A SURVEY OF EMBODIED AI: FROM SIMULATORS TO RESEARCH TASKS

Jiafei Duan[⋆] *Samson Yu*[*] *Hui Li Tan*[†] *Hongyuan Zhu*[†] *Cheston Tan*[†]

duan0038@e.ntu.edu.sg, samson_yu@mymail.sutd.edu.sg, {hltan,zhuh, cheston-tan}@i2r.a-star.edu.sg
[⋆] Nanyang Technological University, Singapore
[*] Singapore University of Technology and Design
[†]Institute for Infocomm Research, A*STAR

ABSTRACT

There has been an emerging paradigm shift from the era of "internet AI" to "embodied AI", whereby AI algorithms and agents no longer simply learn from datasets of images, videos or text curated primarily from the internet. Instead, they learn through embodied physical interactions with their environments, whether real or simulated. Consequently, there has been substantial growth in the demand for embodied AI simulators to support a diversity of embodied AI research tasks. This growing interest in embodied AI is beneficial to the greater pursuit of artificial general intelligence, but there is no contemporary and comprehensive survey of this field. This paper comprehensively surveys state-of-the-art embodied AI simulators and research, mapping connections between these. By benchmarking nine state-of-the-art embodied AI simulators in terms of seven features, this paper aims to understand the simulators in their provision for use in embodied AI research. Finally, based upon the simulators and a pyramidal hierarchy of embodied AI research tasks, this paper surveys the main research tasks in embodied AI – visual exploration, visual navigation and embodied question answering (QA), covering the state-of-the-art approaches, evaluation and datasets.

Index Terms— Embodied AI, Simulators, 3D environment, Embodied Question Answering

1. INTRODUCTION

Recent advances in deep learning, reinforcement learning, computer graphics and robotics have garnered growing interest in developing general-purpose AI systems. As a result, there has been a shift from "internet AI" that focuses on learning from datasets of images, videos and text curated from the internet, towards "embodied AI" which enables artificial agents to learn through interactions with their surrounding environments. The concept of embodied AI can be seen in traces of GOFAI ("Good Old-Fashioned Artificial Intelligence") [1] previously. Many scientists consider embodiment a necessary condition for the development of true intelligence in machines [2]. However, embodied AI now primarily focuses on infusing traditional intelligence concepts such as vision, language, and reasoning into an artificial agent in a virtual environment for deployment in the physical world.

Modern techniques in machine learning, computer vision, natural language processing and robotics have achieved great successes in their respective fields, and the resulting applications have enhanced many aspects of technology and human life in general [3, 4, 5]. However, there are still considerable limitations in existing techniques. Typically known as "weak AI" [6], existing techniques are confined to pre-defined settings, where the nature of the environment does not change significantly [1]. However, the real world is much more complicated and hence further amplifies the difficulties by vast margins. Despite monumental technological advancements such as the AI systems that can beat most of the world champions in their respective games – such as chess [7], Go [8], and Atari games [3] – existing AI systems still do not possess the effectiveness and sophistication of a low-intelligence animal [9].

Growing interest in embodied AI has led to significant progress in embodied AI simulators that aim to faithfully replicate the physical world. These simulated worlds serve as virtual testbeds to train and test embodied AI frameworks before deploying them into the real world. These embodied AI simulators also facilitate the collection of of task-based dataset [10, 11] which are tedious to collect in real-world as it requires an extensive amount of manual labour to replicate the same setting as in the virtual world. While there have been several survey papers in the field of embodied AI [1, 12, 2], they are mostly outdated as they were published before the modern deep learning era, which started around 2009 [13, 14, 15, 16, 8]. To the best of our knowledge, there has been only one survey paper [17] on the research effort for embodied AI in simulators.

This paper makes three major contributions to the field of embodied AI. Firstly, the paper surveys the state-of-the-art embodied AI simulators and provide insights into the specification and selection process of simulators for research tasks. Secondly, the paper provides a systematic look into embodied

AI research directions, and the different stages of embodied AI research that are currently available. Lastly, the paper establishes the linkages between embodied AI simulators' development and the progress of embodied AI research.

2. EMBODIED AI SIMULATORS TO RESEARCH

There is a tight connection between embodied AI simulators and research tasks, as the simulators serve to create ideal virtual testbeds for training and testing of embodied AI frameworks before they are deployed into the physical world. This paper will focus on the following nine popular embodied AI simulators that were developed over the past four years: DeepMind Lab [18], AI2-THOR [19], CHALET [20], VirtualHome [21], VRKitchen [22], Habitat-Sim [23], iGibson [24], SAPIEN [25], and ThreeDWorld [26]. These simulators are designed for general-purpose intelligence tasks, unlike game simulators [27] which are only used for training reinforcement learning agents. These nine embodied AI simulators provide realistic representations of the real world in computer simulations, mainly taking the configurations of rooms or apartments that provide some forms of constraint to the environment. Majority of these simulators minimally comprise physics engine, Python API, and artificial agent that can be controlled or manipulated within the environment.

These embodied AI simulators have given rise to a series of potential embodied AI research tasks, such as *visual exploration*, *visual navigation* and *embodied QA*. The tasks being discussed in this paper have been implemented in at least one of the nine embodied AI simulators covered in the paper. The areas of Sim2Real [28, 29, 30] and robotics will not be covered in this paper.

In this paper, we will provide an contemporary and comprehensive overview of embodied AI simulators and research through understanding the trends and gaps in embodied AI simulators and research. In section 2, this paper outlines state-of-the-art embodied AI simulators and research, drawing connections between the simulators and research. In section 3, this paper benchmarks nine state-of-the-art embodied AI simulators to understand their provision for realism, scalabilty, interactivity and hence use in embodied AI research. Finally, based upon the simulators, in section 4, this paper surveys three main research tasks in embodied AI - visual exploration, visual navigation and embodied QA, covering the state-of-the-art approaches, evaluation, and datasets.

3. SIMULATORS FOR EMBODIED AI

In this section, the backgrounds of the embodied AI simulators will be presented in section 3.1, and the features of the embodied AI simulators will be compared and discussed in Section 3.2.

3.1. Embodied AI Simulators

In this section, we present the backgrounds of the nine embodied AI simulators: DeepMind Lab, AI2-THOR, SAPIEN, VirtualHome, VRKitchen, ThreeDWorld, CHALET, iGibson, and Habitat-Sim. Readers can refer to the corresponding references for details.

DeepMind Lab [18] is the first proof-of-concept of an embodied AI simulator. It is a first-person 3D game platform that is solely developed to research general artificial intelligence and machine learning systems. It was developed out of id Software's Quake III Arena engine. It provides researchers with an environment to perform navigation tasks, fruit collection, movement through narrow spaces, and even laser tag. All of the tasks are inspired by neuroscience experiments. The artificial agent in the environment can perform basic navigation manoeuvres. A reinforcement learning API is being constructed for the environment to better assist with reinforcement learning tasks. The environment is mainly divided into three levels which are meant for different tasks, ranging from fruit gathering and navigation to laser tag. Unlike DeepMind's Arcade Learning Environment (Atari) [27] is made for reinforcement learning research, DeepMind Lab is established to set a benchmark for further embodied AI simulators.

AI2-THOR [19] is a simulator consisting of 120 near photo-realistic 3D scenes of four room categories: kitchen, living room, bed and bathroom. AI2-THOR was built on the Unity 3D game engine, and it provides users with a Python API to perform interactions with the objects in the rooms. One of the main features of AI2-THOR is their actionable objects, which can change their states upon specific actions by the agent. AI2-THOR also provides users with a wide range of manipulation capabilities for their agent, even down to low-level robotics manipulation. AI2-THOR also supports a multi-agent setting for research in multi-agent reinforcement learning. With the previous success in AI2-THOR, the Allen Institute of Artificial Intelligence has further improved the AI2-THOR system and pushed out RoboTHOR [31]. RoboTHOR is an extension of AI2-THOR, where some of the rooms in the AI2-THOR environment have been reconstructed in the real world, allowing users to deploy their trained agent in the real-world.

CHALET Cornell House Agent Learning Environment [20] is an interactive home-based environment which allows for navigation and manipulation of both the objects and the environment. It was developed using the Unity game engine and provides the user with a few deployment versions such as WebGL, the standalone simulator, and a client-based framework. CHALET consists of 58 rooms organized into ten houses with 150 object types. The object types are also mixed with different textures to produce 330 different objects. The artificial agent sees the environment from a first-person perspective. It has very similar features to AI2-THOR.

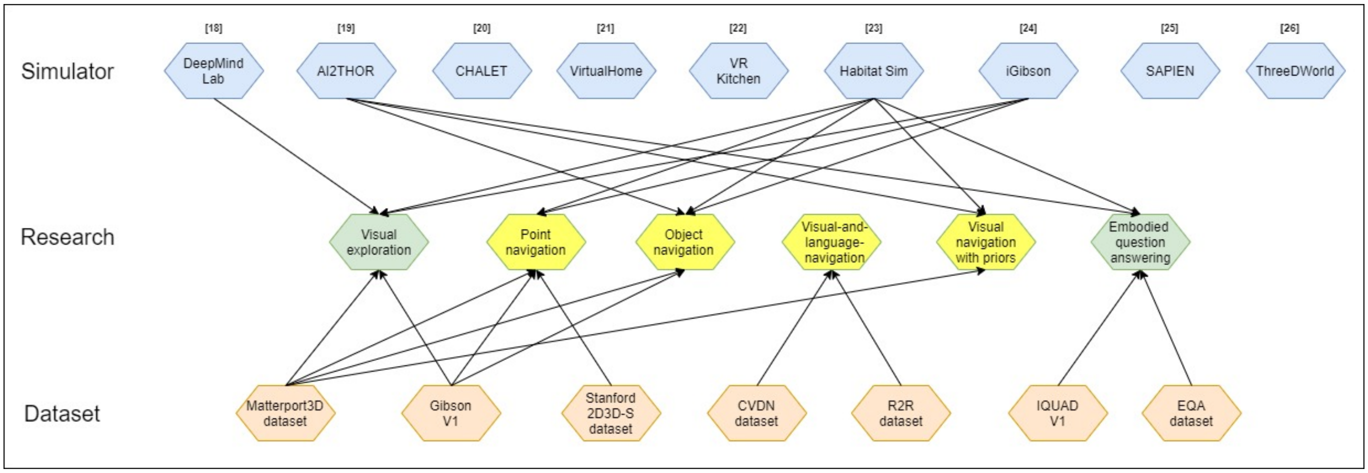

Fig. 1. Connections between Embodied AI simulators to research. (Top) Nine up-to-date embodied AI simulators. (Middle) The various embodied AI research tasks as a result of the nine embodied AI simulators. The yellow colored research tasks are grouped under the visual navigation category while the rest of the green colored tasks are the other research categories. (Bottom) The evaluation dataset used in the evaluation of the research tasks in one of the nine embodied AI simulators.

