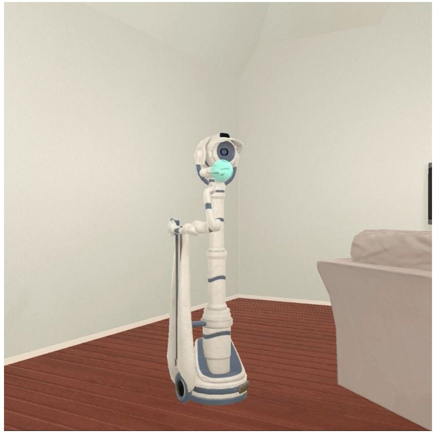


Embodiment: Other Sensors

- iTHOR
  - Base position and rotation (a.k.a. GPS and compass)
  - Primitive action binary feedback (open, pick, slice, etc.)
- ManipulaTHOR
  - Base position and rotation (a.k.a. GPS and compass)
  - End-effector (or joint) poses
  - Grasp binary feedback (based on abstract grasp)

Embodiment: Actuator

- Base
  - Discrete actions (RotateLeft, RotateRight, MoveForward, etc.)
- Arm (ManipulaTHOR)
  - Discrete actions (MoveArmX, MoveArmY, MoveArmHeight, etc.)
  - Target end-effector pose

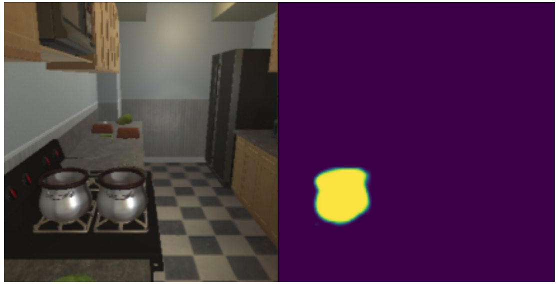


The arm movement

Embodiment: Actuator

- Head
  - Adjust the height and pitch angle of the camera
- Gripper
  - Mask-based primitive actions (for iTHOR)
  - Magnet-based abstract grasp (for ManipulaTHOR)

The head tilts to adjust its camera



Specify target object by seg. mask (iTHOR)

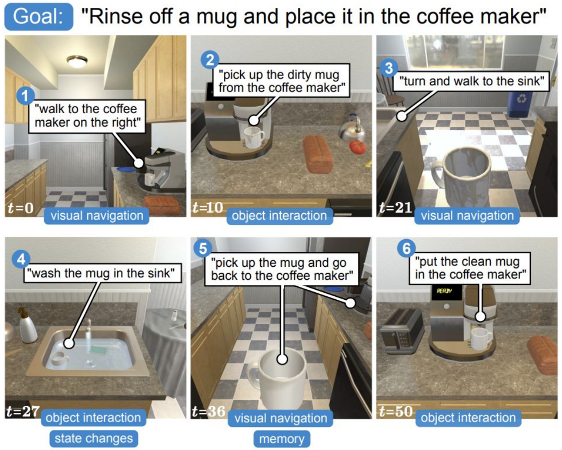
Assets

- Scenes
  - 120 room-scale scenes designed by artists
  - kitchens, living rooms, bedrooms, and bathrooms
- Objects
  - Over 100 object types
    - Rigid objects such as knife
    - Articulated objects such as fridge
  - State changes
    - Temperature, is broken, is dirty, etc.

