iGibson 2.0: Object-Centric Simulation for Robot Learning of Everyday Household Tasks

Anonymous Author(s)
Affiliation
Address
e-mail

Abstract: Recent research in embodied AI has been boosted by the use of simulation environments to develop and train robot learning approaches. However, the use of simulation has skewed the attention to tasks that only require what robotics simulators can simulate: motion and physical contact. We present iGibson 2.0, an open-source simulation environment that supports the simulation of a more diverse set of household tasks through three key innovations. First, iGibson 2.0 supports object states, including temperature, wetness level, cleanliness level, and toggled and sliced states, necessary to cover a wider range of tasks. Second, iGibson 2.0 implements a set of predicate logic functions that map the simulator states to logic states like Cooked or Soaked. Additionally, given a logic state, iGibson 2.0 can sample valid physical states that satisfy it. This functionality can generate potentially infinite instances of tasks with minimal effort from the users. The sampling mechanism allows our scenes to be more densely populated with small objects in semantically meaningful locations. Third, iGibson 2.0 includes a virtual reality (VR) interface to immerse humans in its scenes to collect demonstrations. As a result, we can collect demonstrations from humans on these new types of tasks, and use them for imitation learning. We evaluate the new capabilities of iGibson 2.0 to enable robot learning of novel tasks, in the hope of demonstrating the potential of this new simulator to support new research in embodied AI. iGibson 2.0 and its new dataset will be publicly available after the anonymous reviewing process.

1 Introduction

Recent years, we have seen the emergence of many simulation environments for robotics and embodied AI research [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. The main function of these simulators is to compute the motion resulting from the physical contact-interaction between (rigid) bodies, as this is the main process that allows robots to navigate and manipulate the environment. This kinodynamic simulation is sufficient for pick-and-place and rearrangement tasks [11, 12, 13, 14]; however, as the field advances, researchers are taking on more diverse and complex tasks that cannot be performed in these simulators, e.g., household activities that involve changing the temperature of objects, their dirtiness and wetness levels. There is a need for new simulation environments that can maintain and update new types of object states to broaden the diversity of activities that can be studied.

We present iGibson 2.0, an open-source extension of the kinodynamic simulator iGibson with several novel functionalities. First and foremost, iGibson 2.0 maintains and updates new extended physical states resulting from the approximation of additional physical processes. These states include not only kinodynamics (pose, motion, forces), but also object’s temperature, wetness level, cleanliness level, toggled and sliced state (functional states). These states have a direct effect on the appearance of the objects, captured by the high-quality virtual sensor signals rendered by the simulator.

Second, iGibson 2.0 provides a set of logical predicates that can be evaluated with a single object (e.g. Cooked) or a pair of objects (e.g. InsideOf). These logical predicates discriminate the continuous physical state maintained by the simulator into semantically meaningful logical states (e.g. Cooked is True if the temperature is above a certain threshold). Complementary to the discriminative functions, iGibson 2.0 implements generative functions that sample valid simulated physical states based on logical states. Scene initialization can then be described as a set of logical states that the simulator can

Submitted to the 5th Conference on Robot Learning (CoRL 2021). Do not distribute.
translate into valid physical instances. This enables faster prototyping and specification of scenes in iGibson 2.0, facilitating the training of embodied AI agents in diverse instances of the same tasks. We demonstrate the potential of this generative functionality with a new dataset of home scenes densely populated with small objects. We generate this new dataset by applying a set of hand-designed semantic-logic rules to the original scenes of iGibson 1.0.

Third, to facilitate the development of new embodied AI solutions to new tasks in these new scenes, iGibson 2.0 includes a new virtual reality interface (VR) compatible with the two main commercially available VR systems. All states are logged during execution and can be replayed deterministically in the simulator, enabling the generation a posteriori of additional virtual sensor signals or visualizations of the interactions and the development of imitation learning solutions.

We evaluate the new functionalities of iGibson 2.0 on six novel tasks for embodied AI agents, and apply state-of-the-art robot learning algorithms to solve them. These tasks were not possible before in iGibson or in alternative simulation environments. Additionally, we evaluate the use of the new iGibson 2.0 VR interface to collect human demonstrations to train an imitation learning policy for bimanual operations. While the previous version of iGibson and other simulators provide interfaces to control an agent with a keyboard and/or a mouse, these interfaces are insufficient for bimanual manipulation.