VirtualHome [21] is a simulator built using the Unity game engine. It possesses in-built kinematics, physics and a navigation model. All the objects built into the simulator comes from the Unity Asset Store. Hence, VirtualHome simulator consists of six apartments and four rigged humanoid models. Each apartment consists of 357 object instances. The VirtualHome simulator requires a program script from the annotators before it can animate the corresponding interaction or tasks that can be performed within its virtual environment.

VRKitchen [22] is a virtual kitchen environment that is constructed using three modules: a physics-based and photo-realistic kitchen environment which is constructed using Unreal Engine 4 (UE4), a user interface module which allows user to perform controls using a virtual reality device or a Python API, and a Python-UE4 bridge, which allows the user to send interactive commands. The artificial agent can perform basic interactions and navigation. VRKitchen consists of 16 fully interactive kitchen scenes, where the 3D models of furniture and appliances were imported from the SUNCG dataset [32]. One of the novelties of VRKitchen is the object state changes that it provides. Hence, some of the objects within VRKitchen can change their states when actions are done to them.

Habitat-Sim [23] is a flexible and high-performance 3D simulator that consist of configurable agents, sensors and 3D datasets. Habitat-Sim can render scenes from both the Matterport3D [33] and Gibson V1 datasets, and is hence very flexible in supporting different 3D environment datasets. Habitat-Sim will load 3D scenes from a dataset and return sensory data from the scenes. Habitat-Sim also provides an API layer that is a modular high-level library aimed at the development of embodied AI, something like the OpenAI Gym. However, the objects imported from Gibson V1 and Matterport3D are from a real-world 3D scan, and they cannot be interacted with.

iGibson [24] is a high fidelity visual-based indoor simulator that provides a high level of physical dynamics between the artificial agent and the objects in the scene. The Interactive Gibson Environment (iGibson) is an improved version of the Gibson V1, as the iGibson presents the user with a new rendering engine that can render dynamical environments and performs much better than the Gibson V1 [34]. Secondly, the iGibson built on top of the Gibson V1, which can augment 106 scenes with 1984 interact-able CAD models under five different object categories: chairs, desks, doors, sofas and tables. With their asset annotation process, they also manage to generate interact-able objects from a single environment mesh. This technique is a massive breakthrough for embodied AI simulators that use photogrammetry for their room construction. iGibson provides users with ten fully functional robotic agents such as MuJoCo's [35] Humanoid and Ant, Freight, JackRabbot V1, TurtleBot V2, Minitaur and Fetch.

SAPIEN A SimulAted Part-based Interactive Environment [25] is a realistic and physics-rich simulated environment that can host a large set of articulated objects. SAPIEN taps into the PartNet-Mobility dataset [36] which contains 14K movable parts over 2346 3D articulated 3D models from 46 standard indoor object classes. One of SAPIEN features is that the robotic agent in SAPIEN possesses a Robot Operating System (ROS) interface that supports three levels of abstraction: direct force control, ROS controllers, and motion planning interface. This feature provides favourable conditions for continuous control, which is favourable for reinforcement learning-based training.

ThreeDWorld [26] is the most recent work on an interactive embodied AI simulator with both photo-realistic scenes in both indoor and outdoor settings. It is also constructed with the Unity game engine using a library of 3D model assets of over 2000 objects spanning 200 categories, such as furniture, appliances, animals, vehicles and toys. However, it has a few additional features that are unique to it. Its high-level physics simulation not only includes rigid-body physics, but also soft-body physics, cloth and fluids. It also has acoustic stimulation during object-to-object or object-to-environment interactions. For user interaction, it enables three ways of interaction: direct API-based, avatar-based and human-centric VR-based. Lastly, it allows multi-agent settings. Despite being one of the most advanced embodied AI simulators, it still has limitations. It lacks articulated objects and a robotics-based avatar system that can perform low-level manipulation.

3.2. Features of Embodied AI Simulators

This section comprehensively compares the nine embodied AI simulators based on seven technical features. Referring to Table 1, the seven features are: Environment, Physics, Object Type, Object Property, Controller, Action, and Multi-Agent.

Environment There are two main methods of constructing the embodied AI simulator environment: game-based scene construction (G) and world-based scene construction (W). Referring to Fig. 2, the game-based scenes are constructed from 3D assets, while world-based scenes are constructed from real-world scans of the objects and the environment. A 3D environment constructed entirely out of 3D assets often have built-in physics features and object classes that are well-segmented when compared to a 3D mesh of an environment made from real-world scanning. The clear object segmentation for the 3D assets makes it easy to model them as articulated objects with movable joints, such as the 3D models provided in PartNet [36]. In contrast, the real-world scans of environments and objects provide higher fidelity and more accurate representation of the real-world, facilitating better transfer of agent performance from simulation to the real world. As observed in Table 1, most simulators other than Habitat-Sim and iGibson have game-based scenes, since significantly more resources are required for world-based scene construction.

Physics A simulator have to construct not only realistic environments but also realistic interactions between agents and objects or objects and objects that model real-world physics properties. We study the simulators' physics features, which we broadly classify into basic physics features (B) and advanced physics features (A). Referring to Fig. 3, basic physics features include collision, rigid-body dynamics, and gravity modelling while advanced physics features include cloth, fluid, and soft-body physics. As most embodied AI simulators construct game-based scenes with in-built physics engines, they are equipped with the basic physics features. On the other hand, for simulators like ThreeDWorld, where the goal is to understand how the complex physics environment can shape the decisions of the artificial agent in the environment, they are equipped with more advanced physics capabilities. For simulators that focus on interactive navigation-based tasks, basic physics features are generally sufficient.

Object Type As shown in Fig. 4, there are two main sources for objects that are used to create the simulators. The first type is the dataset driven environment, where the objects are mainly from existing object datasets such as the SUNCG dataset, the Matterport3D dataset and the Gibson dataset. The second type is the asset driven environment, where the objects are from the net such as the Unity 3D game asset store. A difference between the two sources is the sustainability of the object dataset. The dataset driven objects are more costly to collect than the asset driven objects, as anyone can contribute to the 3D object models online. However, it is harder to ensure the quality of the 3D object models in the asset driven objects than in the dataset driven objects. Based on our review, the game-based embodied AI simulators are more likely to obtain their object datasets from asset stores, whereas the world-based simulators tend to import their object datasets from existing 3D object datasets.

Object Property Some simulators only enable objects with basic interactivity such as collision. Advanced simulators enable objects with more fine-grained interactivity such as multiple-state changes. For instance, when an apple is sliced, it will undergo a state change into apple slices. Hence, we categorize these different levels of object interaction into simulators with interact-able objects (I) and multiple-state objects (M). Referring to Table 1, a few simulators, such as AI2-THOR and VRKitchen, enable multiple state changes, providing a platform for understanding how objects will react and change their states when acted upon in the real world.

Controller Referring to Fig. 5, there are different types of controller interface between the user and simulator, from direct Python API controller (P) and robotic embodiment (R) to virtual reality controller (V). Robotics embodiment allows for virtual interaction of existing real-world robots such as Universal Robot 5 (UR5) and TurtleBot V2, and can be controlled directly using a ROS interface. The virtual reality controller interfaces provide more immersive human-computer interaction and facilitate deployment using their real-world counterparts. For instance, simulators such as iGibson and AI2-THOR, which are primarily designed for visual navigation, are also equipped with robotic embodiment for ease of deployment in their real-world counterparts such as iGibson's Castro [37] and RoboTHOR [31] respectively.

Action There are differences in the complexity of an artificial agent's action capabilities in the embodied AI simulator, ranging from being only able to perform primary navigation manoeuvres to higher-level human-computer actions via virtual reality interfaces. This paper classifies them into three tiers of robotics manipulation: navigation (N), atomic action (A) and human-computer interaction (H). Navigation is the lowest tier and is a common feature in all embodied AI simu-

Table 1. Benchmark for embodied AI Simulators. Environment: game-based scene construction (G) and world-based scene construction (W). Physics: basic physics features (B) and advanced physics features (A). Object Type: dataset driven environments (D) and object assets driven environments (O). Object Property: interact-able objects (I) and multi-state objects (M). Controller: direct Python API controller (P), robotic embodiment (R) and virtual reality controller (V). Action: navigation (N), atomic action (A) and human-computer interaction (H). Multi-agent: avatar-based (AT) and user-based (U). The seven features can be further grouped under three secondary evaluation features; realism, scalability and interactivity.

Year	Embodied AI Simulators	Environment (Realism)	Physics (Realism)	Object Type (Scalability)	Object Property (Interactivity)	Controller (Interactivity)	Action (Interactivity)	Multi-agent (Interactivity)
2016	DeepMind Lab	G	-	-	-	P, R	N	-
2017	AI2-THOR	G	B	O	I, M	P, R	A, N	U
2018	CHALET	G	B	O	I, M	P	A, N	-
2018	VirtualHome	G	-	O	I, M	R	A, N	-
2019	VRKitchen	G	B	O	I, M	P, V	A, N, H	-
2019	Habitat-Sim	W	-	D	-	-	N	-
2019	iGibson	W	B	D	I	P, R	A, N	U
2020	SAPIEN	G	B	D	I, M	P, R	A, N	-
2020	ThreeDWorld	G	B, A	O	I	P, R, V	A, N, H	AT

lators [38]. It is defined by the agent's capability of navigating around its virtual environment. Atomic action provides the artificial agent with a means of performing basic discrete manipulation to an object of interest and is found in most embodied AI simulators. Human-computer interaction is the result of the virtual reality controller as it enables humans to control virtual agents to learn and interact with the simulated world in real time (Gao et al.,2019; Gan et al., 2020a). Most of the larger-scale navigation-based simulators, such as AI2-THOR, iGibson and Habitat-Sim, tend to have navigation, atomic action and ROS [19, 34, 23] which enable them to provide better control and manipulation of objects in the environment while performing tasks such as Point Navigation or Object Navigation. On the other hand, simulators such as ThreeDWorld and VRKitchen [26, 22] fall under the human-computer interaction category as they are constructed to provide a highly realistic physics-based simulation and multiple state changes. This is only possible with human-computer interaction as it provides human-level dexterity when interacting with objects within the simulators.