Scenes from AI2-THOR

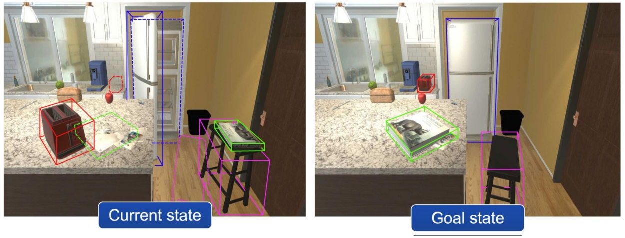
Task: Goal Specification

- Semantic
  - Target object category (RoboTHOR Challenge)
- Language instructions
  - High-level or step-by-step human instructions (ALFRED Challenge)
- Visual observations (Rearrangement Challenge)



Goal: "Rinse off a mug and place it in the coffee maker"
1 "walk to the coffee maker on the right"
t=0 visual navigation
2 "pick up the dirty mug from the coffee maker"
t=10 object interaction
3 "turn and walk to the sink"
t=21 visual navigation
4 "wash the mug in the sink"
t=27 object interaction
state changes
5 "pick up the mug and go back to the coffee maker"
t=36 visual navigation
memory
6 "put the clean mug in the coffee maker"
t=50 object interaction

Example of ALFRED Challenge



Example of Rearrangement Challenge

Success Metrics (iTHOR)

- How to define success?
  - Check the difference between current and target object poses
  - Check the states of target objects (e.g., is food cooked?)
- Metrics
  - Success rate
  - Success rate weighted by path length (SPL)

$$\mathrm{SPL} = \frac{1}{N}\sum_{i=1}^{N} S_i \cdot \frac{\ell_i}{\max(p_i, \ell_i)}$$

Task in ManipulaTHOR

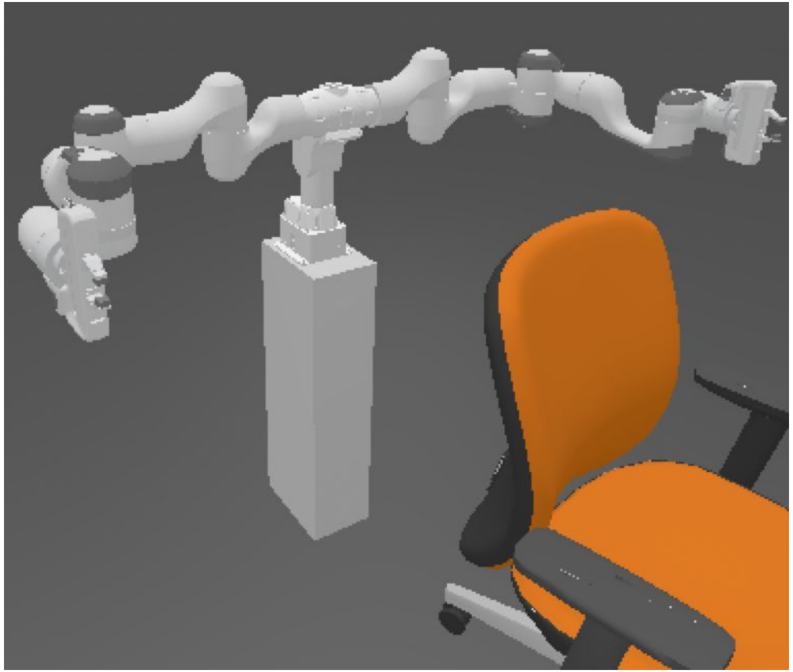
- Supported task: ArmPointNav
- Geometric goal: 3D goal position of the target object
- How to define success?
  - Check the difference between current and target object poses
- Additional metrics
  - Success rate without disturbance/collision