In summary, iGibson 2.0 presents the following contributions:

• A set of new physical properties, e.g. temperature, wetness and cleanliness level, maintained and updated by the simulator; and a set of unary and binary logical predicates that map simulated states to a logical state that have a direct connection to semantics and language,

• A set of generative functions associated with the logical predicates to sample valid simulated states from a given logical state, and a new rule-based mechanism exploiting these functions to populate the iGibson scenes with small objects placed at semantically meaningful locations to increase realism,

• A novel virtual reality interface that allows humans to collect demonstrations for robot learning,

We hope that iGibson 2.0 will open new avenues of research and development in embodied AI, enabling solutions to household activities that have been under-explored before.

2 Related Work

Simulation environments with (mostly) kinodynamic simulation: In the last years, the robotics and AI communities have presented several impressive simulation environments and benchmarks:

iGibson [3, 1], Habitat AI [5], AI2Thor (and variants) [15, 6], ThreeDWorld [7], Sapien [4], Robosuite [9], VirtualHome [16], RLBench [8], MetaWorld [10], and more. They are based in physics engines such as (py)bullet [17], MuJoCo [18], and Nvidia Physx [19, 20], combined with rendering capabilities and usually enriched with a dataset of objects and/or scenes to use to develop embodied AI solutions. While these simulators have fueled research with new possibilities for training, testing and developing robotic solutions, they have skewed, with few exceptions, the exploration towards activities related to what they can simulate accurately: changes in kinematic states of (rigid or flexible) objects, i.e. Rearrangement tasks [11]. However, many everyday activities require the simulation of other physical states that can be modified by the agents, like the temperature of objects and their level of wetness or cleanliness. Recent simulators have attempted realistic simulation of fluids and flexible materials [19, 21, 22, 23] or extended rigid body simulators to approximate the dynamics of...
soft materials [17, 24]. Compared with these simulators, iGibson 2.0 provides a simple but effective mechanism to simulate the temperature of objects that change based on proximity to heat sources. It also simulates fluids through a system of droplets that can be absorbed by objects and change their wetness level. Despite simpler than the accurate simulation of heat transfer, fluids and soft materials used by a few other simulators, iGibson 2.0 object-centric solution leads to realistic robotic behavior and motion in tasks involving changing temperature, handling liquids, or soaking objects.

Simulation environments with object-centric representation: In robotics, some simulators have adopted an object-centric representation with extended physical states, e.g., AI2Thor [15] and VirtualHome [16]. Both are based on Unity [25] and share a common set of functionalities. Actions in these simulators are predefined, discrete and symbolic, and characterized by preconditions (“what conditions need to be fulfilled for this action to be executable?”) and postconditions (“what conditions will change after the execution of this action?”), similar to actions in the planning domain definition language (PDDL [26]) or STRIPS [27] but with the additional link to visual rendering of the predefined action execution and outcome. In AI2Thor and VirtualHome, some of the actions change the temperature of an object between two or more discrete values pre- and post-execution of an action, e.g., raw and cooked. This is fundamentally different to our approach in iGibson 2.0: instead of maintaining only a symbolic state (raw/cooked), we provide a simple simulation of the underlying physical process (e.g., heat transfer) leading to continuously varying values of temperature and other extended states. These states are then mapped into a symbolic representation through predicates (see Sec. 4). This provides a new level of detail in the execution of actions, where the agent can and should control the specific value of the object’s extended states (temperature, wetness, cleanliness) to achieve a task, leading to more complex activities and more realistic execution.

Simulation environments with virtual reality interfaces: Researchers have used virtual reality interfaces before to develop robotic solutions, by using VR to teleoperate real robots [28, 29, 30]. VR has also been used as an interface in simulation environments. VRKitchen [31] collected demonstrations for five cooking tasks in simulation. While realistic looking, activities in VRKitchen are performed with primitive actions similar to the pre-condition/post-condition system of AI2Thor and VirtualHome, falling short to obtain demonstrations to achieve tasks with realistic motion. More physically realistic are the interactions in VR with UnrealROX/RobotriX [32, 33] and ThreeDWorld [7], based on Unreal [34] and Nvidia Physx [20]. Our interface also enables realistic manipulation of objects, with additional features such as gaze tracking and assistive grasping to bridge the differences between simulation and the real-world.