Multi-agent Referring to Table 1, only a few simulators, such as AI2-THOR, iGibson and ThreeDWorld, are equipped with multi-agent setup, as current research involving multi-agent reinforcement learning is scarce. In general, the simulators need to be rich in object content before there is any practical value of constructing such multi-agent features used for both adversarial and collaborative training [39, 40] of artificial agents. As a result of this lack of multi-agent supported simulators, there have been fewer research tasks that utilize the multi-agent feature in these embodied AI simulators.

For multi-agent reinforcement learning based training, they are still currently being done in OpenAI Gym environments [41] . There are two distinct multi-agent settings. The first is the avatar-based (AT) multi-agents in ThreeDWorld [26] that allows for interaction between artificial agents and simulation avatars. The second is the user-based (U) multi-agents in AI2-THOR [19] which can take on the role of a dual learning network and learn from interacting with other artificial agents in the simulation to achieve a common task [42].

3.3. Comparison of embodied AI Simulators

Based upon the seven features above and with reference to a study by Allen Institute of Artificial Intelligence [43], we propose secondary evaluation features for embodied AI simulators which consist of three key features: realism, scalability and interactivity shown in Table 1. The realism of the 3D environments can be attributed to the *environment* and *physics* of the simulators. The environment models the real world's physical appearance while the physics models the complex physical properties within the real world. Scalability of the 3D environments can be attributed to the *object type*. The expansion can be done via collecting more 3D scans of the real world for the dataset driven objects or purchasing more 3D assets for the asset driven objects. Interactivity of the 3D environments can be attributed to *object property*, *controller*, *action* and *multi-agent*.

Based on the secondary evaluation features of embodied AI simulators, the seven primary features from the Table 1 and the Fig. 1, simulators which possess all of the above three secondary features (e.g. AI2-THOR, iGibson and Habitat-

Fig. 2. Comparison between game-based scene (G) and world-based scene (W). The game-based scene (G) focuses on environment that are constructed from 3D object assets, while the world-based scene (W) are constructed based off real-world scans of the environment.

Sim) are more well-received and widely used for a diverse range of embodied AI research tasks. This further supports the notion that an ideal embodied AI simulator should contain the seven primary features or the three secondary evaluation features.

4. RESEARCH IN EMBODIED AI

In this section, we discuss the various embodied AI research tasks that would derive based on the nine embodied AI simulators surveyed in the previous section. The three main types of embodied AI research tasks are *Visual Exploration*, *Visual Navigation* and *Embodied QA*. As shown in Fig. 6, the tasks are increasingly complex towards the peak of the pyramid. We will start with the fundamental visual exploration before moving up the pyramid to visual navigation and embodied QA. Each of the tasks makes up the foundation for the next tasks as it goes up the pyramid. We will highlight important aspects for each task, starting with the summary, then discussing the methodologies, evaluation metrics, and datasets.

4.1. Sensor Setup Considerations

Sensor suite refers to the sensor(s) that the agent is equipped with. Some of the most popular ones include the RGB, depth and RGB-D sensors. Ablation studies are sometimes done to test the effects of different sensors. An interesting point is that having more sensors does not always improve performance in learning-based approaches for navigation tasks [44, 23], and performance is dependent on the specific use cases. It is hypothesized that more sensors might result in overfitting in datasets with more variety (e.g. different houses look very different) due to high dimensional signals [23].

Sensor and actuation noise have become a more important consideration in recent works as a larger emphasis is placed on the transferability of agent performance to the real world [23, 45]. Most notably, Habitat Challenge 2020 has introduced a noise model acquired by benchmarking the LoCoBot robot, and RGB and depth sensor noises for point navigation [37]. Another recent work uses Gaussian mixture models to create sensor and actuation noise models for point navigation [45]. While sensor and actuation noise can be easily set to zero in a simulation (i.e. idealized sensors), it is not easy to do so in the real world.

4.2. Visual Exploration

In *visual exploration* [46, 47], an agent gathers information about a 3D environment, typically through motion and perception, to update its internal model of the environment [48, 17], which might be useful for downstream tasks like visual navigation [49, 50, 47]. The aim is to do this as efficiently as possible (e.g. with as few steps as possible). The internal model can be in forms like a topological graph map [51], semantic map [52], occupancy map [53] or spatial memory [54, 55]. These map-based architectures can capture geometry and semantics, allowing for more efficient policy learning and planning [53] as compared to reactive and recurrent neural network policies [56]. Visual exploration is usually either done before or concurrently with navigation tasks. In the first case, visual exploration builds the internal memory as priors that are useful for path-planning in downstream navigation tasks. The agent is free to explore the environment within a certain budget (e.g. limited number of steps) before the start of navigation [17]. In the latter case, the agent builds the map as it navigates an unseen test environment [57, 58, 44], which

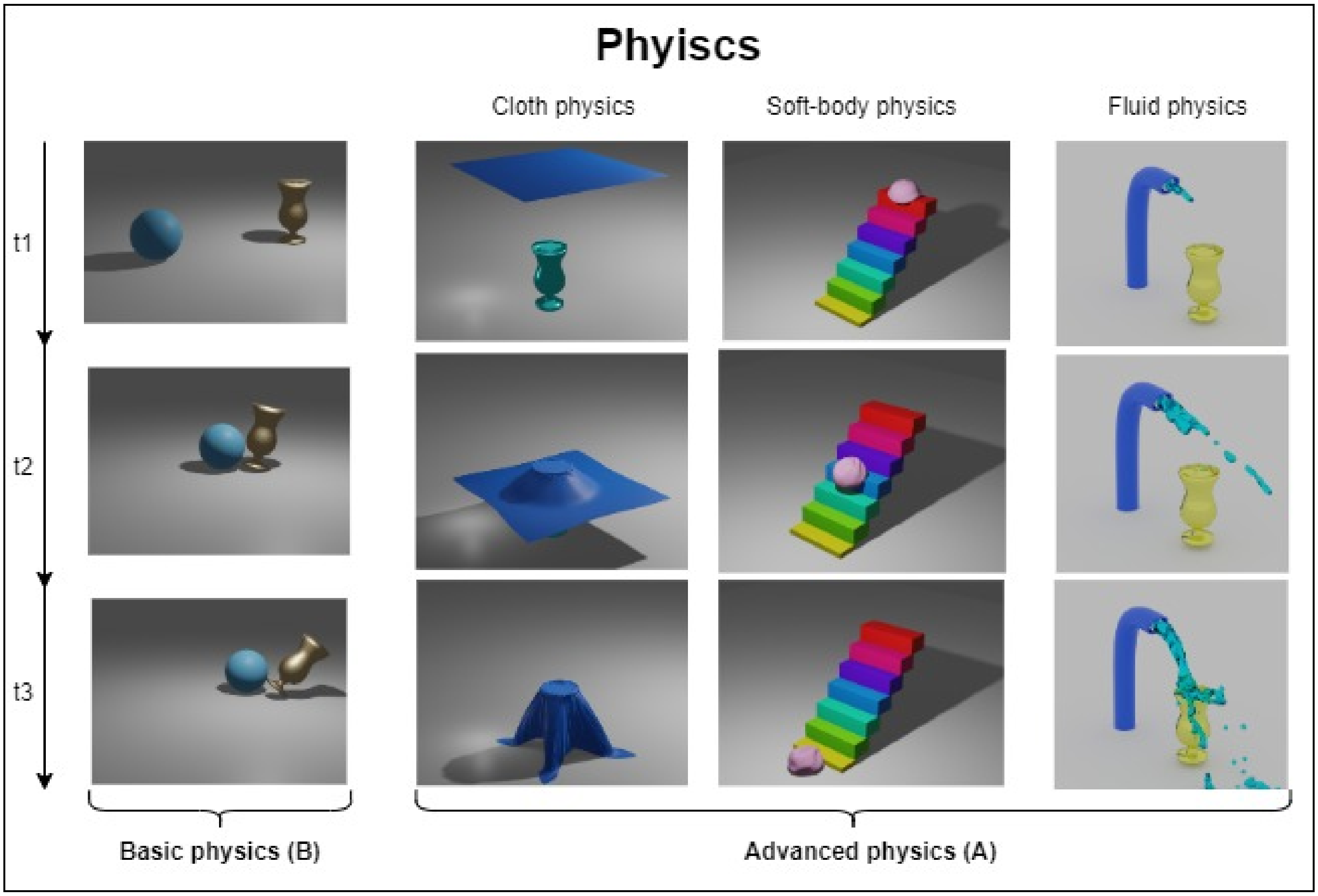

Fig. 3. Comparison between basics physics features such as rigid-body and collision (B) and advanced physics features (A) which includes cloth, soft-body, and fluid physics.

makes it more tightly integrated with the downstream task. In this section, we build upon existing *visual exploration* survey papers [48, 47] to include more recent works and directions.

In classical robotics, exploration is done through passive or active simultaneous localisation and mapping (SLAM) [47, 53] to build a map of the environment. This map is then used with localization and path-planning for navigation tasks. In passive SLAM, the robot is operated by humans to move through the environment, while the robot does it by itself automatically in active SLAM. SLAM is very well-studied [59], but the purely geometric approach has room for improvements. Since they rely on sensors, they are susceptible to measurement noise [47] and would need extensive fine-tuning. On the other hand, learning-based approaches that typically use RGB and/or depth sensors are more robust to noise [45, 47]. Furthermore, learning-based approaches in *visual exploration* allow an artificial agent to incorporate semantic understanding (e.g. object types in the environment) [53] and generalise its knowledge of previously seen environments to help with understanding novel environments in an unsupervised manner. This reduces reliance on humans and thus improves efficiency.

Learning to create useful internal models of the environment in the form of maps can improve the agent's performance [53], whether it is done before (i.e. unspecified downstream tasks) or concurrently with downstream tasks. Intelligent exploration would also be especially useful in cases where the agent has to explore novel environments that dynamically unfold over time [60], such as rescue robots and deep-sea exploration robots.

4.2.1. Approaches

In this section, the non-baseline approaches in *visual exploration* are typically formalized as partially observed Markov decision processes (POMDPs) [61]. A POMDP can be represented by a 7-tuple $(S, A, T, R, \Omega, O, \gamma)$ with state space S, action space A, transition distribution T, reward function R, observation space Ω, observation distribution O and discount factor $\gamma \in [0, 1]$. In general, the non-baseline approaches can be viewed as a particular reward function in the POMDP [48].