Case Study III: ManiSkill

- Physical manipulation + object-level generalization

Embodiment: Robot

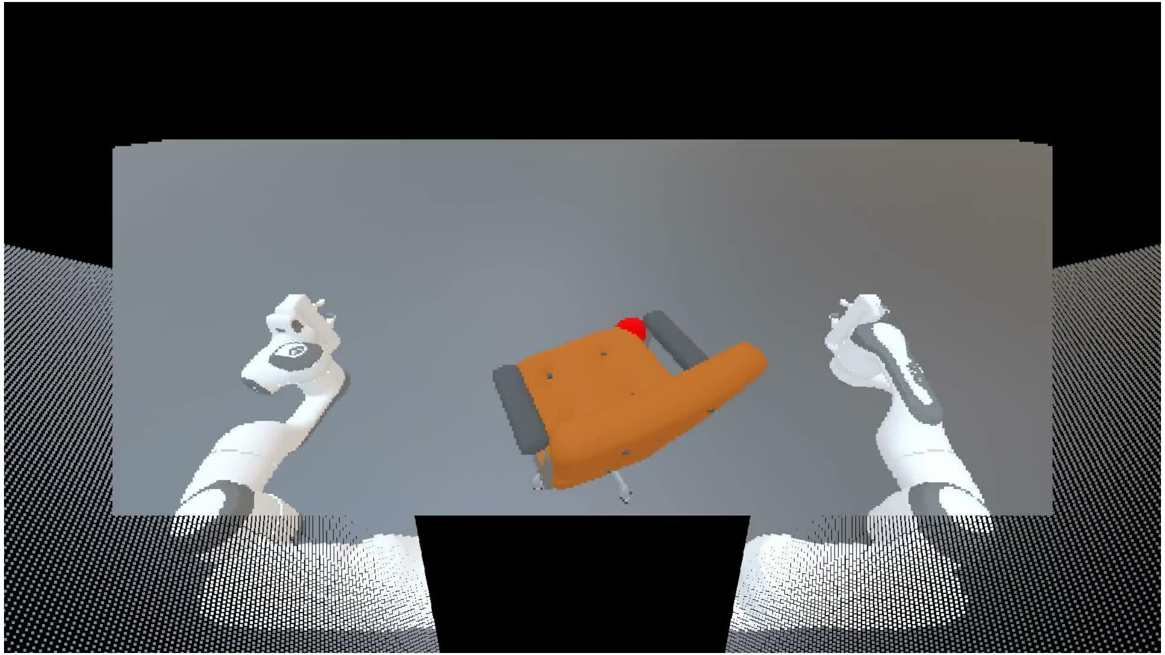
- Mobile platform
  - 3 DoF: x-y-plane translation and z-axis rotation
- Sciurus torso
  - 1 DoF (adjust height)
- Franka Panda arm(s)
  - 7 DoF arm
  - 2 DoF gripper



Embodiment: Sensors

- Visual sensors
  - RGB-D and semantic/instance masks
  - 3 cameras mounted on the top of the robot, with a 120° FOV per camera to provide a panoramic view
- Other sensors
  - Joint positions and velocities
  - Perfect GPS and compass

Embodiment: Visual Sensors

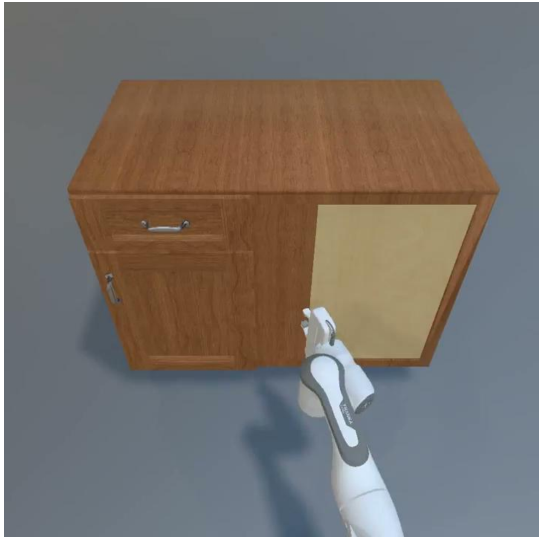


Fused Point Cloud

Embodiment: Actuator

- Input: target velocities of joints
- Output: torques on joints to achieve the target velocities
- It is equivalent to PD joint velocity controller in robotics
- Full physical simulation without abstraction (including grasp)