3 Extended Physical States for Simulation of Everyday Household Tasks

To perform household tasks, an agent needs to change objects’ states beyond their poses. iGibson 2.0 extends objects with five additional states: temperature, $T$, wetness level, $w$, cleanliness level (dustiness level, $d$, or stain level, $s$), toggled state, $TS$, and sliced state, $SS$. While some of these states could be different for different parts of an object, in iGibson 2.0 we simplify their simulation and adopt an object-centric representation: the simulator maintains a single value of each extended state for every simulated object (rigid, flexible, or articulated). This simplification is sufficient to simulate realistically household tasks such as cooking or cleaning. We assume that the extended properties are latent: agents are not able to observe them directly. Therefore, iGibson 2.0 implements a mechanism to change objects’ appearance based on their latent extended states (see Fig. 2), so that visually-guided agents can infer the latent states from sensor signals.

We further impose in iGibson 2.0 that every simulated object should be an instance of an existing object category in WordNet [35]. This semantic structure allows us to associate characteristics to all instances of the same category [36, 37]. For example, we further simplify the simulation of extended states by annotating what extended states each category need. Not all object categories need all five extended states (e.g., the temperature is not necessary/relevant for non-food categories for most tasks of interest). The extended states required by each object category are determined by a crowdsourced annotation procedure in the WordNet hierarchy.

In the following, we explain the details of the five extended states and the way they are updated in iGibson 2.0. For a full list of object states (kinematics and extended), see Table A.4.

Temperature: To update temperature, iGibson 2.0 needs to approximate the dynamics of heat transfer from heat sources (hot) or sinks (cold). To that end, we annotate object categories in the WordNet hierarchy as heat sources or heat sinks. Heat sources elevate the objects’ temperature
Wetness Level: Similar to temperature, iGibson 2.0 maintains the level of wetness for each object with the particle. Once a dirt particle is cleaned, it disappears from objects’ surface. An object that can change its wetness level gets in contact with the droplets and absorbs them, changing its appearance (most right). With the droplets system, iGibson v2.0 provides a simplified liquid simulation sufficient to perform common household activities.

Figure 2: Extended object states I): temperature and wetness level

(b) Object wetness level: (From left to right) A sink –toggleable droplet source– is toggled on, and starts to create droplets (second from left). Droplets can be contained in a receptacle to be poured on other objects (third from left). An object that can change its wetness level gets in contact with the droplets and absorbs them, changing its appearance (most right). The temperature system with heat sources and sinks, enable visually guided execution of household activities such as cooking.

Cleanliness – Dustiness and Stain Level: A common task for robots in homes and offices is to clean dirt. This dirt commonly appears in the form of dust or stains. In iGibson 2.0, the main difference between dust and stains is the way they get cleaned: while dust can be cleaned with a dry cleaning tool like a cleaning cloth, stains can only be cleaned with a soaked cleaning tool like a scrubber. To clean a particle of dirt (dust or stain), the right part of a cleaning tool should get in physical contact with the particle. Once a dirt particle is cleaned, it disappears from objects’ surface.

In iGibson 2.0, objects can be initialized with visible dust or stain particles on its surface. The number of particles at initialization corresponds to a 100% level of dustiness, $d$, or stains, $s$, as we assume that dust/stain particles cannot be generated after initialization. As particles are cleaned, the level
(a) **Object cleanliness level**: (From left to right) An object is initialized with dust particles that can be cleaned with a cloth; Another object is initialized with stains that require a wet tool (a scrubber) to be removed; Realistic behaviors and motion are required in iGibson v2.0 to change the cleanliness level of the objects.

(b) **Object toggling state**: An object with the toggled state, an oven, is initially off. When it is toggled on and its appearance changes accordingly.

(c) **Object slice state**: An object with the sliced state, a peach, is sliced after exerting enough force with the right part of a slicing tool (the sharp edge of a knife).

Figure 3: Extended object states II): cleanliness level (dustiness, stains), toggling, slice state

decreases proportionally to the number of particles removed, reaching a level of 0% dustiness, $d$, or stains when the object is completely clean (no particles left). This extended state allows simulating multiple cleaning tasks in our simulator: the agent needs to exhibit a behavior (motion, use of tools) similar to the one necessary in the real world. Fig. 3a depicts an example of the cleanliness level simulation in iGibson 2.0, for both dust and stain particles.