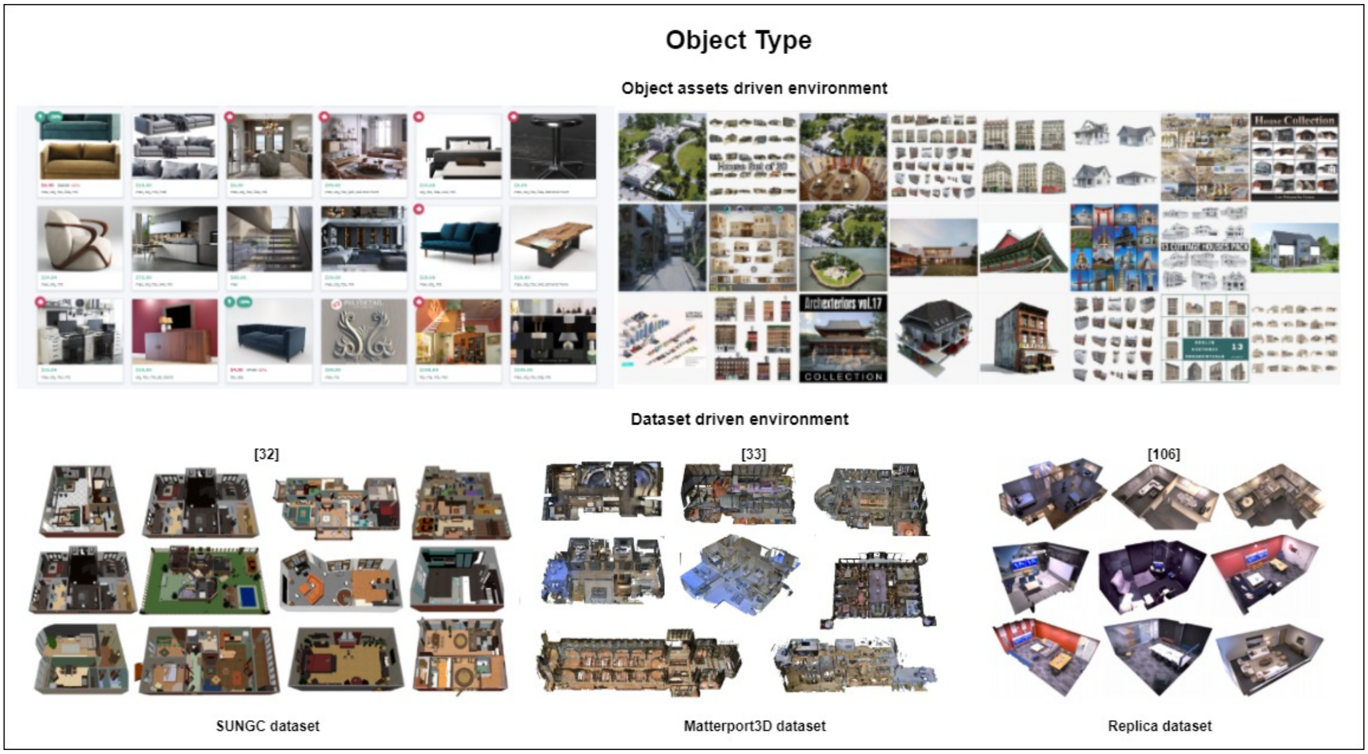

Fig. 4. Comparison between dataset driven environment (D) which are constructed from 3D objects datasets and object assets driven environment (O) are constructed based 3D objects obtain from the assets market.

Baselines. Visual exploration has a few common baselines [48]. For *random-actions* [23], the agent samples from a uniform distribution over all actions. For *forward-action*, the agent always chooses the forward action. For *forward-action+*, the agent always chooses the forward action, but turns left if a collision occurs. For *frontier-exploration*, the agent visits the edges between free and unexplored spaces iteratively using a map [62, 47].

Curiosity. In the *curiosity* approach, the agent seeks states that are difficult to predict. The prediction error is used as the reward signal for reinforcement learning [63, 64]. This focuses on intrinsic rewards and motivation rather than external rewards from the environment, which is beneficial in cases where external rewards are sparse [65]. There is usually a forward-dynamics model that minimises the loss: $L(\hat{s}_{t+1}, s_{t+1})$. In this case, $\hat{s}_{t+1}$ is the *predicted* next state if the agent takes action a_t when it is in state s_t, while s_{t+1} is the *actual* next state that the agent will end up in. Practical considerations for curiosity have been listed in recent work [63], such as using Proximal Policy Optimization (PPO) for policy optimisation. Curiosity has been used to generate more advanced maps like semantic maps in recent work [66]. Stochasticity poses a serious challenge in the curiosity approach, since the forward-dynamics model can exploit stochasticity [63] for high prediction errors (i.e. high rewards). This can arise due to factors like the "noisy-TV" problem or noise in the execution of the agent's actions [65]. One proposed solution is the use of an inverse-dynamics model [46] that estimates the action a_{t-1} taken by the agent to move from its previous state s_{t-1} to its current state s_t, which helps the agent understand what its actions can control in the environment. While this method attempts to address stochasticity due to the environment, it may be insufficient in addressing stochasticity that results from the agent's actions. One example is the agent's use of a remote controller to randomly change TV channels, allowing it to accumulate rewards without progress. To address this more challenging issue specifically, there have been a few methods proposed recently. One method is the Random Distillation Network [67] that predicts the output of a randomly initialized neural network, since the answer is a deterministic function of its inputs. Another method is Exploration by Disagreement [65], where the agent is incentivised to explore the action space where there is maximum disagreement or variance between the predictions of an ensemble of forward-dynamics models. The models in the ensemble converge to mean, which reduces the variance of the ensemble and prevents it from getting stuck in stochasticity traps.

Coverage. In the *coverage* approach, the agent tries to maximise the amount of targets it directly observes. Typically, this would be the area seen in an environment [47, 45, 48]. Since the agent uses egocentric observations,

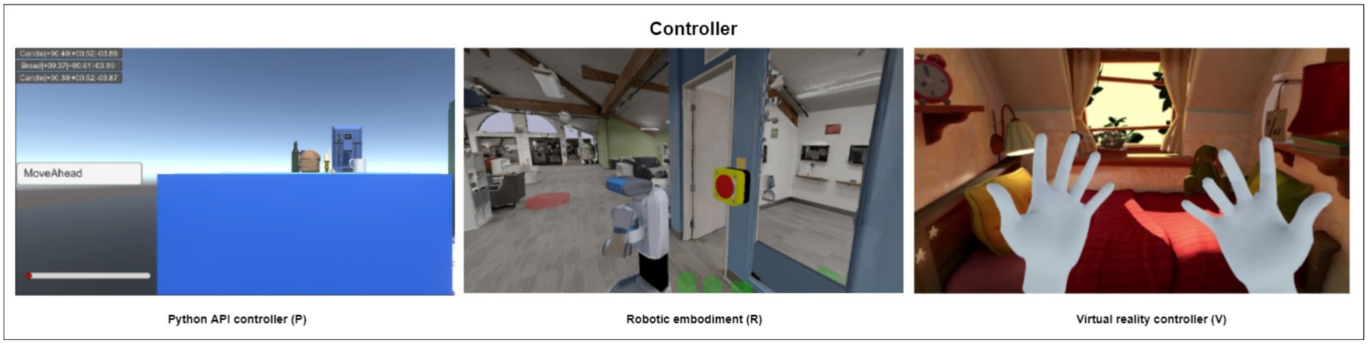

Fig. 5. Comparison between direct Python API controller (P), robotics embodiment (R) which refers to real-world robots with a virtual replica and lastly the virtual reality controller (V).

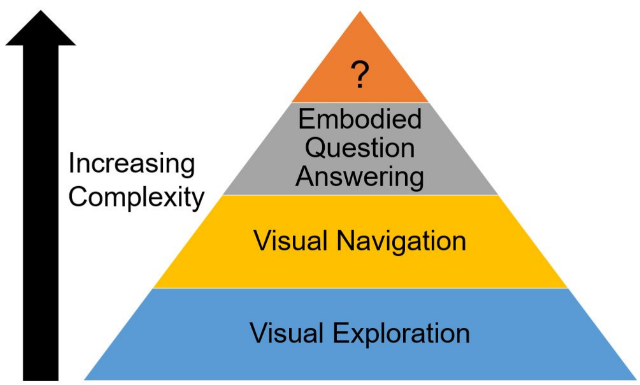

Fig. 6. A hierarchical look into the various embodied AI research tasks with increasing complexity of tasks.

it has to navigate based on possibly obstructive 3D structures. One recent method combines classic and learning-based methods [45]. It uses analytical path planners with a learned SLAM module that maintains a spatial map, to avoid the high sample complexities involved in training end-to-end policies. This method also includes noise models to improve physical realism for generalisability to real-world robotics. Another recent work is a scene memory transformer which uses the self-attention mechanism adapted from the Transformer model [68] over the scene memory in its policy network [56]. The scene memory embeds and stores all encountered observations, allowing for greater flexibility and scalability as compared to a map-like memory that requires inductive biases. A memory factorisation method is used to reduce the overall time complexity of the self-attention block from quadratic to linear.

Reconstruction. In the *reconstruction* approach, the agent tries to recreate other views from an observed view. Past work focuses on pixel-wise reconstructions of 360 degree panoramas and CAD models [69, 70, 71, 72], which are usually curated datasets of human-taken photos [53]. Recent work has adapted this approach for embodied AI, which is more complex because the model has to perform scene reconstruction from the agent's egocentric observations and the control of its own sensors (i.e. active perception). In a recent work, the agent uses its egocentric RGB-D observations to reconstruct the occupancy state beyond visible regions and aggregate its predictions over time to form an accurate occupancy map [53]. The occupancy anticipation is a pixel-wise classification task where each cell in a local area of V x V cells in front of the camera is assigned probabilities of it being explored and occupied. As compared to the *coverage* approach, anticipating the occupancy state allows the agent to deal with regions that are not directly observable. Another recent work focuses on semantic reconstruction rather than pixel-wise reconstruction [48]. The agent is designed to predict whether semantic concepts like "door" are present at sampled query locations. Using a K-means approach, the true reconstruction concepts for a query location are the J nearest cluster centroids to its feature representation. The agent is rewarded if it obtains views that help it predict the true reconstruction concepts for sampled query views.