Task Specification



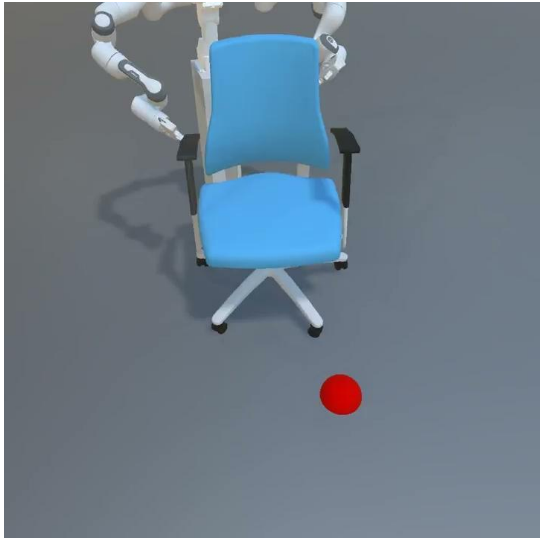
OpenCabinetDoor

OpenCabinetDrawer

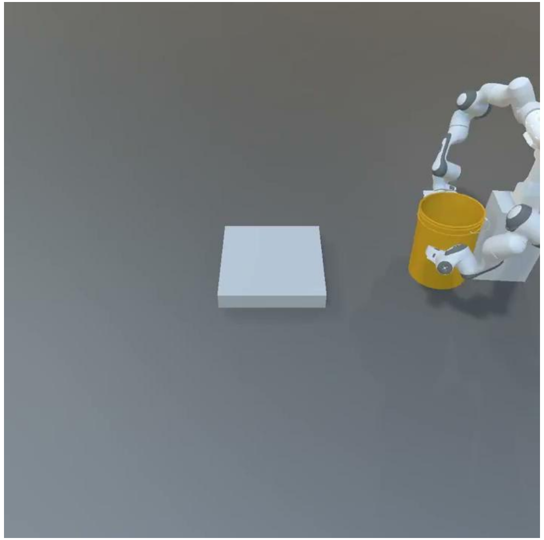
Goal specification: The target link (handle) is specified by the semantic mask

Success metric: The door/drawer is open (the joint state exceeds a threshold)

Task Specification



PushChair



MoveBucket

Goal specification: The goal position (red point or white platform) needs to be inferred from the RGB-D observation

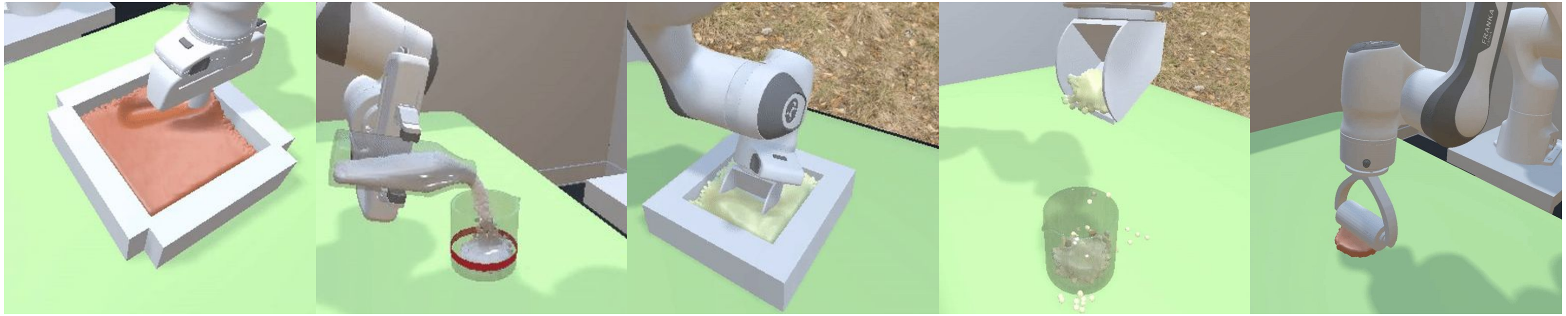
Success metric: The target object is located at the goal position