**Toggled State**: Some object categories in iGibson 2.0 can be toggled on and off. iGibson 2.0 maintains and updates an internal binary functional state for those objects. The functional state can affect the appearance of an object, but also activate/deactivate other processes, e.g. heating food inside a microwave requires the microwave to be toggled on. To toggle the object, a certain area needs to be touched. For object models of categories that can be toggled on/off, we annotate a TogglingLink, an additional virtual fixed link that needs to be touched by the agent to change the toggled state. Fig. 3b depicts an example of an object that can change its toggled state, an oven.

**Sliced State**: Many cooking activities require the agent to slice objects, e.g. food items. Slicing is challenging in simulation environments where objects are assumed to have a fixed (rigid or flexible) 3D structure of vertices and triangular faces. To approximate the effect of slicing, iGibson 2.0 maintains and updates a sliced state in instances of object categories that are annotated as sliceable. When the sliced state transitions to True, the simulator replaces the whole object with two halves. The two halves will be placed at the same location and inherit the extended states from the whole object (e.g. temperature). The transition is not reversible: the object will remain sliced for all upcoming simulated time steps. Objects can only be sliced into two halves, with no further division. The sliced state changes when the object is contacted with enough force (over a slicing force threshold for the object) by a slicing tool, e.g. a knife. Objects of these categories are annotated as SlicingTool. If an object is a slicing tool, it will undergo a second annotation process to obtain a new virtual fixed link that acts as SlicingLink, the part of the slicing tool that can slice an object, e.g., the sharp edge of a knife. Fig. 3c depicts an example of a peach being sliced by a knife. For more information about update rules for all extended physical states, as well as efficient implementation of these update rules, please refer to the appendix.

4 Logical Representation of Physical States

The new extended object states from iGibson 2.0 are sufficient to simulate a new set of household activities in indoor environments. However, there is a semantic gap between the extended states (e.g. temperature or wetness level) and the natural description of activities in a household setup (e.g. cooking apples). To bridge this gap, we define a set of functions that map the extended object states to logical states for single objects and pairs of objects. The logical states are semantically grounded on common natural language representing properties such as cooked or dusty.
268 The list of logical predicates covers kinematic states between pair of objects (InsideOf, OnTopOf, NextTo, InContactWith, Under, OnFloor), states related to the internal degrees of freedom of articulated objects (Open), states based on the object temperature (Cooked, Burnt, Frozen), wetness and cleanliness level (Soaked, Dusty, Stained), and functional state (ToggledOn, Sliced). A complete list of the logic predicates with detailed explanation is included in Table A.1. They allow iGibson 2.0 to map a physical simulated state into an corresponding logical state.

4.1 Generative System based on Logical Predicates

Logical predicates are not only a way to describe the current simulated state with symbolic representation, but also a way to describe initial states that can be used for scene initialization. iGibson 2.0 includes a generative system that uses logical predicates to generate valid instances of tasks described with them. This system facilitates the creation of semantically meaningful initial states for robot learning, without the tedious process of manually annotating the initial distributions per scene.

Different procedures must be followed to generate valid initial states when it comes to different logical predicates. For example, for predicates based on objects’ extended states such as Cooked or ToggledOn, we simply sample values for the extended states that satisfy the requirements of the predicate. For example, to sample a Frozen object, we sample a temperature below the annotated freezing point of that object. On the other hand, to generate initial states with kinematic predicates such as OnTopOf or Inside, additional work must be done. For example, to sample a configuration that satisfies Inside(A, B), we use a ray-casting approach to sample an empty bounding box of the same size as object A, inside object B. A number of challenges arise here: we want to be able to sample positions in different heights inside the object (e.g. for shelves, drawer units, etc.); the bounding box needs to be fully supported on the bottom; the supporting surface needs to be clear of obstacles, roughly flat, and not too far from being horizontal. We use a variety of approaches including ray-casting and analytical methods, explained in further detail in Section A.2 to verify the validity of the sampled positions. Particles for Dusty and Stained are similarly sampled on an object when the logic predicate requires the object to be dusty or stained initially.

4.2 iGibson 2.0 Scenes with Realistic Object Distribution Created by Generative System

One common issue that limits the realism of indoor scenes in simulation is that they are less densely populated than those in the real world. Creating highly populated simulated houses is usually a laborious process that requires manually selecting and placing models of small objects in different locations. Thanks to the generative system in iGibson 2.0, this process can be extremely simplified. The users only need to specify a list of logical predicates that represent a realistic distribution of objects in a house.