4.2.2. *Evaluation Metrics*

Amount of targets visited. Different types of targets are considered, such as area [45, 73] and interesting objects [56, 74]. The area visited metric has a few variants, such as the absolute coverage area in m^2 and the percentage of the area explored in the scene.

Impact on downstream tasks. Visual exploration performance can also be measured by its impact on downstream tasks like visual navigation. This evaluation metric category is more commonly seen in recent works. Examples of downstream tasks that make use of visual exploration outputs (i.e. maps) include Image Navigation [51, 57], Point Navigation [45, 17] and Object Navigation [75, 76, 77]. More details about these navigation tasks can be found in Section 4.3.1

4.2.3. Datasets

For visual exploration, some popular datasets include Matterport3D and Gibson V1. Matterport3D and Gibson V1 are both photorealistic RGB datasets with useful information for embodied AI like depth and semantic segmentations. The Habitat-Sim simulator allows for the usage of these datasets with extra functionalities like configurable agents and multiple sensors. Gibson V1 has also been enhanced with features like interactions and realistic robot control to form iGibson. However, more recent 3D simulators like those mentioned in Section 3 can all be used for visual exploration, since they all offer RGB observations at the very least.

4.3. Visual Navigation

In *visual navigation*, an agent navigates a 3D environment to a goal with or without external priors or natural language instruction. Many types of goals have been used for this task, such as points, objects, images [78, 79] and areas [17]. We will focus on points and objects as goals for visual navigation in this paper, as they are the most common and fundamental goals. They can be further combined with specifications like perceptual inputs and language to build towards more complex visual navigation tasks, such as *Navigation with Priors*, *Vision-and-Language Navigation* and even *Embodied QA*. Under point navigation [80], the agent is tasked to navigate to a specific point while in object navigation [81, 38], the agent is tasked to navigate to an object of a specific class.

While classic navigation approaches [82] are usually composed of hand-engineered sub-components like localization, mapping [83], path planning [84, 85] and locomotion, the visual navigation in embodied AI aims to learn these navigation systems from data. This helps to reduce case-specific hand-engineering, thereby easing integration with downstream tasks having superior performance with the data-driven learning methods, such as question answering [86]. There have also been hybrid approaches [45] that aim to combine the best of both worlds. As previously mentioned in Section 3, learning-based approaches are more robust to sensor measurement noise as they use RGB and/or depth sensors and are able to incorporate semantic understanding of an environment. Furthermore, they enable an agent to generalise its knowledge of previously seen environments to help understand novel environments in an unsupervised manner, reducing human effort.

Along with the increase in research in recent years, challenges have also been organised for visual navigation in the fundamental point navigation and object navigation tasks to benchmark and accelerate progress in embodied AI [38]. The most notable challenges are the iGibson Sim2Real Challenge, Habitat Challenge [37] and RoboTHOR Challenge. For each challenge, we will describe the 2020 version of the challenges, which is the latest as of this paper. In all three challenges, the agent is limited to egocentric RGB-D observations. For the iGibson Sim2Real Challenge 2020, the specific task is point navigation. 73 high-quality Gibson 3D scenes are used for training, while the Castro scene, the reconstruction of a real world apartment, will be used for training, development and testing. There are three scenarios: when the environment is free of obstacles, contains obstacles that the agent can interact with, and/or is populated with other moving agents. For the Habitat Challenge 2020, there are both point navigation and object navigation tasks. Gibson 3D scenes with Gibson dataset splits are used for the point navigation task, while 90 Matterport3D scenes with the 61/11/18 training/validation/test house splits specified by the original dataset [17, 33] are used for the object navigation task. For the RoboTHOR Challenge 2020, there is only the object navigation task. The training and evaluation are split into three phases. In the first phase, the agent is trained on 60 simulated apartments and its performance is validated on 15 other simulated apartments. In the second phase, the agent will be evaluated on four simulated apartments and their real-world counterparts, to test its generalisation to the real world. In the last phase, the agent will be evaluated on 10 real-world apartments.

In this section, we build upon existing *visual navigation* survey papers [17, 44, 86] to include more recent works and directions.

4.3.1. Types of Visual Navigation

Point Navigation has been one of the foundational and more popular tasks [45] in recent visual navigation literature. In point navigation, an agent is tasked to navigate to any position within a certain fixed distance from a specific point [17]. Generally, the agent is initialized at the origin $(0, 0, 0)$ in an environment, and the fixed goal point is specified by 3D coordinates (x, y, z) relative to the origin/initial location [17]. For the task to be completed successfully, the artificial agent would need to possess a diverse range of skillsets such as visual perception, episodic memory construction, reasoning/planning, and navigation. The agent is usually equipped with a GPS and compass that allows it to access to their location coordinates, and implicitly their orientation relative to the goal position [23, 80]. The target's relative goal coordinates can either be static (i.e. given only once, at the beginning of the episode) or dynamic (i.e. given at every time-step) [23]. More recently, with imperfect localization in indoor environments in the real world, Habitat Challenge 2020 has moved on to the more challenging task [87] of RGBD-based online localization without the GPS and compass.

There have been many learning-based approaches to point navigation in recent literature. One of the earlier works [44] uses an end-to-end approach to tackle point navigation in a realistic autonomous navigation setting (i.e. unseen environment with no ground-truth maps and no ground-truth agent's poses) with different sensory inputs. The base navigation algorithm is the Direct Future Prediction (DFP) [88] where relevant inputs such as color image, depth map and actions from the four most recent observations are processed by appropriate neural networks (e.g. convolutional networks for sensory inputs) and concatenated to be passed into a two-stream fully connected action-expectation network. The outputs are the future measurement predictions for all actions and future time steps at once.

The authors also introduce the Belief DFP (BDFP), which is intended to make the DFP's black-box policy more interpretable by introducing an intermediate map-like representation in future measurement prediction. This is inspired by the attention mechanism in neural networks, and successor representations [89, 90] and features [91] in reinforcement learning. Experiments show that the BDFP outperforms the DFP in most cases, classic navigation approaches generally outperform learning-based ones with RGB-D inputs. [92] provides a more modular approach. For point navigation, SplitNet's architecture consists of one visual encoder and multiple decoders for different auxiliary tasks (e.g. egomotion prediction) and the policy. These decoders aim to learn meaningful representations. With the same PPO algorithm [93] and behavioral cloning training, SplitNet has been shown to outperform comparable end-to-end methods in previously unseen environments.

Another work presents a modular architecture for simultaneous mapping and target-driven navigation in indoors environments [58]. In this work, the authors build upon MapNet [55] to include 2.5D memory with semantically-informed features and train a LSTM for the navigation policy. They show that this method outperforms a learned LSTM policy without a map [94] in previously unseen environments.

With the introduction of the *Habitat Challenge* in 2019 and its standardized evaluation, dataset and sensor setups, the more recent approaches have been evaluated with the *Habitat Challenge 2019*. The first work comes from the team behind Habitat, and uses the PPO algorithm, the actor-critic model structure and a CNN for producing embeddings for visual inputs. An ablation study is done with different sensors like the depth and RGB-D sensors to set a Reinforcement Learning baseline for the *Habitat Challenge 2019*. It is observed that the depth sensor alone outperforms the other sensor setups and learning-based approaches outperformed classic approaches for point navigation when the agent has more learning steps and data. A follow-up work provides an "existence proof" that near-perfect results can be achieved for the point navigation task for agents with a GPS, a compass and huge learning steps (2.5 billion steps as compared to Habitat's first PPO work with 75 million steps) in unseen environments in simulations [87]. Specifically, the best agent's performance is within 3-5$\%$ of the shortest path oracle. This work uses a modified PPO with Generalized Advantage Estimation [95] algorithm that is suited for distributed reinforcement learning in resource-intensive simulated environments, namely the Decentralized Distributed Proximal Policy Optimization (DD-PPO). At every time-step, the agent receives an egocentric observation (depth or RGB), gets embeddings with a CNN, utilizes its GPS and compass to update the target position to be relative to its current position, then finally outputs the next action and an estimate of the value function. The experiments show that the agents continue to improve for long time, and the results nearly match that of a shortest-path oracle.

The next work aims to improve on this resource-intensive work by increasing sample and time efficiency with auxiliary tasks [80]. Using the same DD-PPO baseline architecture from the previous work, this work adds three auxiliary tasks: action-conditional contrastive predictive coding (CPC—A) [96], inverse dynamics [46] and temporal distance estimation. The authors experiment with different ways of combining the representations. At 40 million frames, the best performing agent achieves the same performance as the previous work $5.5X$ faster and even has improved performance. The winner of the Habitat Challenge 2019 for both the RGB and the RGB-D tracks [45] provides a hybrid solution that combines both classic and learning-based approaches as end-to-end learning-based approaches are computationally expensive. This work incorporates learning in a modular fashion into a "classic navigation pipeline", thus implicitly incorporating the knowledge of obstacle avoidance and control in low-level navigation. The architecture consists of a learned Neural SLAM module, a global policy, a local policy and an analytical path planner. The Neural SLAM module predicts a map and agent pose estimate using observations and sensors. The global policy always outputs the target coordinates as the long-term goal, which is converted to a short-term goal using the analytic path planner. Finally, a local policy is trained to navigate to this short-term goal. The modular design and use of analytical planning help to reduce the search space during training significantly.

Object Navigation is one of the most straightforward tasks, yet one of the most challenging tasks in embodied AI. Object navigation focuses on the fundamental idea of navigating to an object specified by its label in an unexplored environment [38]. The agent will be initialized at a random position and will be tasked to find an instance of an object category within that environment. Object navigation is generally more complex than point navigation, since it not only requires many of the same skillsets such as visual perception and episodic memory construction, but also semantic understanding. These are what makes the object navigation task much more challenging, but also rewarding to solve.

The task of object navigation can be demonstrated or learnt through adapting, which helps to generalize navigation in an environment without any direct supervision. This work [97] achieve that through a meta-reinforcement learning approach, as the agent learns a self-supervised interaction loss which helps to encourage effective navigation. Unlike the conventional navigation approaches for which the agents freeze the learning model during inference, this work allows the agent learns to adapt itself in a self-supervised manner and adjust or correct its mistake afterwards. This approach prevents an agent from making too many mistakes before realizing and make the necessary correction. Another method is to learn the object relationship between objects before executing the planning of navigation. This work [76] implements an object relation graph (ORG) which is not from external prior knowledge but rather a knowledge graph that is built during the visual exploration phase. The graph consists of object relationships such as category closeness and spatial correlations. It also has a Trial-driven imitation learning modules together with a memory-augmented tentative policy network (TPN) to aid in preventing the learning agent from being trapped in a deadlock.