Assets: Objects

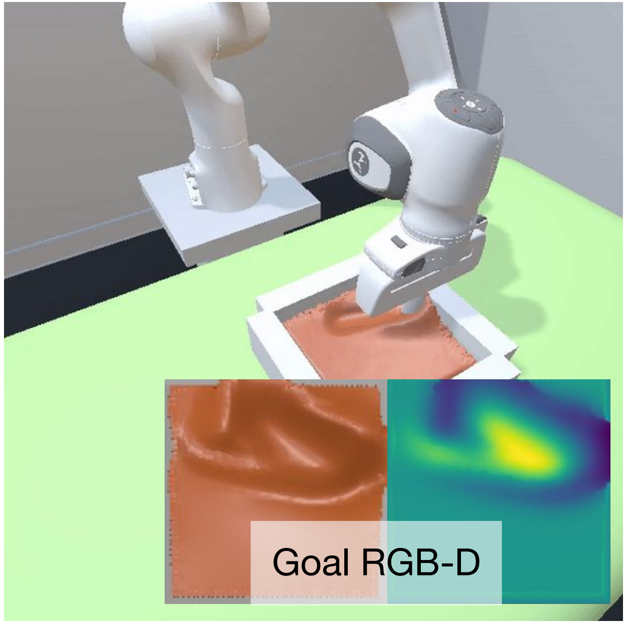
- 162 objects over 4 categories (from PartNet-Mobility)
- Large topology, geometry, and appearance variations

Case Study III*: ManiSkill-Softbody

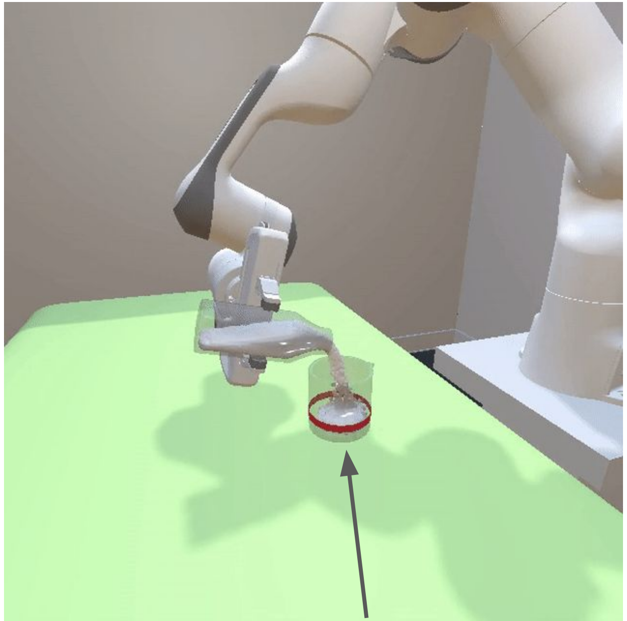
- Full physical rigid+soft body manipulation
- 5 deformable manipulation tasks (to be released in July)
- Embodiment: same as ManiSkill, except for special robot end-effectors — rod, bucket, and rolling pin