We provide as part of iGibson 2.0 a set of semantic rules to generate more populated scenes, and a new version of the iGibson original 15 fully interactive scenes, populated with additional small objects as a result of the application of the rules. Given an indoor scene with multiple types of rooms (kitchen, bathroom, ...) that are populated with furniture containers and appliances (fridge, cabinet, ...), the semantic rules define the probabilities for object instances of diverse categories to be sampled in certain container type in a given room type, e.g. \( p(\text{InsideOf}(\text{Beer}, \text{Fridge}) \land \text{InsideOf}(\text{Fridge}, \text{Kitchen})) \). To generate the more densely populated versions of the 15 scenes, we first collect a large number of small 3D object models created by artists and we annotate them with semantic categories (e.g. cereal, apple, bowl) and realistic dimensions. Then, we apply in a random sequence the logic predicates using the generative system explained in Sec. 4.1, increasing the number of objects in the scene by over 100. The result is depicted in Fig. 4.
Figure 5: Evaluation: We create six novel household tasks in iGibson 2.0 where the agents have to manipulate extended states. These tasks cannot be studied in other simulation environments.

5 Virtual Reality Interface

(iGibson 2.0’s new functionalities enable modeling new household activities and generating multiple instances in more densely populated scenes. To facilitate research in these new, complex tasks, iGibson 2.0 includes a novel virtual reality (VR) interface compatible with major commercially available VR headset through OpenVR [38]. One of the goals is to allow researchers to collect human demonstrations, and use them to develop new solutions via imitation (see Sec. 6).

iGibson 2.0’s VR interface creates an immersive experience: humans embody an avatar in the same scene and for the same task as the AI agents. The virtual reality avatar (see Fig. 1) is composed of a main body, two hands and a head. The human controls the motion of the head and the two hands via the VR headset and hand controllers with an optional, additional tracker for control of the main body. Humans receive stereo images as generated from the point of view of the head of the virtual avatar, at least 30 fps (up to 90 fps) using the PBR rendering functionalities of iGibson [1].

Grasping in VR: Grasping in the real-world, while natural to adult humans, is a complex experience that proves difficult to reproduce with virtual reality controllers. Our empirical observations revealed that while using solely physical grasping, user dexterity was significantly impaired relative to the real-world resulting in unnatural behavior when manipulating in VR. To provide a more natural grasping experience, we implement an assistive grasp (AG) mechanism that enables an additional constraint between the palm and a target object after the user passes a grasp threshold (50% actuation), given the object is in contact with the hand, between the fingers and the palm. This facilitates grasping of small objects, and prevents object slippage. To not render grasping artificially trivial, the AG connection can break if the constraint is violated beyond a set threshold, such as while lifting heavy objects or during intense acceleration, encouraging natural task execution that leverages careful motions and bimanual manipulation. For additional information about the AG, see Sec. A.1 in the Appendix.

Navigating in VR: Navigation of the avatar is controlled by the locomotion of the human. However, the VR space is much smaller than the typical size of iGibson 2.0 scenes. To navigate between rooms, we configured a touchpad in the hand controller that humans can use to translate the avatar.

6 Evaluation

In our evaluation, we test the new functionalities of iGibson 2.0 explained above and that sets it apart from other existing environments. First, we create a set of tasks that showcase the new extended states (see Fig. 5a) as their modification is required to achieve the tasks. In our experiments, we make use of our discriminative and generative logical engine to detect task completion and create multiple instances of each task for training. Then, we use our novel VR interface to collect human demonstrations to train an imitation learning policy for a bimanual task.

Experimental Setup: To evaluate and demonstrate the new capabilities of iGibson 2.0, we create the following six novel tasks for embodied AI agents related to household domains: 1) Grasping Book: the agent has to grasp a book OnTopOf a table and lift it. This task demonstrates the kinematic logical states; 2) Soaking Towel: the agent has to soak a cleaning tool (a towel) with water droplets coming out of a sink. This task demonstrates the liquid/droplets system. 3) Cleaning Stained Shelf:
the agent has to clean a stained shelf using a cleaning tool. This task showcases the capabilities to
update the cleanliness level of an object. 4) Cooking Meat: the agent has to cook a piece of meat by
placing it on a heat source (a burning stove) and waiting for enough time for the temperature to rise.
This task showcases the capabilities to update the temperature of an object. 5) Slicing Fruit: the agent
has to slice a fruit by exerting enough force with the SlicingLink of a slicing tool (a knife). This
task showcases the capabilities to simulate sliceable objects and to model the interaction between
cutting tools and sliceable objects.