Navigation with Priors focuses on the idea of injecting semantic knowledge or priors in the form of multimodal inputs such as knowledge graph or audio input or to aid in the training of navigation tasks for embodied AI agents in both seen and unseen environments. Past work [98] that use human priors of knowledge integrated into a deep reinforcement learning framework has shown that artificial agent can tap onto human-like semantic/functional priors to aid the agent in learning to navigate and find unseen objects in the unseen environment. Such example taps onto the understanding that the items of interest, such as finding an apple in the kitchen, humans will tend to look at logical locations to begin our search. These knowledge are encoded in a graph network and trained upon in a deep reinforcement learning framework.

There are other examples of using human priors such as human's ability to perceive and capture correspondences between an audio signal modal and the physical location of objects hence to perform navigation to the source of the signal. In this work [99], artificial agents pick multiple sensory observations such as vision and sound signal of the target objects and figure out the shortest trajectory to navigation from its starting location to the source of the sounds. This work achieves it through having a visual perception mapper, sound perception module and dynamic path planners.

Vision-and-Language Navigation (VLN) is a task where agents learn to navigate the environment by following natural language instruction. The challenging aspect of this task is to perceive both the visual scene and language sequentially. VLN remains a challenging task as it requires agents to make predictions of future action based on past actions and instruction [17]. It is also tricky as agents might not be able to align their trajectory seamlessly with natural language instruction. Although Visual-and-Language Navigation and Visual Question Answering (VQA) might seem to be very much similar, there are major differences in both tasks. Both tasks can be formulated as visually grounded, sequence-to-sequence transcoding problems. But VLN sequences are much longer and require a constant feeding input of vision data and the ability to manipulate camera viewpoints, which is unlike VQA which takes in a single input question and performs a series of actions to determine the answer to the question.The notion that we might be able to give out a general, natural language instruction to a robot and expect them to execute or perform the task is now possible. These are [100, 10, 11] achieved with the advancedment of recurrent neural network methods for joint interpretation of both visual and natural language input and datasets that are designed for simplifying processes of task-based instruction in navigation and performing of tasks in the 3D environment.

One of such approaches for VLN is to use Auxiliary Reasoning Navigation framework [101]. It tackles four auxiliary reasoning tasks which are trajectory retelling task, progress estimation task,angle prediction task and cross-modal matching task. The agent learns to reason about the previous actions and predicts future information through these tasks. Vision-dialog navigation is the newest holy-grail tasks for the general task of VLN as it aims to train an agent to develop the ability to engage in a constant conversation in natural language with human to aid in navigation. The current work [101] in this area focuses on having a Cross-modal Memory Network (CMN) to help with remembering and understanding of the rich information related to the past navigation actions and make decisions for the current steps in navigation.

4.3.2. Evaluation Metrics

Apart from VLN, visual navigation uses *success weighted by path length* and *success rate* as the main evaluation metrics [17]. Success weighted by path length can be defined as: $\frac{1}{N}\sum_{i=1}^{N} S_i \frac{l_i}{max(p_i,l_i)}$. S_i is a success indicator for episode i, p_i is the agent's path length, l_i is the shortest path length and N is the number of episodes. It is noteworthy that there are some known issues with success weighted by path length [38]. Success rate is the fraction of the episodes in which the agent reaches the goal within the time budget [44]. There are also other evaluation metrics [17, 44, 58, 80] in addition to the two mentioned.

Besides using shortest path length (SPL), there are also four popular metrics are used to evaluate VLN agents. They are: (1) success rate, which measure the percentage of final position a certain distance away from the goal; (2) Oracle success rate, the rate for which the agent stops at the closet point to the goal; (3) goal progress, which is the average agent progress towards goal location; (4) Oracle path success rate, which is the success rate if the agent can stop at the closet point to goal along the shortest path. In general for VLN tasks, the best metric is still SPL as it takes into account of the path taken and not just the goal.

4.3.3. Datasets

As in visual exploration, Matterport3D and Gibson V1 and the most popular dataset. More details can be found in section 3.2.3. It is noteworthy that the scenes in Gibson V1 are smaller and usually have shorter episodes (lower GDSP from start position to goal position).

Unlike the rest of the visual navigation tasks, VLN requires a different kind of dataset. Most of the VLN works use the R2R dataset from Matterport3D Simulator. It consists of 21,567 navigation instruction with an average length of 29 words. While some like [101] uses the CVDN dataset which comprises collects 2050 human-to-human dialogs and over 7k trajectories within the Matterport3D simulator.

4.4. Embodied Question Answering

The task of embodied QA in recent embodied AI simulators has been a significant advancement in the field of general-purpose intelligence system, as in order to perform a question and answering in a physical embodiment, an artificial agent would need to possess a wide range of AI capabilities such as visual recognition, language understanding, commonsense reasoning, task planning, and goal-driven navigation. Hence, embodied QA can be considered the most onerous and most complicated tasks in embodied AI research.

4.4.1. Methods

A common EQA framework is to divide the task into two sub-tasks: a navigation task and a QA task. The navigation module is essential since the agent needs to explore the environment to see the objects before answering questions about them.

For example, [102] proposed Planner-Controller Navigation Module (PACMAN), which is a hierarchical structure for the navigation module, with a planner that selects actions (directions) and a controller that decides how far to move following this action. Once the agents decide to stop, the question answering module is executed by using the sequence frames along the paths. The navigation module and visual question answering model are first trained individually and then jointly trained by REINFORCE [103]. [104] and [105] further improved the PACMAN model by the Neural Modular Control (NMC) where the higher-level master policy proposes semantic sub-goals to be executed by sub-policies.

Similarly, [106] proposed using a Hierarchical Interactive Memory Network (HIMN) which is factorized into a hierarchy of controllers to help the system operate, learn and reason across multiple time scales, while simultaneously reduce the complexity of each sub-task. An Egocentric Spatial Gated Recurrent Unit (GRU) is used to act as a memory unit for retaining spatial and semantic information of the environment. The planner module will have control over the other modules such as a navigator which runs an A* search to find the shortest path to the goal, a scanner which performs rotation to the agent for detecting new images, a manipulator that is invoked to carry out actions to change the state of the environment and lastly an answerer that will ask the question posted to the artificial agent.

Recently, [107] studied Interactive Question Answering from an multi-agent perspective, where several agents explore the scene jointly to answer a question. [107] proposed a multi-layer structural and semantic memories as scene memories to be shared by multiple agents to first reconstruct the 3D scenes and then perform question answering.

4.4.2. Evaluation Metrics

Embodied QA involves two sub-tasks: *1) Question Answering*, and *2) Navigation*, therefore need to evaluate the performance in these two aspects separately:

1) Question Answering Accuracy is typically measured by the mean rank (MR) of the ground-truth answer of all test questions and environments over all possible answers (colors, rooms, objects).

2) Navigation Accuracy can be further measured by four metrics: (1) *distance to target* which measures the distance to target object at the end of navigation; (2) *change in distance to target* from initial to final position; (3) *smallest distance to target* along the shortest path to the target; and (4) *percentage of episodes agent to terminate navigation for answering* before reaching the maximum episode.

4.4.3. Datasets

EQA [102] dataset is based on House3D, a subset of recent popular SUNCG dataset with synthesis rooms and layouts which is similar to the Replica dataset [108]. House3D converts SUNCG's static environment into virtual environment, where the agent can navigate with physical constraints (e.g. can't pass through walls or objects). To test the agent's capabilities in language grounding, commonsense reasoning and environment navigation, [102] uses a series of functional program in CLEVR [109] to synthesize questions and anwers regarding object's color, existence, location and relative preposition etc. In total, there are 5000 questions in 750 environments with reference to 45 unique objects in 7 unique room types.

Recently,[105] proposed a new embodied QA task: Multi-target embodied QA, which studies questions that that multiple targets in them, e.g. "Is the apple in the bedroom bigger than the orange in the living room?" where the agent has to navigate to "bedroom" and "living room" to localize the "apple" and "orange" and perform comparison to answer the questions.

IQA [106] is another work that focuses on tackling the task of embodied QA in AI2-THOR environment. IQA is different from EQA not only because it is more realistic synthesis but its requirement to interact with the objects to resolve certain questions (e.g. the agent need to open the refrigerator to answer the existence question "if there is an egg in the fridge?")

On top of all these, the authors annotated a large scale Question and Answer dataset (IQUAD V1) which consist of 75,000 multiple-choice questions. Similar to EQA v1 dataset, IQUAD contains questions regarding the existence, counting and spatial relationships.

4.4.4. Challenges

Currently embodied QA faces following two challenges. First, most existing embodied QA systems are generally divided into a navigation module and a QA module. These two modules are individually optimized and then jointly trained by reinforcement learning. Second, the QA bottleneck stems from the worse navigation in the unseen environment, i.e. an agent is hardly able to reach a point to observe the target object in a new environment, let alone answer the question. It suggests that the embodied QA should further consider the generalization problem to new environment.

5. CHALLENGES

In this section, we discuss some challenges in embodied AI simulators and research.

5.1. Challenges in Embodied AI Simulators

With the advancement in computer graphics and state-of-the-art physics engines [110, 111], current embodied AI simulators have all reached a level that separates them from just a conventional game-based simulation for reinforcement learning. They all possess one form of virtual embodiment, which allows for basic controls and interacts with the virtual duplicates of the real-world objects embedded into the virtual environment. There are several existing challenges in embodied AI simulators as deduced from our key findings.

Realism. In terms of realism, there is a lack of quality *world-based scene* simulator that can better bring out the real world's high fidelity and help bridge the gaps between the simulation and real-world. Currently, AI2-THOR [19], Habitat Sim [23], iGibson [24] are the front runner ups to bridging the gaps between simulation and real-world. This is largely because of their continuous effort in the yearly embodied AI challenges [112], which tackles embodied AI problems in visual navigation in both simulation and the real world.

Scalability. In terms of scalability, there is a lack of methodologies to collect large-scale 3D object datasets, unlike image-based datasets [13] which are widely available on the internet and only requires manual annotation of the data. However,it is significantly harder to obtain 3D object datasets, as it requires special techniques such as photogrammetry [113] and other neural rendering approaches [114] to be synthesized before it can be annotated.