Goal Specification

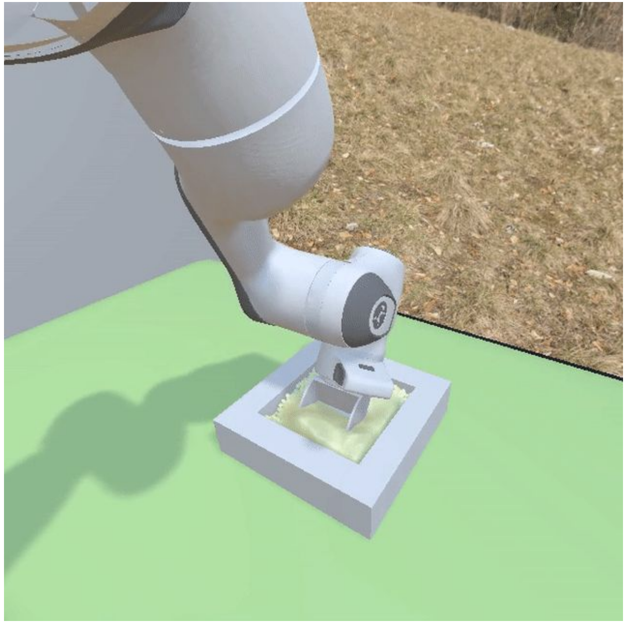


Writer: target shape specified as top-view RGB-D image

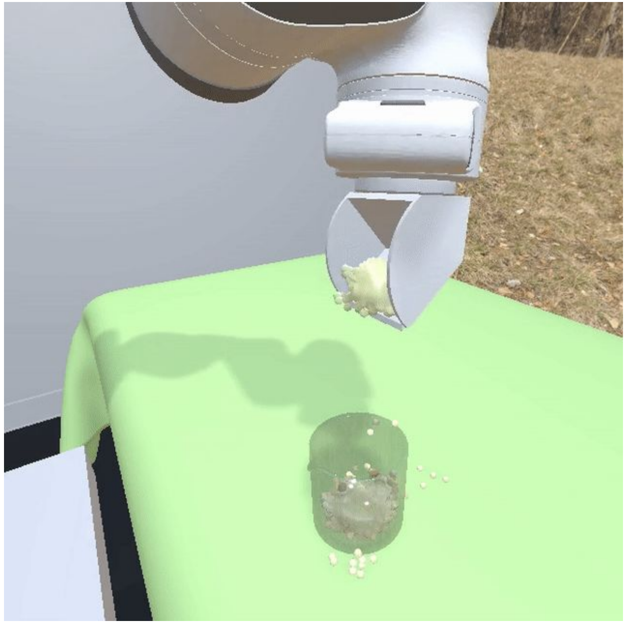


Pouring: desired fill level specified with the red marker ring

Goal Specification



Excavation: target scooped mass specified as a number



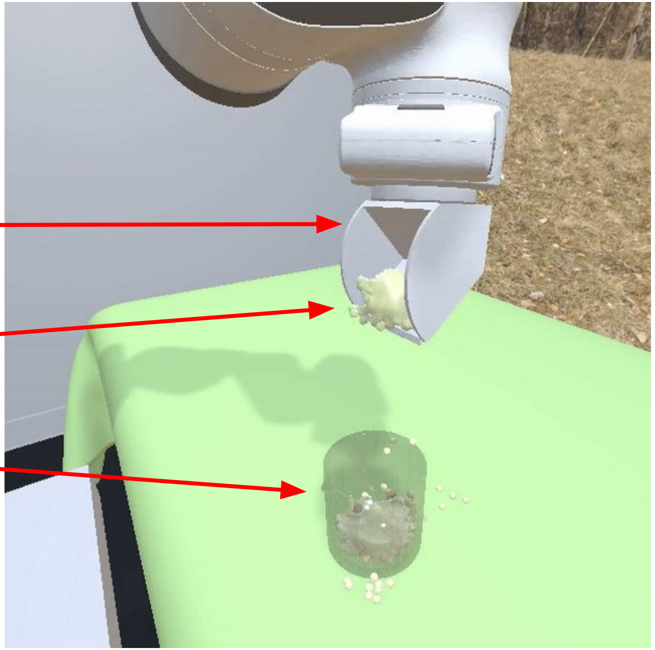
Filling: goal is inferred from placement of the beaker

Goal Specification

Rolling pin: target bounding box of the dough

Assets

- Special robot models
- Soft body represented by particles
- Rigid body containers



Summary

Name	Task Focus	Navigation actuation	Manipulation actuation	Assets
Habitat 1.0	Navigation	Non-physical	N/A	Static scenes
Habitat 2.0	Mobile manipulation	Non-physical	Partially physical	Interactive scenes
iTHOR	Language grounding Interactive navigation	Non-physical	Non-physical	Interactive scenes
ManipulaTHOR	Mobile manipulation	Non-physical	Partially physical	Interactive scenes
ManiSkill	Object manipulation	Physical	Physical	Articulated objects