6) Bi-Manual Grasping: The agent has to pick up a heavy object (a cauldron) and place it in another
location. Manipulating this object requires bimanual interaction as its mass is 50 kg and each hand
can only exert at most 300 N. Demonstrating bimanual manipulation is natural and easy with our
novel VR interface, which is not the case with previous interfaces such as mouse and keyboard [1].

In all tasks we use as agent embodiment the same embodiment used in VR, a bimanual humanoid
agent. For evaluation of the tasks 1)-5), we train agents with a state-of-the-art reinforcement learning
(RL) algorithm, Soft-Actor Critic (SAC [39]). The agent receives RGB-D images from its onboard
sensors and proprioception information as observation and outputs the desired linear and angular
velocities of the right hand (assuming the rest of the agent is stationary). We adopt the “sticky mitten”
simplification [11] for grasping that creates a fixed constraint between the hand and the object when
they get into contact. For the evaluation of task 6), we collected 5 demonstrations of 1400+ frames
in total (45 s) and train an imitation learning policy based on behavioral cloning [40]. The agent
receives ground-truth states of the objects and proprioception information as observation, and outputs
the desired linear and angular velocities of both hands. Additional information about the experimental
setup can be found in Sec. A.3.

**Experimental Results:** RL training in tasks 1)-5): The agents trained with SAC are able to find a
solution and achieve 100% success rate for Grasping Book, Soaking Towel, Cleaning Stained Shelf,
Cooking Meat tasks. The agent that does not achieve success always is the one trained in Slicing
Fruit: its success rate is 15% as this task requires very accurate alignment of the knife blade and the
fruit to be accomplished. Beyond the concrete success, these results of RL supports the idea that
iGibson 2.0 enables to study and train embodied AI agents in new household activities that were
traditionally under-explored, as they cannot be simulated by traditional kinodynamic simulators. The
attached video includes visualization of the trained policies performing the tasks thanks to the support
of iGibson 2.0 of new extended physical states and logical predicates that help generating the tasks
and discriminating their completion. We also conducted an ablation study in which we train policies
with three different levels of variability in task 2) (no variations, different poses, different objects and
poses) and evaluate them on an unseen setup (an unseen object with randomized initial pose). The
policies achieve success rate of 19%, 79%, and 87%, respectively. This study shows that it’s essential
to train with diverse object models and initial states to obtain robust policies, and the generative
system of iGibson 2.0 facilitates it by specifying a few logical states that describe the initial scene.

IL training in task 6) from VR demonstrations: The trained policy in the full task diverges and
fails. This is expected for such a long horizon task (300+ steps) with just a handful of demonstrations.
The agent suffers from covariate shift [41], visiting areas not covered by demonstrations. We then
evaluate if the IL policy can successfully perform the last part of the task. We take a successful human
demonstration and we initialize a few seconds before the task completion. We then query the IL
policy from this point onward to control the agent. The results of this experiment are shown in Fig 3b.
The IL policy achieves 20% and 60% success rate when starting 6 s and 3 s away from the goal.
This experiment demonstrates the potential of the new VR interface of iGibson 2.0 for generating
demonstrations for IL. We believe that our new interface will open new avenues for research in
bimanual manipulation, hand-eye coordination, and coordination of robot base and robot arm.

7 Conclusion

We presented iGibson 2.0, a new open-source simulation environment for household task with several
key novel features: 1) an object-centric representation and extended object states (e.g. temperature,
wetness and cleanliness level, toggled and sliced state), 2) logical predicates mapping simulation
states to logical states, and generative mechanism to create simulated worlds based on a given logical
description, and 3) a virtual reality interface to easily collect human demonstrations for imitation.
We demonstrate in multiple experiments the new avenues for research enabled by iGibson 2.0. We
hope iGibson 2.0 becomes a useful tool for the community, reducing the burden of creating simulated
worlds for more diverse tasks, and facilitating the development of novel embodied AI solutions.
References