Interactivity. In terms of interactivity, there is a lack of rich dynamics physics between objects to object and agent to object interaction within the virtual environment. This is especially significant as simulators with quality interaction between the agent and virtual object would be allowed for an easier deployment of the trained models into the real world. Even the best physics-based embodied AI simulator which is ThreeDWorld [26], are still lacking in complex physics such as particle physics and a mixture of multiple physical properties in a single object (e.g. how a chair would have different object affordance and texture at the different parts of the chair).

Lastly, with the growing interest in the field of embodied AI, computer graphics, 3D objects datasets, embodied AI research will be expected to grow significantly. Hence,the nature of those embodied AI simulators will be vital to support those embodied AI research and open up doors to many exciting research tasks that are yet to be unveiled.

5.2. Challenges in Embodied AI Research

The domain of embodied AI research is vast, stretching from visual exploration to embodied QA, with each task having its own set of challenges and problems to be addressed. The pyramid of embodied AI research's fundamental blocks serves to provide for more complex blocks up the pyramid. A foreseeable trend for the pyramid of embodied AI research is a task-based interactive question answering (TIQA), which aims to integrate tasks with answering specific questions. For example, such questions can be *How long would it take for an egg to boil? Is there an apple in the cabinet?*. These are questions that cannot be answered through the conventional approaches [102, 106] due to a lack of the capability to perform general-purpose tasks in an environment to unlock new insights and infer, which aids in answering those questions. For TIQA agents to answer those questions, it has to navigate the room to keep track of the spatial or existing relationships of the objects-of-interest and execute specifics task in the environment. From performing these tasks, it can observe the result and conclude the answer to the posted questions. Such embodied QA system's implications may hold the key to generalising task-planning and developing general-purpose artificial agent in simulations that can later be deployed into the real world.

Conclusion

Recent advances in embodied AI simulators have been a key driver of progress in embodied AI research. Aiming to understand the trends and gaps in embodied AI simulators and research, this paper provides a contemporary and comprehensive overview of embodied AI simulators and research. The paper surveys state-of-the-art embodied AI simulators and their connections in serving and driving recent innovations in research tasks for embodied AI. By benchmarking nine state-of-the-art embodied AI simulators in terms of seven features, we seek to understand their provision for realism, scalability and interactivity, and hence use in embodied AI research. The three main tasks supporting the pyramid of embodied AI research – visual exploration, visual navigation and embodied QA, are examined in terms of the state-of-the-art approaches, evaluation, and datasets. This is to review and benchmark the existing approaches in tackling these categories of embodied AI research tasks in the various embodied AI simulators. Based on the findings and discussions, we seek to aid AI researchers in the selection of embodied AI simulators for their research tasks, as well as computer graphics researchers in developing embodied AI simulators that are aligned with and support the current embodied AI research trends.

Acknowledgments

This research is supported by the Agency for Science, Technology and Research (A*STAR), Singapore under its AME Programmatic Funding Scheme (Award #A18A2b0046) and the National Research Foundation, Singapore under its NRF-ISF Joint Call (Award NRF2015-NRF-ISF001-2541).

6. REFERENCES

[1] Rolf Pfeifer and Fumiya Iida, "Embodied artificial intelligence: Trends and challenges," in *Embodied artificial intelligence*, pp. 1–26. Springer, 2004.

[2] Rolf Pfeifer and Josh C. Bongard, "How the body shapes the way we think - a new view on intelligence," 2006.

[3] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[4] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al., "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[5] Mohammed AlQuraishi, "Alphafold at casp13," *Bioinformatics*, vol. 35, no. 22, pp. 4862–4865, 2019.

[6] Nils J Nilsson, "Human-level artificial intelligence? be serious!," *AI magazine*, vol. 26, no. 4, pp. 68–68, 2005.

[7] Murray Campbell, A Joseph Hoane Jr, and Feng-hsiung Hsu, "Deep blue," *Artificial intelligence*, vol. 134, no. 1-2, pp. 57–83, 2002.

[8] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al., "Mastering the game of go without human knowledge," *nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[9] Francisco Rubio, Francisco Valero, and Carlos Llopis-Albert, "A review of mobile robots: Concepts, methods, theoretical framework, and applications," *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, pp. 1729881419839596, 2019.

[10] Jiafei Duan, Samson Yu, Hui Li Tan, and Cheston Tan, "Actionet: An interactive end-to-end platform for task-based data collection and augmentation in 3d environment," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 1566–1570.

[11] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox, "Alfred: A benchmark for interpreting grounded instructions for everyday tasks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10740–10749.

[12] John Haugeland, "Artificial intelligence: The very idea, cambridge, ma, bradford," 1985.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[17] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al., "On evaluation of embodied navigation agents," *arXiv preprint arXiv:1807.06757*, 2018.

[18] Charles Beattie, Joel Z Leibo, Denis Teplyashin, Tom Ward, Marcus Wainwright, Heinrich Küttler, Andrew Lefrancq, Simon Green, Víctor Valdés, Amir

Sadik, et al., "Deepmind lab," *arXiv preprint arXiv:1612.03801*, 2016.

[19] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi, "Ai2-thor: An interactive 3d environment for visual ai," *arXiv preprint arXiv:1712.05474*, 2017.

[20] Claudia Yan, Dipendra Misra, Andrew Bennnett, Aaron Walsman, Yonatan Bisk, and Yoav Artzi, "Chalet: Cornell house agent learning environment," *arXiv preprint arXiv:1801.07357*, 2018.

[21] Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba, "Virtualhome: Simulating household activities via programs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8494–8502.

[22] Xiaofeng Gao, Ran Gong, Tianmin Shu, Xu Xie, Shu Wang, and Song-Chun Zhu, "Vrkitchen: an interactive 3d virtual environment for task-oriented learning," *arXiv preprint arXiv:1903.05757*, 2019.

[23] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al., "Habitat: A platform for embodied ai research," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9339–9347.

[24] Fei Xia, William B Shen, Chengshu Li, Priya Kasimbeg, Micael Edmond Tchapmi, Alexander Toshev, Roberto Martín-Martín, and Silvio Savarese, "Interactive gibson benchmark: A benchmark for interactive navigation in cluttered environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 713–720, 2020.

[25] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al., "Sapien: A simulated part-based interactive environment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11097–11107.

[26] Chuang Gan, Jeremy Schwartz, Seth Alter, Martin Schrimpf, James Traer, Julian De Freitas, Jonas Kubilius, Abhishek Bhandwaldar, Nick Haber, Megumi Sano, et al., "Threedworld: A platform for interactive multi-modal physical simulation," *arXiv preprint arXiv:2007.04954*, 2020.

[27] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling, "The arcade learning environment: An evaluation platform for general agents," *Journal of Artificial Intelligence Research*, vol. 47, pp. 253–279, 2013.

[28] Abhishek Kadian, Joanne Truong, Aaron Gokaslan, Alexander Clegg, Erik Wijmans, Stefan Lee, Manolis Savva, Sonia Chernova, and Dhruv Batra, "Sim2real predictivity: Does evaluation in simulation predict real-world performance?," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6670–6677, 2020.

[29] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 1–8.

[30] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 23–30.

[31] Matt Deitke, Winson Han, Alvaro Herrasti, Aniruddha Kembhavi, Eric Kolve, Roozbeh Mottaghi, Jordi Salvador, Dustin Schwenk, Eli VanderBilt, Matthew Wallingford, et al., "Robothor: An open simulation-to-real embodied ai platform," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3164–3174.

[32] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser, "Semantic scene completion from a single depth image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1746–1754.

[33] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," *arXiv preprint arXiv:1709.06158*, 2017.

[34] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese, "Gibson env: Real-world perception for embodied agents," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9068–9079.

[35] Emanuel Todorov, Tom Erez, and Yuval Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.

[36] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su, "Part-

net: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 909–918.

[37] Abhishek Kadian*, Joanne Truong*, Aaron Gokaslan, Alexander Clegg, Erik Wijmans, Stefan Lee, Manolis Savva, Sonia Chernova, and Dhruv Batra, "Are We Making Real Progress in Simulated Environments? Measuring the Sim2Real Gap in Embodied Visual Navigation," in *arXiv:1912.06321*, 2019.

[38] Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans, "Objectnav revisited: On evaluation of embodied agents navigating to objects," *arXiv preprint arXiv:2006.13171*, 2020.

[39] Unnat Jain, Luca Weihs, Eric Kolve, Ali Farhadi, Svetlana Lazebnik, Aniruddha Kembhavi, and Alexander Schwing, "A cordial sync: Going beyond marginal policies for multi-agent embodied tasks supplementary material," .

[40] Unnat Jain, Luca Weihs, Eric Kolve, Mohammad Rastegari, Svetlana Lazebnik, Ali Farhadi, Alexander G. Schwing, and Aniruddha Kembhavi, "Two body problem: Collaborative visual task completion," in *CVPR*, 2019, first two authors contributed equally.

[41] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[42] Unnat Jain, Luca Weihs, Eric Kolve, Mohammad Rastegari, Svetlana Lazebnik, Ali Farhadi, Alexander G Schwing, and Aniruddha Kembhavi, "Two body problem: Collaborative visual task completion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6689–6699.

[43] Luca Weihs, Jordi Salvador, Klemen Kotar, Unnat Jain, Kuo-Hao Zeng, Roozbeh Mottaghi, and Aniruddha Kembhavi, "Allenact: A framework for embodied ai research," *arXiv*, 2020.

[44] Dmytro Mishkin, Alexey Dosovitskiy, and Vladlen Koltun, "Benchmarking classic and learned navigation in complex 3d environments," *arXiv preprint arXiv:1901.10915*, 2019.

[45] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov, "Learning to explore using active neural slam," *arXiv preprint arXiv:2004.05155*, 2020.

[46] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell, "Curiosity-driven exploration by self-supervised prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 16–17.

[47] Tao Chen, Saurabh Gupta, and Abhinav Gupta, "Learning exploration policies for navigation," *arXiv preprint arXiv:1903.01959*, 2019.

[48] Santhosh K Ramakrishnan, Dinesh Jayaraman, and Kristen Grauman, "An exploration of embodied visual exploration," *arXiv preprint arXiv:2001.02192*, 2020.

[49] Saurabh Gupta, David Fouhey, Sergey Levine, and Jitendra Malik, "Unifying map and landmark based representations for visual navigation," *arXiv preprint arXiv:1712.08125*, 2017.

[50] Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun, "Semi-parametric topological memory for navigation," *arXiv preprint arXiv:1803.00653*, 2018.

[51] Edward Beeching, Jilles Dibangoye, Olivier Simonin, and Christian Wolf, "Learning to plan with uncertain topological maps," *arXiv preprint arXiv:2007.05270*, 2020.

[52] Medhini Narasimhan, Erik Wijmans, Xinlei Chen, Trevor Darrell, Dhruv Batra, Devi Parikh, and Amanpreet Singh, "Seeing the un-scene: Learning amodal semantic maps for room navigation," *arXiv preprint arXiv:2007.09841*, 2020.

[53] Santhosh K Ramakrishnan, Ziad Al-Halah, and Kristen Grauman, "Occupancy anticipation for efficient exploration and navigation," *arXiv preprint arXiv:2008.09285*, 2020.

[54] Saurabh Gupta, James Davidson, Sergey Levine, Rahul Sukthankar, and Jitendra Malik, "Cognitive mapping and planning for visual navigation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2616–2625.

[55] Joao F Henriques and Andrea Vedaldi, "Mapnet: An allocentric spatial memory for mapping environments," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8476–8484.

[56] Kuan Fang, Alexander Toshev, Li Fei-Fei, and Silvio Savarese, "Scene memory transformer for embodied agents in long-horizon tasks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 538–547.

[57] Lina Mezghani, Sainbayar Sukhbaatar, Arthur Szlam, Armand Joulin, and Piotr Bojanowski, "Learning to visually navigate in photorealistic environments without any supervision," *arXiv preprint arXiv:2004.04954*, 2020.

[58] Georgios Georgakis, Yimeng Li, and Jana Kosecka, "Simultaneous mapping and target driven navigation," *arXiv preprint arXiv:1911.07980*, 2019.

[59] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[60] Santhosh K Ramakrishnan, Dinesh Jayaraman, and Kristen Grauman, "Emergence of exploratory look-around behaviors through active observation completion," *Science Robotics*, vol. 4, no. 30, 2019.

[61] William S Lovejoy, "A survey of algorithmic methods for partially observed markov decision processes," *Annals of Operations Research*, vol. 28, no. 1, pp. 47–65, 1991.

[62] Brian Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*. IEEE, 1997, pp. 146–151.

[63] Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros, "Large-scale study of curiosity-driven learning," *arXiv preprint arXiv:1808.04355*, 2018.

[64] Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel, "Vime: Variational information maximizing exploration," in *Advances in Neural Information Processing Systems*, 2016, pp. 1109–1117.

[65] Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta, "Self-supervised exploration via disagreement," *arXiv preprint arXiv:1906.04161*, 2019.

[66] Devendra Singh Chaplot, Helen Jiang, Saurabh Gupta, and Abhinav Gupta, "Semantic curiosity for active visual learning," *arXiv preprint arXiv:2006.09367*, 2020.

[67] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov, "Exploration by random network distillation," *arXiv preprint arXiv:1810.12894*, 2018.

[68] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[69] Soroush Seifi and Tinne Tuytelaars, "Where to look next: Unsupervised active visual exploration on 360 degree input," *arXiv preprint arXiv:1909.10304*, 2019.

[70] Dinesh Jayaraman and Kristen Grauman, "Learning to look around: Intelligently exploring unseen environments for unknown tasks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1238–1247.

[71] Shuran Song, Andy Zeng, Angel X Chang, Manolis Savva, Silvio Savarese, and Thomas Funkhouser, "Im2pano3d: Extrapolating 360 structure and semantics beyond the field of view," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3847–3856.

[72] Santhosh K Ramakrishnan and Kristen Grauman, "Sidekick policy learning for active visual exploration," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 413–430.

[73] Nikolay Savinov, Anton Raichuk, Raphaël Marinier, Damien Vincent, Marc Pollefeys, Timothy Lillicrap, and Sylvain Gelly, "Episodic curiosity through reachability," *arXiv preprint arXiv:1810.02274*, 2018.

[74] Nick Haber, Damian Mrowca, Stephanie Wang, Li F Fei-Fei, and Daniel L Yamins, "Learning to play with intrinsically-motivated, self-aware agents," in *Advances in Neural Information Processing Systems*, 2018, pp. 8388–8399.

[75] Ayzaan Wahid, Austin Stone, Kevin Chen, Brian Ichter, and Alexander Toshev, "Learning object-conditioned exploration using distributed soft actor critic," *arXiv preprint arXiv:2007.14545*, 2020.

[76] Heming Du, Xin Yu, and Liang Zheng, "Learning object relation graph and tentative policy for visual navigation," *arXiv preprint arXiv:2007.11018*, 2020.

[77] Devendra Singh Chaplot, Dhiraj Gandhi, Abhinav Gupta, and Ruslan Salakhutdinov, "Object goal navigation using goal-oriented semantic exploration," *arXiv preprint arXiv:2007.00643*, 2020.

[78] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 3357–3364.

[79] Devendra Singh Chaplot, Ruslan Salakhutdinov, Abhinav Gupta, and Saurabh Gupta, "Neural topological slam for visual navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12875–12884.

[80] Joel Ye, Dhruv Batra, Erik Wijmans, and Abhishek Das, "Auxiliary tasks speed up learning pointgoal navigation," *arXiv preprint arXiv:2007.04561*, 2020.

[81] Tommaso Campari, Paolo Eccher, Luciano Serafini, and Lamberto Ballan, "Exploiting scene-specific features for object goal navigation," *arXiv preprint arXiv:2008.09403*, 2020.

[82] Francisco Bonin-Font, Alberto Ortiz, and Gabriel Oliver, "Visual navigation for mobile robots: A survey," *Journal of intelligent and robotic systems*, vol. 53, no. 3, pp. 263, 2008.

[83] Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha, "Visual simultaneous localization and mapping: a survey," *Artificial intelligence review*, vol. 43, no. 1, pp. 55–81, 2015.

[84] Lydia E Kavraki, Petr Svestka, J-C Latombe, and Mark H Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[85] Steven M LaValle and James J Kuffner, "Rapidly-exploring random trees: Progress and prospects," *Algorithmic and computational robotics: new directions*, , no. 5, pp. 293–308, 2001.

[86] Xin Ye and Yezhou Yang, "From seeing to moving: A survey on learning for visual indoor navigation (vin)," *arXiv preprint arXiv:2002.11310*, 2020.

[87] Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra, "Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames," *arXiv*, pp. arXiv–1911, 2019.

[88] Alexey Dosovitskiy and Vladlen Koltun, "Learning to act by predicting the future," *arXiv preprint arXiv:1611.01779*, 2016.

[89] Peter Dayan, "Improving generalization for temporal difference learning: The successor representation," *Neural Computation*, vol. 5, no. 4, pp. 613–624, 1993.

[90] Yuke Zhu, Daniel Gordon, Eric Kolve, Dieter Fox, Li Fei-Fei, Abhinav Gupta, Roozbeh Mottaghi, and Ali Farhadi, "Visual semantic planning using deep successor representations," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 483–492.

[91] André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver, "Successor features for transfer in reinforcement learning," *Advances in neural information processing systems*, vol. 30, pp. 4055–4065, 2017.

[92] Daniel Gordon, Abhishek Kadian, Devi Parikh, Judy Hoffman, and Dhruv Batra, "Splitnet: Sim2sim and task2task transfer for embodied visual navigation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1022–1031.

[93] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov, "Proximal policy optimization algorithms. arxiv 2017," *arXiv preprint arXiv:1707.06347*.

[94] Arsalan Mousavian, Alexander Toshev, Marek Fišer, Jana Košecká, Ayzaan Wahid, and James Davidson, "Visual representations for semantic target driven navigation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8846–8852.

[95] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[96] Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Bernardo A Pires, and Rémi Munos, "Neural predictive belief representations," *arXiv preprint arXiv:1811.06407*, 2018.

[97] Mitchell Wortsman, Kiana Ehsani, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi, "Learning to learn how to learn: Self-adaptive visual navigation using meta-learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6750–6759.

[98] Wei Yang, Xiaolong Wang, Ali Farhadi, Abhinav Gupta, and Roozbeh Mottaghi, "Visual semantic navigation using scene priors," *arXiv preprint arXiv:1810.06543*, 2018.

[99] Chuang Gan, Yiwei Zhang, Jiajun Wu, Boqing Gong, and Joshua B Tenenbaum, "Look, listen, and act: Towards audio-visual embodied navigation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9701–9707.

[100] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen

Gould, and Anton van den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3674–3683.

[101] Yi Zhu, Fengda Zhu, Zhaohuan Zhan, Bingqian Lin, Jianbin Jiao, Xiaojun Chang, and Xiaodan Liang, "Vision-dialog navigation by exploring cross-modal memory," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10730–10739.

[102] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra, "Embodied question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 2054–2063.

[103] Ronald J Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[104] Abhishek Das, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra, "Neural modular control for embodied question answering," *arXiv preprint arXiv:1810.11181*, 2018.

[105] Licheng Yu, Xinlei Chen, Georgia Gkioxari, Mohit Bansal, Tamara L. Berg, and Dhruv Batra, "Multi-target embodied question answering," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2019, pp. 6309–6318.

[106] Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi, "Iqa: Visual question answering in interactive environments," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4089–4098.

[107] Sinan Tan, Weilai Xiang, Huaping Liu, Di Guo, and Fuchun Sun, "Multi-agent embodied question answering in interactive environments," in *ECCV 2020 - 16th European Conference, Glasgow,UK, August 23-28, 2020, Proceedings*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, Eds., 2020, pp. 663–678.

[108] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al., "The replica dataset: A digital replica of indoor spaces," *arXiv preprint arXiv:1906.05797*, 2019.

[109] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick, "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2901–2910.

[110] Erwin Coumans and Yunfei Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," 2016.

[111] Yuke Zhu, Josiah Wong, Ajay Mandlekar, and Roberto Martín-Martín, "robosuite: A modular simulation framework and benchmark for robot learning," *arXiv preprint arXiv:2009.12293*, 2020.

[112] CVPR, "Embodied AI Workshop, year = 2020, url = https://embodied-ai.org/, urldate = 2020-09-30," .

[113] Edward M Mikhail, James S Bethel, and J Chris McGlone, "Introduction to modern photogrammetry," *New York*, p. 19, 2001.

[114] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth, "Nerf in the wild: Neural radiance fields for unconstrained photo collections," *arXiv preprint arXiv:2008.02268*, 2020.